**Project's Title**

The objective of this project is to implement the simulator for Smart Home based on the design proposal from Unit 3 of the Secure Systems Architecture module. This README file will guide you through the features and functionalities of the application.

**Question**

How do the mitigation strategies proposed in the document ensure the continuous availability of smart home services and functionalities in the face of various cyber-attacks?

Argument -

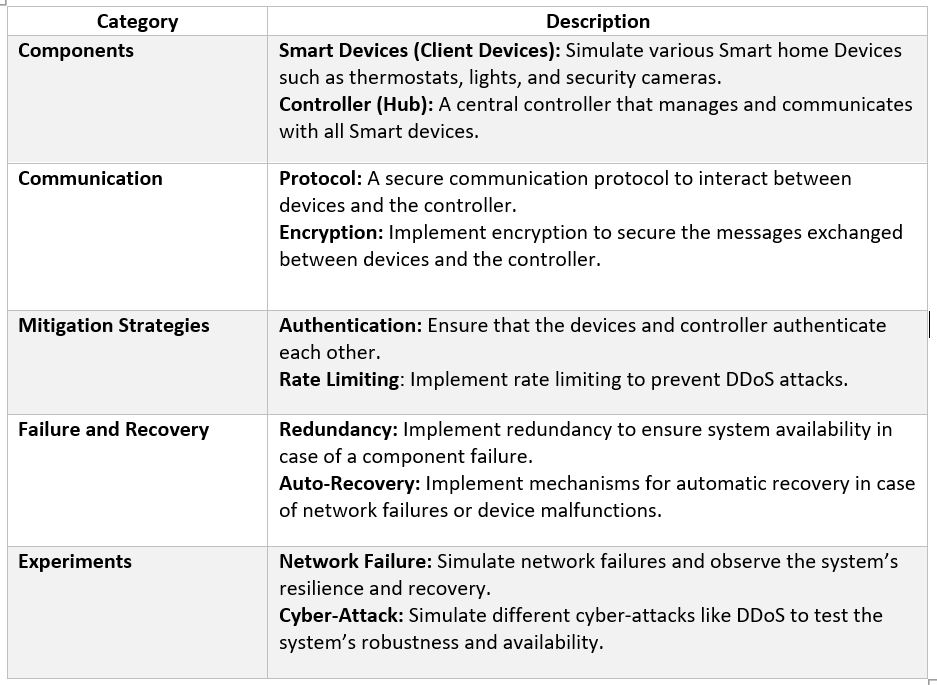
* Feasibility -Testing availability involves simulating cyber-attacks and observing whether the system remains operational.
* Measurability - The success of the mitigation strategies can be measured by the time period of the system services and the user’s ability to access and use them during the simulation of attacks.
* Practical Relevance - Ensuring that smart home services remain available during cyber-attacks is crucial for user experience and the practical utility of smart home technologies.

Figure : Model Specification

**Project Description and Program functions**

* Classes: Three main classes are defined: Vulnerability, Attack, and Mitigation.

1. Vulnerability represents individual security vulnerabilities.
2. Attack represents attacks that consist of multiple vulnerabilities.
3. Mitigation represents mitigation strategies applied to attacks to reduce vulnerability risks.
4. Objects creation: Instances of vulnerabilities, attacks, and mitigations are created with specific attributes.

* Risk calculation: The remaining overall risk of the attack is calculated after applying all mitigations, and the result is printed.
* SmartDevice and controller Classes: These classes simulate the operation of a smart home environment, including device registration, authentication, and encrypted communication.

A computer screen shot of a program code

Description automatically generated

A computer screen with text on it

Description automatically generated

Figure : Class file for Client side

* Latency and message Loss: These aspects are simulated within the send\_status method of the SmartDevice class, adding realism to the communication between devices and the controller.
* Main simulation code: This part of the code creates instances of the classes, simulates the operation of the smart home environment, applies mitigation strategies, and evaluates the remaining risks.

|  |  |
| --- | --- |
| Category | Description |
| Smart Devices (Python) | Attributes**:** Device\_ID, Status, Last\_Communication\_Time Methods:Send\_Status, Receive\_Command |
| Controller (Python) | Attributes: Registered\_Devices, Command\_Queue. Methods: Send\_Command, Receive\_Status, Authenticate\_Device |
| Communication | Implement a secure messaging protocol using encryption for message exchanges. |
| Mitigation Strategies | Implement authentication mechanisms and rate-limiting in the communication protocol. |
| Failure and Recovery | Implement redundancy and auto-recovery mechanisms to handle network failures and device malfunctions. |

**Detailed Design:**

**Prerequisites**

This project has been developed in PyCharm.

1. PyCharm Professional is needed to run the project.

2. Go to [PyCharm](https://www.jetbrains.com/pycharm/download/) (JetBrains, 2023b) to install PyCharm Professional; select "Free 30-day trial" or "Pay for subscription".

3. PyCharm website will automatically detect your Operating System type and provide a suitable link to download; alternatively, choose between Windows, macOS and Linux.

4. Before the download, select the location folder to save the file.

5. Once downloaded, click on "Install" to install PyCharm program.

**How to install libraries**

- Go to File > Settings > Project: Smart\_Simulator.py > Python Interpreter > + > Install package > to install the following packages:

* \*\*datetime:\*\* A library supplying classes to work with time and date - an extract for time, date and time intervals is provided, and date/time object is created (PSF, 2023c).
* \*\*hashlib:\*\* A library providing a common interface to hash and message algorithms, e.g., SHA256, SHA512, MD5, etc. (PSF, 2023d).

**How to run the "main" program**

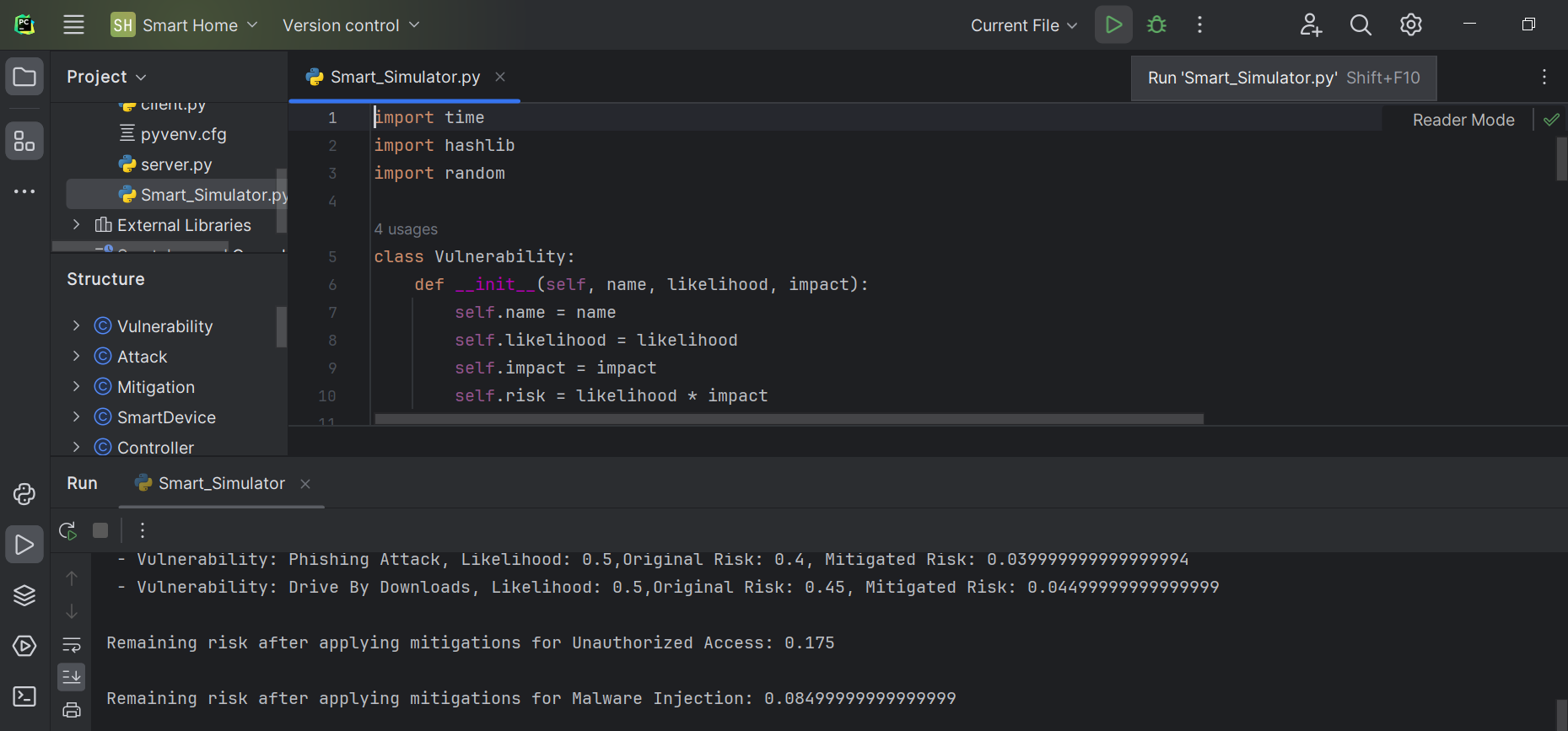
1. Click on the “\*\*Run\*\*” in the dropdown navigation and click on “\*\*Run Main\*\*” as seen below:

Figure : Program Window

**Output from demonstrations**

Impact score is generated based on the assumptions.**A screenshot of a computer error

Description automatically generated**

Figure : Test Result