Web Application Testing

Overview:

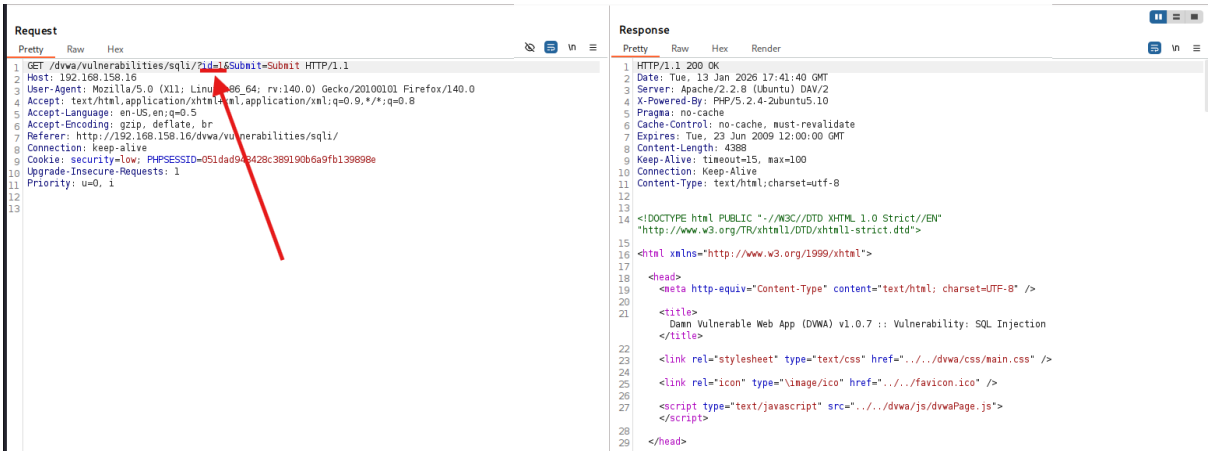
Testing a web application on a target (192.168.158.16) and confirmed several critical weaknesses that need an immediate patch , The critical vulnerabilities are SQL injection, reflected XSS, and unrestricted file uploads. Using a “Burp Suite” for manual testing and using “Ozap” for automated testing and manual verification, used various techniques and tools to exploit the SQLi for data extraction; sqlmap dumped the users table. Session cookies lacked security flags. Critical risks require parameterized queries, output encoding, and file validation

Sno	Vulnerability	Severity	Target URL
1	SQL Injection	Critical	http://192.168.74.16/dvwa/vulnerabilities/sqli/?id=1
2	XSS Reflected	High	http://192.168.74.16/dvwa/vulnerabilities/xss_r/?name=
3	File Upload RCE	Critical	http://192.168.74.16/dvwa/vulnerabilities/upload/
4	Command Injection	High	http://192.168.74.16/dvwa/vulnerabilities/exec/
5	Weak Session Management	Medium	http://192.168.74.16/dvwa/login.php

Manual Testing:

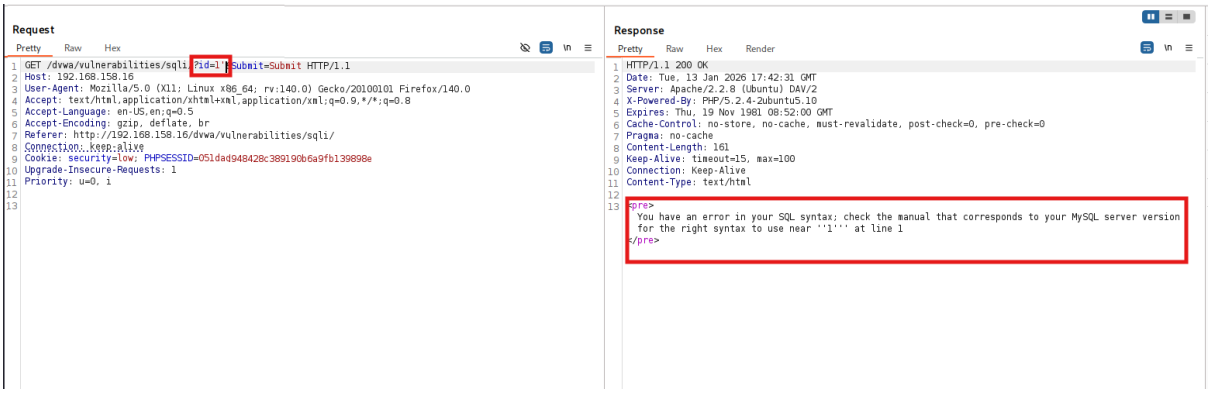
IDOR:

1. Found an IDOR vulnerability on the parameter /?id=1



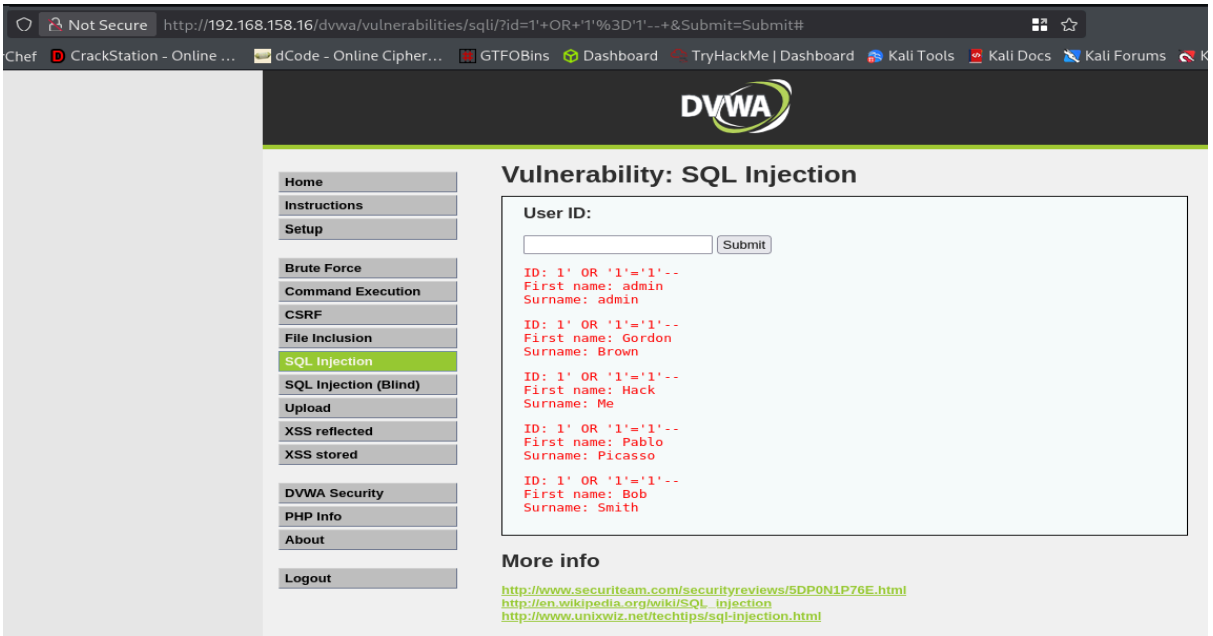
SQLi:

1. The parameter is also vulnerable to SQL vulnerable parameter /?id=1



2. The output confirmed the SQL injection

Manual injection:

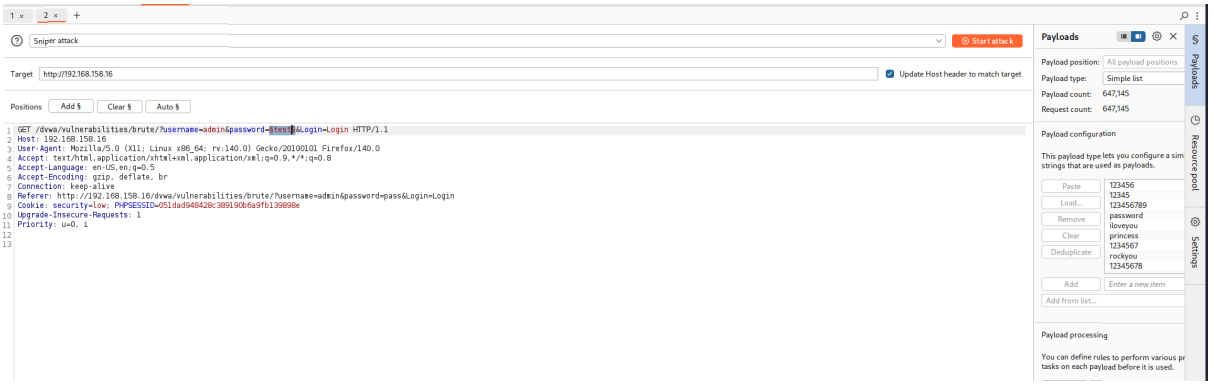


3. Using Sqlmap to get the database
Automated Injection:

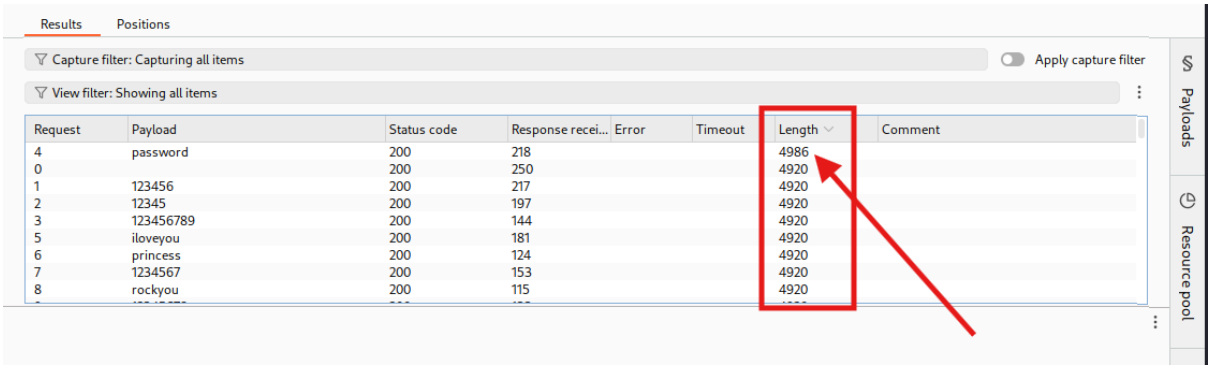
```
    Title: MySQL UNION query (NULL) - 2 columns
    Payload: id=1' UNION ALL SELECT CONCAT(0x7178787671,0x544e4451696f78436c447678735275 6c4849706f4f7570454854654e785a
---
[13:49:47] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu 8.04 (Hardy Heron)
web application technology: Apache 2.2.8, PHP 5.2.4
back-end DBMS: MySQL >= 4.1
[13:49:47] [INFO] fetching database names
available databases [7]:
[*] dvwa
[*] information_schema
[*] metasploit
[*] mysql
[*] owasp10
[*] tikiwiki
[*] tikiwiki195

[13:49:47] [INFO] fetched data logged to text files under '/home/kali/.local/share/sqlmap/output/192.168.158.16'

[*] ending @ 13:49:47 /2026-01-13/
```

No Rate Limiting:

1. Bruteforcing the admins password using intruder

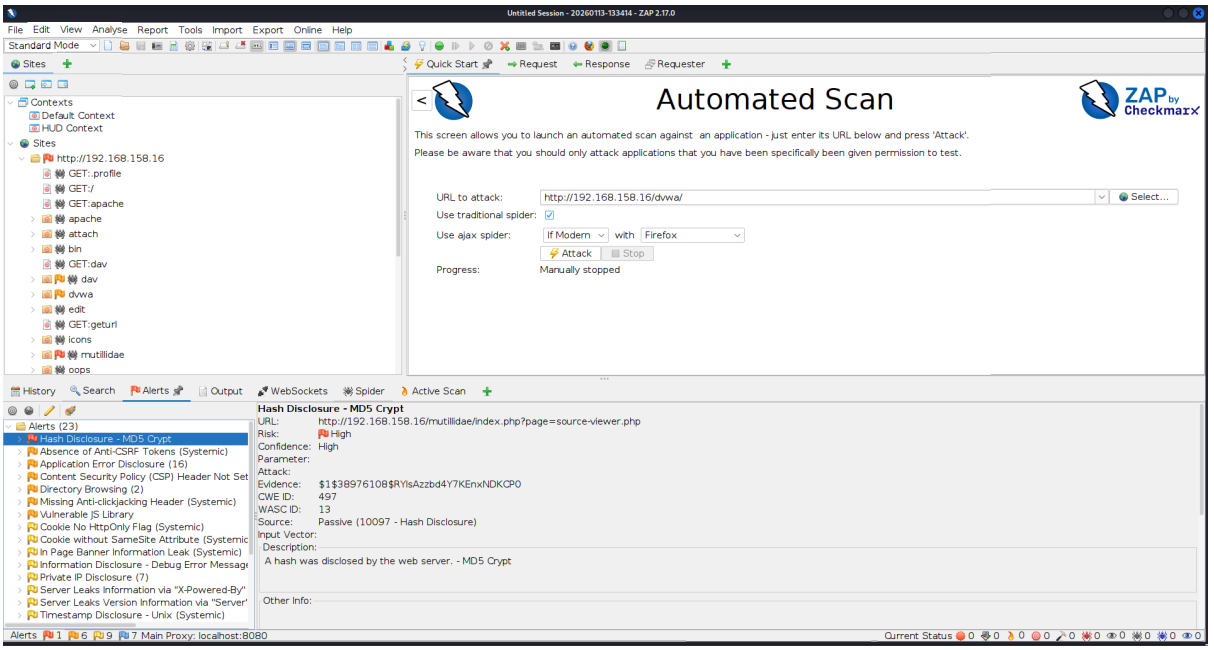


2. Found the admin users password by filtering the length

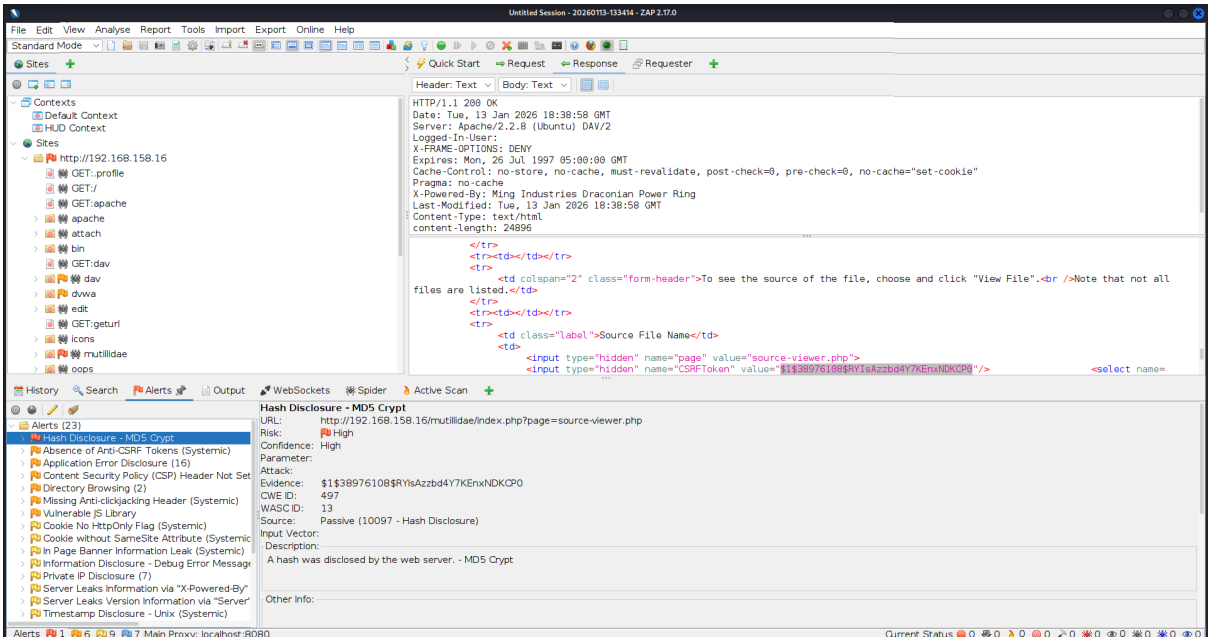


Automated Testing:

1. Using Ozap tool to test the web application



2. Findings by automated scanner



Summary:

Upon testing the web application there are several critical and high level vulnerabilities which need an immediate patch. Used manual and automated method for finding the vulnerability for manual the tool Burp Suite had been used , for automated finding Ozap tool had been used.