

|  |
| --- |
| **Web Application penetration Security Testing**  **Project Plan**  **To**  [THE Ci GROUP](https://www.thecigroup.com.au/) |
| **13/02/2023** |

**Table of Contents**

1. [Executive Summary](#_bookmark0) [3](#_bookmark0)
   1. [Introduction](#_bookmark1) [3](#_bookmark1)
2. [Scope and Objectives](#_bookmark2) [3](#_bookmark2)
   1. [Scope Inclusions](#_bookmark3) [3](#_bookmark3)
   2. [Scope Exclusions](#_bookmark4) [3](#_bookmark4)
   3. [Objectives](#_bookmark5) [3](#_bookmark5)
3. [Execution Process for Web Application Security Assessment](#_bookmark6) [4](#_bookmark6)
   1. [Web Application Security Assessment Methodology](#_bookmark7) [4](#_bookmark7)
   2. [Entry Criteria](#_bookmark8) [5](#_bookmark8)
   3. [Effort estimate for Web Application Security Assessment](#_bookmark9) [6](#_bookmark9)
   4. [Technical Tools and Utilities](#_bookmark10) [10](#_bookmark10)
   5. [Roles and Responsibilities](#_bookmark11) [10](#_bookmark11)
4. [Assumptions and Dependencies](#_bookmark12) [12](#_bookmark12)
   1. [Assumptions](#_bookmark13) [12](#_bookmark13)
   2. [Dependencies](#_bookmark14) [12](#_bookmark14)
5. [Communication and Reporting](#_bookmark15) [13](#_bookmark15)

# Executive Summary

# Introduction

This document includes the plan for Security testing of the web Application penetration testing. It describes the test process, procedures, methodology, and configuration that will be used for the Security testing activity to achieve the defined test objectives and produce the contracted deliverable.

# Scope and Objectives

It is important to summarize the scope of the services that AsquareIT Team will be providing to within the context of this document. The sections below outline the key service inclusions and exclusions.

# Scope Inclusions

The following activities are in the scope of this engagement:

▶ Security Assessment of ’s web Applications is included if any to understand its susceptibility to tests listed in section 3.3.1

▶ Web Application Security Assessment on the below URL’s List of URL’s –1

# Scope Exclusions

The following activities are not in scope of this engagement:

▶ Social Engineering

▶ Accreditation

▶ Retesting infrastructure after applying security fixes

▶ Regression

▶ Source Code Review

▶ Performance Testing

▶ Functional Testing

# Objectives

The key objectives of this work are summarized below:

▶ Conduct security assessment to determine presence of any vulnerabilities in the given Web Application

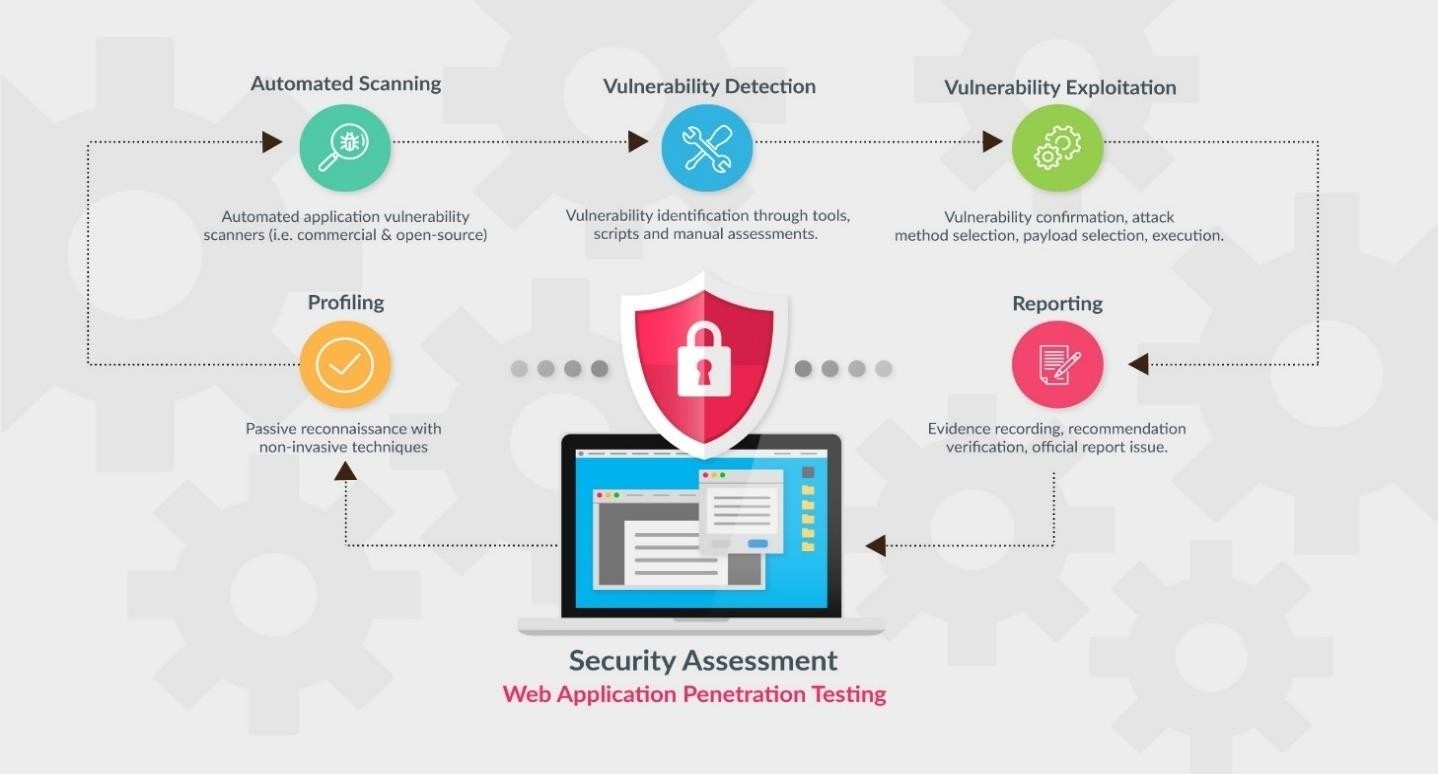
▶ Suggest recommendations to mitigate vulnerabilities discovered

▶ Exploiting the vulnerabilities wherever applicable

# Execution Process for Web Application Security Assessment

AsquareIT plans to conduct ’s Web Application at Asquare STM Lab. Sections below elucidate AsqareIt approach to the security assessment engagement.

# Web Application Security Assessment Methodology



# Entry Criteria

* Target URL
* Whitelisting IP 14.143.79.206 (We request that you whitelist our Security test lab IP addresses in your IPS device, such that malicious activity it detects from us will be logged but our IP address will never be blocked)
* User Roles and Credentials (at least 2 credentials for each role)

|  |  |  |
| --- | --- | --- |
| **Stage 1** | **Profiling** | * In this stage, profiling of the target web application is performed by identifying user entry points. * identifying roles with varying trust levels and determine the data flow path with indication on privilege boundaries. |
| **Stage 2** | **Automated Scanning** | * Automated application vulnerability scanners (i.e. commercial and open-source) are used to scan for application specific vulnerabilities covering all OWASP, WASC and SANS references. |
| **Stage 3** | **Vulnerability Detection** | * This phase involved a complete hybrid approach of identifying the web application security vulnerabilities with automated tools and scripts along with manual assessment to eliminate false positives and negatives. * Manual assessment uses various vulnerability databases to identify vulnerabilities that were missed during automated scans in addition to security verification of business logic flaws, broken access controls and more. |
| **Stage 4** | **Vulnerability Exploitation** | * The primary focus in this phase is on using manual security testing techniques to exploit the systems, which included several exploits to assess the application hardening measures, cryptography issues, authentication & authorization controls, session management module, business logic flaws and various validation measures. * Attack scenarios for production environment will use a combination of exploit payloads in strict accordance with agreed rules of engagement. |
| **Stage 5** | **Reporting** | * All exploitable security vulnerabilities in the target web application are recorded with associated CVSS v3 based scores are reported to the client. * The identified security vulnerability is thoroughly assessed and reported along with appropriate recommendation or mitigation measures. |

# Effort estimate for Web Application Security Assessment

Table below depicts the execution strategy applicable for Web Application Security Assessment for

**Web Application Security Assessment Resource Allocation& Cost**

|  |  |  |
| --- | --- | --- |
| **S. No** | **Activity** | **Duration (in business days)** |
| 1 | Information gathering/Profiling | **1 Day** |
| 2 | Planning & Analysis/ Automated Scanning | **2 Days** |
| 3 | Vulnerability Detection | **5 Days** |
| 4 | Exploitation of identified vulnerabilities | **30 Days** |
| 5 | Reporting | **2 Days** |
| 6 | Re-assessment | **5 Days** |
| **Total Number of days** | | **40Days** |

# Security Test Cases for

All applicable tests would be executed against the Web Application and detailed results with sufficient evidences and mitigations would be provided to

|  |  |
| --- | --- |
| **Web Application Checklist for** | |
| **S.No** | **Test Name** |
|  | **Information Gathering** |
| 1 | Conduct Search Engine Discovery and Reconnaissance for Information Leakage |
| 2 | Fingerprint Web Server |
| 3 | Review Webserver Metafiles for Information Leakage |
| 4 | Enumerate Applications on Webserver |
| 5 | Review Webpage Comments and Metadata for Information Leakage |
| 6 | Identify application entry points |
| 7 | Map execution paths through application |
| 8 | Fingerprint Web Application Framework |
| 9 | Fingerprint Web Application |
| 10 | Map Application Architecture |
|  | **Configuration and Deploy Management Testing** |
| 11 | Test Network/Infrastructure Configuration |
| 12 | Test Application Platform Configuration |

|  |  |
| --- | --- |
| 13 | Test File Extensions Handling for Sensitive Information |
| 14 | Backup and Unreferenced Files for Sensitive Information |
| 15 | Enumerate Infrastructure and Application Admin Interfaces |
| 16 | Test HTTP Methods |
| 17 | Test HTTP Strict Transport Security |
| 18 | Test RIA cross domain policy |
|  | **Identity Management Testing** |
| 19 | Test Role Definitions |
| 20 | Test User Registration Process |
| 21 | Test Account Provisioning Process |
| 22 | Testing for Account Enumeration and Guessable User Account |
| 23 | Testing for Weak or unenforced username policy |
| 24 | Test Permissions of Guest/Training Accounts |
| 25 | Test Account Suspension/Resumption Process |
|  | **Authentication Testing** |
| 26 | Testing for Credentials Transported over an Encrypted Channel |
| 27 | Testing for default credentials |
| 28 | Testing for Weak lock out mechanism |
| 29 | Testing for bypassing authentication schema |
| 30 | Test remember password functionality |
| 31 | Testing for Browser cache weakness |
| 32 | Testing for Weak password policy |
| 33 | Testing for Weak security question/answer |
| 34 | Testing for weak password change or reset functionalities |
| 35 | Testing for Weaker authentication in alternative channel |
|  | **Authorization Testing** |
| 36 | Testing Directory traversal/file include |
| 37 | Testing for bypassing authorization schema |
| 38 | Testing for Privilege Escalation |
| 39 | Testing for Insecure Direct Object References |
|  | **Session Management Testing** |
| 40 | Testing for Bypassing Session Management Schema |
| 41 | Testing for Cookies attributes |

|  |  |
| --- | --- |
| 42 | Testing for Session Fixation |
| 43 | Testing for Exposed Session Variables |
| 44 | Testing for Cross Site Request Forgery |
| 45 | Testing for logout functionality |
| 46 | Test Session Timeout |
| 47 | Testing for Session puzzling |
|  | **Data Validation Testing** |
| 48 | Testing for Reflected Cross Site Scripting |
| 49 | Testing for Stored Cross Site Scripting |
| 50 | Testing for HTTP Verb Tampering |
| 51 | Testing for HTTP Parameter pollution |
| 52 | Testing for SQL Injection |
| 53 | Oracle Testing |
| 54 | MySQL Testing |
| 55 | SQL Server Testing |
| 56 | Testing PostgreSQL |
| 57 | MS Access Testing |
| 58 | Testing for NoSQL injection |
| 59 | Testing for LDAP Injection |
| 60 | Testing for ORM Injection |
| 61 | Testing for XML Injection |
| 62 | Testing for SSI Injection |
| 63 | Testing for XPath Injection |
| 64 | IMAP/SMTP Injection |
| 65 | Testing for Code Injection |
| 66 | Testing for Local File Inclusion |
| 67 | Testing for Remote File Inclusion |
| 68 | Testing for Command Injection |
| 69 | Testing for Buffer overflow |
| 70 | Testing for Heap overflow |
| 71 | Testing for Stack overflow |
| 72 | Testing for Format string |
| 73 | Testing for incubated vulnerabilities |

|  |  |
| --- | --- |
| 74 | Testing for HTTP Splitting/Smuggling |
|  | **Error Handling** |
| 75 | Analysis of Error Codes |
| 76 | Analysis of Stack Traces |
|  | **Cryptography** |
| 77 | Testing for Weak SSL/TSL Ciphers, Insufficient Transport Layer Protection |
| 78 | Testing for Padding Oracle |
| 79 | Testing for Sensitive information sent via unencrypted channels |
|  | **Business Logic Testing** |
| 80 | Test Business Logic Data Validation |
| 81 | Test Ability to Forge Requests |
| 82 | Test Integrity Checks |
| 83 | Test for Process Timing |
| 84 | Test Number of Times a Function Can be Used Limits |
| 85 | Testing for the Circumvention of Workflows |
| 86 | Test Defenses Against Application Misuse |
| 87 | Test Upload of Unexpected File Types |
| 88 | Test Upload of Malicious Files |
|  | **Client Side Testing** |
| 89 | Testing for DOM based Cross Site Scripting |
| 90 | Testing for JavaScript Execution |
| 91 | Testing for HTML Injection |
| 92 | Testing for Client Side URL Redirect |
| 93 | Testing for CSS Injection |
| 94 | Testing for Client Side Resource Manipulation |
| 95 | Test Cross Origin Resource Sharing |
| 96 | Testing for Cross Site Flashing |
| 97 | Testing for Clickjacking |
| 98 | Testing WebSockets |
| 99 | Test Web Messaging |
| 100 | Test Local Storage |

# Deliverables

On completion of Web Application assessment, STM Team will deliver a comprehensive report consisting of the following:

▶ Executive summary report

▶ Detailed vulnerability analysis report

▶ Recommendations with prioritized action plan

▶ Risk Registry

# Technical Tools and Utilities

AsquareIT STM team will use utilities/tools for carrying out the tests including but not limited to the following utilities:

# Commercial Tools

* Qualys Guard
* Nessus
* Nexpose
* Burp Suite

# Open Source Tools

* OWASP ZAP
* Netcat
* Nmap
* Kali Os for pentesting
* NSlookup
* Dig
* SSL Digger

# Roles and Responsibilities

|  |  |
| --- | --- |
| **Role** | **Responsibility** |
| Sr.Security Analyst | * Create new tests to identify vulnerabilities across the application. * Find vulnerabilities in popular, common software as well as proprietary applications. * Pinpoint entry points for hackers * comparing the latest security threats and malware with existing application * Review current corporate policies and help redefine procedures for better security. * Enhance current software with implementations of better security standards. * Documenting feedback and reports for review of main managers * Pinpoint methods that attackers could use to exploit weaknesses and logic flaws. * Employ social engineering to uncover security holes (e.g., poor user security practices or password policies) * Review and define requirements for information security solutions. * Work on improvements for security services, including the continuous enhancement of existing methodology material and supporting assets. * Provide feedback and verification as an organization fixes security issues. * Rate the identified vulnerabilities with CVSS Scoring system. * Detailed Report with proper Evidence along with step to reproduce |

# Assumptions and Dependencies

It is important at this stage that and AsquareIT are clear and agree the assumptions and dependencies by both parties.

# Assumptions

In outlining the solution, the following assumptions have been made:

▶ Whilst best efforts are made to identify security vulnerabilities in the application in question, AsquareIT cannot be held responsible for any security breaches occurring after or during the assessment.

▶ The Web Application will be adjudged based on the current threat, exploits and vulnerabilities. Emergence of a new vulnerability after the completion of the testing process will not account to the scope of current exercise.

▶ The scope and estimates given in this plan covers a single run of penetration testing on each of the application. Any changes to scope shall be treated as new engagement and require different estimates.

▶ It is assumed that no changes will be made to the application when the assessment process is in progress. Any changes will be considered in the subsequent assessment round as a separate engagement.

▶ AsquareIT will execute the test on Web Application and report the issues. Any future regression test or Web Applications will be taken up as additional engagements.

▶ will appoint a single point of contact who will act as the key interface and escalation point for the Asqaure team. In addition, may need to make available certain technical staff that have the necessary knowledge and expertise to collaborate with and assist the AsqareIT team on resolving project issues. These persons shall be made available in the required time to facilitate issue resolution and completion of the assessment on schedule.

# Dependencies

In outlining the solution, the following dependencies have been made:

▶ Dates for regression testing are to be mutually agreed upon between AsquareIT and. AsquareIT expects a lead time of a week to commence regression test cycle.

▶ shall make available the access information for Web Application in scope for security assessment.

▶ shall provide a stable and well managed environment to conduct tests. AsquareIT follows standard practices and procedures to ensure that this environment is used securely and shall not be liable for any security incidents.

▶ Test credentials based on the roles and end user types are to be provided by

# Communication and Reporting

For the services outlined in Section 4, AsquareIT will provide the following management team.