# SRC上分秘诀+实战挖掘+挖洞技巧+新手上路+详细讲解

[thelm](https://bbs.zkaq.cn/u/thelm.html)  · 发表于 2021-09-04 00:39:33 · [技术文章](https://bbs.zkaq.cn/f/200.html)

[](https://bbs.zkaq.cn/u/thelm.html)

#### 9月SRC马上到来 可能有些好兄弟们还没有头绪 只会做一些靶场 并没有什么实战经验

##### 所以这篇文章给大家分享一下我挖洞2个月的经验分享 适合新手上路

如何找站？

## 谷歌搜索 谷歌搜索 谷歌搜索

#### SQL注入+XSS+所有漏洞

inurl:.php?id=xx 公司`  
`inurl:.asp?id=xx 公司`  
`inurl:.jsp?id=xx 公司

id传参的数字也可以变换 这样就可以找到不同的站点



Figure :

当然也可以在后面加上地区 比如：`  
`inurl:.php?id=xx 公司 陕西



Figure :

##### SQL注入的那些见不得人的小秘密：

首先就是Bypass了 如何绕：  
Bypass（绕狗）:  
  
大小写加负数绕过：  
ANd -1=-1  
  
编码绕过:  
跟DOM型XSS一样  
  
+号连接绕过：  
and+1=1  
union+select+1+2  
  
缓冲区POST：  
post传递的参数可以有很多  
有些安全狗只检测某个长度之前的字符 之后的并不检测  
  
白名单（管理员权限或者是127.0.0.1本地访问不拦截，设置X-Forwarded-For）

、

如果在挖SQL注入的时候碰到了这种安全狗：



Figure :

想都别想 这个时候你应该开心起来  
如果这个站点存在SQL注入的话 那就可以用这种方式绕过  
  
无闭合方式：  
http://localhost/?id=1 --+/\*%0aand 1=1 --+\*/  
  
有闭合方式：  
http://localhost/?id=1 --+/\*%0a'and 1=1 --+ --+\*/  
  
所以只要在挖SQL注入的时候看到这类安全狗 就跟没有一样  
有的话直接提交 没有换下一个

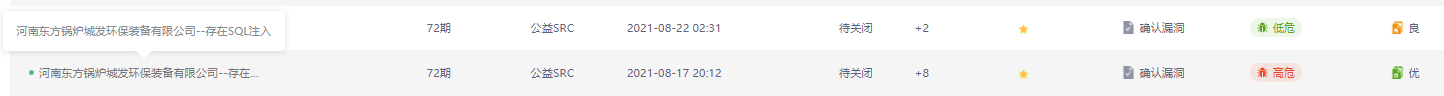


Figure :

##### 网站内部过滤特殊代码：

测试网站是否存在SQL注入的时候 很简单 直接and 1=1呗  
但是可能会遇到这种站点

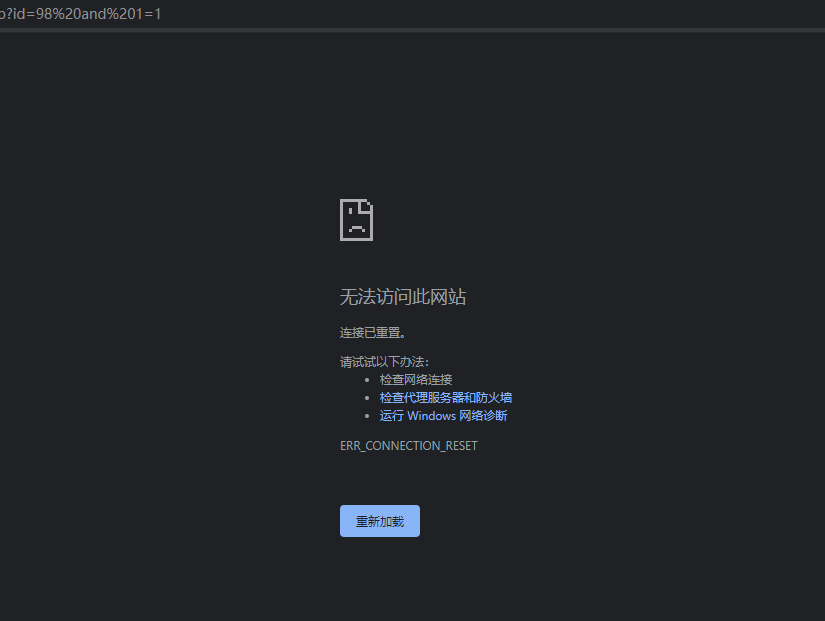


Figure :

它直接来了一个链接出错 有些人看到这种情况直接走了 换下一个站点  
如果是这样的想法 那这个站点的SQL注入就与你无缘了  
  
遇到这种情况还挺气人  
而且这个页面时不时还自动刷新一下 感觉它像再说：我还有救 快来测试我！  
  
正如以上说的那样 还真有救。

在id后面加上-false 有数据

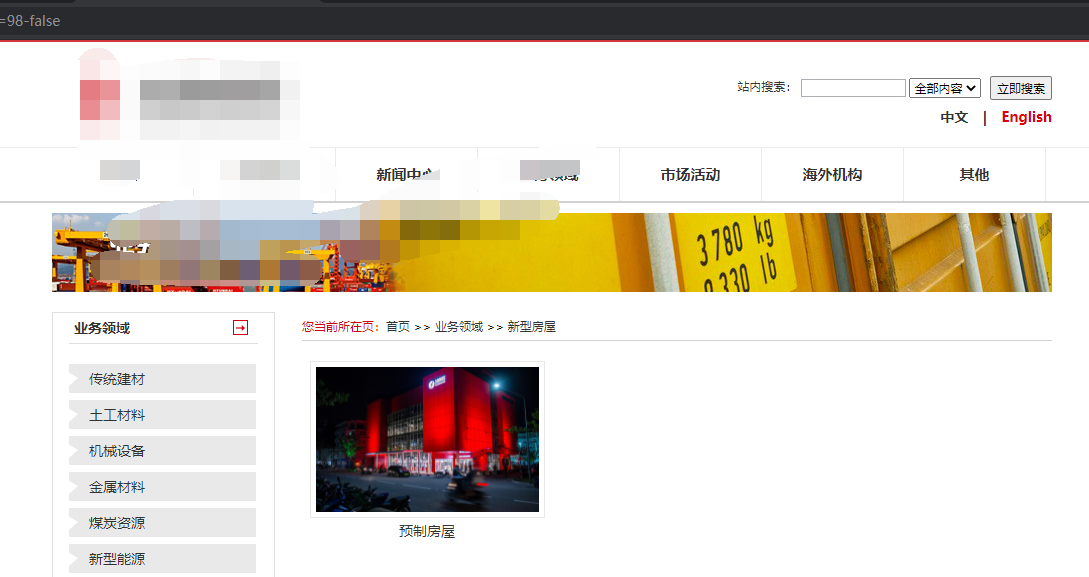


Figure :

加上-true 没数据



Figure :

说明这里存在SQL注入  
那怎么办 不止有显错注入 还有盲注  
  
可以使用:  
length(database())>1  
ascii(substr(database(),1,1))>1   
  
但是这个需要and啊 and 被检测了  
and 其实可以转换为 HAVING  
HAVING可以代替and

、

测试结果如下  
HAVING length(database())=7  
当前库名长度为：7

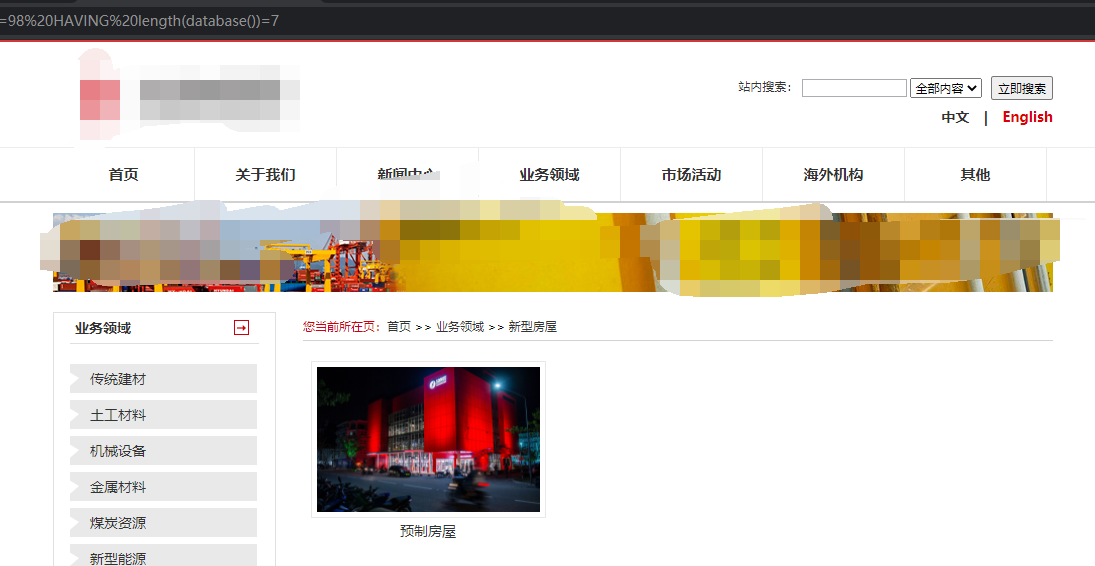


Figure :

不要遗忘了使用盲注 有时候 显错注入做不出来的时候 就可以用盲注  
  
有时候网站对接的sql语句太复杂 虽然是单引号闭合 当你开始查库名的时候就开始报错了

ps：  
绕狗确实真实有效 D盾除外  
D盾也就只能输入个group by来测试一下字段 想输入其他敏感语句还是算了吧 人生苦短 何必跟D盾过不去

##### 伪静态

伪静态也是一样的  
比如：  
  
http://localhost/index.html  
http://localhost/index and 1=1.html

总结：  
不管什么SQL注入都要结合着来 就算这个地方存在显错注入 你也可以试试用延时注入做出来 这样做的好处就是可以增强对盲注语法认识的强烈性  
  
如果一直用显错 一直不用其他注入的话 可能下次连靶场都做不出来了  
只能看看自己之前写的作业 然后重新认识语法  
  
聂风说过一句话：能用显错为什么要用盲注呢？  
其实挖掘漏洞就是为了提升自己 哈哈 我也没有怼聂风

如何选择：  
一般人都去挖php站点了 也可以试试挖asp和jsp  
  
这些站点做起来比php站点难 步骤多 大佬也懒得写那么多步骤 最后也就加个4分左右  
这类站点大多部分存在cookie注入 逻辑漏洞 存储型XSS 以及各种莫名其妙的漏洞  
  
还记得我挖到的第一个外网漏洞 就是学完存储型XSS 就在一个asp站点上找到的

### XSS

XSS不用专门去找 一般挖SQL注入漏洞的时候 直接把XSS语句往id传参值哪里一粘贴 或者搜索框粘贴  
  
如果有 那就有 。如果没有 那就没有喽  
但是一般情况下 我挖到的SQL注入站点上 80%都存在XSS

一定要见框就插 就是因为这个 我挖到了一个gov站点的反射型XSS  
还是挺幸运的 挖洞就是运气最重要  
如下：

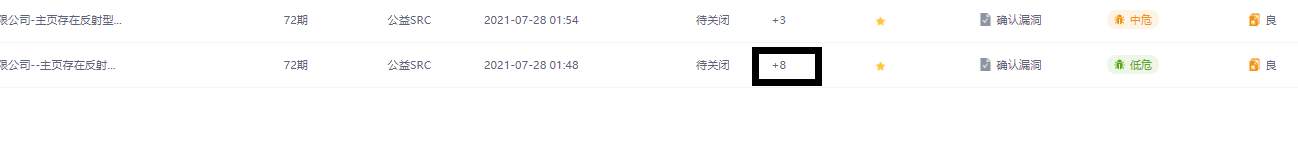


Figure :

反射加8分 没见过吧 毕竟是gov站点

注意：  
有些时候直接粘贴XSS语句 然后网站就把你ip封了 访问不了网站  
这种情况不需要挂代理 只需要重启路由器就可以了  
路由器地址一般都是192.168.1.1



Figure :

### 弱密码+爆破+SQL注入

inurl:后台管理  
弱密码也可以试试 有时候admin admin 就进去了  
这个时候就可以提交 反正这也是个漏洞 虽然说没有技术性 但是还很难找  
不过还可以加个2-3分



Figure :

当然有时间的话 也可以选择爆破 并且抓包分析 试试有没有万能密码

ps：  
当你挖掘所有漏洞的时候 假如说你什么都没有挖到 那么就在试试找一下网站后台 比如：  
  
admin  
webadmin  
......

### 支付漏洞:

info: 商城 and 积分商城  
通过抓包去改参数  
比如：  
点击充值金额，抓住返回包，第一个包通常都不需要去改，直接放掉  
第二个返回包和第三个返回包可以仔细观察观察，那些都是接口的数据包  
通常把接口数据包里的金额值 改成其他

### 文件下载漏洞

inurl:.php?file 公司`  
其实文件下载漏洞不用刻意的去挖  
挖其他漏洞的时候可以看看网站最下方有没有资料下载  
有的话就直接测试，这波岂不是血赚

只要有站点可以下载东西 可以现下载一下 看一下URL是否出现这种东西  
比如：  
http://localhost/1.php?file=download/web.pdf  
像这种 有file=赋值的 就很有可能有逻辑缺陷 就会存在任意文件下载  
  
但是·测试时可能会遇到这种站点：  
http://www.xxx.com/file/download/xxx.zip  
  
一般我遇到过的任意文件下载在后面输入../就可以下载到其他文件  
但是这类站点输入../后并不会下载到这类文件 只是会跳转到这个页面  
  
比如：  
https://www.xx.com/upload/file/../../index.php  
  
他会直接跳转到  
https://www.xx.com/index.php

### 文件上传漏洞

通常出现在很老的系统，但是也是有的  
比如：  
攻击者上传了一个可执行文件如木马，病毒，恶意脚本，WebShell等到服务器执行，并最终获得网站控制权限

找到上传点



Figure :

前端验证 那么就上传jpg文件 抓包改后缀

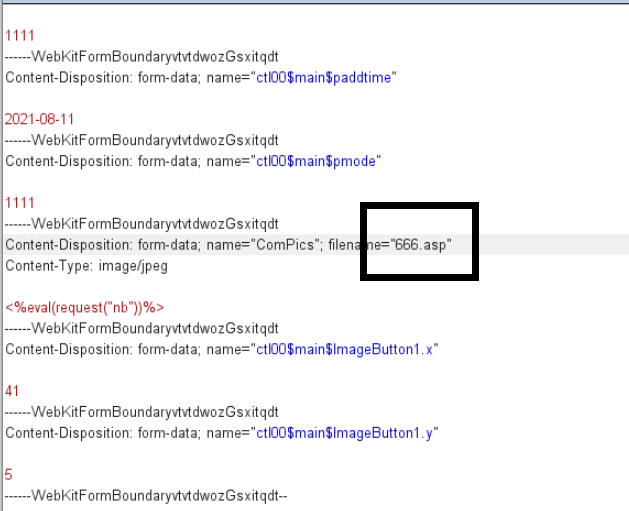


Figure :

最后复制图片地址 asp木马就成功上传到服务器了

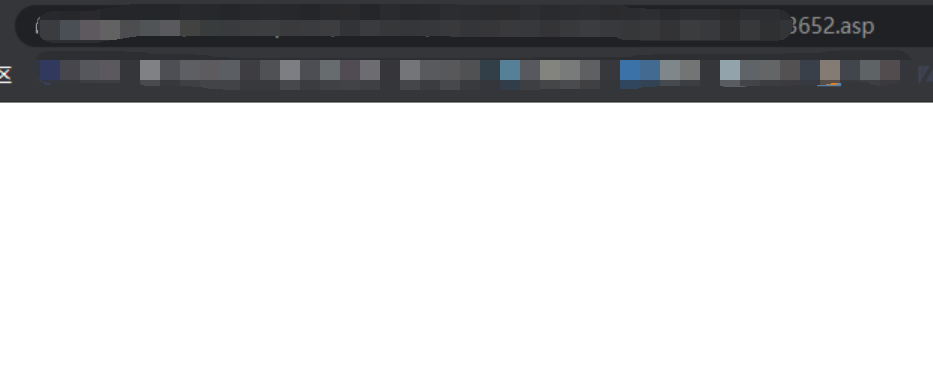


Figure :

总结：  
碰到文件上传的地方 都可以试试上传一个一句话木马

#### 逻辑漏洞:

size: 出版社

任意密码修改：  
  
逻辑漏洞也不需要刻意去挖  
找第一类站的时候 就会碰到很多有逻辑漏洞的站点  
  
例如：  
用户在修改自己密码的时候可以将modid改成其他 即可修改他人的账号密码  
  
在登录账号1的情况下，修改密码，将modid=383改成384即可修改账户2的密码



Figure :

任意用户资料修改：  
  
注册登录一个账号后 可以修改自己资料的同时抓个返回包  
同时修改返回包里的userid传参就可以修改到别人的资料



Figure :

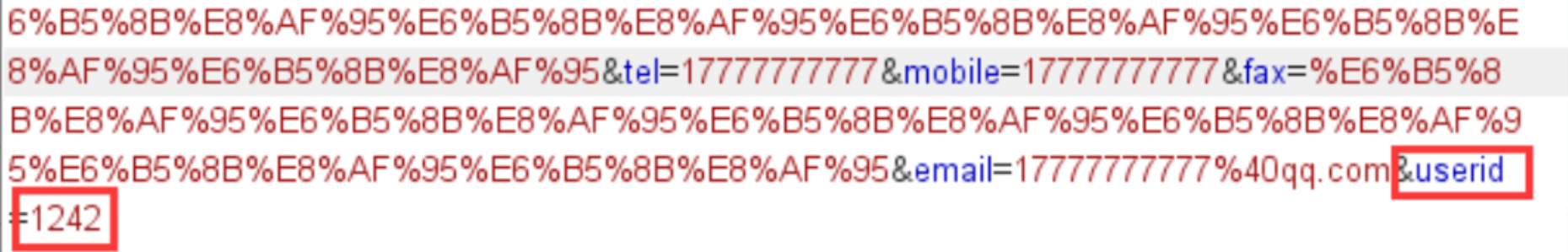


Figure :

ps：逻辑漏洞是上分的最好选择  
  
为什么：  
实操性强 步骤多  
  
步骤如下：任意用户密码修改  
首先得创建个a用户，找到某个功能点，测试某个功能点，抓包分析某个功能点，测试某个功能点  
其次创建b用户，验证a用户是否能影响到b用户  
  
这么多个步骤下来 你提交漏洞盒子不得写上个50个字左右？图片不得5-10个左右  
所以这样下来 漏洞报告质量不是优就是良  
  
  
但是我还要说一点：  
就是再写漏洞复现的时候一定要把测试的账号密码写到步骤当中  
写上a用户的账户密码 和b用户的账号密码  
为什么啊？ 答：因为这样更容易得优 漏洞盒子审核人员得验证你的漏洞 所以他们还得注册 你直接提供给他这样不是更好  
这个是我成绩



Figure :

逻辑漏洞全靠操作性 不像XSS一样 直接输入一个alert(1) 步骤最多也就三步吧？1.打开URL 2.输入alert(1) 3.弹窗1  
然后你直接提交盒子 报告质量肯定不是良就是差  
ps：  
虽然说xss不是良就是差 偶尔盒子还说xss漏洞选择性接受 但是上分的话 确实挺有用的

#### 越权漏洞

其实跟逻辑漏洞差不多 最简单的就是修改Cookie

例如：  
账户A通过修改cookie可以登录到账户B的账号上  
  
这样是不是就把ceshi1用户名，修改成ceshi2 就可以登录到这个账号上了

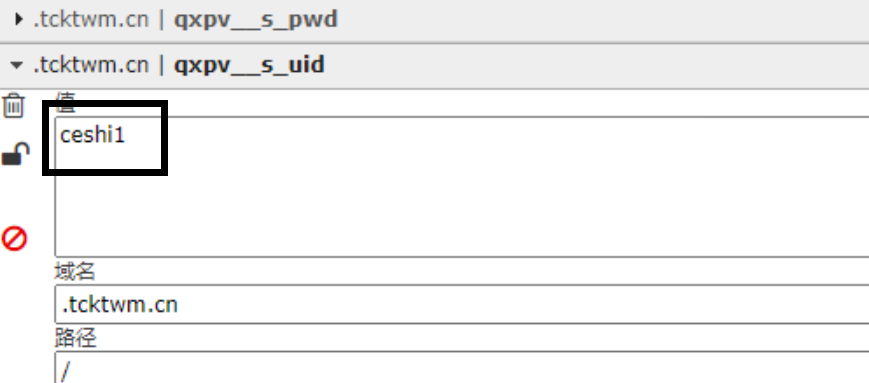


Figure :

#### 认证缺陷漏洞

例如：  
验证码绕过：  
本身不是漏洞 但是只要造成危害就是漏洞 就可以提交  
比如  
爆破到管理员账号密码  
  
任意用户注册：  
1，比如说有一个注册点，没有验证码就直接可以注册  
那么这种漏洞直接可以用Bp跑包 注册几万个账号都没问题  
  
2，跟随验证码绕过  
如果验证码可以绕过 那就跟没有一样  
只要抓个包不放 或者放完包之后直接照着抄新的验证码就行  
  
短信轰炸漏洞：  
有些人可能没听说过这种漏洞，还在想短信轰炸？  
这个不就是那个代刷网嘛 只要掏钱就就可以帮你短信轰炸某某人一天  
  
这个其实就是一个手机号可以在短时间内接受多条验证码  
  
有的站点1分钟才能让你发一次  
有的站点直接没有限制，但这个还是看运气，我运气好就碰到了这种站点  
  
比如：

直接输入手机号



Figure :

随后点击获取验证码的时候 Burp抓一个返回包  
放到单独发包

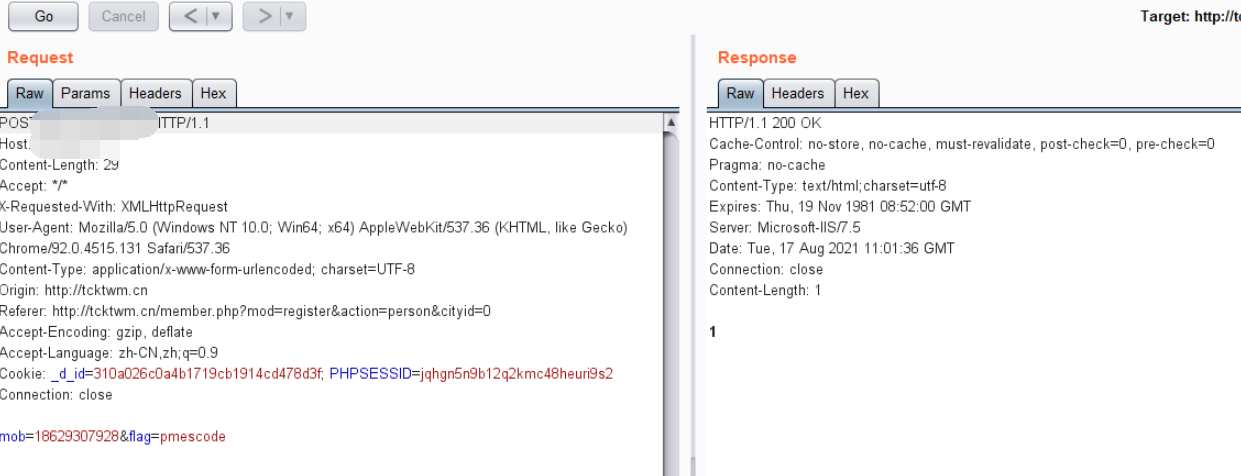


Figure :

Go之后会显示1  
这个时候多Go几次 手机短信就噌噌的响



Figure :

但是我还是感觉会有那种只限制前端的站点  
前端显示60秒倒计时 但是你可以第一次发送验证码的时候 多发送几次 看看结果  
这个我个人感觉会有的 只是没碰到过  
  
  
密码找回漏洞：  
这个漏洞我也不说什么 你们直接看漏洞复现  
  
a用户在知道b用户的邮箱号的情况下，可以修改b用户的密码  
a用户在个人中心 修改邮箱 收到验证码后填写验证码 修改返回包  
将a用户的邮箱号改成b的用户名，这样b用户的邮箱就改成a用户的邮箱号了  
然后a用户 进行找回密码 就可以将b用户的密码进行修改  
  
是不是听的有点绕 我也感觉 逻辑漏洞就是太麻烦了 但是确实写得好 加的分多  
  
因为当时可能网络有点问题 最后一张截图没有截上去   
所以导致盒子不给通过 真的是太烦了 就像自己女朋友丢了一样 步骤多 我也懒得重新提交了



Figure :

ps：我没有女朋友

## FOFA：

一般都是通杀 就是某个系统爆出了某个漏洞 可以使用fofa语法查询此类站点  
利用一个POC 直接漏洞复现 感觉也没太多意思   
反正我自己也很少使用FOFA 也说不了太多经验

感兴趣的可以搜搜这个语法：  
body="DC\_Login/QYSignUp"  
  
ps：前台文件上传 + 后台文件上传

### 到这里漏洞分享也就结束了 所以总结一下重点

这个地方确实存在SQL注入 但是莫名奇妙的报错 试试盲注  
  
网站有装安全狗 试试bypass  
  
输入了敏感字符从而被封ip 重启路由器  
  
没挖到漏洞不要紧 在URL后面加一个amdin 说不定进了后台管理 又有可以测试的地方了

## 篇外之自研小脚本：

#### union.py:

有时候手动测试SQL注入的时候 站点的字段非常多 输入20-60个字段 从而把手累死？  
这个时候 该怎么办呢 别急 专门为这种情况做了一个小脚本  
如下：  
  
a = 0  
b = "and 1=2 union select "  
password = input("请输入要生成的字段数：")  
print(b,end="")  
for i in range(0,int(password)):  
 a += 1  
 print(int(a),end=",")

测试截图：

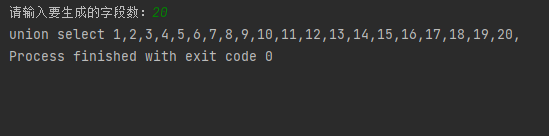


Figure :

#### 谷歌URL批量采集.py：

该脚本可以配合超级SQL注入工具 批量的扫描站点的漏洞  
如何使用：

#第一步：首先在自己的浏览器打开谷歌  
#第二步：启动脚本 脚本启动后会在目录下自动生成一个url.txt 收集到的URL会保存到该文件

import requests  
from lxml import etree  
import random  
open('url.txt','w')  
url = "https://g.luciaz.me/search?q=inurl:.php?id=%s 公司&amp;start=%s" #这里是谷歌语法  
head = {  
 'User-Agent':'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/92.0.4515.159 Safari/537.36'  
}  
password = random.randint(50,200)  
print("当前随机数为：%d"%password)  
print("自动采集id参数为 %d 的公司URL"%password)  
for i in range(20): #默认采集20个，有需求自己改  
 newurl = format(url%(password,(i+1)))  
 shuju = requests.get(url=newurl,headers=head).text  
 data = etree.HTML(shuju)  
 li\_list = data.xpath('//div[@class="eqAnXb"]/div/div/div/div[1]/div/div/div/a/@href')[0]  
 with open("url.txt","a",encoding="utf-8") as ww:  
 ww.write(li\_list+"\n")  
 print("当前已完成：%s"%i)

测试截图：

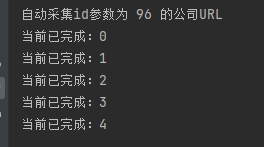


Figure :

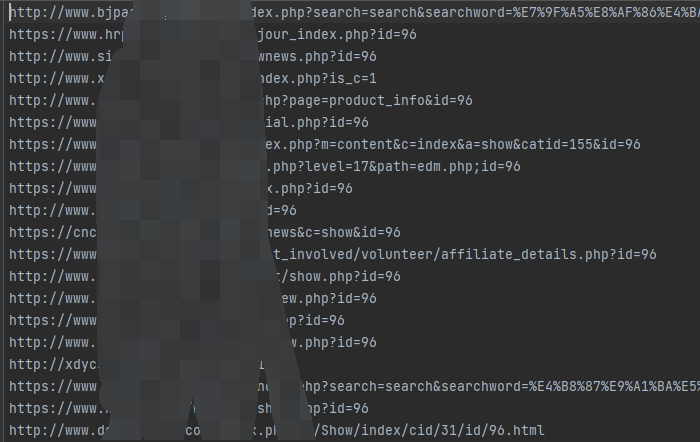


Figure :

ps：  
脚本启动第二次的时候会自动递增到url.txt上

当然了 你也可以选择把爬取到的URL批量的放到sqlmap上跑

### 漏洞盒子自提交脚本

前言：  
曾经我为了研究漏洞盒子自动提交脚本 翻遍了全网 可惜各个脚本都坚持不到最后的那个提交按钮  
  
从此我就下定决心 一定要做一个属于自己的脚本

因为是自研脚本 自己用的习惯就行 所以不好用的话 勿怪 我也才刚学python1个月  
脚本在附件

脚本内容太多 就不把源代码发出来了 请在附件下载  
  
如何使用：  
打开后会有SQL注入模板和XSS模板  
按照模板的内容写你要提交的漏洞  
  
注意：每个模板都不一样  
图片的路径一定要是绝对路径  
并且要双斜杠  
具体该如何高效的使用 请自行琢磨  
  
报错问题：  
如果遇到脚本报错 这就代表这可能自动化操作太快了  
你可以在源代码中加入sleep(xxx)  
哪里延时几秒可以自己写  
  
优势：  
可以同时提交多个漏洞

##### [演示视频 点我观看](http://www.thelm.vip/1.mp4)

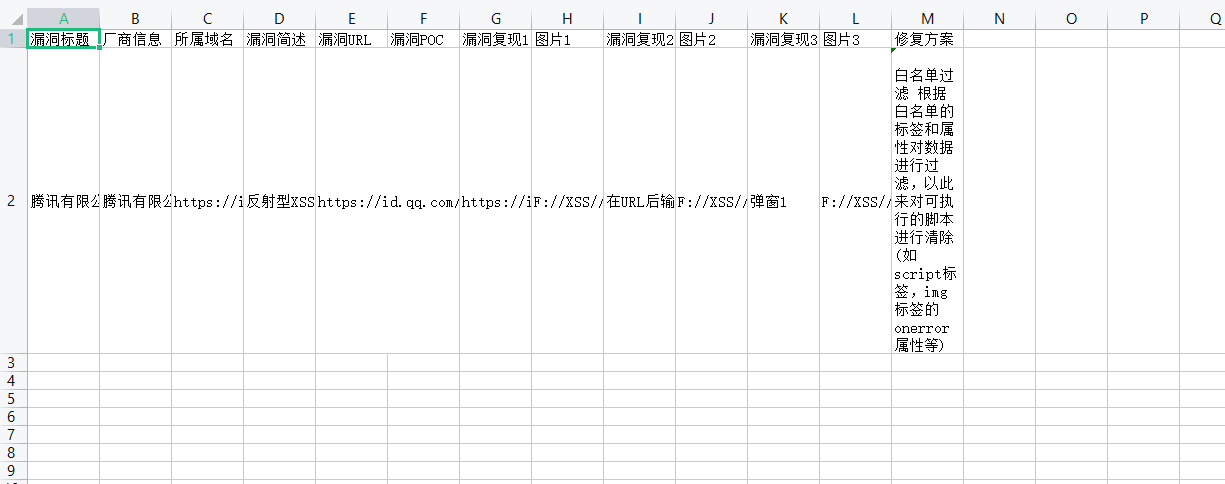


Figure :

ps：  
源代码写的通俗易懂 有兴趣的可以修改代码 制作属于自己的自动化脚本  
  
脚本不能说超级好 但可以称得上这四个字：“能用就行”

### 公益SRC：

通常需要1-10天的时间等待厂商认领及处理。  
也就是说一个漏洞 审核期间+漏洞确认期间 大概10天左右  
  
上分最好的时间点：  
上个月的25-28号-当月的20号  
  
20号以后呢？：  
可以把漏洞都攒起来 到25-28号左右一块提交 然后下个月的排行榜上绝对有你的名字  
  
复现步骤很重要 一定要多写 一个SQL注入在加一个优的报告质量 最多可以加8分  
  
一个SQL注入 良的报告质量加4分 这就不用我多说了吧   
  
漏洞简述和修复方案也很重要 有很多人在修复方案直接写一个 问开发  
这样可能就与优的报告质量无缘了 具体怎么写 参考下面的模板：

反射型XSS：  
  
漏洞简述：  
反射型XSS攻击，危害：通过引诱用户点击一个链接到目标网站的恶意链接来实施攻击  
  
修复方案：  
白名单过滤 根据白名单的标签和属性对数据进行过滤，以此来对可执行的脚本进行清除(如script标签，img标签的onerror属性等)  
  
SQL注入：  
  
漏洞简述：  
（1）攻击者未经授权可以访问数据库中的数据，盗取用户的隐私以及个人信息，造成用户的信息泄露。  
（2）可以对数据库的数据进行增加或删除操作，例如私自添加或删除管理员账号。  
（3）如果网站目录存在写入权限，可以写入网页木马。攻击者进而可以对网页进行篡改，发布一些违法信息等。  
（4）经过提权等步骤，服务器最高权限被攻击者获取。攻击者可以远程控制服务器，安装后门，得以修改或控制操作系统。  
  
修复方案：  
严格限制变量类型，比如整型变量就采用intval()函数过滤，数据库中的存储字段必须对应为int型

这些都是我在网上复制粘贴的 因为我就常挖这两个漏洞 就这两个漏洞的模板了  
如果是其他的漏洞 可以自行百度 问题不大

## 以上就是我都挖洞经验分享

我的经验也就这些了 希望小伙伴们可以少走很多弯路 学习更多知识
  
为了我们的黑客梦 加油