Web Application Security Testing

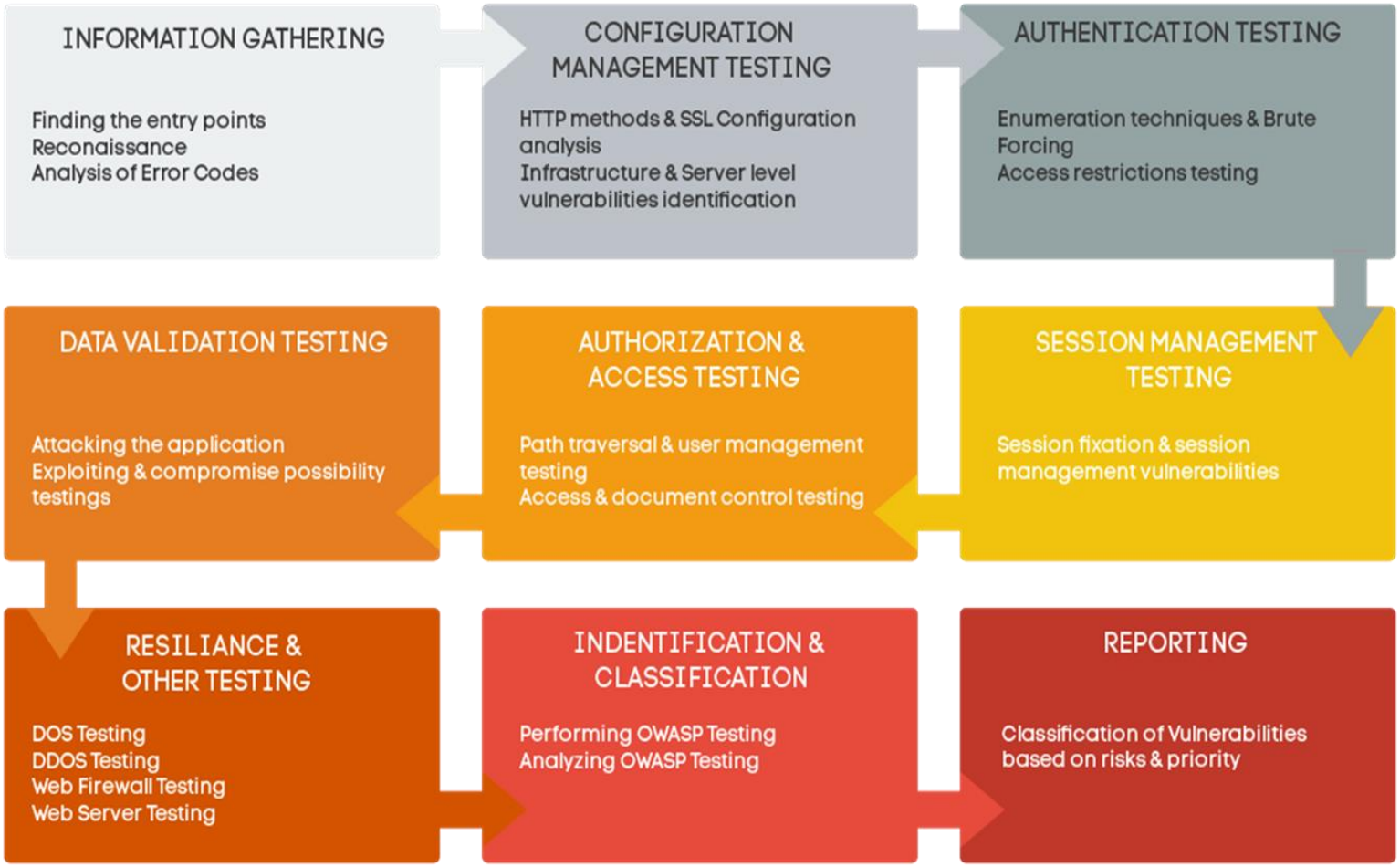
High Level Process

Sample Scope

Generic customer Scope

Area	Description
Functional Understanding of Application Module in scope	Our team will understand functionality and data flow for application in scope and map the testing requirements to control objectives.
Web Application Security Testing (Manual and Automated) as per OWASP guidelines.	We will carry out manual testing ,black box & gray box, to address scripting land application coding level issues for all applicable OWASP security test cases. We will also run automated scans for application security testing and APIs will be covered for its logic
Repot Generation and Evidence Submission	We will submit Executive summary report signed by CISA and detailed technical level findings report along with proof of concept/evidences of reported vulnerabilities.

Web Application Security Testing Methodology



Test Cases

- **Injection – (SQL, OS, HTML Injection):**

Injection flaws, such as SQL & OS injection occur when untrusted data is sent to an interpreter as part of a command or query. The attacker's hostile data can trick the interpreter into executing unintended commands or accessing unauthorized data.

- **Broken Authentication and Session Management – (Session & Cookie Management, Brute force)**

Application functions related to authentication and session management are often not implemented correctly, allowing attackers to compromise passwords, keys, session tokens, or exploit other implementation flaws to assume other users' identities.

- **Cross-Site Scripting (XSS) – (Reflected & Stored XSS)**

XSS flaws occur whenever an application takes untrusted data and sends it to a web browser without proper validation or escaping. XSS allows attackers to execute scripts in the victim's browser which can hijack user sessions, deface web sites, or redirect the user to malicious sites.

Test Cases

- **Insecure Direct Object References – (Directory Traversal)**

A direct object reference occurs when a developer exposes a reference to an internal implementation object, such as a file, directory, or database key. Without an access control check or other protection, attackers can manipulate these references to access unauthorized data.

- **Security Misconfiguration – (File Upload, Application Error Handling)**

Good security requires having a secure configuration defined and deployed for the application, frameworks, application server, web server, database server, and platform. All these settings should be defined, implemented, and maintained as many are not shipped with secure defaults. This includes keeping all software up to date.

- **Sensitive Data Exposure –**

Many web applications do not properly protect sensitive data, such as credit cards, tax ids, and authentication credentials. Attackers may steal or modify such weakly protected data to conduct identity theft, credit card fraud, or other crimes. Sensitive data deserves extra protection such as encryption at rest or in transit, as well as special precautions when exchanged with the browser

Test Cases

- **Missing Function Level Access Control .**

Virtually all web applications verify function level access rights before making that functionality visible in the UI. However, applications need to perform the same access control checks on the server when each function is accessed. If requests are not verified, attackers will be able to forge requests in order to access unauthorized functionality.

- **Cross-Site Request Forgery (CSRF)**

A CSRF attack forces a logged-on victim's browser to send a forged HTTP request, including the victim's session cookie and any other automatically included authentication information, to a vulnerable web application. This allows the attacker to force the victim's browser to generate requests the vulnerable application thinks are legitimate requests from the victim.

- **Using Components with Known Vulnerabilities – (Using automated vulnerability scanners)**

Vulnerable components, such as libraries, frameworks, and other software modules almost always run with full privilege. So, if exploited, they can cause serious data loss or server takeover. Applications using these vulnerable components may undermine their defenses and enable a range of possible attacks and impacts.

Test Cases

- **Non validated Redirects and Forwards – (Arbitrary Redirection, Click Jacking)**

Web applications frequently redirect and forward users to other pages and websites, and use untrusted data to determine the destination pages. Without proper validation, attackers can redirect victims to phishing or malware sites, or use forwards to access unauthorized pages

VMR IRM Team

VMR IRM team comprises of the best in the industry experts qualified to understand and deliver your needs.

• CRISC (Certified in Risk and Information Systems Control) • CISA (Certified Information Systems Auditor) • CISM (Certified Information Security Manager) • RHCE(Red hat certified Engineer)	• CEH (Certified Ethical Hacker) • CCNA (Cisco Certified Network Associate) • MCSE (Microsoft Cert. Systems Engineer) • ISO 27001 Lead Auditor • ITIL Manager and Expert • ISO 20001 LA • BS25999 LA • ISO 9001 LA