CS490 AutoPen AI Project Scope

# // Product Vision Statement:

The AutoPen project aims to advance the field of penetration testing by introducing a system based on artificial intelligence methodologies in the evolving landscape of cybersecurity, where threats are becoming increasingly complex. AutoPen's objectives include the development of an AI algorithm capable of conducting autonomous penetration tests, the creation of an intuitive dashboard for streamlined management and review of test results, and the provision of actionable insights based on identified vulnerabilities. The system is intended to be adaptive, with machine learning techniques used to continuously update testing protocols in response to emerging threat vectors. Furthermore, AutoPen is dedicated to adhering to recognized cybersecurity frameworks and regulations, providing features such as automatic incident report generation and the option of either periodic or on-demand vulnerability assessments. By focusing on these dimensions, AutoPen hopes to provide organizations with a more efficient, cost-effective, and comprehensive approach to identifying and mitigating potential security flaws, thereby contributing to the strengthening of cybersecurity postures in an increasingly perilous digital environment.

**End of Sprint 1 Presentables - 10/5/2023**

1. Technical Requirements Document
2. UI/UX Functioning dashboard
3. Finalize the setup of the penetration testing environment (update tools, ensure network isolation, etc.)
4. Recorded preliminary manual penetration test
5. AI Algorithm feasibility assessment
6. AI Algorithm logic flow pseudocode
7. "Hello World" version of the AI algorithm (a simple script that demonstrates basic functionality).

**End of Sprint 2 Presentables - 10/31/2023**

1. Basic Algorithm & testing now available from the dashboard
2. Data logging
3. Test Results display

**End of Sprint 3 Presentables - 12/07/2023**

1. Complete AI algorithm for autonomous penetration testing
2. AI Algorithm & testing now available from the dashboard
3. Basic Incident report generator
4. User Acceptance Tests (UAT)
5. Standard cybersecurity frameworks and regulations compliance report
6. User guide/manual
7. Phase 2 plan (for a 2nd semester continuation)

# // Section 1: Basic Framework, Research, and Cybersecurity Prep

// Objective: Set up the basic structure of the project, conduct preliminary research, and prepare the environment for penetration testing. Complete a Project Vision Statement.

// Tasks: (1 week)

1. Create a Project Vision Statement.

2. Set up version control (e.g., Git repository).

3. Create a basic project roadmap outlining the sprints and their objectives.

4. Conduct research on existing penetration testing solutions.

5. Research AI methodologies and tools relevant to penetration testing. List pros and cons of each and figure out how we can improve the market.

6. Identify key cybersecurity frameworks and regulations to adhere to.

7. Develop a list of technical requirements for the project (e.g., programming languages, libraries, hardware).

* Use Node.js or python in the django framework for server-side programming language
  + Easily implemented into an html document

8. Set up the development environment.

* VS code using github desktop?

9. Create a risk assessment document.

10. Identify tools and platforms for penetration testing (e.g., Metasploit, Burp Suite).

11. Setup a test environment (e.g., VMs, Docker containers) for penetration testing.

12. Perform a preliminary manual penetration test to understand the test environment.

---

# // Section 2: Algorithm Prototyping

// Objective: Begin development of a rudimentary AI algorithm, finalize the penetration testing environment, and create an initial prototype of the dashboard.

// Tasks: (3 weeks)

1. Sketch the basic UI/UX design for the dashboard.

2. Develop a "Hello World" version of the AI algorithm (a simple script that demonstrates basic functionality).

3. Conduct a feasibility assessment for the AI algorithm.

4. Write pseudocode outlining the AI algorithm's logical flow.

5. Finalize the setup of the penetration testing environment (update tools, ensure network isolation, etc.)

6. Create documentation for the finalized penetration testing environment.

7. Validate the environment by conducting a series of manual penetration tests.

8. Start building the basic UI for the dashboard.

9. Add dummy data to simulate test results in the dashboard.

10. Implement basic navigation features in the dashboard.

11. Set up data logging for the AI algorithm.

12. Conduct initial tests on the algorithm and dashboard.

13. Document the development process and potential issues, with an emphasis on cybersecurity considerations.

---

# // Section 3: Functionality and Presentation

// Objective: Develop a functioning AI algorithm for penetration testing and complete the dashboard functionalities. Present the working product.

// Tasks: (5 weeks)

1. Complete the AI algorithm for autonomous penetration tests.

2. Integrate AI algorithm with the dashboard.

3. Implement features in the dashboard to display test results.

4. Add functionalities to provide actionable insights based on test results.

5. Test the integrated system for bugs and vulnerabilities.

6. Create a basic version of an incident report generator.

7. Conduct user acceptance testing (UAT) with a small group.

8. Make adjustments based on user feedback.

9. Prepare a presentation showcasing the working product.

10. Present the working product.

---

# // Stretch Goal Section 4: Refinement and Optimization

// Objective: Refine and optimize the already functioning AutoPen system, focusing on making it cleaner and more efficient.

// Tasks: (3 weeks)

1. Conduct code reviews to identify areas for improvement.

2. Optimize the AI algorithm for speed and accuracy.

3. Refine the dashboard UI/UX based on user feedback.

4. Add advanced features to the incident report generator.

5. Test for compliance with standard cybersecurity frameworks and regulations.

6. Optimize the system for scalability, focusing on small and medium enterprises (SMEs).

7. Conduct further UAT with a more diverse group.

8. Update system documentation.

9. Finalize user guides and FAQs.

10. Prepare and distribute a project completion report.