Introduction

A system of systems (SoS) is a group of various separate systems collaborating to offer a single service. These have been categorised below with regard to our smart light system (Baldwin & Sauser, 2009). These characteristics align with Microservices Guiding Principles, which interact via an interface. Synergy maximizes the service's extensibility. Because the components are independent, they are simple to enhance and can quickly adapt to changing environments
(Valdes A, 2021).

Project Objective

This is a simple Client-Server project that allows for secure communication between a client and a server (Bekker, 2018). The project is written in Python and implements the Socket module and aligns to the five categories of SoS as follows. The data flow diagram highlights the connectivity in Figure 1 - A simple Data Flow diagrams highlighting the connectivity of a smart bulb in a System of Systems

Characteristic Modelling of a Smart Light

Autonomy – The smart bulb is self-governing and possesses light and motion sensor capability.

Belonging – This is part of a bigger structure of home appliances and fits in with other smart devices such as a fridge or TV. Depending on the larger system and the shared goals, the smart bulbs contribution varies.

Connectivity – The client can transmit messages via a smart device to the front-end application, which will then authenticate with the vendor. The smart light will connect over the network and interact with the controller hub.

Diversity – Not only will the Smart Bulb have the features to switch on and off but could have the ability to detect motion, perform a flickering light in emergency and display RGB colours.

Emergence – The smart bulb is a classic example of an emergent system as it conforms to the norms of system-of-system in the following manner :

- Autonomy in its features
- It is integrated as part of the larger Smart Home system.
- It is interconnected to both the client and server, as well as the third-party OEM (Original Equipment Manufacturer).
- The features of the primary use can be extended as per the users behaviour and choice.

Security Hypothesis

Can session encryption provide additional protection on an already secure channel?

Security Discussion

In order to determine if the hypothesis was correct, two variants of a client server application were developed in Python. One had strong session-level encryption, and the other had none.

Tests were performed on both the Encrypted and Unencrypted variants.
These were multi-varied in nature, conducted with packet inspection via Wireshark to determine the obscurity of the data.

On both applications malicious input was entered, in an attempt to see if further data could be derived.

Test Results

From the results in Figure 6 - Wireshark on Unencrypted Data we can see that the requests are susceptible to Man-In-The-Middle attacks via packet capturing tools like Wireshark. Conversely, in Figure 5 - Wireshark on Encrypted Data we can see that the encrypted data is secure from sniffing.

The hypothesis is then proven true, encryption does indeed provide additional protection.

Features

- The server can receive messages from multiple clients concurrently.
- The server can broadcast messages to all connected clients.
- The project includes error handling for different scenarios.

Installation & Startup

1. Please install the latest version of Python 3
2. Please install the following:
    a) pip install pycrypto
    b) pip install sockets

Getting Started

1. Start the server by running **python ServerEnc.py**.
2. Start the client by running **python ClientEnc.py**.
3. Please choose an option from the Command Line Interface.

Contributions
Contributions are welcome! If you would like to contribute to the project, please follow these steps: 1. Please request access for the private repository: https://github.com/ethical-dev-secops/SSA_PCOM7E23 2. Create a branch using **git checkout -b feature**. 3. Commit using descriptive messages. 4. Use Pull Requests

Authors – Group 1
Yash Roongta Demian Berisford-Maynard Deepak Sidhar Okai Dervishev

Screenshots of Test Results and Dataflow Diagram

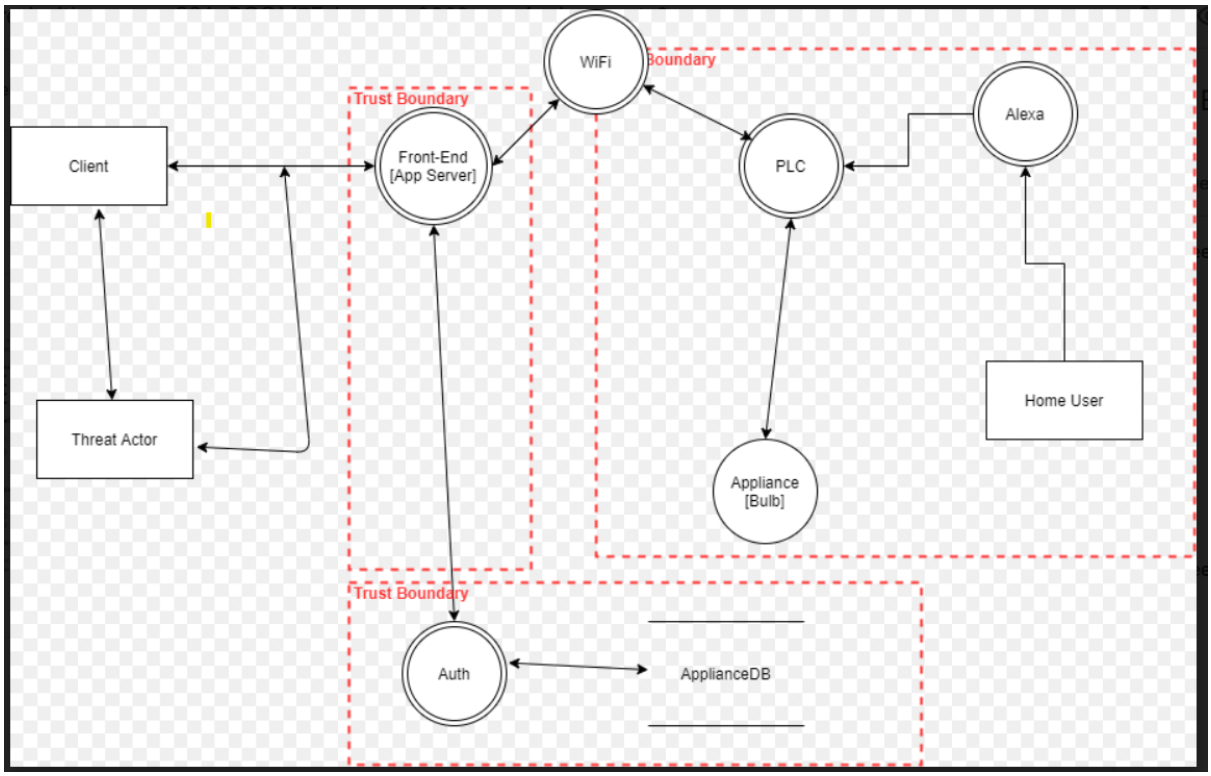

Figure 1 - A simple Data Flow diagrams highlighting the connectivity of a smart bulb in a System of Systems

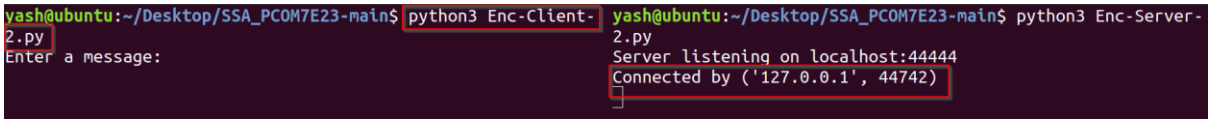

Figure 2 - Encrypted Client - Main Menu

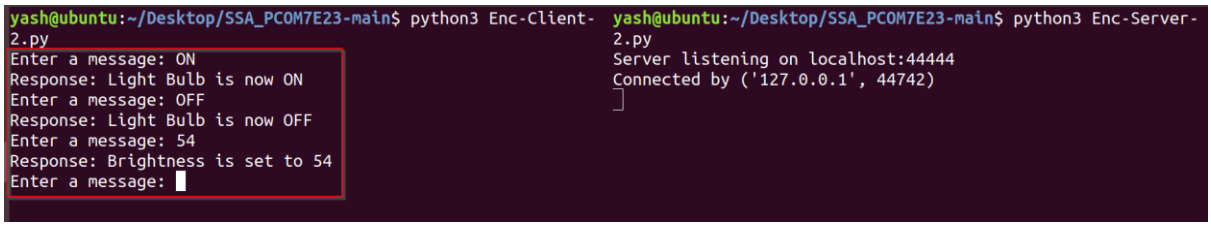

Figure 3 - Encrypted Client - Standard request results

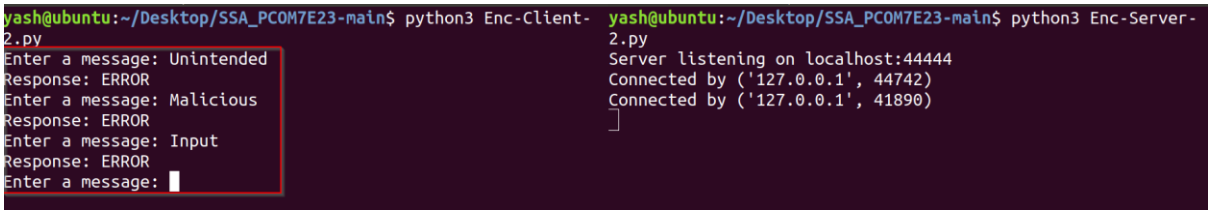

Figure 4 - Encrypted Client - Malicious request results

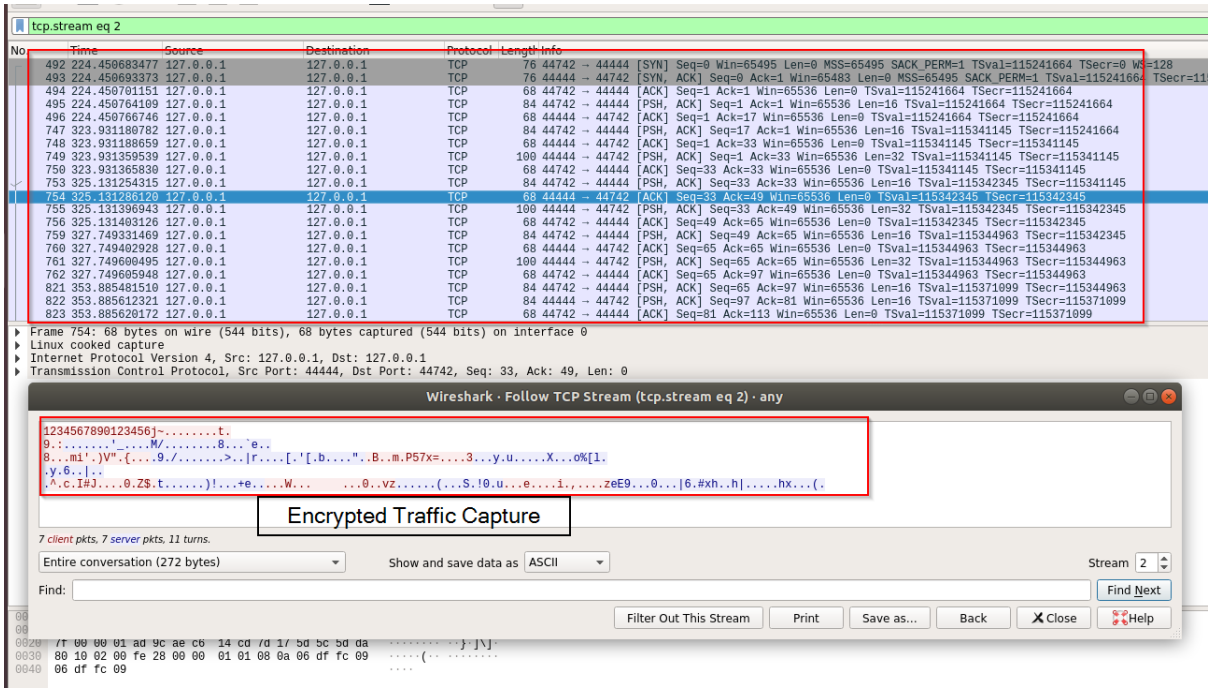

Figure 5 - Wireshark on Encrypted Data

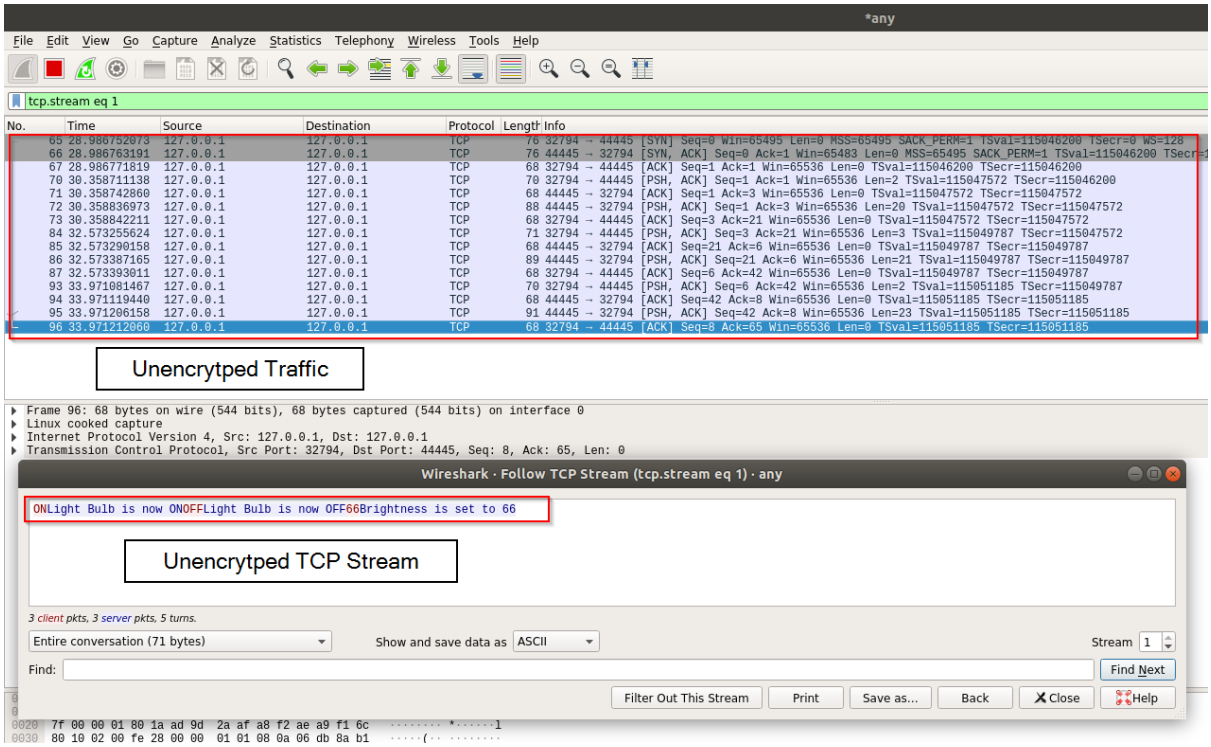

Figure 6 - Wireshark on Unencrypted Data

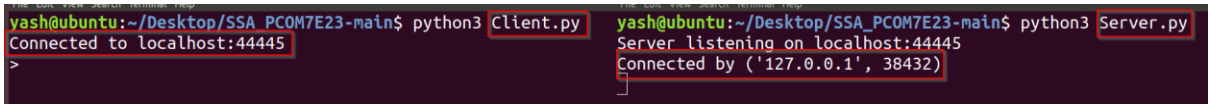

Figure 7 - Unencrypted Client - Main Menu

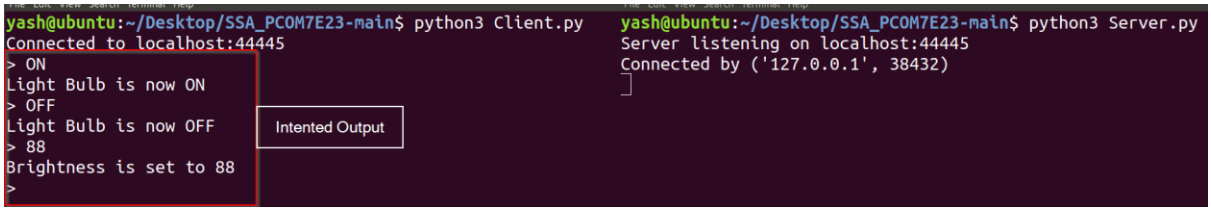

Figure 8 - Unencrypted Client - Main Menu

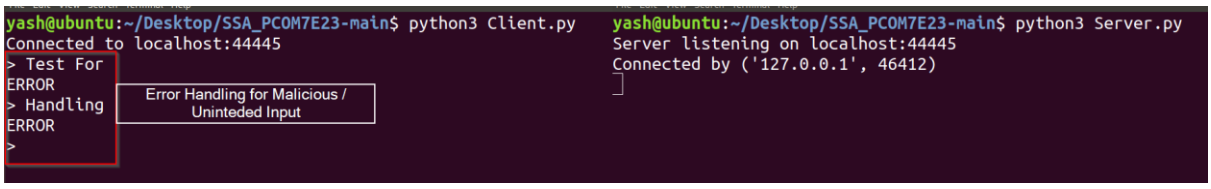

Figure 9 - Malicious request results

References

- Baldwin, W.C. & Sauser, B. (2009). Modeling the Characteristics of System of Systems. In: IEEE International Conference on System of Systems Engineering. IEEE International Conference on System of Systems Engineering : 2–3. Available from: https://www.researchgate.net/publication/224602028_Modeling_the_Characteristics_of_System_of_Systems [Accessed 4 Mar. 2023].

- Bekker, R. (2018) Encryption and Decryption With the PyCrypto Module Using the AES Cipher in Python - Ruan Bekker's Blog. blog.ruanbekker.com. Available from: https://blog.ruanbekker.com/blog/2018/04/30/encryption-and-decryption-with-the-pycrypto-module-using-the-aes-cipher-in-python/ [Accessed 1 Mar. 2023].

- Geeksforgeeks. (2017) Socket Programming in Python - GeeksforGeeks. GeeksforGeeks. Available from: https://www.geeksforgeeks.org/socket-programming-python/ [Accessed 1 Mar. 2023].

- King, M. (2017) Panda Strike. Panda Strike. Available from: https://pandastrike.com/posts/20170105-pycrypto-padding/ [Accessed 1 Mar. 2023].

- Stackoverflow. (2012) python - Encrypt and decrypt using PyCrypto AES-256. Stack Overflow. Available from:

https://stackoverflow.com/questions/12524994/encrypt-and-decrypt-using-pycrypto-aes-256 [Accessed 1 Mar. 2023].

- Valdes, A. (2021) Microservices vs Monolith: The Ultimate Comparison 2022. Available from: https://www.clickittech.com/devops/microservices-vs-monolith/ [Accessed 4 Mar. 2023].