**System Test Plan**

**For**

**AutoPen**

***Caleb Hall***

***Joshua Buscher***

***Calla Robison***

***Myles Scott***

***Michael Allen***

| Version/Author | Date |
| --- | --- |
| 1.0 Joshua Buscher | 10/16/2023 |
| 1.1 Michael Allen | 10/17/2023 |
| 1.2 Caleb Hall | 10/18/2023 |
| 2.0 Michael Allen | 11/14/2023 |
| 2.1 Michael Allen | 12/5/2023 |
| 2.2 Michael Allen | 12/6/2023 |
| 2.3 Caleb Hall | 12/6/2023 |
| 3.0 Joshua Buscher | 1/25/2024 |
| 3.1 Joshua Buscher,  Michael Allen,  Caleb Hall | 2/25/2024 |

Table of Contents

[1. Introduction](#_heading=h.gjdgxs) 2

[1.1 Purpose 2](#_heading=h.30j0zll)

[1.2 Objectives 2](#_heading=h.1fob9te)

[2. Functional Scope 2](#_heading=h.3znysh7)

[3. Overall Strategy and Approach 2](#_heading=h.2et92p0)

[3.1 Testing Strategy 2](#_heading=h.tyjcwt)

[3.2 System Testing Entrance Criteria 3](#_heading=h.3dy6vkm)

[3.3 Testing Types 3](#_heading=h.1t3h5sf)

[3.4 Suspension Criteria and Resumption Requirements](#_heading=h.4d34og8) [4](#_heading=h.esb6ch8nxr47)

[4. Execution Plan 4](#_heading=h.2s8eyo1)

[4.1 Execution Plan 4](#_heading=h.17dp8vu)

[5. Traceability Matrix & Defect Tracking 4](#_heading=h.3rdcrjn)

[5.1 Traceability Matrix 4](#_heading=h.26in1rg)

[5.2 Defect Severity Definitions 5](#_heading=h.lnxbz9)

[6. Environment 5](#_heading=h.35nkun2)

[6.1 Environment 5](#_heading=h.1ksv4uv)

[7. Assumptions 6](#_heading=h.2jxsxqh)

[8. Risks and Contingencies](#_heading=h.z337ya) [6](#_heading=h.3j2qqm3)

# Introduction

## Purpose

This document is a test plan for AutoPen System Testing, produced by the System Testing team. It describes the testing strategy and approach to testing the team will use to verify that the application meets the established requirements of the business prior to release.

## Objectives

* Meets the requirements, specifications and the Business rules.
* Supports the intended business functions and achieves the required standards.
* Satisfies the Entrance Criteria for User Acceptance Testing.

# Functional Scope

The Modules in the scope of testing for the AutoPen System Testing are mentioned in the document attached in the following path :

1. The Software Design Document:  
   <https://docs.google.com/document/d/1bsdYF6pc5wq42goeOdWyzdGO5NMxhz6m/edit?usp=sharing&ouid=107851779739044104235&rtpof=true&sd=true>
2. The System Requirements Specification Document:  
   <https://docs.google.com/document/d/1--zXq_zFsEa_CdH05QXsuDOT1ZxZHEgw/edit?usp=sharing&ouid=107851779739044104235&rtpof=true&sd=true>
3. Section 3.1 of this document

# Overall Strategy and Approach

## Testing Strategy

AutoPen System Testing will include testing of all functionalities that are in scope (Refer Functional Scope Section). System testing activities will include the testing of new functionalities, modified functionalities, screen level validations, work flows, functionality access, testing of internal & external interfaces.

**Objective:** To ensure a comprehensive assessment of the "AutoPen" system by meticulously detailing the functional and non-functional areas that require evaluation.

## Functionalities Testing

**Test Objective:** Test user login, registration, scan start, and scan retrieval

**Technique:** Follow user process. When given correct information, the system shall execute the expected function. When given incorrect information, the system shall not execute.

**Completion Criteria:** All tests pass and no defects remain.

## UI Testing

**Test Objective:** Perform Screen level validations to ensure usability and consistency

**Technique:** Navigate through web pages, noting usability and ease of access.

**Completion Criteria:** Website is easy to understand and navigate.

## Workflow Testing

**Test Objective:** Examine the workflow process for accuracy and efficiency

**Technique:** After a sprint completes, review all work completed

**Completion Criteria:** Tasks accomplished in timely manner, sprint reports (ie burndown) are even

## Access Testing

**Test Objective:** Ensure that different roles have appropriate and restricted access to functionalities as per their permissions

**Technique:** As a user, attempt to access the scan results of another user.

**Completion Criteria:** Users are able to access their own results, but no results of another user.

## 

**Details:**

* Scope of Testing: All functionalities within the defined scope will be tested (Refer to the Functional Scope Section for specifics).
* Functionalities Testing: This involves rigorous testing of new functionalities, altered functionalities, and their interactions within the system.
* UI Testing: Screen level validations will be conducted to ensure usability and consistency across the application.
* Workflow Testing: The process flows and their respective sequences within "AutoPen" will be examined for accuracy and efficiency.
* Access Testing: Evaluations will be done to ensure that different user roles have appropriate and restricted access to functionalities as per their permissions.
* Interface Testing: Both internal and external interfaces of "AutoPen" will be tested to ensure seamless integration and communication.

## System Testing Entrance Criteria

URL : autopentest.com

User/Developer logged in

AutoPen test ready

**Objective:** To define clear prerequisites that must be fulfilled before initiating the system testing phase.

**Criteria for Testing Readiness:**

* URL Access: The test environment should be accessible at autopentest.com.
* User Status: A user or developer must be logged into the system.
* Test Status: 'AutoPen' should be in a 'test-ready' state, implying that the application is stable, all primary features are implemented, and it is ready to undergo the testing process using BurpSuite for comprehensive scanning and penetration testing activities."

## 

## Testing Types

### Usability Testing

**Objective**: To assess the user-friendliness and usability of the "AutoPen" web interface.

**Details**:

* The UI will be examined for its aesthetic appeal, accuracy of content, and intuitive nature.
* The main aim is to guarantee a smooth and logical user experience, with special emphasis on ensuring a coherent navigation flow.
* Elements like access shortcuts, consistent navigation patterns, and legibility of text are paramount.

User interface attributes, and its respective content will be tested for accuracy and usability. The goal is to guarantee a smooth and logical user experience, with special emphasis on ensuring a coherent navigation flow, with elements like access shortcuts, consistent navigation patterns, and legibility of text being paramount.

System Requirement Specification REQ-1: “System shall allow users to input target specifications”

System Requirement Specification REQ-3: System shall provide real-time feedback on the penetration testing progress.

### Functional Testing

**Objective:** Validate that "AutoPen" fulfills all functional requirements and operates in accordance with predefined business rules.

**Details:**

* This testing focuses on individual functionalities to ensure they align with the stipulated Business / Functional Requirements.
* All business rules or conditions are verified to ensure they are correctly implemented in the system.
* Resolutions to issues, feedback, and change requests documented during the project lifecycle will also be subject to functional testing.
* The goal of this form of testing is to ensure each part of the system meets the functional requirements outlined in the following:
* Business / Functional Requirements
* Business rules or conditions
* Resolutions to issues, feedback, and change requests

System Requirement Specification REQ-2: System shall validate the target specifications for correctness and safety.

System Requirement Specification REQ-4: System shall display a summary of vulnerabilities detected and potential fixes.

System Requirement Specification REQ-5: System must handle potential errors, such as unreachable targets, gracefully by notifying the user.

System Requirement Specification REQ-6: System shall ensure user authentication before initiating any penetration tests.

System Requirement Specification REQ-7: System shall provide user registration and login functionality.

System Requirement Specification REQ-8: System should allow users to edit their profile information, including name, email, and contact details.

System Requirement Specification REQ-9: System must securely store and encrypt user passwords.

System Requirement Specification REQ-10: System must allow password recovery through a secure method, such as email verification.

**Previous Tests:**

* The functionality of Kali Linux alongside Metasploit found full functionality of the penetration testing regime. However, AutoPen has moved to Burpsuite for more enhanced scanning and exploitation capabilities, requiring further functionality testing.

## Suspension Criteria and Resumption Requirements

This section will specify the criteria that will be used to suspend all or a portion of the testing activities on the items associated with this test plan.

## Suspension Criteria

**Criteria:**

* Any significant malfunction or incident that renders further testing of "AutoPen" impossible or counterproductive will lead to a suspension of testing activities.
* Alterations made to the software, hardware, or database post a testing suspension will mandate a review.
* Testing should be suspended if the Algorithm manages to escape the firewall used for preliminary research.

**Responsibility:**

* The Testing Manager has the discretion to decide if the entire test plan should be rerun or only specific sections of it, especially after significant modifications.

## Resumption Requirements

### Testing activities will resume only after the problematic functionality, which initially caused the halt, has undergone successful retesting.

### A comprehensive review and verification process will be established to ensure the cause for suspension has been rectified.

* If the suspension is related to the Algorithm escaping the firewall, testing can continue when the way the Algorithm escaped the firewall is found and patched, and the Algorithm is put back into the confines of the firewall.

# Execution Plan

## Execution Plan

The execution plan outlines the procedure for carrying out this project’s test cases. It ensures coverage of all requirements and is adaptable to accommodate any necessary changes during the testing phase. All test cases for the release are logically sequenced based on their interdependencies and the overall testing strategy. This plan is designed to confirm that AutoPen meets its specified requirements for functionality, database integrity, and performance.

The following is the test plan for AutoPen’s system:

* + 1. **UI Testing** - Refer to 3.1.2 for objectives and details.
    2. **Functionalities Testing -** Refer to section 3.1.1 for objectives and details.
    3. **Workflow Testing** - Refer to section 3.1.3 for objectives and details.
    4. **Access Testing** - Refer to section 3.1.4 for objectives and details.

**Test Case 1:** The system shall allow a user to switch between site pages

**Test Case 2:** The system shall allow a user to create a profile

**Test Case 3:** The system shall allow a user to input a target URL

**Test Case 4:** The system shall communicate between the front-end interface and back-end VM API

**Test Case 5:** The system shall output test results

* When scans are completed, the API will forward information collected.
* Forwarded information shall be displayed on the UI interface to the user.

**Test Case 6:** The system shall document and track all prior tests

**Test Case 7:** The system shall be online during penetration tests

**Test Case 8:** The system shall identify and mitigate defects when they occur

* When a defect is identified, it will be documented with relevant details like the defect description, severity, steps to reproduce, and screenshots if applicable.
* The defect will be assigned to the respective developer/team for resolution.
* The status of the defect (open, in-progress, resolved, closed) will be regularly updated in the defect tracking tool.
* Once the defect has been fixed, it will be retested to ensure its resolution. If the defect is not resolved, it will be reassigned with appropriate comments.
* Regular defect review meetings will be held to discuss the status of the defects and the plan of action.

# Traceability Matrix & Defect Tracking

## Traceability Matrix

| **Requirement ID** | **Requirement Description** | **Test Case ID** | **Test Case Description** |
| --- | --- | --- | --- |
| REQ-001 | User Registration Functionality | TC-001 | Verify user can register with valid email and password. |
| REQ-002 | Password Strength for Registration | TC-002 | Confirm password meets minimum strength requirements. |
| REQ-003 | Penetration Test Configuration Screen | TC-003 | Validate entry of target URL and test depth selection. |
| REQ-004 | User Login | TC-004 | Ensure user can login with registered credentials. |
| REQ-005 | Report Generation after Test | TC-005 | Validate that a detailed report is generated post-test. |
| REQ-006 | External System Interfaces | TC-006 | Verify the interface connections to external systems. |
| REQ-007 | Error Handling in Penetration Test | TC-007 | Check system's response to unexpected inputs during a test. |
| REQ-008 | Payment System Integration | TC-008 | Validate secure payment gateway integration. |
| REQ-009 | User Profile Modification | TC-009 | Ensure users can modify their profiles post-registration. |
| REQ-010 | System Load and Stress Testing | TC-010 | Evaluate system performance under heavy user load. |

## Defect Severity Definitions

| **Critical** | The defect causes a catastrophic or severe error that results in major problems and the functionality rendered is unavailable to the user. A manual procedure cannot be either implemented or a high effort is required to remedy the defect. Examples of a critical defect are as follows:   * System abends * Data cannot flow through a business function/lifecycle * Data is corrupted or cannot post to the database |
| --- | --- |
| **Medium** | The defect does not seriously impair system function and can be categorized as a medium Defect. A manual procedure requiring medium effort can be implemented to remedy the defect. Examples of a medium defect are as follows:   * Form navigation is incorrect * Field labels are not consistent with global terminology |
| **Low** | The defect is cosmetic or has little to no impact on system functionality. A manual procedure requiring low effort can be implemented to remedy the defect. Examples of a low defect are as follows:   * Repositioning of fields on screens * Text font on reports is incorrect |

# Environment

## Environment

The AutoPen testing environment has undergone a significant transition, moving from a local setup to a cloud-based infrastructure provided by Digital Ocean. This shift marks an evolution from using an HP laptop with an Intel® Core™ i5-1035G1 processor and 12 gigabytes of RAM, which ran a virtual machine through VMware Workstation Player. The previous VM, equipped with two dedicated cores and 4 gigabytes of RAM, operated on Kali Linux 2023.3 and hosted the Damn Vulnerable Web Application (DVWA) for testing purposes.

Digital Ocean's architecture centers around the concept of 'droplets', which are essentially scalable virtual machines (VMs) allocated resources from a physical host machine. These droplets function under a virtual machine monitor, or hypervisor, ensuring each droplet receives its designated virtual resources like vCPU. The droplets are divided into two categories: Shared CPU and Dedicated CPU, with the former offering dynamic allocation of hyper-threads based on load and the latter providing guaranteed access to the full hyper-thread at all times.

For AutoPen's testing requirements, a Premium AMD shared CPU droplet with 4 GB / 2 AMD CPUs, 80 GB NVMe SSD storage, and 4 TB Transfer limit, and 512 MB of RAM has been chosen. This configuration ensures robust performance and reliability, crucial for the intensive processes of system testing. The droplet hosts the DVWA web app, which continues to use Apache2 and MariaDB on the backend. This new setup, encompassing significantly enhanced processing power and memory, is tailored to meet the demanding needs of AutoPen's penetration testing, ensuring precision and efficiency in the testing process.

General testing usage

* HP laptop with an Intel® Core™ i5-1035G1 processor
* 12 gigabytes of RAM
* Have VMware Workstation Player download

Digital ocean sections

* Using droplets (scalable virtual VM)
* Shared CPU (has dynamic allocation of hyper-threads based on load)

Cloud usage

* 2 AMD CPUs
* 4 GB RAM
* 80 GB NVMe SSD storage
* 4 TB Transfer limit
* Be able to use DVWA web app
* Be able to use NGINX
* Be able to use Postgresql

# Assumptions

* Assume that the Algorithm can and will break out and a kill switch will be activated if needed.
* Assume that there is a possibility of not being able to do all testing on the required schedule, and attempt to prioritize testing actions.

# Risks and Contingencies

* Environment Instability: Test environments may not mimic production perfectly or may be unstable, leading to false positives or negatives.
* Resource Unavailability: Key testing resources, both human and technical, might become unavailable during crucial testing phases.
* Data Integrity Issues: Test data might not be comprehensive or accurate, potentially leading to skewed results.
* Inadequate Coverage: Some critical scenarios or functionalities might be missed out during testing.
* Software Dependencies: There might be external systems or applications that AutoPen depends on, and any instability in those can affect the testing.

| Risk # | Risk | Impact | Contingency plan |
| --- | --- | --- | --- |
| 1 | Environment Instability | High | Backup files on external device, reinstall software |
| 2 | Resource Unavailability | High | Contact support |
| 3 | Data Integrity Issues | High | Restore from recent backup |
| 4 | Inadequate Coverage | Medium | Remediate any findings brought to the teams attention |
| 5 | Software Dependencies | Low | Find new library or create custom scripts |

# Appendices