Individual Reflection

Our team was given a scenario to examine, which modelled the activities associated with networked devices in a smart home. We had to submit a proposal design and the development of a prototype.

Using an Attack-Defence Tree (AD-Tree), we had to identify and analyse potential vulnerabilities of a smart home control system for the proposal document. My contribution was to look for potential vulnerabilities of the smart home and to relatedly describe them. The analysis job appeared uncomplicated at first since my analytical abilities had increased as a result of prior modules in this course. However, while researching academic materials, I discovered a number of comprehensive information on the potential vulnerabilities associated with networked devices in a smart home, which was a challenge because I needed to extract only the most essential information and justify my decisions to my team. Some arguments for the identified vulnerabilities were not satisfactory to my team members after in-depth talks, thereby they were eliminated.

My personal opinion on the design document is that, despite the fact that I believed my analysis abilities had improved, I was challenged by my given tasks, and therefore, I am looking to improve the latter by reading additional academic materials.
Figure 1 shows how our team divided and allocated the tasks for the design document among the team members.

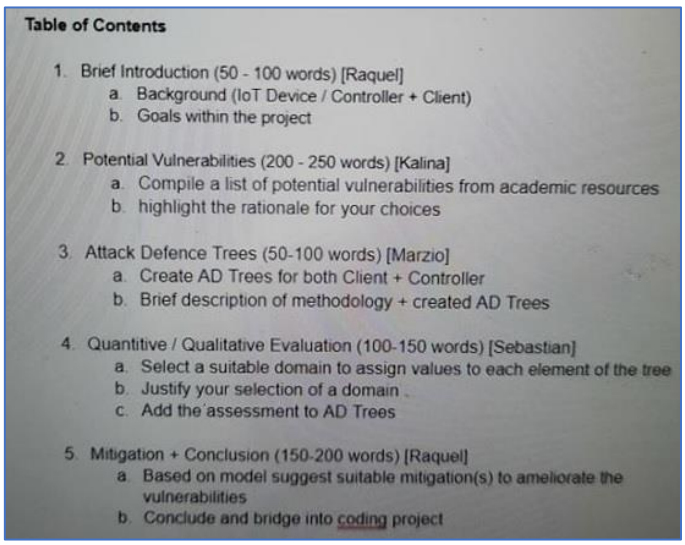
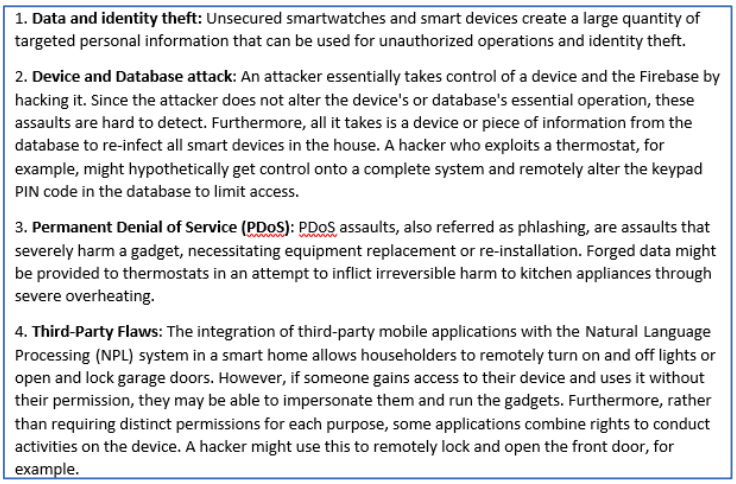


Figure 1 - Distribution of tasks for design document

Figure 2 shows my research and analysis of the potential vulnerabilities from academic resources (Smart Home: Threats and Countermeasures, 2022).



1. **Data and identity theft:** Unsecured smartwatches and smart devices create a large quantity of targeted personal information that can be used for unauthorized operations and identity theft.

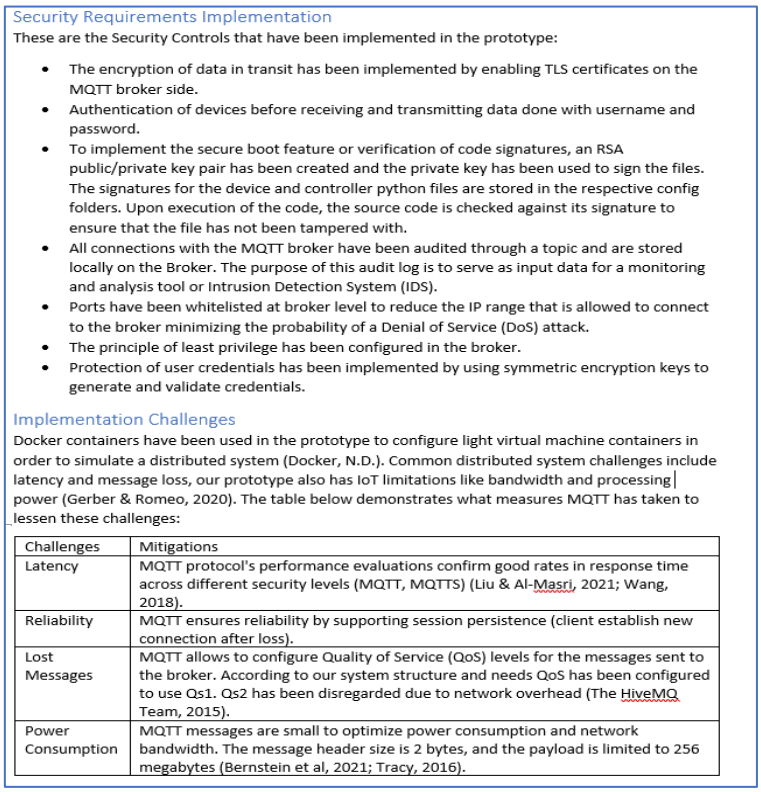
2. **Device and Database attack**: An attacker essentially takes control of a device and the Firebase by hacking it. Since the attacker does not alter the device's or database's essential operation, these assaults are hard to detect. Furthermore, all it takes is a device or piece of information from the database to re-infect all smart devices in the house. A hacker who exploits a thermostat, for example, might hypothetically get control onto a complete system and remotely alter the keypad PIN code in the database to limit access.

3. **Permanent Denial of Service (PDoS)**: PDoS assaults, also referred as phlashing, are assaults that severely harm a gadget, necessitating equipment replacement or re-installation. Forged data might be provided to thermostats in an attempt to inflict irreversible harm to kitchen appliances through severe overheating.

4. **Third-Party Flaws**: The integration of third-party mobile applications with the Natural Language Processing (NPL) system in a smart home allows householders to remotely turn on and off lights or open and lock garage doors. However, if someone gains access to their device and uses it without their permission, they may be able to impersonate them and run the gadgets. Furthermore, rather than requiring distinct permissions for each purpose, some applications combine rights to conduct activities on the device. A hacker might use this to remotely lock and open the front door, for example.

Figure 2 - My research and analysis of potential vulnerabilities from academic resources

Then we began developing our system prototype, which would consist of a single client device that would manage one or more thermostat controller nodes in a smart house. The implementation phase was also broken into two phases, with me and Raquel working on the 'README' documentation and Marzio and Sebastian working on the prototype coding. My contribution to the documentation was to describe the project and to look into the challenges of implementing security as shown in ***Figure 3***.



Security Requirements Implementation

These are the Security Controls that have been implemented in the prototype:

- The encryption of data in transit has been implemented by enabling TLS certificates on the MQTT broker side.
- Authentication of devices before receiving and transmitting data done with username and password.
- To implement the secure boot feature or verification of code signatures, an RSA public/private key pair has been created and the private key has been used to sign the files. The signatures for the device and controller python files are stored in the respective config folders. Upon execution of the code, the source code is checked against its signature to ensure that the file has not been tampered with.
- All connections with the MQTT broker have been audited through a topic and are stored locally on the Broker. The purpose of this audit log is to serve as input data for a monitoring and analysis tool or Intrusion Detection System (IDS).
- Ports have been whitelisted at broker level to reduce the IP range that is allowed to connect to the broker minimizing the probability of a Denial of Service (DoS) attack.
- The principle of least privilege has been configured in the broker.
- Protection of user credentials has been implemented by using symmetric encryption keys to generate and validate credentials.

Implementation Challenges

Docker containers have been used in the prototype to configure light virtual machine containers in order to simulate a distributed system (Docker, N.D.). Common distributed system challenges include latency and message loss, our prototype also has IoT limitations like bandwidth and processing power (Gerber & Romeo, 2020). The table below demonstrates what measures MQTT has taken to lessen these challenges:

Challenges	Mitigations
Latency	MQTT protocol's performance evaluations confirm good rates in response time across different security levels (MQTT, MQTTS) (Liu & Al-Masri, 2021; Wang, 2018).
Reliability	MQTT ensures reliability by supporting session persistence (client establish new connection after loss).
Lost Messages	MQTT allows to configure Quality of Service (QoS) levels for the messages sent to the broker. According to our system structure and needs QoS has been configured to use Qs1. Qs2 has been disregarded due to network overhead (The HiveMQ Team, 2015).
Power Consumption	MQTT messages are small to optimize power consumption and network bandwidth. The message header size is 2 bytes, and the payload is limited to 256 megabytes (Bernstein et al, 2021; Tracy, 2016).

Figure 3 - My contribution in README documentation

Furthermore, every team member was involved in the majority of the phases of our team's Software Development Life Cycle (SDLC) for the prototype, particularly requirement analysis, architectural design, and testing. Despite the fact that Raquel and I were not involved in the prototype's development, we took part in the testing process as non-IT users. As a result, we were able to modify the prototype, making it more user-friendly and providing timely messaging recommendations.

Following the prototype development, my personal reflection is that prototyping not only involves code, but also documentation and user acceptability concerns. Our team had all of the necessary abilities; therefore, it was subdivided into teams to produce the prototype.

Considering the proposal document's feedback, which said that the latter should contain more potential vulnerabilities, I realized that I needed to enhance my analytical skills greater in order to extract more information from resources. As a result, I verified that the same problem did not recur in the README documentation by drafting all of my analyses and conducting a thorough study on the security requirement implementation.

References

Docker (N.D.) Use containers to Build, Share and Run your applications. Available from: https://www.docker.com/resources/what-container

Gerber, A. & Romeo, J. (2020) Connecting all the things in the Internet of Things. https://developer.ibm.com/articles/iot-lp101-connectivity-network-protocols/

Rambus. 2022. Smart Home: Threats and Countermeasures. [online] Available at: <https://www.rambus.com/iot/smart-home/?fbclid=IwAR3h9kddLxVSgtHQjyXArJY-BbA-dlYTSO4bXbJPOZBN4d4lW3vvWtQFumo>

UKEssays.com. 2021. Example Reflective Essay using Rolfe Reflective Model. [online] Available at: <https://www.ukessays.com/essays/nursing/rolfe-reflective-model.php>.