Leviton Manufacturing Co., Inc.

Chalida (Anita) Ruangrotsakun

Computer Science – Systems Option

Summer-Fall 2021 Internship

10 December 2021

Table of Contents

Introduction

Company History

Leviton Manufacturing Co., Inc. was founded in 1906 by Isidor Leviton and to this day remains a private, family-owned business offering over 25,000 products electrical, lighting, energy management, and data networking products for residential and commercial markets in over 100 countries. Leviton has 31 locations around the world and currently employs almost 7000 people. The Tualatin office specializes in the Controls portion of the Lighting & Controls division and houses a manufacturing facility in addition to the R&D Engineering Department.

Intern Role

I was an intern in the R&D Engineering Department at Leviton. I primarily worked with one team to complete several projects for a room controller system they were developing, but I also had the opportunity to work with another team to evaluate several web vulnerability scanning tools.

Project List

- Certificate Manager
  - Researched TLS/SSL certificate and key infrastructure
  - Evaluated Lemur, open source certificate management framework
  - Implemented authentication proxy for integrating Lemur and Leviton mobile application
  - Created Tkinter GUI to demonstrate how the Lemur API can be used by the mobile app team
- Room Controller Firmware Update
  - Implemented faster room controller firmware upgrade process
- AcquiSuite Web Application Scanning
  - Evaluated a variety of commercial and open-source tools that offer fuzzing and penetration testing of web applications
- ST Time of Flight Sensor AI Exploration
  - Learned about time of flight sensor technology and use cases
  - Learned about machine learning processes and tools
  - Collected data using a time of flight sensor
  - Wrote Python scripts for processing the raw data
  - Tried out a variety of model training approaches and tools (Edge Impulse, Nano Edge AI Studio, Keras, Autokeras)

Executive Summary

Overview of Projects

My main project involved setting up Lemur, a certificate management system capable of generating and organizing all of the TLS certificates and keys that will be used to install room controllers for lighting systems. After Lemur was set up, I implemented some proof of concept applications using Tkinter and Flask to show how the certificate manager can be integrated with the other lighting system components.

Additionally, I completed several smaller projects over the course of my internship. I programmed the ability for the room controllers to receive firmware updates over a secure wireless protocol. I evaluated whether data collected using an ST Time of Flight sensor can produce a workable machine learning model for detecting the presence of people. And finally, I evaluated several web application vulnerability scanning tools—specifically Qualys WAS, WebInspect Fortify, and OWASP ZAP because they provided fuzzing and penetration testing capabilities.

Results and Accomplishments

My first major accomplishment was successfully setting up the certificate management system and providing several supporting applications to show how it can be integrated with other important components of the room controller system. About four months into the internship, I presented a working demo and detailed overview of this project to my mentors, managers, and members of the remote mobile app team.

My second major accomplishment was successfully implementing a way to upgrade the room controller firmware using a secure wireless protocol. This process consistently takes about one minute rather than the current method, which reportedly takes around five minutes. This feature will greatly reduce the time that field technicians spend upgrading the firmware on the room controllers, freeing up their time to focus on other important tasks.

My third major accomplishment was providing my evaluations of web application scanning tools to another team. I presented them with options for quickly finding critical vulnerabilities in their web application, allowing them to fix those bugs before their project went into production.

My last major accomplishment was providing some of the preliminary groundwork for determining the feasibility of using machine learning and time of flight sensors for people detection.

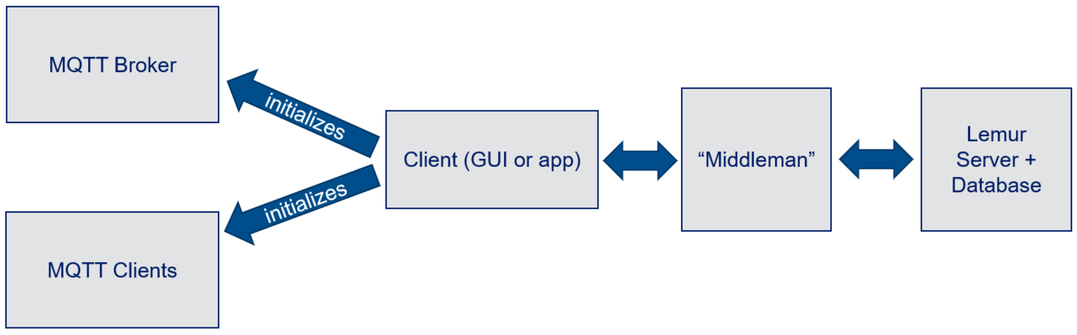
Project Descriptions

Certificate Manager

At the start of my internship, my team was in the process of developing a new feature that would allow room controllers to communicate with each other. However, there was no easy way to quickly generate TLS certificates and keys in order to secure these communications. My first project involved setting up Lemur, an open-source certficiate management framework developed by Netflix.

I installed Lemur on an AWS EC2 instance, obtained a public-facing Let's Encrypt certificate, and configured Lemur to be accessible from an HTTPS URL. Lemur comes with its own web browser interface, but the project managers wanted to know how the existing Leviton mobile app would be able to connect to Lemur. I developed a very basic GUI using the Python Tkinter library to demonstrate how any type of client interface can make use of Lemur's API endpoints in order to create, retrieve, and delete certificates and keys. Because the GUI was simply a collection of Python scripts, I showed how the application can also be run from a tablet using the Pydroid3 app.

The final major concern about integrating Lemur into the room controller ecosystem was user authentication. We knew that the app team maintains a database of users, projects, and permissions and that we did not want users to log in to Lemur see the certificates and keys from projects they are not authorized to see. The solution my mentors and I came up with was to set up a "middleman" in between the app and the Lemur server. I implemented a proof of concept application using Flask and SQLite to show how the "middleman" can maintain a database to correspond app user logins with a single, organization-wide Lemur login and enforce user permissions. This "middleman" ensures that users with operator permissions would be allowed only to retrieve certificates and keys; users with admin permissions would be allowed to retrieve, create, and delete certificates and keys; and all users would see only the resources that belong to their organization.



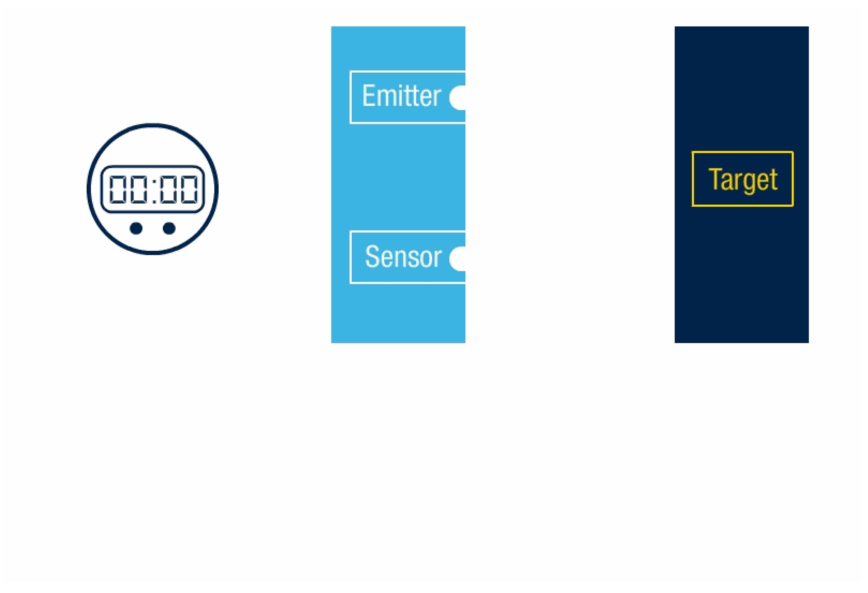
Room Controller Firmware Update

I was also tasked with figuring out a way to complete firmware updates for the room controllers using a secure wireless protocol. One of my mentors was curious to know if it would be quicker than the existing firmware upgrade process.

I first had to implement a toy example of transmitting files of about 1.7MB and 3.2MB to a sender and a simulated room controller using Python scripts. I wrote a function to calculate a simple CRC value from all of the received bytes as the simulated room controller received them. I later created a C version of that function that was incorporated into the room controller's firmware. After a series of tests, my mentor and I confirmed that the room controllers were capable of receiving firmware update files using the secure wireless protocol and the entire upgrade process consistently took under one minute, which is significantly quicker than using the existing method.

ST Time of Flight Sensor AI Exploration

This exploratory project was borne from my manager's curiosity in using machine learning and time of flight sensors to detect the presence of people in a room more accurately while still preserving privacy. A time of flight sensor emits photons that bounce off of surfaces back to the sensor's receivers, and the sensor can determine the distance between itself and the object(s) in front of it based using the speed of light and the amount of time it took for the photons to return. Using a time of flight sensor for people detection would help preserve the privacy of a room's occupants as it does not collect audio or visual data.



Time of Flight (ToF) Sensors, FlightSense sensors, ToF sensors - STMicroelectronics

My manager provided me with the resources from STMicroelectronics: a time of flight sensor, a simple GUI application for data collection, and some documentation. I used these resources as a starting point as I researched different tools and approaches I could use to train a people detection model. In the end, I tried four different tools and had varying degrees of success.

I first created a few datasets using the ST GUI: detect_hand, detect_body, and detect_bear. For each dataset, I recorded short samples in which the object (hand, body, or teddy bear) was in front of the time of flight sensor, as well as samples in which other objects or no objects were in front of the sensor. For each of the tools that I tried, I quickly realized that the data format that was outputted by the ST GUI was not the format that was expected or accepted, so I wrote a couple of Python scripts that transformed the raw data into the appropriate formats.

The first category of tools I explored were low-code machine learning platforms, which are applications designed for developers who are not machine learning experts. The first one I tried was Edge Impulse, a website for developing machine learning models for embedded devices that is free for developers. I was able to successfully upload my data, but the website could not process the data in order to generate features required for model training. The second one I tried was Nano Edge AI Studio, developed by Cartesiam, which was acquired by STMicroelectronics. I obtained a time-limited trial version of their software and successfully trained two models using the detect_hand and detect_body datasets I created. However, this application hid a lot more of the model training aspects from the user and had very limited options for evaluating the models despite strong concerns about them overfitting to the data.

The second category of tools I explored were machine learning libraries—specifically, Keras and Autokeras. One of my initial impressions was that time of flight data can be categorized as time series data. I looked for and followed tutorials the showed me how to use Keras create a long-short term memory (LSTM) model that is well-suited for learning from time series data. However, no matter the data I fed to the model, it would not train to an accuracy beyond 0.5, which means its predictions are no better than random guessing. My final approach was to try training an image classifier using Autokeras. I had to transform the data into the format expected of images, which meant I was treating the time of flight sensor data like low-resolution images. However, I once again failed to train the model beyond an accuracy of 0.5.

In sum, only one of my four approaches to training a person detection model from time of flight data was successful, but my main accomplishment here was laying out some of the groundwork for evaluating the feasibility of this type of project. I wrote several data processing and model training Python scripts that can be utilized again in the future, potentially on more robust datasets or to evaluate a different use case.

Conclusion

Lessons Learned

Over the course of this internship, I learned a lot about how real-world, multi-disciplinary, product-oriented projects are planned and executed. Being part of the room controller team, I experienced first-hand how my mentors and I had to pivot our development efforts to meet updated specifications from project managers, as well as coordinate our efforts with those of remote teams to ensure that all of our work will be interoperable down the road. I have also learned a lot about the importance of thorough testing to help our team find the majority of the bugs as we develop, thus increasing the quality and robustness of the product long before it ever reaches the customer.

Intern Benefits

This internship has helped me greatly develop my technical and communication skills. I have learned to take a big-picture approach to engineering projects to see the how different components of the entire system will work together to provide a solution to an existing problem. This has helped me to see how my assigned projects help contribute to the larger solution. I have also learned the importance of effective communication—specifically, in understanding the gap in what my audience—such as managers or colleagues from different teams—knows compared with what I know about my project and figuring out how best to lay out the information that will get us all on the same page so we can have a meaningful discussion.

Company Benefits

I hope that the various projects that I worked on will prove useful to Leviton, particularly the work I did for the room controller system. I have high hopes that the addition of Lemur will eliminate the need for anyone to manually generate and safeguard their own TLS certificates and keys, which will make the room controller installation process quicker and the entire lighting system more secure. Demonstrating that firmware upgrades are possible over a secure wireless protocol will also help reduce the amount of time dedicated to maintaining and upgrading the room controllers systems that will be deployed. Teams that must develop a web browser interface now knows of several options for automatically finding vulnerabilities in their software before deploying it. And finally, my manager has a starting point for further exploring the feasibility of creating person detection devices using time of flight sensors.

Buzzwords

Autokeras – An AutoML system developed by DATA Lab at Texas A&M University to make machine learning more accessible (AutoKeras)

AWS – Amazon Web Services, which offers many cloud services

C – A programming language commonly used for embedded programming

CRC – Cyclic Redundancy Check; a technique used to detect errors in data

Docker – Software platform that allows developers to more quickly and automatically build, test, and deploy applications in containers

Flask – A micro web framework written in Python

Keras – An open-source deep learning Python library (Keras)

Python – A high-level programming language

SQLite – A library that allows developers to create a self-contained, serverless SQL database (SQLite)

Time of Flight Sensor – A device that measures distances by calculating the time it takes to send and receive reflected photons

Tkinter – A Python library for creating a graphical user interface (GUI)