Abertay University

TAP-MFA

Christopher Di-Nozzi, 1800317

CMP408

Introduction

Multi Factor Authentication (MFA) is one of the most effective methods of securing accounts. In its most common form, it consists of a temporary code that is sent to the user via an app or messaging service that is used as part of authentication, usually an initial logon for a new location. With the rise of work-from-home, secure login is more important than ever. However, this app based approach is not the most secure implementation of MFA. It is still susceptible to phishing attacks as well as more untraditional attack methods, including simply annoying the user until they permit access, (Cimpanu, 2021).

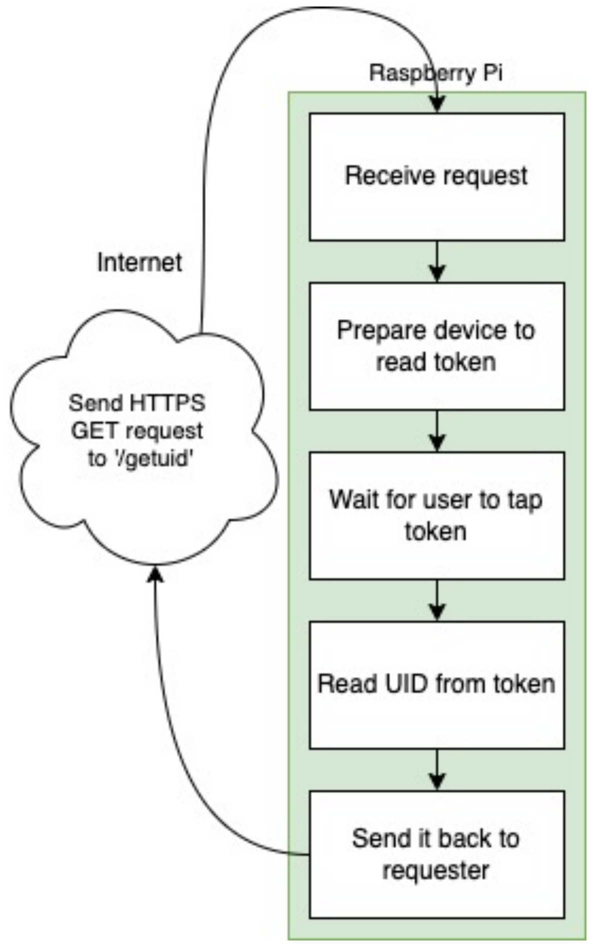
The TAP-MFA project aims to solve this by creating a low cost, hardware based MFA system that works with cloud based applications. The project will have been a success if it can meet the following three objectives:

1. Design and create software to read and process a RFID token.
2. Design and create a website that implements password and hardware token MFA.
3. Design and deploy cloud infrastructure to host the website and supporting database, making the system easily expandable.

Methodology

A Raspberry Pi Zero W was used in combination with an RFID reading HAT to allow the device to read a unique identifier from the RFID token. Two LEDs were used to indicate a 'ready' state of the device, as well as visual feedback when the RFID token had been read. The HAT also included a small OLED display that was used to communicate with the user as to when to tap their token against the device, and the current stage of operation. An image of the final hardware can be seen in the 'Project Highlights' section.

The hardware device was controlled by a short python script. The script ran a Flask webserver over HTTPS that would respond to a GET request sent to '/getuid'. On receiving the GET request, a function was run that readied the RFID reader. Once the user tapped their token against the reader, the UID was sent back to the requester in JSON format. A third-party library was used to write messages onto the OLED screen, and a custom made Linux kernel module was used to turn the LEDs on and off as operation demanded. The general flow of the program can be seen in the figure below left.



Raspberry Pi

Receive request

Prepare device to read token

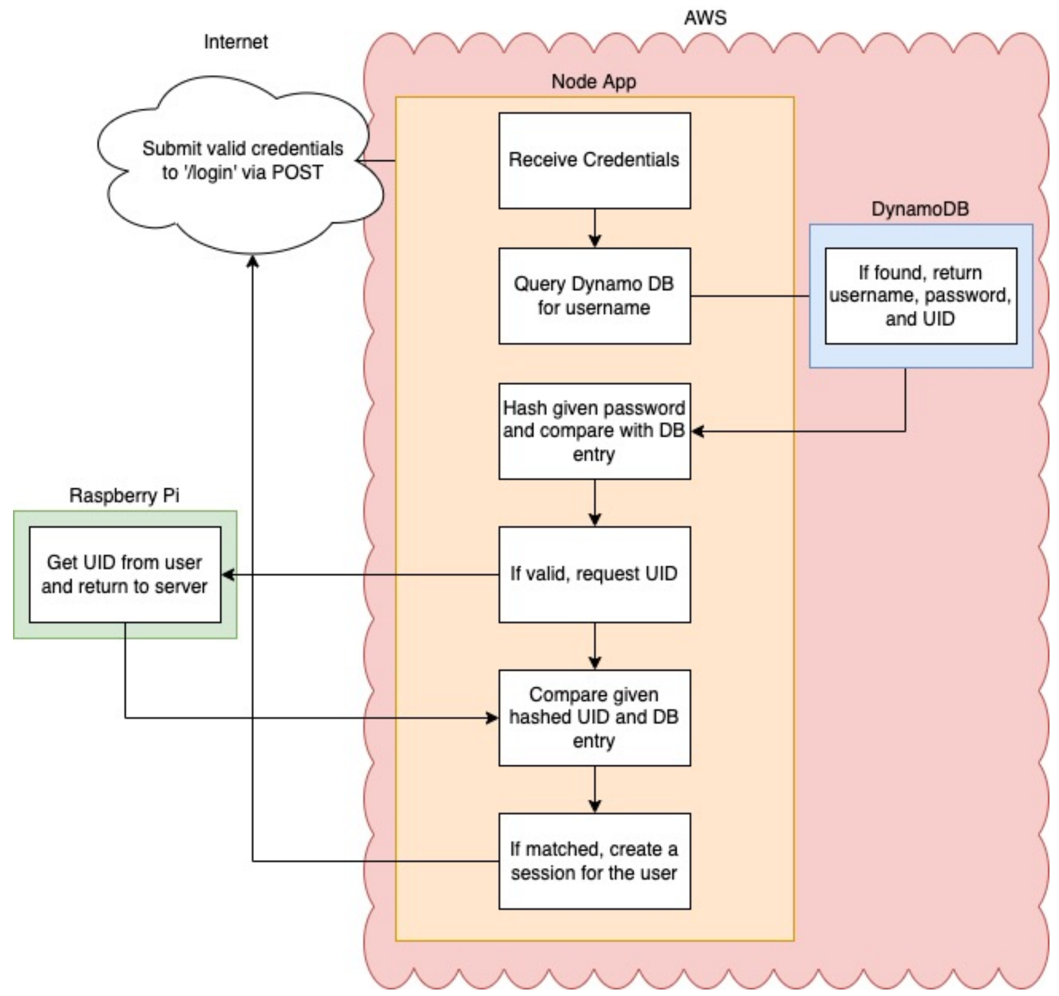
Wait for user to tap token

Read UID from token

Send it back to requester

Internet

Send HTTPS GET request to '/getuid'



AWS

Internet

Submit valid credentials to '/login' via POST

Node App

Receive Credentials

Query Dynamo DB for username

DynamoDB

If found, return username, password, and UID

Hash given password and compare with DB entry

If valid, request UID

Raspberry Pi

Get UID from user and return to server

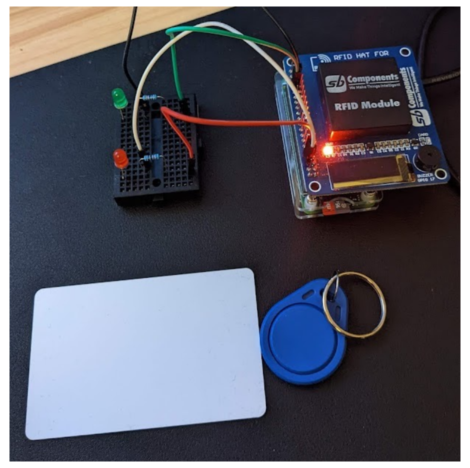
Compare given hashed UID and DB entry

If matched, create a session for the user

A Node JS web app was run as an Elastic Beanstalk instance in AWS and hosted a signup and login system that use the Express library to handle sing up and login requests. For example, after receiving a login POST request with valid credentials, the server would query a Dynamo DB instance, via the AWS SDK, to retrieve the username, and hashed password and UID stored within the DB. The password entered by the user would then be hashed and compared. If they matched, a request would be sent to receive the UID from the device which would also be hashed and compared. If they also matched, a session would be created for the user. This process can be seen within the above right figure.

Project Highlights

This project successfully created and demonstrate a hardware based MFA system, using only low cost parts, that could interact with cloud based applications. This system in its current state could be used as part of an entry system or for accessing highly secure, one-of-a-kind systems inside an enterprise.



The project was also secure, using BCrypt for hashing of passwords and UIDs, as well as HTTPS for the raspberry pi's webserver and security groups within AWS.

Future Work

Implement HTTPS into the web application

Add a buzzer for vocal feedback when scanning

Research expandability and what changes could be made to allow multiple devices to be used by different users in the same network.

References

Cimpanu, C., 2021. Russian hackers bypass 2FA by annoying victims with repeated push notifications. [Online]
Available at:
https://therecord.media/russian-hackers-bypass-2fa-by-annoying-victims-with-repeated-push-notifications/