

SQLMAPAPI-一个被遗忘的API接口《第一章：初识SQLMAP API和命令行中调用》

原创 Gcow-悠 零维安全 4月28日

```
$ python sqlmap.py -u "http://debiandev/sqlmap/mysql/get_int.php?id=1" --batch

  H
  |
  | {1.3.4.44#dev}
  |
  | http://sqlmap.org
  |
  | V...

[!] 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 fed
eral laws. Developers assume no liability and are not responsible for any misuse or damage
caused by this program

[*] starting @ 10:44:53 /2019-04-30/

[10:44:54] [INFO] testing connection to the target URL
[10:44:54] [INFO] heuristics detected web page charset 'ascii'
[10:44:54] [INFO] checking if the target is protected by some kind of WAF/IPS
[10:44:54] [INFO] testing if the target URL content is stable
[10:44:55] [INFO] target URL content is stable
[10:44:55] [INFO] testing if GET parameter 'id' is dynamic
[10:44:55] [INFO] GET parameter 'id' appears to be dynamic
[10:44:55] [INFO] heuristic (basic) test shows that GET parameter 'id' might be injectable
(possible DBMS: 'MySQL')
```

“本文前言：

最近在写一款漏洞检测软件的时候，发现对于SQL注入的一些判断还有测试不是很精准，于是乎我就想到了本文的一个主角，SQLMAPAPI，这个API是SQLMAP官方提供的一个调用SQLMAP里面服务的一个API，以前觉得SQLMAP自己玩得挺好的了，但是当我接触了SQLMAP API后发现自己还是太年轻了。同时国内对于SQLMAP的API的一些记录不是很多或者很细，于是乎就有了这篇文章，来总结和记录一个过程。同时写的不好的话还望大佬们手下留情！（P.S. 本文为了方便理解很多东西都是用大白话去讲解的,还望各位大佬们海涵）”

本文目录：

- 1.对SQLMAP API的一个介绍
2. sqlmapapi.py的使用帮助
3. SQLMAP API的两种模式

01

对SQLMAP API的一个介绍

P.S. 介绍废话有点多各位可以酌情看

为什么要使用SQLMAP API？

有的读者就要问了，我们的-m不是可以批量检测吗？为什么还要来调用SQLMAP API呢？虽然-m参数可以批量扫描URL，但是他的一个运行方式是一个扫描完成后开始下一个任务。但是我们通过api接口，直接下发扫描任务后台就可以直接静默开始运行，无需开启一个新的命令行窗

口。这样的话对于我们进行大量测试的时候就可以批量去提交任务了。这样的话对我们的工作就可以起到很大的一个辅助。同时对于一些国外的站点的话我们在进行一些测试的时候就可以把我们的API放在我们的VPS上，这样的话就可以直接来加快我们的测试。

02

详细对SQLMAP API的真容进行一探究竟

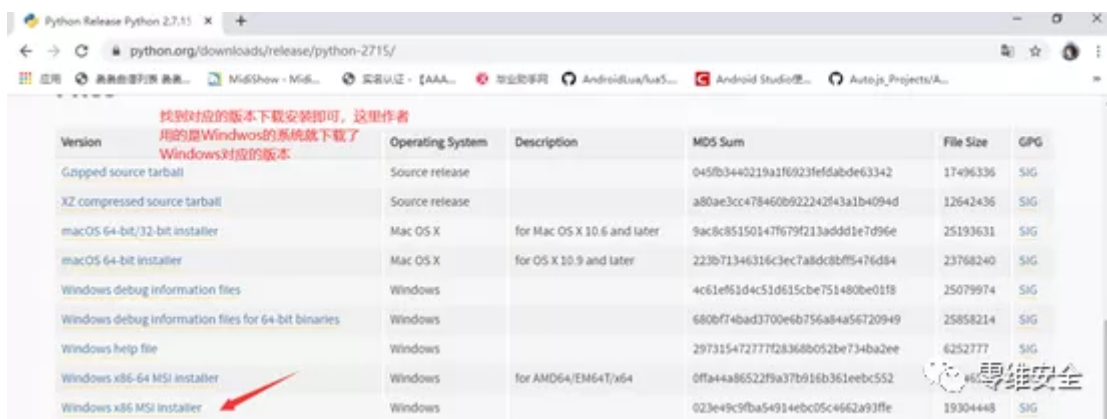
1.环境与搭建：

我们只需要下载完成Python2.7和安装配置后，再下载我们的SQLMAP里面就直接自带了API。

附上一些下载地址：

最新的Python 2.7.x 下载地址：

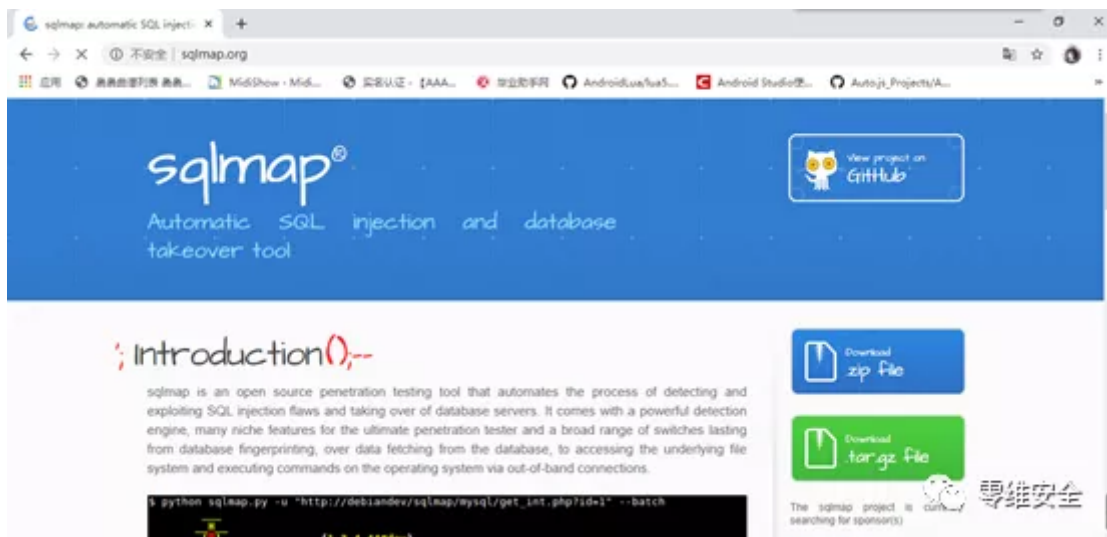
1 <https://www.python.org/downloads/release/python-2718/>



Version	Operating System	Description	MD5 Sum	File Size	GPG
Gzipped source tarball	Source release		045fb3440219a1f6923f6dabde63342	17496336	SIG
XZ compressed source tarball	Source release		a80ae3cc478460b92242f43a1b4094d	12642436	SIG
macOS 64-bit/32-bit installer	Mac OS X	for Mac OS X 10.6 and later	9ac8c85150147f679f213a0dd1e7d96e	25193631	SIG
macOS 64-bit installer	Mac OS X	for OS X 10.9 and later	223b71346316c3ec7a8dc8b7f5476d84	23768240	SIG
Windows debug information files	Windows		4c61e61d4c51d615cbe751480be01f8	25079974	SIG
Windows debug information files for 64-bit binaries	Windows		680bf74bad3700e6b756a84a56720949	25858214	SIG
Windows help file	Windows		29731547277728368b052be734ba2ee	6252777	SIG
Windows x86-64 MSI installer	Windows	for AMD64/EM64T/x64	0ffa44a86522f9a37b916b361eebc552		
Windows x86 MSI installer	Windows		023e49c3fba54914ebc05c4662a93ffe	19304448	SIG

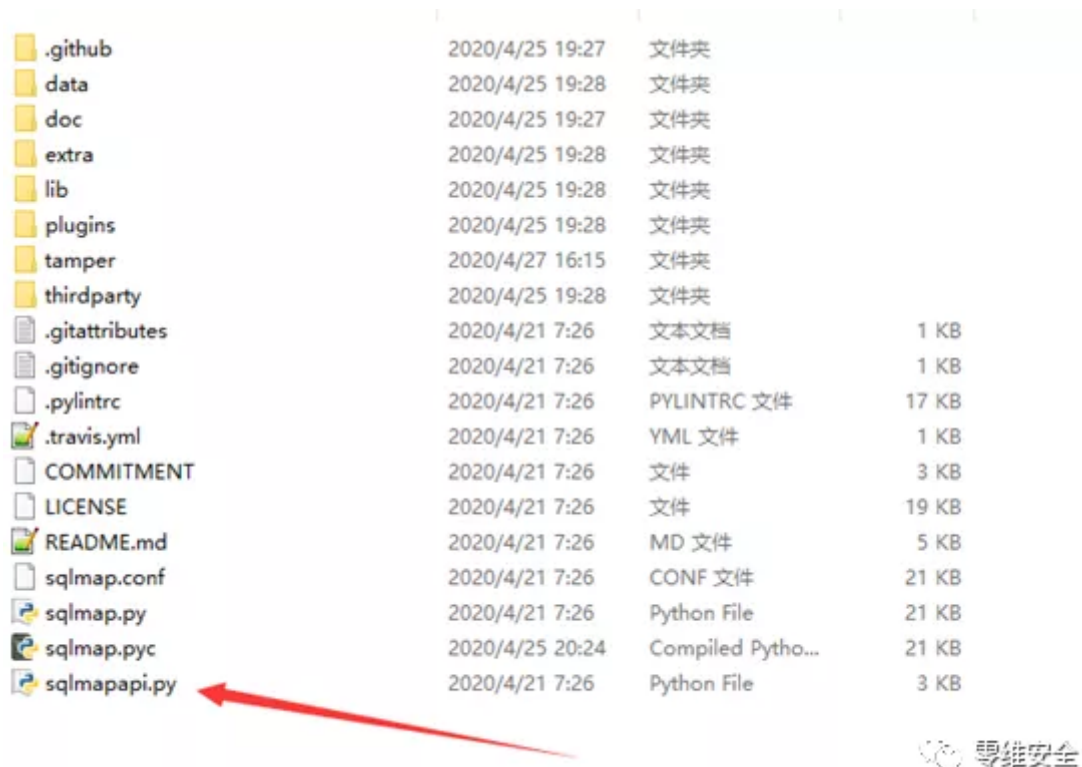
SQLMAP 下载地址：

1 <https://github.com/sqlmapproject/sqlmap/zipball/master>



2.SQLMAP API的真容

那说了那么多，到底api如何使用呢？在下载安装SQLMAP后，你会在sqlmap安装目录中找到一个 sqlmapapi.py 的文件，这个 sqlmapapi.py 文件就是sqlmap api。我们的SQLMAP API就静静的在这里，你看着他，他也在看着你。



同时我们的sqlmap api分为服务端和客户端，sqlmap api有两种模式，一种是基于HTTP协议的接口模式，一种是基于命令行的接口模式。

我们下面可以来看看SQLMAP API官方的一些帮助：

```
C:\Users\Administrator>sqlmapapi
Usage: sqlmapapi.py [options]

Options:
-h, --help            show this help message and exit
-s, --server          Run as a REST-JSON API server
-c, --client          Run as a REST-JSON API client
-H HOST, --host=HOST  Host of the REST-JSON API server (default "127.0.0.1")
-p PORT, --port=PORT  Port of the REST-JSON API server (default 8775)
--adapter=ADAPTER     Server (bottle) adapter to use (default "wsgiref")
--username=USERNAME   Basic authentication username (optional)
--password=PASSWORD   Basic authentication password (optional)
```

这里我们来看看每一个参数详细的一个介绍

- 1 Usage: sqlmapapi.py [options]
- 2 Options: -h, --help 显示帮助信息并退出
- 3 -s, --server 作为api服务端运行
- 4 -c, --client 作为api客户端运行
- 5 -H HOST, --host=HOST 指定服务端IP地址（默认IP是 "127.0.0.1"）
- 6 -p PORT, --port=PORT 指定服务端端口（默认端口8775）
- 7 --adapter=ADAPTER #服务端标准接口（默认是"wsgiref"）
- 8 --username=USERNAME #可空，设置用户名
- 9 --password=PASSWORD #可空，设置密码

3.启动API服务

1. 服务端模式：

无论是基于HTTP协议的接口模式还是基于命令行的接口模式，首先都是需要开启api服务端的。通过输入以下命令即可开启api服务端：

```
1 python sqlmapapi.py -s
```

命令执行成功后我们的返回信息如下：

```
C:\Users\Administrator>sqlmapapi -s
[11:53:03] [INFO] Running REST-JSON API server at '127.0.0.1:8775'...
[11:53:03] [INFO] Admin (secret) token: 0e8943723fcbc108ff608e413632d6d2
[11:53:03] [INFO] IPC database: 'c:\users\admini~1\appdata\local\temp\sqlmapipc-cb3i2w'
[11:53:03] [INFO] REST-JSON API server connected to IPC database
[11:53:04] [INFO] Using adapter 'wsgiref' to run bottle
```

运行的地址和我们的端口

管理者用的一个T 零维安全

其他的一些信息就是IPC数据库的位置，api服务端已经和IPC数据库连接上了，正在使用bottle 框架wsgiref标准接口。这里呢我们就不过多的去解释了

但是通过上面的这种方式开启api服务端有一个缺点，当服务端和客户端不是一台主机会连接不上，这个只是针对于我们本地调用的时候使用的，因此如果要解决这个问题，可以通过输入以下命令来开启api服务端。

```
1 python sqlmapapi.py -s -H "0.0.0.0" -p 8775
```

这样的话就可以直接指定创建一个开放的地址和端口了。

2.客户端模式（命令行接口模式）：

有的时候我们只是需要调用，而不需要我们的程序调用的时候，就可以直接用我们的客户端模式来进行连接（P.S. 我们的API服务还是要开启）

然后通过下面代码就可以连接到我们的SQLMAP API了

```
1 python sqlmapapi.py -c -H(必须大写，小写的是帮助) 地址 -p 端口
```

```
C:\Users\Administrator>sqlmapapi -c -H 127.0.0.1 -p 8775
[12:00:53] [INFO] Example client access from command line:
$ taskid=$(curl http://127.0.0.1:8775/task/new 2>1 | grep -o -I '[a-f0-9]\{16\}') && echo $taskid
$ curl -H "Content-Type: application/json" -X POST -d '{"url": "http://testphp.vulnweb.com/artists.php?artist=1"}' http://127.0.0.1:8775/scan/$taskid/start
$ curl http://127.0.0.1:8775/scan/$taskid/data
$ curl http://127.0.0.1:8775/scan/$taskid/log
[12:00:53] [INFO] Starting REST-JSON API client to 'http://127.0.0.1:8775'...
[12:00:53] [INFO] Calling 'http://127.0.0.1:8775'
[12:00:53] [INFO] Type 'help' or '?' for list of available commands
api>
```

零维安全

这样的话我们就进入了我们的命令行。

同时我们可以执行的命令有这些：

- 1 **help**显示帮助信息
- 2 **new** ARGS[我们的一些参数等等] 开启一个新的扫描任务
- 3 **use** TASKID 切换taskid
- 4 **data**获取当前任务返回的数据

- 5 **log** 获取当前任务的扫描日志
- 6 **status** 获取当前任务的扫描状态
- 7 **option OPTION** 获取当前任务的选项
- 8 **options** 获取当前任务的所有配置信息
- 9 **stop** 停止当前任务
- 10 **kill** 杀死当前任务
- 11 **list** 显示所有任务列表
- 12 **flush** 清空所有任务
- 13 **exit** 退出客户端

之后我们可以用

- 1 [例如:**new-u** “url”]这样就可以直接开始扫描新的任务

我们就可以通过这个来进行创建我们的新的任务（如下图）

```

Microsoft Windows [版本 10.0.16299.1087]
(c) 2017 Microsoft Corporation. 保留所有权利。

C:\Users\Administrator>sqlmapapi -s
[12:50:32] [INFO] Running REST-JSOM API server at '127.0.0.1:8775'...
[12:50:32] [INFO] Admin (secret): token: c7b4d6e2394c25fb7e865f1b7b3ab01
[12:50:32] [INFO] IPC database: 'c:\users\admini~1\appdata\local\temp\sqlmapipc-jl3oba'
[12:50:32] [INFO] REST-JSOM API server connected to IPC database
[12:50:32] [INFO] Using adapter 'wsiref' to run bottle
[12:51:08] [INFO] Created new task: '2625bd0537e8f75f'
[12:51:08] [INFO] (2625bd0537e8f75f) Started scan

C:\Users\Administrator>sqlmapapi -c -H 127.0.0.1 -p 8775
[12:50:58] [INFO] Example client access from command line:
$ taskid=$(curl http://127.0.0.1:8775/task/new 2>1 | g
$ curl -H "Content-Type: application/json" -X POST -d '
' http://127.0.0.1:8775/scan/$taskid/start
$ curl http://127.0.0.1:8775/scan/$taskid/data
$ curl http://127.0.0.1:8775/scan/$taskid/log
[12:50:58] [INFO] Starting REST-JSOM API client to http://127.
[12:50:58] [INFO] Calling 'http://127.0.0.1:8775'
[12:50:58] [INFO] Type 'help' or '?' for list of available com
api> new -u "127.0.0.1/index.php?id=1"
[12:51:08] [INFO] Calling 'http://127.0.0.1:8775/task/new'
[12:51:08] [INFO] New task ID is '2625bd0537e8f75f'
[12:51:08] [INFO] Calling 'http://127.0.0.1:8775/scan/2625bd05
[12:51:08] [INFO] Scanning started
api (2625bd0537e8f75f)>
  
```

并且已经切换到了我们的这个任务的ID.

(P.S. 我们的每一个任务只能是一个单独测试点，每个任务对应一个ID)

创建成功后就会这样，之后我们可以通过输入status来获取当前的一个运行情况

```

[12:51:08] [INFO] Calling 'http://127.0.0.1:8775/task/new'
[12:51:08] [INFO] New task ID is '2625bd0537e8f75f'
[12:51:08] [INFO] Calling 'http://127.0.0.1:8775/scan/2625bd0537e8f75f/start'
[12:51:08] [INFO] Scanning started
api (2625bd0537e8f75f)> status
[12:53:35] [INFO] Calling 'http://127.0.0.1:8775/scan/2625bd0537e8f75f/status'
{
  "status": "terminated",
  "returncode": 0,
  "success": true
}
  
```

这个就是我们的状态

我们的状态分为

running 正在扫描 (running的一个截图)

```

[12:54:51] [INFO] Calling 'http://127.0.0.1:8775/scan/9d1cd055925f935a/status'
{
  "status": "running",
  "returncode": null,
  "success": true
}
  
```

描完成

(P.S. SQLMAP API扫描完成后，不会进行主动推送完成信息)


```
api (9dlcd055925f935a)> new -u "127.0.0.1/index2.php?id=1"
[12:56:07] [INFO] Calling 'http://127.0.0.1:8775/task/new'
[12:56:07] [INFO] New task ID is 'c00a8fec54c0fa42'
[12:56:07] [INFO] Calling 'http://127.0.0.1:8775/scan/c00a8fec54c0fa42/start'
[12:56:07] [INFO] Scanning started
api (c00a8fec54c0fa42)> status
[12:56:09] [INFO] Calling 'http://127.0.0.1:8775/scan/c00a8fec54c0fa42/status'
```



我们的每一步，其实都是调用了http的对应的接口的。那么我们就可以通过HTTP接口来让其他的程序也可调用我们的SQLMAP API来进行注入的测试。

同时我们的api接口都是静默运行的只会有部分的调用信息在我们的命令框中

```
C:\Users\Administrator>sqlmapapi -s
[12:50:33] [INFO] Running REST-JSON API server at '127.0.0.1:8775'..
[12:50:33] [INFO] Admin (secret) token: c7b4a6e2394c25fb3be855f1b7b3db01
[12:50:33] [INFO] IPC database: 'c:\users\admini~1\appdata\local\temp\sqlmapipc-ji3oba'
[12:50:33] [INFO] REST-JSON API server connected to IPC database
[12:50:33] [INFO] Using adapter 'wsgiref' to run bottle
[12:51:08] [INFO] Created new task: '2625bd0537e8f75f'
[12:51:08] [INFO] (2625bd0537e8f75f) Started scan
[12:53:35] [INFO] (2625bd0537e8f75f) Retrieved scan status
[12:54:48] [INFO] Created new task: '9dlcd055925f935a'
[12:54:48] [INFO] (9dlcd055925f935a) Started scan
[12:54:51] [INFO] (9dlcd055925f935a) Retrieved scan status
[12:56:07] [INFO] Created new task: 'c00a8fec54c0fa42'
[12:56:07] [INFO] (c00a8fec54c0fa42) Started scan
[12:56:09] [INFO] (c00a8fec54c0fa42) Retrieved scan status
[12:56:10] [INFO] (c00a8fec54c0fa42) Retrieved scan status
[12:56:23] [INFO] Created new task: '5c39411b3052dc8f'
[12:56:23] [INFO] (5c39411b3052dc8f) Started scan
[12:56:26] [INFO] (5c39411b3052dc8f) Retrieved scan status
[12:56:27] [INFO] (5c39411b3052dc8f) Retrieved scan status
[12:56:28] [INFO] (5c39411b3052dc8f) Retrieved scan status
[12:57:26] [INFO] (5c39411b3052dc8f) Retrieved scan data and error messages
```



这里如果我们执行了多任务的话list可以来进行查看我们的所有当前的执行任务

```
api (c8346ab18320784f)> list
[16:57:30] [INFO] Calling 'http://127.0.0.1:8775/admin/list'
{
  "tasks": {
    "ccb2477a2ceb2838": "running",
    "c8346ab18320784f": "running",
    "4d6d76e86852f912": "running"
  },
  "tasks_num": 3,
  "success": true
}
api (c8346ab18320784f)> _
```



- 1 我们可以用: `use + taskid #`来进行切换任务的切换

```
api (c8346ab18320784f)> use 4d6d76e86852f912
[16:58:29] [INFO] Switching to task ID '4d6d76e86852f912'
api (4d6d76e86852f912)>
```



以上就是我们对于命令行模式中的一些使用了，其他的命令呢，各位有兴趣都可以自己去测试一下。这一篇呢也就到这里就结束了，之后下一篇我会更新基于HTTP协议的调用方式和对SQLMAP API的代码去进行分析。感兴趣的小伙伴们可以持续关注哦！

如果喜欢本文的话可以关注零维安全公众号哦，我们会不定期更新部分小白技术文章给各位新人哦！

