Web Application Pentesting Roadmap

Phase 1: Pre-engagement

- **Define Scope:**
  - Clearly define the scope of the web application penetration test, including specific URLs, features, and functionalities.
- **Legal and Compliance:**
  - Ensure compliance with legal and ethical standards. Obtain written permission from the website owner or organization.
- **Gather Information:**
  - Collect information about the web application, including its purpose, technologies used, and any publicly available information. Review any relevant documentation.

Phase 2: Reconnaissance

- **Target Analysis:**
  - Identify the target web application and understand its technology stack, server infrastructure, and potential attack surfaces.
- **Domain and Subdomain Enumeration:**
  - Enumerate domain names and subdomains associated with the web application.
- **Network and Infrastructure Discovery:**
  - Identify IP addresses, web servers, and network infrastructure related to the web application.

Phase 3: Mapping

- **URL Enumeration:**
  - Enumerate and map the web application's URLs, including hidden or less-accessible pages.
- **Site Map Creation:**
  - Develop a comprehensive site map that outlines the structure and functionality of the web application.
- **Technology Stack Identification:**
  - Identify the technologies, frameworks, and CMS platforms used by the web application.

Phase 4: Vulnerability Analysis

- **Automated Scanning:**
  - Utilize automated tools for vulnerability scanning, focusing on common vulnerabilities such as SQL injection, cross-site scripting (XSS), and security misconfigurations.
- **Manual Testing:**
  - Conduct manual testing to identify vulnerabilities that automated tools may miss. Pay attention to business logic flaws, authentication issues, and authorization weaknesses.
- **API Security Testing:**

    - If applicable, test the security of APIs used by the web application, including authentication, authorization, and data integrity.

Phase 5: Exploitation

- **Authentication Bypass:**
    - Attempt to bypass authentication mechanisms, including password-based and token-based systems.
- **Injection Attacks:**
    - Test for injection vulnerabilities, such as SQL injection, NoSQL injection, and command injection.
- **Cross-Site Scripting (XSS):**
    - Exploit and demonstrate the impact of XSS vulnerabilities, including stored and reflected XSS.

Phase 6: Post-exploitation

- **Session Management:**
    - Assess the web application's session management mechanisms for weaknesses and potential session hijacking risks.
- **Sensitive Data Exposure:**
    - Check for sensitive data exposure issues, such as the insecure transmission or storage of personally identifiable information (PII).
- **Backdoor Testing:**
    - Test for the presence of backdoors or hidden functionalities that could pose a security risk.

Phase 7: Reporting

- **Document Findings:**
    - Compile a detailed report outlining vulnerabilities, their severity, and potential impact.
- **Risk Prioritization:**
    - Prioritize findings based on risk and potential impact, providing recommendations for remediation.
- **Mitigation Strategies:**
    - Offer clear and actionable mitigation strategies to address identified vulnerabilities.

Phase 8: Debriefing

- **Client Debrief:**
    - Present findings and recommendations to the client. Discuss any additional insights gained during the testing.
- **Lessons Learned:**
    - Conduct an internal review to identify lessons learned and improve future testing processes.