**HOW TO USE SQLMAP FOR BEGINNERS**

**SHARE**[0](https://neoslab.com/2018/07/16/how-to-use-sqlmap-for-beginners/)

Using SQLmap can be tricky when you are not familiar with it. This SQLmap tutorial aims to present the most important functionalities of this popular SQL injection tool in a quick and simple way. Before using SQLmap you must first get the latest release of the tool and install a Python interpreter. Most Linux distributions have python installed by default. If it’s not the case or if you are not using Linux, you will need to download and install python. Finally, you will need a vulnerable website to test. In this tutorial we are using our simulation environment (hosted on the local machine and available on port 8888).

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.

**Source:** [sqlinjection.net](http://www.sqlinjection.net/sqlmap/tutorial/)

**HOW CAN YOU USE IT?**

First, you’ll have to download it, if it’s not already installed and configured. You can visit SQLmap’s official website for checking out the latest version and [download](http://sqlmap.org/) it.

One thing to keep in mind is that SQLmap is a Python based tool, this means it will usually run on any system with python however we like Linux and specifically Ubuntu, it simply makes it easy to get stuff done. Python comes already installed in Ubuntu. To get started with SQLmap it is a matter of downloading the tool, unpacking it and running the command with the necessary options.

If you are running Microsoft Windows as your main operating system you will likely find it the most convenient and simple to run an install of Ubuntu Linux (or Kali Linux) in a virtual machine. You can then play with SQLmap, Nmap, Nikto and Openvas along with a hundred other powerful open source security tools. If you would like to perform remote scanning such as that provided by Hackertarget.com you could pay for a cheap Ubuntu based VPS from one of hundreds of providers, Linode is great for this, providing high quality and solid systems for the price.

**FEATURES**

* Full support for **MySQL, Oracle, PostgreSQL, Microsoft SQL Server, Microsoft Access, IBM DB2, SQLite, Firebird, Sybase, SAP MaxDB, HSQLDB and Informix** database management systems.
* Full support for six SQL injection techniques: **boolean-based blind, time-based blind, error-based, UNION query-based, stacked queries and out-of-band.**
* Support to **directly connect to the database** without passing via a SQL injection, by providing DBMS credentials, IP address, port and database name.
* Support to enumerate **users, password hashes, privileges, roles, databases, tables and columns.**
* Automatic recognition of password hash formats and support for **cracking them using a dictionary-based attack.**
* Support to **dump database tables** entirely, a range of entries or specific columns as per user’s choice. The user can also choose to dump only a range of characters from each column’s entry.
* Support to **search for specific database names, specific tables across all databases or specific columns across all databases’ tables**. This is useful, for instance, to identify tables containing custom application credentials where relevant columns’ names contain string like name and pass.
* Support to **download and upload any file** from the database server underlying file system when the database software is MySQL, PostgreSQL or Microsoft SQL Server.
* Support to **execute arbitrary commands and retrieve their standard output** on the database server underlying operating system when the database software is MySQL, PostgreSQL or Microsoft SQL Server.
* Support to **establish an out-of-band stateful TCP connection between the attacker machine and the database server** underlying operating system. This channel can be an interactive command prompt, a Meterpreter session or a graphical user interface (VNC) session as per user’s choice.
* Support for **database process’ user privilege escalation** via Metasploit’s Meterpreter getsystem command.

**Source:** [sqlmap.org](http://sqlmap.org/)

**DOWNLOAD**

**Step 1: Get a Linux based Operating System**

If you are going to run SQLmap on Windows with Python, make sure you have Python installed and skip down to the next step, otherwise get your Linux system fired up. Either install a Linux virtual machine (Ubuntu or Kali recommended) on Windows (Virtualbox / VMware / Parrallels) or boot up your Linux desktop.

**Step 2: SQLmap Installation**

Python is pre-installed in Ubuntu so all you need to do is download SQLmap from Sourceforge, unpack it into a directory and start your testing.

Otherwise, if you have git already installed you can grab the latest copy with the below command. You can also directly download the latest ZIP from SQLmap’s official website.

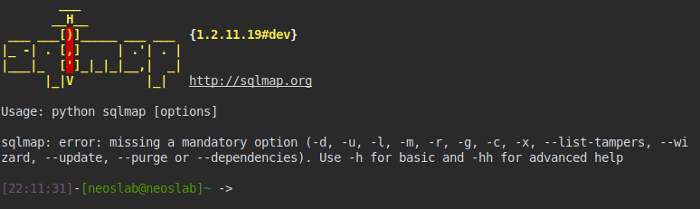
|  |  |
| --- | --- |
| $ git clone --depth 1 <https://github.com/SQLmapproject/SQLmap.git> sqlmap | |
| Cloning into 'sqlmap'... |

|  |
| --- |
| remote: Counting objects: 633, done. |
| remote: Compressing objects: 100% (602/602), done. | |

|  |  |
| --- | --- |
| remote: Total 633 (delta 170), reused 171 (delta 20), pack-reused 0 | |
| Receiving objects: 100% (633/633), 7.17 MiB | 2.44 MiB/s, done. |

|  |  |
| --- | --- |
| Resolving deltas: 100% (170/170), done. | |
| $ cd sqlmap |

|  |
| --- |
| $ python SQLmap.py |

[[](https://neoslab.com/2018/07/16/how-to-use-sqlmap-for-beginners/)click to enlarge](https://neoslab.com/2018/07/16/how-to-use-sqlmap-for-beginners/)

If you do not see something like the output above, make sure python is working (try **python -V**) to check the version and if you are in the right location. Now let’s move on this was the easy part.

To get a full list of the options available run **python sqlmap.py -h.**

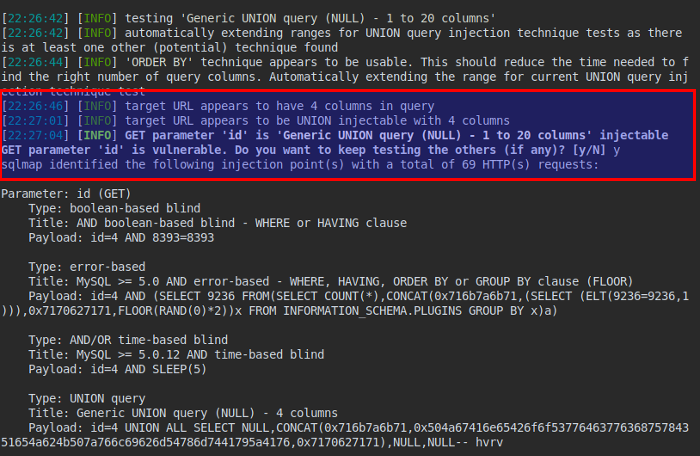
**RUNNING SQLMAP**

**Simple HTTP GET based test**

In this simple test we will use a standard **HTTP GET** based request against a URI with a parameter (?id=5). This will test different SQL injection methods against the **id** parameter.

|  |
| --- |
| $ python SQLmap.py -u '<http://mytestsite.com/page.php?id=5>' |

In the results we can see the different methods used against the parameter.

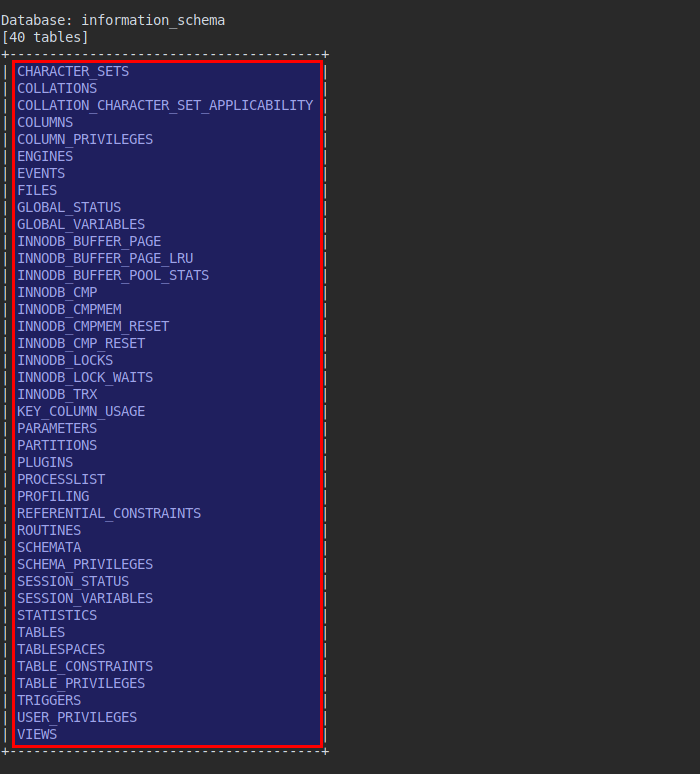
[[](https://neoslab.com/2018/07/16/how-to-use-sqlmap-for-beginners/)click to enlarge](https://neoslab.com/2018/07/16/how-to-use-sqlmap-for-beginners/)

**Retrieve the Database Tables**

SQLmap can be used to not only test but also to exploit SQL Injection, doing things such as extracting data from databases, updating tables and even **popping shells on remote hosts** if all the ducks are in line.

Let’s retrieve the tables from the database using the SQL Injection vulnerability we confirmed above. As you will see in the output below we are able to continue testing against the target without having to retest the vulnerability. SQLmap uses information it knows about the site to further exploit the target database. To retrieve data we simply add a parameter to the previous command. By adding *–tables* we are able to attempt to retrieve all the tables.

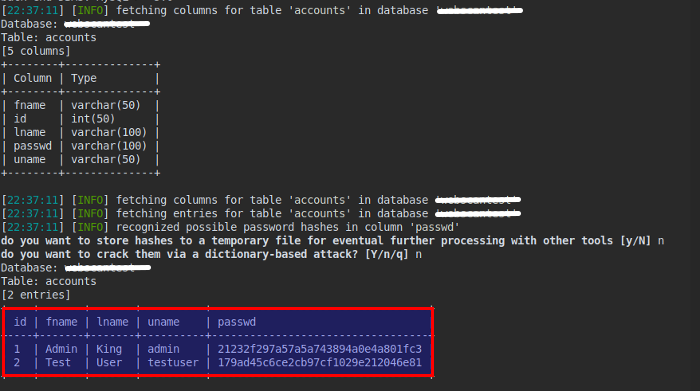
|  |
| --- |
| $ python SQLmap.py -u '<http://mytestsite.com/page.php?id=5>' --tables |

[[](https://neoslab.com/2018/07/16/how-to-use-sqlmap-for-beginners/)click to enlarge](https://neoslab.com/2018/07/16/how-to-use-sqlmap-for-beginners/)

**Dump the data**

To get data we simply extend our command. Adding -T users will focus in on the users table where we might be able to get some credentials. Adding –dump will tell SQLmap to grab all the data from the users table, first the columns will be enumerated and then the data will be dumped from the columns.

|  |
| --- |
| $ python SQLmap.py -u '<http://mytestsite.com/page.php?id=5>' --columns -D DATABASE-NAME -T TABLE-NAME --dump |

[[](https://neoslab.com/2018/07/16/how-to-use-sqlmap-for-beginners/)click to enlarge](https://neoslab.com/2018/07/16/how-to-use-sqlmap-for-beginners/)

**SQLMAP TIPS**

**Verbose Output**

When testing for SQL Injection it is often necessary to dig into the requests manually in order to determine problems with the test or to confirm or even further exploit a discovered injection. Being able to increase the verbosity of your SQLmap output will help with this testing.

By increasing the verbosity to 4 you are able to get the HTTP requests, with 5 you also see the HTTP response headers and 6 will show the full HTTP response. Obviously this can get super noisy but sometimes you need to see what is happening.

**Got Database Credentials**

This handy tip allows you to connect to the database directly and dump data such as users, databases or tables. The nice thing about this is you don’t have to remember the SQL syntax for the database or have a client installed. SQLmap will do the heavy lifting acting as a Database Client to dump data.

**DB Connection strings**

MySQL, Oracle, Microsoft SQL Server, PostgreSQL.

|  |
| --- |
| [DBMS://USER:PASSWORD@DBMS\_IP:DBMS\_PORT/DATABASE\_NAME](dbms://USER:PASSWORD@DBMS_IP:DBMS_PORT/DATABASE_NAME) |

SQLite, Microsoft Access

|  |
| --- |
| [DBMS://DATABASE\_FILEPATH](dbms://DATABASE_FILEPATH) |

**Popping Shells and More**

SQLmap has a ridiculous number of options, and is an amazing tool for becoming one with a database. Apart from popping shells on the target host, you can send requests through tor, find injections in page responses automatically, spider sites and of course perform HTTP POST based testing. These examples just scratch the surface more examples are available on the excellent github wiki page.