**Milestone 03: Methodology**

How have your literature and technology reviews influenced your project choices and strategies? (e.g., technologies selected, project scope, aims and objectives)

The literature and technology reviews have significantly influenced the choices and strategies for my project on identifying potential vulnerabilities in Android-based infotainment systems emulated within Android Studio.

##### Technologies Selected:

1. **OWASP ZAP**: Based on its extensive use in vulnerability scanning and web application security, ZAP was chosen for identifying potential security issues in the emulated infotainment system.
2. **Nessus**: Selected for its robust network vulnerability scanning capabilities, Nessus helps in detecting vulnerabilities in network configurations and services within the infotainment system.

##### Project Scope:

* The reviews highlighted the importance of both app-level and system-level security. Therefore, the project scope includes a comprehensive analysis covering various layers of the infotainment system.
* Emphasis on WiFi vulnerabilities was added after reviewing the relevant literature, expanding the project scope to include network security aspects.

##### Influence of Literature:

* **Security Analysis of Android Automotive**: Reinforced the need for examining system architecture and third-party app risks.
* **Vulnerability Analysis of Android Auto Infotainment Apps**: Highlighted common app vulnerabilities, influencing the focus on app behavior analysis.
* **Penetration Testing of an In-Vehicle Infotainment System**: Provided insights into practical testing techniques and real-world application of security measures.
* **Vulnerability Analysis of an Automotive Infotainment System's WIFI Capability**: Brought attention to the critical nature of WiFi security, leading to the inclusion of WiFi vulnerability analysis in the project.

##### Strategies:

* Comprehensive coverage of security layers, from app to network level.
* Regular updates on emerging threats and evolving attack vectors.
* Continuous refinement of mitigation strategies based on the latest research and testing outcomes.

These reviews have provided a solid foundation for the project, ensuring a holistic approach to securing Android-based infotainment systems.

What tools and strategies will you use to manage your project? (e.g., GitHub, MS Teams, Trello and an agile project management methodology etc.,)

To manage my project effectively, I will use the following tools and strategies:

##### Tools:

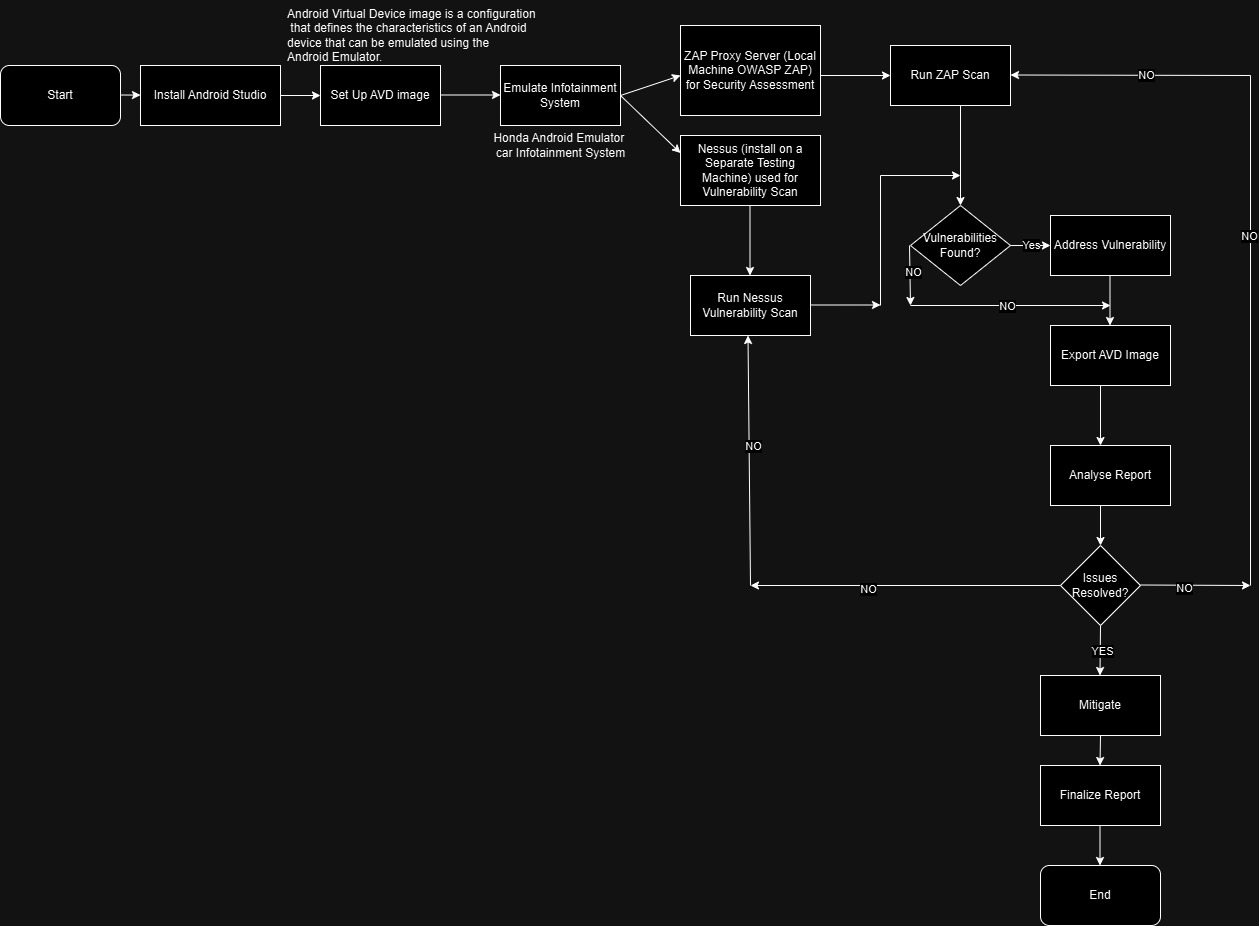
1. **GitHub**: For version control and collaboration on code and documentation.
2. **MS Teams**:  file sharing, and virtual meetings.
3. **Trello**: For task management and tracking project progress.

##### Strategies:

1. **Agile Methodology**: Implementing agile principles to ensure flexibility and iterative progress. This includes regular sprints, stand-up meetings, and continuous feedback.
2. **Scrum Framework**: Using Scrum to manage the development process, with defined roles such as Product Owner, Scrum Master, and development team.
3. **Regular Reviews and Retrospectives**: Conducting regular reviews and retrospectives to evaluate progress, address challenges, and continuously improve processes.
4. **Risk Management**: Identifying potential risks early and developing mitigation strategies to address them proactively

What resources will you require for your project? (e.g., technology stack, hardware (including version numbers), software and plugins (including version numbers), applications (including version numbers), network resources, internet access, API resources etc). Resources Required for the Project  
**Hardware**1. Computer  
     - Processor: Intel CORE i3 10th GEN or higher  
     - Memory: 8 GB RAM or higher  
     - Storage: SSD with at least 256 GB of free space  
   - Purpose: To run Android Studio, ZAP, and Nessus smoothly.  
2. Mobile Device (Optional but Ideal)  
   - Specifications: A physical Android device matching the target infotainment system's specifications.  
   - Purpose: For more realistic emulation and to potentially uncover hardware-specific vulnerabilities.  
**Software**  
1.Operating System  
   -Options:   
     - Windows 10/11  
     - macOS 10.15 (Catalina) or later  
     - Ubuntu 20.04 LTS or later  
   -Purpose: To support the installation and running of Android Studio, ZAP, and Nessus.  
2. Android Studio  
   - Version: Latest version (2024)  
   - Purpose: The primary development environment for building and testing Android applications.  
   - Includes: Tools for emulating Android devices.  
3. ZAP (Zed Attack Proxy)  
   - Version: Latest version (2024)  
   -Purpose: An open-source web application security scanner used to capture and analyze traffic between the emulator and the internet.  
4. Nessus  
   -Version: Latest version (2024)  
   -Purpose: A commercial vulnerability scanner that can identify a broader range of vulnerabilities beyond web applications.  
   -Edition: Nessus Essentials or Nessus Professional.  
5. Android Virtual Device (AVD) Image  
   -Purpose: Represents the target infotainment system. Pre-built images can be found online or created using Android Studio.  
**Network Resources**  
1.Reliable Network Connection  
   -Purpose: Required for downloading software, tools, and updates, as well as for running network-based scans and tests.  
**Internet Access**Purpose:   
   - To download and update software and tools.  
   - To access online resources, documentation, and support.  
   - To perform web-based vulnerability scans using ZAP and Nessus.  
**API Resources**  
1.Google Maps API  
   -Purpose: If the infotainment system includes navigation features, testing the integration and security of API calls.  
**Version Control**  
1. Git  
   - Version: Latest version (2024)  
   - Purpose: For version control of the code and project documentation.  
   - Repository: GitHub or GitLab for collaboration and backup.  
By ensuring that all these resources are in place, the project can be executed efficiently and effectively, leading to comprehensive identification and analysis of potential vulnerabilities in the emulated Android-based car infotainment system.

What is the design of your project and how will the resources listed be used? (e.g., express the project in the form of architectural diagram accompanied by a description of how the resources will be used. Include wireframes, diagrams, user persona etc. If you use data, where will it come from and have you secured ethical approval. If in doubt, ask your supervisors).



##### User Persona

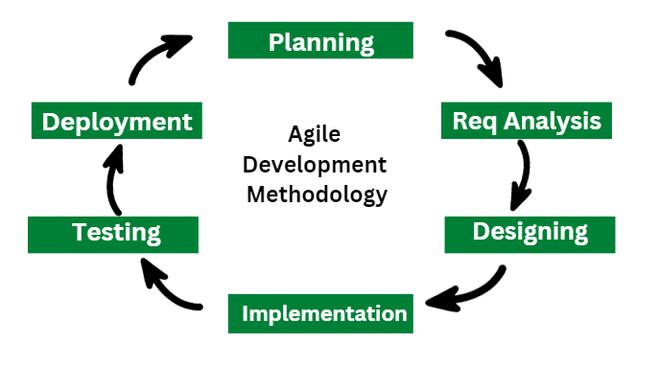
1. **Name**: Razeena Rafeek

* **Role**: Cyber Security Risk Analyst
* **Goals**: Identify and mitigate vulnerabilities in automotive infotainment systems to enhance security. Complex and evolving threat landscape, need for comprehensive testing tools and up-to-date information.

##### Strategies and Implementation

1.      **Agile Methodology**:

* **Description**: Use agile principles to manage the project, ensuring flexibility and iterative progress



2.      **Risk Management**:

* **Description**: Identify and assess potential risks early. Develop mitigation strategies to address these risks proactively.
* **Tools**: Trello for tracking and managing risk-related tasks and actions.

3.      **Regular Reviews and Retrospectives**:

* **Description**: Conduct regular reviews and retrospectives to evaluate progress, address challenges, and continuously improve processes.
* **Tools**: MS Teams for conducting virtual meetings and discussions.

How will you evaluate the project artefact and know if it is fit for purpose? (e.g., this might include unit testing, user testing, performance testing etc. You will also need to consider authenticity of the artefact evaluation and who carries out the tests).

To ensure the project artefact is fit for purpose, the following evaluation methods will be employed:

##### 1. Unit Testing

* **Purpose**: Verify that individual components of the infotainment system function correctly.
* **Implementation**: Automated tests using frameworks like JUnit to test specific modules of the emulated system.

##### 2. Integration Testing

* **Purpose**: Ensure that different modules of the system work together as expected.
* **Implementation**: Automated and manual testing to check interactions between various components.

##### 3. Vulnerability Scanning

* **Purpose**: Identify security flaws in the system.
* **Implementation**: Using OWASP ZAP and Nessus to perform thorough scans of the emulated environment.

##### 4. User Testing

* **Purpose**: Assess usability and functionality from an end-user perspective.
* **Implementation**: Real-world scenarios and feedback collection from users.

##### Evaluation Process

1.    **Preparation**

* Define test cases and scenarios for each evaluation method.
* Set up the testing environment and tools.
* Select testers and gather necessary resources.

2.    **Execution**

* Perform unit, integration, and performance tests using automated scripts and manual procedures.
* Conduct vulnerability scans with OWASP ZAP and Nessus.
* Carry out user testing with real users, collecting feedback and observing interactions.

3.    **Analysis**

* Review test results, identify issues, and categorize them based on severity.
* Analyze user feedback to determine usability issues and areas for improvement.
* Compile performance metrics to assess system efficiency and stability.

4.    **Reporting**

* Document findings from all tests and evaluations.
* Provide detailed reports on vulnerabilities, performance issues, and user feedback.
* Present recommendations for improvements and necessary actions.

5.    **Action**

* Address identified issues and implement improvements.
* Re-evaluate after making changes to ensure all issues have been resolved.
* Repeat testing cycles as necessary to achieve the desired level of quality and security.