Attack Scenarios and Mitigation Notes

Web Application Security:

Objective: Identify and exploit OWASP Top 10 vulnerabilities in a controlled lab environment.

1. SQL Injection

- Scenario: Injected ' OR ‘ 1'='1 into DVWA login form to bypass authentication and extract credentials using SQLmap.

- Mitigation:

  - Use parameterized queries

  - Validate and sanitize user input

  - Disable detailed SQL error messages

2. Cross-Site Scripting (XSS)

- Scenario: Injected <script> payload into comment field to hijack session using BeEF.

- Mitigation:

  - Sanitize input and encode output

  - Implement Content Security Policy (CSP)

  - Use secure frameworks with built-in XSS protection

3. Cross-Site Request Forgery (CSRF)

- Scenario: Created auto-submitting HTML form to change victim’s password in DVWA.

- Mitigation:

  - Use anti-CSRF tokens

  - Validate Origin and Referer headers

  - Set SameSite attribute on cookies

4. File Inclusion (LFI/RFI)

- Scenario: Accessed /etc/passwd using ?page=../../../../etc/passwd in DVWA.
- Mitigation:
  - Whitelist allowed files
  - Disable remote file inclusion
  - Validate file paths securely

5. Command Injection
- Scenario: Injected ; cat /etc/passwd and executed reverse shell via Netcat.
- Mitigation:
  - Use safe APIs
  - Sanitize input
  - Run applications with least privilege

6. Broken Authentication
- Scenario: Bypassed login using default credentials and predicted session IDs.
- Mitigation:
  - Enforce strong password policies
  - Regenerate session IDs after login
  - Implement multi-factor authentication

7. Insecure Direct Object Reference (IDOR)
- Scenario: Accessed another user's invoice via URL manipulation (/invoice?id=102).
- Mitigation:
  - Implement access control checks
  - Use indirect references
  - Log and monitor access attempts

Advanced Web Application Security:

Objective: Identify and exploit advanced web application vulnerabilities.

8. Burp Suite Advanced

- Scenario: Intercepted and modified HTTP requests using Repeater, Intruder, and Decoder.
- Mitigation:
  - Use secure session management
  - Validate all client-side inputs server-side
  - Monitor for abnormal request patterns

9. Directory Traversal

- Scenario: Accessed Apache log files using ../../../../var/log/apache2/access.log.
- Mitigation:
  - Restrict file path access
  - Use secure file handling functions
  - Disable directory listing

10. File Upload Vulnerability

- Scenario: Bypassed file type restrictions and uploaded reverse shell.
- Mitigation:
  - Validate MIME types and file extensions
  - Store files outside web root
  - Rename uploaded files and scan for malware

11. Web Server Misconfiguration

- Scenario: Modified HTTP headers in Apache config to expose internal details.
- Mitigation:
  - Harden server configurations
  - Remove default files and directories
  - Disable unnecessary modules

12. Broken Access Control

- Scenario: Escalated privileges via IDOR and accessed admin functionality.

- Mitigation:

  - Enforce role-based access control

  - Validate user permissions server-side

  - Log access violations

13. Vulnerable Web App Exploitation

- Scenario: Exploited DVWA, bWAPP, and Juice Shop vulnerabilities and documented findings.

- Mitigation:

  - Regularly update applications

  - Apply security patches

  - Conduct periodic vulnerability assessments

Exploitation & System Security:

Objective: Perform penetration testing workflow and exploit vulnerable systems.

14. Penetration Testing Methodology

- Workflow:

  - Information Gathering → Exploitation → Post-Exploitation → Reporting

15. Exploitation of Services & Software

- Scenario: Used Metasploit to exploit vsftpd 2.3.4 and EternalBlue on Metasploitable2.

- Mitigation:

  - Patch vulnerable services

  - Disable unused ports

  - Monitor network traffic

16. Reverse Shell

- Scenario: Gained shell access using Netcat and MSFvenom payloads.

- Mitigation:

  - Use egress filtering

  - Monitor outbound connections

  - Restrict shell execution privileges

17. Privilege Escalation

- Scenario: Escalated privileges using Dirty COW kernel exploit and SUID binaries.

- Mitigation:

  - Patch kernel vulnerabilities

  - Audit SUID binaries

  - Use least privilege principle

18. Metasploit Framework

- Scenario: Exploited vulnerable services and software in Metasploitable2 using Metasploit modules.

- Mitigation:

  - Update Metasploit and system packages

  - Restrict access to vulnerable systems

  - Implement intrusion detection systems