Chock-O-Lock: Tech Edition

by
Lexie Aytes, Christian McGee, Tavin Singh

Submitted to
the Faculty of the School of Information Technology
in Partial Fulfillment of the Requirements for
the Degree of Bachelor of Science
in Information Technology

Lexie Aytes 4/20/21

Lexie Aytes Date

Christian McGee 4/22/21

Christian McGee Date

Tavin Singh 4/20/21

Tavin Singh Date

Tony Iacobelli 4/20/21

Tony Iacobelli, Faculty Advisor Date

University of Cincinnati
College of
Education, Criminal Justice, and Human Services

April 2021

TABLE OF CONTENTS

LIST OF FIGURES

ABSTRACT

Insiders pose a real threat to the security of data centers. The Chock-o-Lock is an auxiliary security system that uses facial recognition to authenticate users. The system will require the user to authenticate first at the door to the server room and then again to gain access to each individual server rack. The system will log all attempts to access the data center and server racks, along with other important security information. Security teams will be able to easily control who can enter the data center and regulate the number of people in the data center at one time, restrict the access of each user only to the specific server racks they need to work on, audit the logs recording all data center accesses, and gain a clear picture of data center usage.

1. INTRODUCTION

1.1 Introduction

It can cost information technology companies lots of money to keep up security for data centers. However, keeping data centers safe is important for companies' reputations and continued operations. The best way to keep costs down would be to reduce human resources as much as possible. To this end, our company uses biometrics to verify if users are supposed to be allowed access into the server room. Choc-O-Lock uses facial recognition to determine what kind of access each person going into the server room has, reducing the amount of human resources that are needed and increasing the security of our client's data center.

1.2 Problem

Insiders pose a real threat to the security of data centers and the sensitive information contained therein. According to PandaSecurity, the number of insider threat incidents has increased by 47% over the last two years (Deyan G, TechJury).. The cost of insider threats for organizations in 2020 is $2.79 million (Deyan G, TechJury). Current authentication systems based on factors like PINs or employee badges are not optimal, since both PINs and badges can be forgotten or stolen. Moreover, modern security systems might only require authentication when entering the data center. Anyone who can enter then has unrestricted access to all the server racks inside, even though most employees probably only need to access a few server racks pertaining to their work. This violates the principle of least privilege and suggests that there could be a better way to secure data centers.

1.3 Solution

Choc-o-Lock is an auxiliary security system that uses biometric data to authenticate users. First, the system scans a user's face at the entrance of the data center to determine whether they have access. If the user is authorized to enter the data center, the system will then require the user to authenticate a second time with facial authentication to gain access to each individual server rack. The system will log all attempts to access the data center and server racks, along with timestamps that capture the beginning and end of an access event such as when the user started and stopped using a server rack. The Choc-o-Lock system will enable security teams to easily:

- Control who can enter the data center and regulate the number of people in the data center at one time,
- Restrict the access of each user only to the specific server racks they need to work on,
- Audit the logs recording all data center accesses, and
- Gain a clear picture of data center usage.

1.4 Project Goals

The main goals for Choc-O-Lock are to create a security system that uses biometric data to secure data rooms and server racks. This will happen by using various digital and physical tools.

1.5 Overview

The final report will provide information on how the project was completed. This will include in-depth processes and includes the following sections: design objectives, methodology, budget, timeline, problems encountered, and future recommendations.

2. DISCUSSION

2.1 Project Concept

Our team consists of two Cybersecurity students and one Computer Engineering student. We wanted to create a product that incorporated all our skills. The Cybersecurity students noticed a server room's security could be more efficient by including more automated technology. With the help of the Computer Engineering student's hardware skills and the Cybersecurity students software skills, we plan on developing a product that will improve the efficiency of a server room's security system.

2.2 Design Objectives

Chock-O-Lock is an auxiliary security system that uses biometric data to authenticate users. We will be using Amazon Rekognition to be able to detect and authenticate users. The system will be built around a Raspberry Pi 4. The Pi will contain a camera, relay module, solenoid lock.

Together this will lead to:

- Facial recognition lock to unlock server room door for authorized user
- Security alert if the person who opened the server rack door leaves the room
- Security alert if server rack door is left open for a predetermined amount of time
- Event log database of rack door open/close and who accessed it
- Security alert if more than one person enters the room per authentication

2.3 Methodology and Technical Approach

Choc-O-Lock utilizes many technological features. It involves using facial recognition and authentication, a permission database, and hardware.

Facial Recognition and Authentication

Choc-O-Lock will be utilizing Python to programmably interact with Amazon Rekognition. It will be utilizing Amazon's Rekognition service to detect a face using the video stream from the camera. In conjunction with the permissions database it will be authenticating the user and evaluating permissions.

Permissions Database

The permission database will include employee information, server information, and permissions. It will also use python to connect to the rest of the system, but will be built in mysql.

Hardware

One of the goals with the hardware is to keep it minimum and low cost without negatively affecting the performance of the system. The components that make up the system are essential for it to be efficient. They also keep the system lightweight and modular. Below is the list of the components that create Chock-O-Lock:

- Raspberry Pi 4
- Camera module
- Solenoid lock

- Relay module
- 12 volt power source
- Jumper wires

2.4 User Profile

There are three types of users utilizing Chock-O-Lock. There are two end users: Security Team & Data Center Employee. Data Center Employee is the person that will be accessing the data center or data racks. The Security Team will be utilizing the logs that Chock-O-Lock produces to secure the Data Center. The System Administrator will be ensuring Chock-O-Lock is running smoothly and updating user pictures and privileges as necessary.

Table 1: *Data Center Employee User Profile*

User Profile **Form 1**
Application: Camera
Potential Users: Data Center Employee
Software and Interface Experience: The Data Center Employee will require no extensive experience. The only training they will need is a high-level explanation on how the camera authenticates, ex: Employee walks up to the camera, camera authenticates, the door unlocks.
Experience with Similar Applications: IPhone facial recognition

Task Experience: The employee walks up to the camera, the camera authenticates, the door unlocks.
Frequency of Use: Whenever the employee needs to access the data center and racks. Could be daily, hourly, weekly.
Key Interface Design Requirements that the Profile Suggests: The interface needs to be easy to use. The camera needs to recognize face right away and authenticate it.

Table 2: *Security Team User Profile*

User Profile **Form 2**
Application: Access to database, MYSQL, Python
Potential Users: Security Team
Software and Interface Experience: Running python applications.
Experience with Similar Applications: Needs database and logging experience.

Task Experience: Reacting to system alerts as well as using the log to figure out the security breach.
Frequency of Use: Whenever there is a system alert.
Key Interface Design Requirements that the Profile Suggests: The security team will need to be able to access the security logs whenever they need to.

Table 3: *System Administrator User Profile*

User Profile **Form 3**
Application: AWS, Amazon S3, Amazon DynamoDB, Amazon Rekognition, Amazon Kinesis,
Potential Users: System Administrator
Software and Interface Experience: Users should be experienced with a command-line interface and AWS products. Running python applications.
Experience with Similar Applications: Other command-line interfaces tools,

Task Experience: Using the tools and applications to upkeep employees' photos and privileges.
Frequency of Use: The administrator will need to use the system whenever needed to add new employees, remove employees, and update privileges as needed.
Key Interface Design Requirements that the Profile Suggests: The system administrator will need to be familiar with python and interfaces to ensure the system is working correctly.

2.5 Use Case Diagram

The following diagram, Figure 1, demonstrates the use case for Chock-O-Lock. The diagram shows all possible users with corresponding tasks. It includes Data Center Employee, Security Team, and System Administrator.

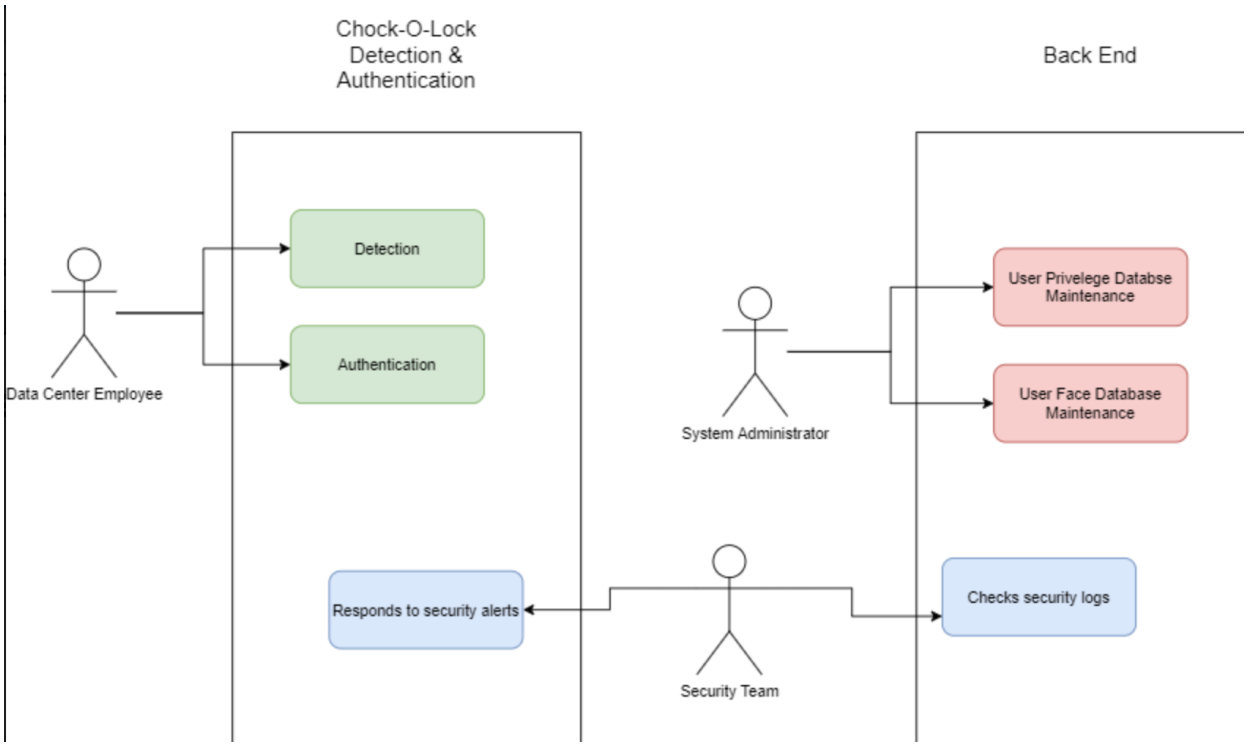


Figure 1: Use Case Diagram

2.6 Technical Architecture

The Choc-O-Lock system will leverage several key technologies: MySQL, Python, Amazon Web Services, and the Raspberry Pi hardware platform. Choc-O-Lock will be utilizing Python to programmably interact with Amazon Rekognition. It will be using Amazon's Rekognition service to detect a face using the Kinesis video stream from the camera. It will then pull from the MySQL permissions database to authenticate the user and evaluate permissions.

The following image shows the hardware set up for Choc-o-lock. It consists of, starting from the right:

- Raspberry Pi 4 with Camera
- Relay Module

- Push button
- Solenoid Lock
- DC Power Source(We used a 9 volt battery)
- Jumper wires

Image 1: Choc-O-Lock hardware setup.

2.7 Testing

The testing plan and report includes the necessary components to fully test the Choc-O-Lock system. This includes our testing methodology and the team's approach to testing. It also includes the use cases and features that were tested, how we tested these functionalities, and the results of these tests. It concludes with what was learned during testing, what was reworked, and what the team would have done differently moving forward.

One aspect of testing will include physical tests such as having a user attempt to access the system and test the responsiveness of our system. Another will be testing the back end of the system, to make sure the user can add or remove users, permissions, and other fields as needed. The database was tested for the most efficient way to log the events that need to be tracked. The database was also tested to see which units of data were least needed to reduce the number of bugs encountered. We utilized unit testing that was predetermined testing. We will also incorporate a group of beta users to test the product. The results of the testing showed that users can edit any and all permissions in our system if needed, as well as access the specific areas they have access to.

To test the functionality of the main Choc-O-Lock functionality, unit tests were created via the GitHub repository. The GitHub also includes the percentage of the unit tests that are passed. Because of time restraints, unit tests weren't completed for the entirety of the code. These unit tests show that the code is working as we expected. Unit testing helps find bugs relating to functionally errors rather than syntax errors.

Image 1 shows the percentage of lines in each file covered by a unit test. The ultimate goal is to have 100% for each file. Choc-O-Lock was not able to get this due to running out of time.

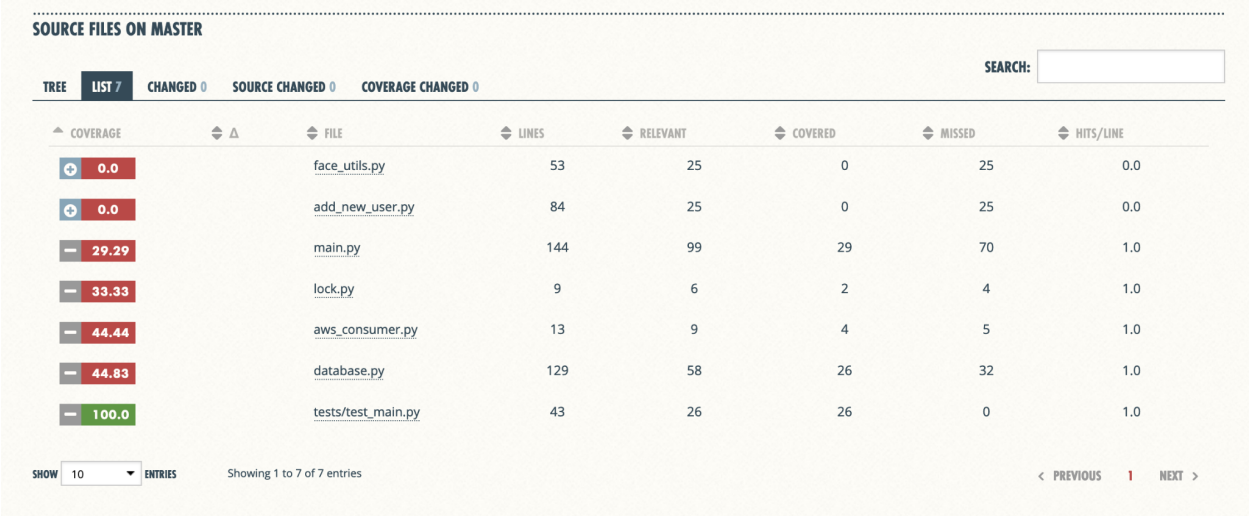


Image 1: Unit test results showing the percentage of lines in each file covered by a unit test.

2.8 Budget

The table below displays the estimation budget for our project. Combining Software, Labor, and Hardware there is a total cost of around $1,214.03. The 'Amazon Kinesis' price is estimated as that is up to the Security Team for how much data they want to save.

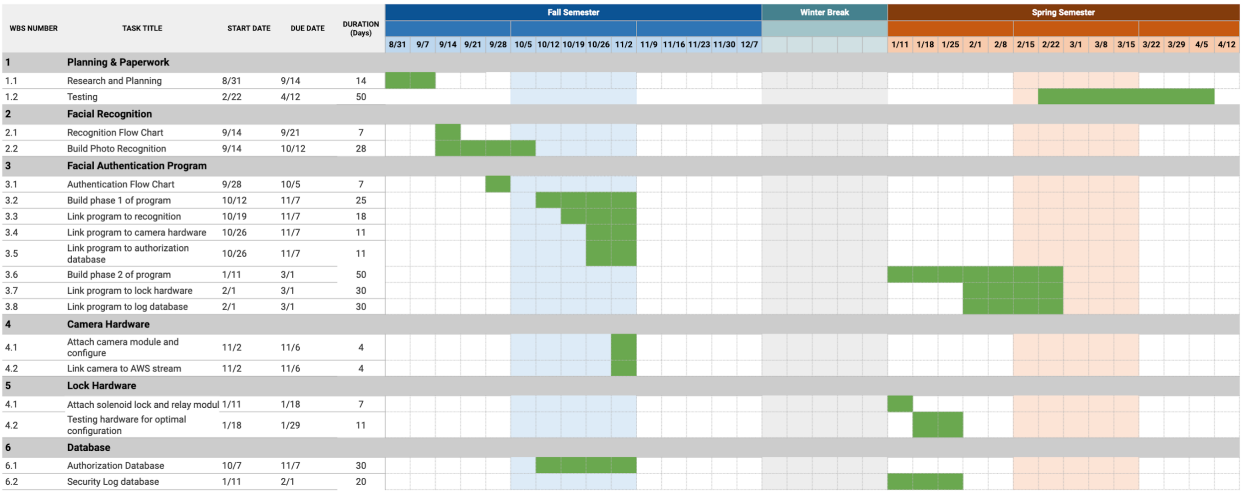
Chock-O-Lock: Tech Edition Budget				
NO.	ITEM	UNIT, HOURS	UNIT PRICE	TOTAL
SOFTWARE				
1	Amazon Kinesis	100 (estimate)	$0.015 per provisioned shard-hour	$156
	Subtotal			$156
LABOR				
4	Logo Design	1	$100 (flat fee)	$100
5	Database Upkeep	1	$800 per month	$800
	Subtotal			$900
Hardware				
	Raspberry Pi 4	1	$99.99	$99.99
	Raspberry Pi Camera	1	$20.58	$20.58
	Solenoid Lock	1	$13.89	$13.89
	Relay Module	1	$6.79	$6.79
	External DC Power Source	$10.99	$10.99	$10.99
	Jumper Cables	$5.79	$5.79	$5.79

	Subtotal			$158.03
	Total			**$1,214.03**

Table 4: Project Budget

2.9 Gantt Chart

Figure 2, shows the Gantt Chart for Choc-O-Lock. Work is divided into the two semesters with winter break in between.



WBS NUMBER	TASK TITLE	START DATE	DUE DATE	DURATION (Days)
1	**Planning & Paperwork**			
1.1	Research and Planning	8/31	9/14	14
1.2	Testing	2/22	4/12	50
2	**Facial Recognition**			
2.1	Recognition Flow Chart	9/14	9/21	7
2.2	Build Photo Recognition	9/14	10/12	28
3	**Facial Authentication Program**			
3.1	Authentication Flow Chart	9/28	10/5	7
3.2	Build phase 1 of program	10/12	11/7	25
3.3	Link program to recognition	10/19	11/7	18
3.4	Link program to camera hardware	10/26	11/7	11
3.5	Link program to authorization database	10/26	11/7	11
3.6	Build phase 2 of program	1/11	3/1	50
3.7	Link program to lock hardware	2/1	3/1	30
3.8	Link program to log database	2/1	3/1	30
4	**Camera Hardware**			
4.1	Attach camera module and configure	11/2	11/6	4
4.2	Link camera to AWS stream	11/2	11/6	4
5	**Lock Hardware**			
4.1	Attach solenoid lock and relay modul	1/11	1/18	7
4.2	Testing hardware for optimal configuration	1/18	1/29	11
6	**Database**			
6.1	Authorization Database	10/7	11/7	30
6.2	Security Log database	1/11	2/1	20

Figure 2: Gantt Chart

2.10 Problems Encountered and Analysis of Problems Solved

We didn't have a great deal of problems as everything ran smoothly for phase one of the project. We did run into a couple of issues relating to documentation and database.

A larger problem we had was the documentation for using Kinesis and Rekogntion was out of date. This required us to do a lot of research outside of the official documentation. A

problem built off this one was that the documentation was built for Python 2.7. I had to do a lot of research to switch it to Python 3.8.

Another issue was when attempting to build GStreamer and Amazon AWS Producer SDK Sample APP for macOS Mojave. This kept failing to install the required dependencies (openssl error). I then tried to build it for Ubuntu 20 using VirtualBox, but when we had to pass the built-in webcam through to the VM, we received an error. Finally, I switched to macOS Catalina and was able to successfully stream footage to a Kinesis Video Streamer.

An issue encountered was connecting the database to a GUI to allow for editing of the backend of the project. Another issue we faced was having to move our database from SQL server to MySQL due to technical constraints.

2.11 Future Recommendations

From the hardware perspective, we would have used a better quality camera. The camera we implemented doesn't always recognize faces in areas with limited lighting. We would also implement a more reliable power source. The 9 volt batteries we currently use have a lifespan of roughly 30 minutes. If we had more time, we would have experimented with different hardware components and/or brands. Although the Raspberry Pi 4 did a great job, we would look into a more powerful processing unit as the workload increases.

For the database, if it had to be done all over again, the database would've first been created using mySQL, instead of being created in SQL Server then having to be transferred over. There would also be improvements made to the efficiency and flexibility of the code used to log

events and edit the database. As of right now there are no plans to continue this project after graduation.

3. CONCLUSION

As a group, we completed a lot for Choc-O-Lock in the Fall semester. We worked with softwares and hardware that we have never worked with before. We encountered some problems along the way but were able to overcome them.

3.1 Lessons Learned

Throughout this project, we learned many valuable lessons about building a product. We realized how important planning and documentation is for a project to be successful. We saw that in the long run, this process can save us time and create an accurate completion date for the project. Having clear requirements and functionalities helps the process go along a lot smoother. Budgeting is also something we learned to consider when creating a product. We had to learn how to decide what functionalities are essential for the product and how to implement it while still keeping the price low. These are all things we learned throughout our project.

3.2 Abilities and Skills Developed Throughout Project

We learned how to develop a python application and bring together external data. We were exposed to lots of new softwares throughout this project. This includes Python, Amazon Rekognition, Amazon Kinesis, and Amazon S3.

REFERENCES

4.1 References

G, Deyan. “20 Insider Threat Statistics to Look Out For in 2020.” *TechJury*, 17 Aug. 2020, techjury.net/blog/insider-threat-statistics/.