**PROJECT PROPOSAL**

**Identifying Potential Vulnerabilities in a generic Android based Infotainment Systems Emulated within Android Studio**

Describe the problem that your project will aim to address and why this problem is important?

Car infotainment systems are becoming increasingly sophisticated, often running Android as their operating system. These systems offer features like internet connectivity, navigation, and in-vehicle apps. However, the complexity of these systems introduces potential security vulnerabilities. Malicious actors could exploit these vulnerabilities to gain unauthorized access to the car's systems, steal sensitive data, or even disrupt critical driving functions.

**Importance:**

* **Safety Risk:** A compromised infotainment system could be used to tamper with car controls, putting drivers and passengers at risk (e.g., disabling brakes, altering steering).
* **Privacy Concerns:** Infotainment systems might store personal data like navigation history, music preferences, or even connected phone information. A vulnerability could expose this data to unauthorized access.
* **Financial Loss:** Hackers could potentially exploit vulnerabilities to steal financial information linked to in-vehicle payment systems.
* **Reputational Damage:** A security breach in car infotainment systems could damage a car manufacturer's reputation and erode consumer trust.  
  This project aims to identify potential vulnerabilities in generic Android car infotainment systems using ZAP and Nessus.

What are the aims of your project?

**AIM:**

* To identify theoretical weaknesses in a generic Android-based car infotainment system emulated using Android Studio.
* To analyze the potential consequences of these vulnerabilities, considering the severity.
* To raise awareness about the importance of car infotainment system security, highlighting potential risks for car manufacturers and consumers

What are the objectives of your project?

**Objectives**

Configure the emulated car infotainment system to use a specific Android version

* To Run a ZAP scan against the emulated car infotainment system and analyze the results for identified vulnerabilities.
* To Run a Nessus scan targeting Android vulnerabilities and analyze the report.
* To Run a ZAP scan against the emulated car infotainment system, analyzing results for identified vulnerabilities.
* To Analyze the combined ZAP and Nessus reports, identifying at least 3 high-risk vulnerabilities affecting the emulated car infotainment system.
* To identify Potential impact on a real car infotainment system (e.g., unauthorized access, data theft, disruption of functionalities)

What resources will you require in order to complete your project?

 **Android Studio:** This free, open-source IDE (Integrated Development Environment) from Google allows  to develop and test Android applications. It includes the Android Emulator, which can be used to simulate a car infotainment system. <https://developer.android.com/studio/intro>

 **ZAP (Zed Attack Proxy):** This free, open-source web application security scanner will capture and analyze traffic between the emulator and the "internet" within  test environment. <https://www.zaproxy.org/>

 **Nessus Community Edition:** This free version of the popular vulnerability scanner by Tenable allows you to scan the emulated car infotainment system for known vulnerabilities. <https://www.tenable.com/products/nessus/nessus-essentials>

Include links to research, oreilly.com, YouTube videos and other websites that are relevant to your project. For each link, how you plan to use it for your project?

**Android Studio Tutorial**

**Link:**

Play Video

**Use**: This YouTube tutorial will be used to guide the setup and configuration of Android Studio for emulating the car infotainment system. It provides a step-by-step introduction to using Android Studio, which is essential for creating a realistic testing environment

**Research Paper:**Attacks and Defences on Intelligent Connected Vehicles

**Link:**[**Attacks and defences on intelligent connected vehicles: a survey - ScienceDirect**](https://www.sciencedirect.com/science/article/pii/S235286481930197X)

**Use:**This paper will provide insights into various attacks and defense mechanisms relevant to connected vehicles. It will help frame the potential vulnerabilities within the broader context of automotive cybersecurity and guide the development of mitigation strategies

**Research Paper:**Security Analysis of Android Automotive

**Link:**[**(PDF) Security Analysis of Android Automotive (researchgate.net)**](https://www.researchgate.net/publication/340632296_Security_Analysis_of_Android_Automotive)

**Use:**This research paper will inform the security analysis of the Android-based infotainment system by highlighting common vulnerabilities and countermeasures specific to Android Automotive platforms

**Research Paper:**Penetration Testing of an In-Vehicle Infotainment System

**Link:**[**Penetration Testing of an In-Vehicle Infotainment System (diva-portal.org)**](https://www.diva-portal.org/smash/get/diva2:1708534/FULLTEXT01.pdf)

**Use:**This document will be used to understand the methodologies and findings from previous penetration testing efforts on infotainment systems. It will help in designing the penetration testing framework and interpreting results for the project.

**Research Paper:**Vulnerability Analysis of Android Auto Infotainment Apps

**Link:**[**Vulnerability analysis of Android auto infotainment apps | Proceedings of the 15th ACM International Conference on Computing Frontiers**](https://dl.acm.org/doi/abs/10.1145/3203217.3203278)

**Use:**This paper provides an analysis of vulnerabilities in Android Auto apps, which will be useful for identifying app-specific security issues within the infotainment system. It will also help in comparing vulnerabilities across different infotainment apps.

**Research Paper:**Vulnerability Analysis of an Automotive Infotainment System's Wi-Fi Capabilities

**Link:**[**Vulnerability Analysis of an Automotive Infotainment System's WIFI Capability | IEEE Conference Publication | IEEE Xplore**](https://ieeexplore.ieee.org/abstract/document/8673049)

**Use:**This paper focuses on vulnerabilities related to Wi-Fi capabilities in infotainment systems. It will provide insights into wireless communication threats and help identify potential Wi-Fi-related vulnerabilities in the emulated system

**Honda Android Automotive OS Emulator**

Link [Honda Android Automotive OS Emulator | Honda Global](https://global.honda/en/cars-apps/)

Describe any risks, ethical issues or other factors that have been considered as part of developing this project proposal.

#### Risks

1. **Accidental Damage**
   * **Risk**: Using tools like ZAP for testing might unintentionally disrupt the functionality of the car infotainment system, potentially causing inconvenience or safety hazards.
   * **Mitigation**: Testing within a controlled and isolated environment and ensuring thorough backups and documentation of system states before conducting tests.
2. **Resource Constraints**
   * **Risk**: Limited resources in terms of time, expertise, and access to real car hardware.
   * **Mitigation**: Conducting testing in an emulator to be more efficient and potentially less resource-intensive, while acknowledging the limitations of this approach.
3. **Technical Risks**
   * **Risk**: Potential challenges such as scalability issues or emulator limitations.
   * **Mitigation**: Having contingency plans, such as alternative testing tools or methods, and documenting any technical limitations encountered.
4. **Legal and Compliance Risks**
   * **Risk**: Ensuring compliance with legal requirements, such as end-user license agreements (EULAs) and intellectual property laws.
   * **Mitigation**: Reviewing and adhering to EULAs, and limiting testing to features and functionalities that are publicly documented or available.

#### EDI Considerations (Equity, Diversity, and Inclusion)

1. **Accessibility and Usability**
   * **Consideration**: Ensuring the system is usable and accessible to a diverse range of users.
   * **Mitigation**: Conducting user testing with a diverse group of participants to identify and address usability issues across different demographics.
2. **Representation and Bias Mitigation**
   * **Consideration**: Avoiding reinforcement of stereotypes in project visuals, content, or interface design.
   * **Mitigation**: Reviewing project materials and designs to ensure they are inclusive and free from bias.
3. **Community Engagement and Outreach**
   * **Consideration**: Engaging underrepresented groups in the project to ensure diverse perspectives and participation.
   * **Mitigation**: Developing outreach strategies to involve and encourage participation from a wide range of communities.

### Specific Risk Mitigations

1. **Accidental Damage**
   * **Use of Isolated Environment**: Ensuring that the test environment is completely isolated from any real-world systems to prevent unintended disruptions.
   * **Regular Backups**: Keeping regular backups of the emulated system to quickly restore functionality in case of accidental damage.
2. **Data Privacy**
   * **Anonymization Techniques**: Implementing anonymization techniques for any data captured during testing to prevent any potential data privacy issues.
   * **Secure Deletion**: Ensuring that all data is securely deleted after analysis to protect privacy.
3. **Responsible Disclosure**
   * **Disclosure Protocols**: Following established responsible disclosure protocols if major vulnerabilities are identified, by reporting them to Google or the relevant app developers before public disclosure.
   * **Documentation**: Keeping detailed documentation of all findings and disclosure communications to ensure transparency and accountability.
4. **Resource Constraints**
   * **Efficiency in Emulation**: Leveraging the efficiency of emulators to conduct initial tests, saving time and resources compared to testing on actual hardware.
   * **Prioritization**: Prioritizing open-source emulators and documented functionalities to avoid legal and resource constraints.
5. **Impact Analysis**
   * **Contextual Evaluation**: Evaluating the potential impact of discovered vulnerabilities within the context of the emulated system and considering whether they would require additional factors present in a real car.
   * **Detailed Reporting**: Providing detailed reports on the impact analysis to offer clear insights into the relevance and severity of each vulnerability.
6. **End User License Agreements (EULAs)**
   * **Review and Compliance**: Thoroughly reviewing EULAs to ensure that testing activities do not violate any terms and ensuring compliance with all legal restrictions.
   * **Documentation**: Documenting all activities to demonstrate adherence to EULA terms and conditions.

By addressing these risks, ethical considerations, and other factors, the project can be conducted responsibly and effectively, ensuring meaningful outcomes while maintaining integrity and compliance with ethical standards.