Security in Internet of Things Devices

October 2020

By

Kai Christopher Michael Tindall

Student number 201842748

Word count: 2902

Contents

1. Project background and purpose

1.1. Background

Internet of Things (IoT) devices are becoming affordable and widespread throughout the market. In 2019 the number of IoT devices on the web became 26.66 Billion and is expected to rise to 31 Billion in 2020 (Maayan, 2020).

With this new mass of devices comes security risks, and the average person isn't aware of good security practices. Therefore, it is important, especially when dealing with sensitive information, that proper methods are put in place to protect people's data.

I will be attempting to solve this problem with a client-server model, this is because the nature of IoT devices mean there will potentially be 10s of devices, maybe even a hundred if it's a particularly big house, so I will be connecting the server to them all, and clients can come and go and not have to worry how many IoT devices are connected.

This will also mean that people should be able to access their internet of things devices' data remotely as it would be possible to port-forward an external router port to the internal IoT server. This would make access to remote location's data very convenient and would be harder to do on a peer-to-peer basis.

1.2. Objectives

The project will seek to achieve in creating a quick and easy to use environment that will allow users to manage their IoT (Internet of Things) devices securely.

Primary Objective

Primary Objective	Justification
To create a driver software for the IoT devices to be able to talk to the command and control server.	The driver software allows the IoT devices to communicate to the command and control server securely using encryption. This is needed for data to not be usable if intercepted in transit by a hacker.
To create a command and control server to manage and communicates to the various IoT devices.	By having the command and control server do all the management of the IoT devices it lightens the load of the client, which means it will then be "dumber" and therefore easier to develop for end users.
To create and document an API that is exposed by the command and control server.	Using an API to transmit and receive commands and data between the client and the command and control server means that each implementation of the client doesn't need to make sure it handles the business logic correctly and means that data will not be different between clients.
To create an example client that will communicate to the command and control server via the API.	This will allow users to be able to see and use the system without having to invest time developing a client for their chosen platform first.

Secondary Objective

Secondary Objective	Justification
To create three interchangeable secure communication methods that the user can choose from within the client.	This will then allow the user to pick and choose how they want their data to be encrypted depending on their individual circumstances.
Evaluate the best method of encryption for speed.	This will then allow me to recommend the chosen method to users who need a lot of data encrypted, but don't necessarily need it to be the most secure.
Evaluate the best method of encryption for security.	This will then allow me to recommend the chosen method to users who need the highest level of security available for their data and don't necessarily care if it makes it a bit slower.

1.3. Scope

I will not be producing my own new encryption method, instead I will primarily be using AES (Advanced Encryption Standard) because it is widely used by authorities on security such as the US NSA.

I will also not be writing my own implementation of the encryption methods, instead using reputable libraries, this is for a similar reason to my previous point, as I may make mistakes in my implementation that leaves my encryptions either broken or vulnerable.

I will be using a client-server architecture for communication between IoT devices and the command and control server and also for the communication between the client and the command and control server, this is to separate the business logic and management of the devices from the user client to the command and control server.

I will be creating a standalone command and control server program that will expose an API for clients to connect with.

I will only be creating one client; however, I will design the environment so that the server does not care or even know what the client is. This means that any number of different clients for different platforms could be written.

I will initially only be allowing one kind of encryption to be used, but if I complete my primary objectives, I will be writing two more that can be interchangeably used to encrypt traffic at the user's discretion.

I will be using the Raspberry Pi model 3B for the IoT devices because they are cheaper than the current best model, but also have the resources that would be required, most importantly they come with WiFi built-in which most IoT devices would be.

I will also be using a Raspberry Pi model 3B for the command and control server because it is still very powerful in a small form factor while being more affordable.

I will be writing the IoT driver and the command and control server software in C++ because it is very portable and can be compiled almost anywhere, and because it is a fast language, which is important in encryption. I have also chosen C++ over C because I think the slight difference in speed C would provide over C++ is compensated enough in development time with the addition of classes in C++.

1.4. Deliverables

I will be delivering three pieces of code and documentation to go with it:

IoT Driver

This is the code that will be running on the IoT devices, the main purposes of this code will be to transmit and receive communications with the command and control server, and then act upon commands received, such as gather data and transmit it, or to receive and deploy a new workload.

Command and Control Server

The command and control server will act as a manager of the IoT devices and will issue commands to them based on what input it gets from the API.

End user client application

The client application can will be the main interface between the user and the system. It will get all its data from the command and control server using the API to poll the command and control server in a set interval. Given the intended application of the environment isn't that many devices, bandwidth should not be an issue for the command and control server to send all the data at once every so often.

Documentation

I will also be delivering two pieces of documentation as well. These consist of a user guide, detailing how to set up and use the environment, and an API guide which will outline all of the API calls for the user to be able to create their own client program if they want to.

1.5. Assumptions

I will be assuming that the end user would be competent enough to set up the environment with some instructions. This may include some command line use or using secure shell (SSH) to remotely access a device. I think this assumption is justified because most people interesting in setting up an IoT network have usually had some experience in this already. There would also be instructions too to further eliminate this factor.

I will be assuming every packet sent over a network is being listened too by someone who is competent and would want to use data gained for malicious purposes, even if such an entity doesn't exist. This is because it will constantly hold the encryption to the highest standard.

I will be assuming that I will have access to Raspberry Pis and that the global supply of them doesn't diminish to a point I won't be able to get a hold of some.

2. Project rationale and operation

2.1. Project benefits

Benefit	Beneficiaries
A secure way to communicate with IoT devices	The end user
A project that can hopefully be graded highly.	Myself, my employer, the university
A framework to create a client program for any platform.	The end user
Expanding my knowledge of C++	Myself, my employer
Expanding my knowledge of working in Linux	Myself, my employer
An environment that will allow users to deploy workloads to an IoT device without having to remote into the device.	The end user

2.2. Project operation

I will be using an agile methodology to develop the project and the software, being agile means that instead of planning everything out in one big go and never being able to turn back, you can change and alter the design as you go and you essentially end up creating prototypes after each sprint that will get progressively better and better until you reach a prototype that can be called the finished product.

This methodology is very good at being able to adapt to changes, be that disruptions or changes in the requirements, etc. This is very useful considering the turbulent times currently occurring due to Covid-19 where there is a very big risk of changing circumstances.

I will be measuring the success of using agile by checking my tasks against the Gantt chart and making sure that I am not falling behind.

For source control, I will be using git; git is a free tool that allows users to stage and commit code into a repository, which can then be backed up remotely. This allows users to go back through commits to “rewind” their code. It also allows users to make a branch of their code, which allows them to make changes to the code and not worry about it as if it goes wrong they can always revert to the master branch and try again, however if it succeeds then it can be merged back into the master branch.

Source control is very important because it means that if my code ever breaks, I always have a previous working version to fall back on. With remote backing up too with a service like GitHub, it means that even if I lose everything on my personal device I still have a copy in the cloud.

2.3. Options

There is a wide variety of Integrated Development Environment (IDE) tools that I can pick from for developing C++. I will look at the most popular available and choose based on my own experience with them and how easy they are to use and how well they might fit my project.

I could also choose between Test Driven Development (TDD) and writing unit tests after the code has been written. I will choose between these two methodologies once I have had some experience with the unit testing framework I will be using and see how easily unit tests can be implemented.

I could also choose between making my example client on Linux or Windows, I have more experience making desktop applications on Windows but if it is easy enough to do in Linux then I might consider making it on Linux as then the entire ecosystem would be Linux based then which would make it easier for people to access as they wouldn't require a Windows machine.

2.4. Risk analysis

Risk	**Likelihood (1 – 3)**	**Severity (1 – 5)**	**Impact**	**Mitigation Plan**	**Recovery Plan**
Covid-19 restrictions increase to a point I can't physically get onto campus	2	1	2	Use digital resources wherever possible and do not rely on having to use university hardware.	Extract all required data from university and talk with supervisor for potential of loaning any needed equipment.
I might fall ill and not be able to develop for a while.	1	4	4	Try not to be in places that may cause me to contract an illness. Secondly, build time into the project plan to allow for the timeline to shift.	Consult Gantt chart to adjust the timeline. If very close to hand in date and illness is particularly severe might be able to negotiate a deadline extension.
Data corruption / Loss of work locally.	1	5	5	Use digital backup tools such as GitHub to make sure that work always has a place that it can be recovered from.	Pull down the copy of the repository from GitHub.
Project veering off course from intended goal.	3	2	6	Make sure I'm always working to a Kanban task that has connections to a requirement. Secondly, my weekly meeting with my project supervisor will give a	Revert to a previous commit and re-assess the relevance of the tasks on the Kanban board.

				second pair of eyes to make sure I’m keeping within scope.	
Not having enough development time to implement the full system.	2	4	8	Regularly consult the Gannt chart and plan extra time to allow for this. If extremely severe then reduce the scope of the project.	Extend the timeline to give myself more time for development. If severe, cut out functionality yet to be implemented.

2.5. Resources required

I require two Raspberry Pi 3Bs to act as my IoT devices to communicate to. If I can't manage to get a hold of them, I could always run them as a virtual machine, however I think it would be better to run them "bare metal" as sometimes virtual machines might not act 100% accurate to their real life counterpart.

I can run the rest of the software on already available hardware that I have access to.

3. Project methodology and outcomes

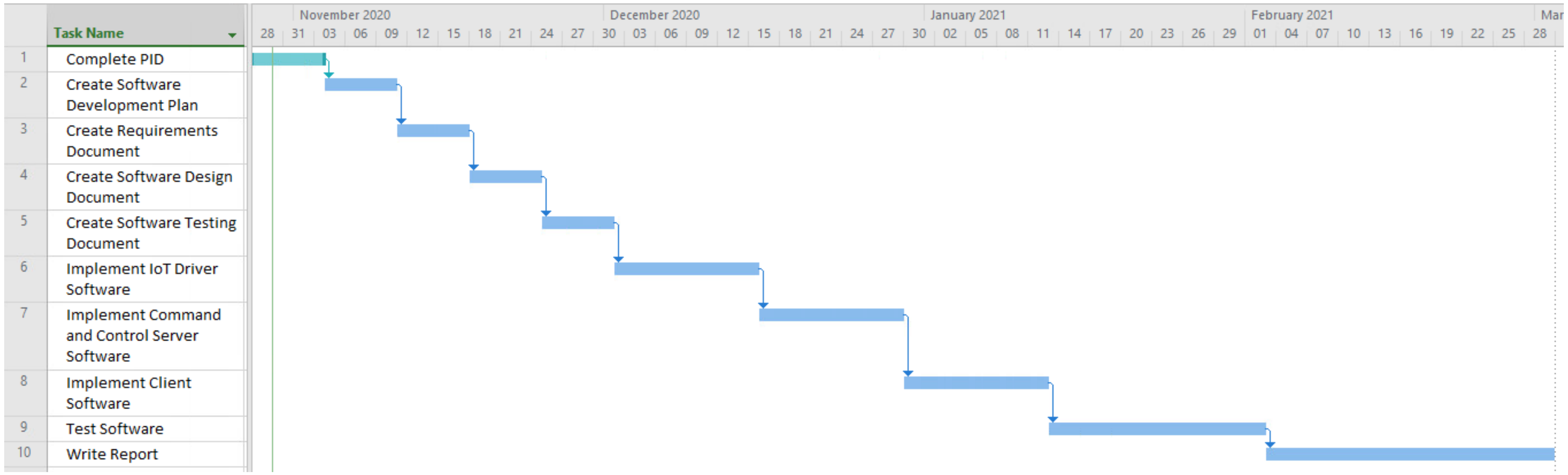
3.1. Initial project plan

3.1.1. Tasks and milestones

Task	Description
Complete PID	Finish writing the Project Initiation Document (This document).
Create Software Development Plan	Create a document that outlines how I will be developing this suite of software.
Create Requirements Document	Create a document that outlines all the requirements that each of the programs will be defined by.
Create Software Design Document	Create a document that shows the design of each of the programs.
Create Software Testing Document	Create a document that outlines the tests that will be ran to fully qualify the programs.
Implement IoT Driver Software	Write the software that will run on the Raspberry Pis that will allow them to communicate with the command and control server.
Implement Command and Control Server software	Write the software that will run on the command and control server device that will expose the API and communicate to the IoT devices.
Implement Client software	Write the example client program that the user will use to communicate with the command and control server using the exposed API
Test software	Testing the software against the software testing document to fully qualify the programs.
Write report	Write the final report for university that will detail how my project has gone.

Milestone	Description
PID Submitted	Submit the Project Initiation Document.
Documentation Complete	Completed all the initial engineering documents.
Implemented Software	Have written the code for all the programs.
Software Fully Qualified	Have all the programs passing the tests laid out in the software testing document.
Report Submitted	Submit the final report.

3.1.2. Schedule Gantt chart



	Task Name
1	Complete PID
2	Create Software Development Plan
3	Create Requirements Document
4	Create Software Design Document
5	Create Software Testing Document
6	Implement IoT Driver Software
7	Implement Command and Control Server Software
8	Implement Client Software
9	Test Software
10	Write Report

3.2. Project control

Because I'm using an agile methodology, I will be implementing this by using GitHub projects. GitHub projects allows users to create project boards on your git repositories, I will be utilising this to create a Kanban board for each development sprint. My sprints will be 1 week long, so this should mean that for a task that might take three weeks there will be three sprints.

I will also be able to link branches and pull requests into tasks on the project board to be able to automate the workflow of the board, meaning I won't have to manage it because as I commit and pull code in, it should manage itself.

I will be keeping track of these tasks against my Gantt chat to make sure I am keeping up on the pace I should be working on. This is how I will measure if I am being successful in my use of the agile methodology.

I will also be using GitHub actions as a continuous integration mechanism which should mean that as I commit and push my code to GitHub it should try to check that the code builds, and alert me if it doesn't.

3.3. Project evaluation

I will be evaluating my project by continually running unit tests throughout development first and foremost. This will make sure that I am heading in the right direction while developing and will help to make me aware if any of my new code breaks the functionality.

I will also be testing my code against a software test plan that I will be creating, this will be used as the final check against the software that it should be working as intended. If those tests pass, then I shall move on to evaluating whether the project was successful.

I will first of all be evaluating the project based on if I managed to get a functional piece of code developed in time, and then move on to reflect on how the methodologies I used either helped or hindered me. I will try to create some lessons learned based on how I feel different stages of the project went and how the tools I used worked for me.

4. References

Maayan, G.D., 2020. The IoT Rundown For 2020: Stats, Risks, and Solutions - [WWW Document]. Security Today. URL https://securitytoday.com/articles/2020/01/13/the-iot-rundown-for-2020.aspx (accessed 11.3.20).