COMP3000

Computing Project

2020/2021

Project Title

Eye of Horus

Links

Source code: *https://github.com/M-Davies/eye-of-horus*

Backlog Board Invite Link:
https://outlook.office365.com/owa/eyeofhorusPlanner@group.plymouth.ac.uk/groupsubscription.ashx?action=join&source=MSExchange/LokiServer&guid=a322b80f-c38a-44dd-b956-e9b43f82ec87

Project Vision

For:

Users looking to protect their assets within safes or rooms and need a more reliable and secure solution than an easily stolen key or forgetful passphrase.

Whose:

Problem of those who forget their forms of authentication or wish for only a single group of authorised people access to an asset.

The:

Eye of Horus.

Is A:

Facial and gesture recognition hybrid smart lock camera system.

That:

Combines two common biometric security features into one camera. It provides a secure, helpful authorisation system into a room, safe, house, you name it! Without the need to remember keys or passphrases, this camera uses facial recognition to recognise you as an authorised party to the asset as well as gesture recognition to accept a premade unlock pattern. To lock again, simply exit the asset and commit your locking gesture while looking up at the camera, exactly the same as what you've already done! Forgot either gesture? No problem! Simply use our app to scan a QR code that will reset your gesture there and then so you can make a new one!

Different From:

Nest Hello:

https://store.google.com/gb/product/nest_hello_doorbell
A motion sensing internet connected camera (for use as a doorbell). While it does have the potential to be used as a facial/gesture recognition camera, it does not come with it out of the box. It's also not open sourced and the system requires you to be registered with a Google account.

Nuki:
https://nuki.io/en/
A smart lock system, it unlocks the door for you when your smartphone detects that you're close to your home. Not great as anyone could steal your smartphone and unlock your house. It also has no camera, so no potential for face or gesture recognition, and isn't open sourced (although it's documentation API is).

Gate8:
https://www.amazon.co.uk/Gate-Labs-GSLK101-Equipped-Brushed/dp/B077YG79FN
A smart lock system with camera, it allows you to see who is at your door where you can then create a custom pin code for them that unlocks the door. My system differs as it removes the potentially compromised pin code, instead allowing you to add people to access groups, each with their own access levels and gestures.

Risk Plan

The open source gesture libraries are not compatible with the camera's hardware

- There are many open source libraries available for both facial and gesture recognition. If I design the codebase to be dynamic, it should be easier to switch between libraries should one prove to be incompatible with my chosen camera.
- I can also look into changing camera to something that has a friendlier interface

The system is focused on hardware and I have little experience working with hardware

- Most of the libraries I am looking at are well documented, as well as the code language I am planning on using. Should I get still get stuck I can reach out to the community for assistance. From what I have seen, there are a lot of tutorials online for interfacing with peripheral devices.

The camera is connected to the internet by a physical cable, not wireless, leaving it easy to be exploited

- Design the system with the assumption the camera may lose access to the internet. I will be able to build it so that it can detect this eventuality and switch to a backup solution using the mobile app that comes with the system. By displaying a backup QR code to the camera, a user can authenticate themselves offline. The camera will reset this backup code and recover itself when it comes back online.

The software and network security on the camera may not be sufficient for modern standards

- The project is a prototype at the end of the day so it should be acceptable if the security is not uncrackable as it's not a production version. However, I attend to get as close to production level security (physical, software and network) as possible. By

working with my supervisor and considering past mistakes made by IOT companies, I feel I can get close to a "secure enough" solution. Considering the ways this could be attacked:

- The physical USB interface could be interfered with or disrupted, so I would need to consider the restrictions to place on any exposed ports (This includes the power supply); even if the project goes 100% wireless. Maintaining the availability and integrity of the system will be crucial in producing a working solution since these are arguably the two most important CIA concepts in my project.
- Should the code introduce a vulnerability, it will likely tie into the network route, so I'll address both attack vectors here. A mistake in the code could allow any of the CIA principles to be breached if an attacker has the capability to meddle with it. By altering the execution flow of the program via a simple misread conditional, an antagonist could alter the saved faces or cause the system to crash completely (forcing it to rely on the less secure, backup QR code recovery method). They could also potentially disrupt the system's connection to the internet, again resulting in the camera resorting to the backup solution that is easier to crack.

Keywords

Provide keywords for your project to enable searching

Camera, security camera, facial, gesture, recognition, QR, security, mobile, app, lock, unlock, biometrics, authentication, all-in-one, hybrid, passphrase, key, hardware, iot, embedded