Web Application Penetration Testing Methodology

Our web application security testing methodology (which includes manual + automated assessments) is designed to cover both traditional and modern web applications across various architectures. The methodology ensures comprehensive coverage of the OWASP Top 10 risks and beyond. Below is the methodology that is used for a comprehensive security review of web applications: -

Application Overview/Pre-Requisites

As a foundational step, we take an overview of the application – core functionalities, business logic, use cases, and integration points. This includes reviewing the user/data setup, inter-tenant scenarios and data required to initiate penetration testing. Gaining this context is critical to understanding the intended behaviour of the application, which in turn enables accurate identification of security risks that may arise from misuse, abuse, or misconfigurations.

Threat Modelling

Before initiating the actual assessment, a structured threat modelling exercise is conducted to map out potential attack vectors across the application landscape. This includes identifying key assets, trust boundaries, entry and exit points, data sensitivity levels, third-party integrations and user roles. The model considers Role-Based Access Control (RBAC) configurations to evaluate whether access permissions are properly enforced across different privilege levels. The objective is to anticipate exploitation paths and guide the assessment toward both technical vulnerabilities and business logic flaws that are specific to the application's architecture and usage patterns.

Security Control and Test Cases

1. **Authentication & Session Management -** This category focuses on verifying that authentication and session handling mechanisms including Single Sign-On (SSO) implementations and federated identity integrations are implemented correctly. It includes testing for lack of proper validation of SAML/OAuth/OpenID Connect assertions, signature validations, replay attacks, no authentication to sensitive resources etc.
2. **Access Control & Authorization** - This category ensures that users can only access resources and functionalities permitted to their roles or privileges. Common issues include insecure direct object references (IDOR), horizontal or vertical or diagonal privilege escalation, bypassing access control checks via parameter tampering, and bypass of business logic-level restrictions.
3. **Input Validation & Injection -** This category includes detection of improperly sanitized input leading to injection flaws. This covers all forms of injection attacks like SQL, NoSQL, Cross Site Scripting (XSS), remote code execution, XML, JSON, template or expression

language injections etc. This would also cover lack of output encoding leading to such attacks.

4. **Security Misconfigurations & Error Handling -** This category evaluates the configuration and deployment aspects of the web application. From security response headers to debug mode, verbose errors, server metafiles, repositories accessible to end users or unnecessary services being enabled on the server etc.
5. **SSL/TLS & Communication Security** - This category ensures that communication channels between the client and server are secure. It includes checks for enforcement of HTTPS across all pages, weak or deprecated TLS protocols, and use of weak ciphers, invalid SSL certificates, and transmission of sensitive information over non-secure channels.
6. **File Upload** – This category focuses on risks associated with unrestricted file uploads, file uploads leading to remote code execution, improper storage locations allowing direct access, and insufficient validation of uploaded files containing embedded scripts or malware.

The list provided is not limited to but represents core test cases, but the high level methodology varies based on the logic/business use case of the services being tested.

Vulnerability Assessment

On the basis of resource attributes and control categories, a thorough vulnerability assessment is performed using human intelligence with the help of supportive semi-automatic and automatic tools. This detects the vulnerabilities residing in the applications, thus giving the actionable item list from application security standpoint.

Automated Scans

A comprehensive automated scan (excluding Denial of Service, Buffer Overflow & Brute-force) is performed on the application functionalities to uncover any vulnerabilities along with the human-intelligence driven manual penetration testing.

Mitigation Strategies

Based on the identified vulnerabilities, weaknesses, and overall risk posture—along with the system architecture and industry best practices—a comprehensive mitigation plan is formulated. This includes a set of actionable, prioritized recommendations outlining the security measures required to effectively harden the environment.

Actionable Report with Zero False Positives

A key deliverable of the assessment is a highly actionable, well-structured report designed to drive immediate remediation. The report is curated to maintain zero false positives and includes the following critical components: -

- Executive Summary
- Description of Discovered Vulnerabilities

- Risk Rating (curated after business impact assessment and industry security standards like CVSS/CWE/CVE)
- Evidence of Vulnerabilities (screenshots, HTTP traffic, vulnerable parameter, exploit vector, tool results, reproduction steps etc.)
- Exploit Evidence of Vulnerabilities (if required)
- Mitigation Strategies and Defence Approaches (catered to help Developers)
- Report Readout and Guidance

Tools

Blueinfy uses its own tools along with open source tools and products during the assessment process. Blueinfy has its own tools and utilities for performing manual penetration testing. Some of these tools are available at https://www.blueinfy.com/tools.html

