**Simplified AI-Driven Supply Chain Attack Detection System**

**Project Overview**

Develop a basic AI-powered system to detect potential supply chain attacks in software packages, inspired by incidents like the SolarWinds attack. The system will analyze software package metadata and contents to identify suspicious modifications.

**Team Structure**

**3 Undergraduate Students:** Students A, B, and C  
**Role Distribution:** All students will work across the entire stack, with focused responsibilities.

**Dataset**

**Synthetic Software Package Dataset**

* **To Be Generated:** By the team
* **Contents:** Metadata and simplified content of software packages
* **Variety:** Mix of "normal" updates and simulated malicious modifications
* **Format:** CSV for easy handling

**Technology Stack**

* **Frontend:** Streamlit (for simplicity)
* **Backend:** Python with Flask
* **Database:** SQLite
* **AI:** Scikit-learn for machine learning

**Project Timeline (8 Weeks)**

**Week 1: Project Setup and Planning**

* **All Members:**
  + Project kickoff, tool selection, and environment setup
  + Research on supply chain attacks and detection methods
  + Begin designing the synthetic dataset structure

**Week 2-3: Data Generation and Preprocessing**

* **Student A (Lead):** Develop scripts to generate synthetic dataset
* **Student B:** Implement data preprocessing and feature extraction
* **Student C:** Set up SQLite database and data storage methods
* **All Members:** Generate initial dataset and perform exploratory data analysis

**Week 4-5: AI Model Development**

* **Student B (Lead):** Develop and train initial machine learning model (e.g., Random Forest)
* **Student A:** Implement feature engineering and selection
* **Student C:** Develop model evaluation metrics and testing framework
* **All Members:** Iterate on model improvement

**Week 6: Basic Detection System**

* **Student C (Lead):** Develop Flask backend for the detection system
* **Student A:** Implement file upload and analysis functionality
* **Student B:** Integrate ML model into the backend system
* **All Members:** Basic system testing and refinement

**Week 7: Frontend Development and Integration**

* **Student A (Lead):** Develop Streamlit frontend for user interaction
* **Student B:** Create visualizations for analysis results
* **Student C:** Integrate frontend with Flask backend
* **All Members:** System integration and end-to-end testing

**Week 8: Final Testing, Documentation, and Presentation**

* **All Members:**
  + Conduct thorough system testing
  + Write technical documentation and user guide
  + Prepare final presentation and demonstration
  + Final code review and project wrap-up

**Key Features to Implement**

1. **Synthetic dataset generation tool**
2. **Machine learning model** for detecting suspicious packages
3. **File upload and analysis system**
4. **Basic visualization** of analysis results
5. **Simple user interface** for interacting with the system

**Simplified ML Approach**

1. **Feature Extraction** from package metadata (e.g., size, version number, upload time)
2. **Basic Content Analysis** (e.g., hash of contents, presence of certain keywords)
3. **Use Random Forest** or similar algorithm for classification
4. **Binary Classification:** "Suspicious" vs "Normal" packages

**Deliverables**

1. **Functional prototype** of the supply chain attack detection system
2. **Synthetic dataset generation script**
3. **Technical report** on the ML model and system architecture
4. **User guide** for operating the system
5. **Project presentation** with a live demonstration

**Learning Objectives**

* **Understand basic concepts** of supply chain attacks
* **Gain experience** in synthetic data generation for cybersecurity
* **Develop skills** in machine learning for security applications
* **Learn full-stack development basics** (Python, Flask, Streamlit)
* **Practice project management and teamwork** in a software project

**Potential Simple Extensions (if time allows)**

* **Implement a basic alert system** for detected suspicious packages
* **Add user authentication** for the web interface
* **Expand the ML model** to classify different types of suspicious modifications

**Resources**

1. **Scikit-learn Documentation:** <https://scikit-learn.org/stable/>
2. **Flask Documentation:** <https://flask.palletsprojects.com/>
3. **Streamlit Documentation:** <https://docs.streamlit.io/>
4. **MITRE ATT&CK (Supply Chain Compromise):** <https://attack.mitre.org/techniques/T1195/>
5. **Python Data Science Handbook:** <https://jakevdp.github.io/PythonDataScienceHandbook/>

This format is structured to be easily transferred into a Word document. Copy and paste the content into Word and make any necessary adjustments to headings, fonts, and layout.