CENTRALIZED HUB FOR SECURING A NETWORK OF IoT DEVICES

2020-086

Project Proposal Report

IT 17078306 – Jayawardhane H.N

Bachelor of Science (Hons) Degree in Information Technology
Specialized in Cyber Security

Department of Information Systems Engineering

Sri Lanka Institute of Information Technology

Sri Lanka

February 2020

CENTRALIZED HUB FOR SECURING A NETWORK OF IoT DEVICES

2020-086

Project Proposal Report

Bachelor of Science (Hons) Degree in Information Technology
Specialized in Cyber Security

Department of Information Systems Engineering

Sri Lanka Institute of Information Technology

Sri Lanka

February 2020

Declaration

I declare that this is my own work and this proposal does not incorporate without acknowledgement any material previously submitted for a degree or diploma in any other university or Institute of higher learning and to the best of knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text.

Name	Student ID	Signature
Jayawardhane H. N	IT 17078306	

The above candidate is carrying out research for the undergraduate Dissertation under my supervision.

Signature of the Supervisor: Date: 2020/02/25

Abstract

With the advancement in technology, network of devices that are interconnected but doesn't require human interaction came into life, namely "Internet of Things". Along with wireless communication, arose the threat of the security of the information being compromised over the network. Despite of the evolution in the security mechanisms used to compensate and mitigate threats arose there are many loopholes which the modern-day attackers have discovered which puts the users into a vulnerable state. In the context of a small office, home office (SOHO) environment where IOT devices are vastly in use securing the user's information is of top priority. It is evident that most attacks occur due to the devices not being up to date. Therefore, this area of research will be mainly targeting as to how to mitigate these potential risks from occurring by automating the functionality of updating the device. Going further into the post-updating phase, the research also proposes a mechanism using blockchain and machine learning modules to detect any instability in the system post-update to bring back the system into its fully functioning state.

Keywords: Internet of Things, Information Security, Blockchain, Machine Learning, Wireless Communication

TABLE OF CONTENTS

Declaration ..... **i**

List of Figures ..... **iv**

List of Tables ..... **iv**

List of Abbreviations ..... **iv**

1. Introduction **1**

1.1. Background 1

1.2. Literature Review 2

1.3. Research gap 6

1.4. Research problem 7

2. Objectives **8**

2.1. Main Objective 8

2.2. Specific Objectives 8

3. Methodology **10**

3.1. Automating and optimizing the process of updating devices 11

3.2. Establishing an update verification mechanism 12

3.3. Establishing a rollback mechanism in case of detecting system instability 12

3.4. Gantt Chart 13

4. Project Requirement **14**

4.1. Functional requirements 14

4.2. Non-functional requirements 14

5. Budget and justification **15**

5.1. Total Expenses 15

6. References **16**

List of Figures

Figure 1.1: Block of a blockchain....4
Figure 1.2: Hash value generation of a block....4
Figure 1.3: Machine Learning techniques and algorithms....5
Figure 1.4: Abnormal behavior detection based on Machine Learning....5
Figure 1.5: Comparison on features of devices currently used for IoT security....6
Figure 3.1: Overall system diagram....10
Figure 3.2: Update automation....11
Figure 3.3: Rollback mechanism....12
Figure 3.4 Gantt chart....13

List of Tables

Table 1: Total expenses 15

List of Abbreviations

IoT – Internet of Things

SOHO – Small Office/ Home office

OWASP - Open Web Application Security Project

SVM – Support Vector Machine

FCM – Fuzzy C- Means

OS – Operating System

IDS – Intrusion Detection System

IPS – Intrusion Prevention System

1. Introduction

1.1. Background

Known vulnerabilities are regularly patched by vendors as a method to provide security to devices. If these updates aren't installed into the devices by the user, it puts the device and all the data saved on it in a vulnerable state. If in case an attacker with proper knowledge of the vulnerability finds a way to exploit it the user will have to face great losses. Therefore, it is essential that the updates are installed once received on the devices to mitigate potential risks. In a small office/home office (SOHO) environment it is necessary to keep their information and data secure or it could put the business at a risk. If in case gone to the hands of unwanted parties, the business will have to bear all financial losses as well as loss of customer confidence and reputation up to even legal penalties which might end up in bankruptcy or having to wind up the business. Hence keeping user information secure at all times should be major concern. One aspect of achieving this will be by keeping the devices up to date. Due to the hectic nature of a SOHO environment users tend to defer and overlook the process of updating their devices, but by automating the process without user involvement will be able to reduce such events from taking place.

Therefore, this system that we propose will automate the process of updating the devices within the network of devices without an involvement of a user. Going deeper into the automating process, this will have additional features not only to ensure that the process of new software update installation is legitimate and not a hoax sent to the device by an attacker, confirmation on if the installation has been done successfully but also to make sure that if the system reaches an unstable state because of the process of updating, it is reversed and the system will be able to perform its functionalities as anticipated. Addition of these features rather than only automating the updating process are expected to enhance availability and integrity of

the system by great margins. Therefore, less down time in the system will in return create greater productivity will help a business generate better revenue as well as gain customer confidence and significant increase in reputation within the market when considered in a business aspect.

1.2. Literature Review

Stepping into the future of technology, IoT, as suggested by Haller et al, *"A world where physical objects are seamlessly integrated into the information network, and where the physical objects can become active participants in business process"* carry many security problems [1] First and foremost amongst the issues are, vulnerabilities produced by poor designing of the program, which will in return create backdoor installations and malware insertion opportunities for attackers. [1] Overlooking these issues in security has the ability to compromise the availability [1] as well as integrity and confidentiality of "IoT". As discussed by the Open Web Application Security Project (OWASP), insecure software/firmware comes under top 10 vulnerabilities identified for the architecture of IoT [2]. Hence, this paper discusses how timely and regular updating of the devices can help mitigate a network of IoT devices from being compromised.

It is recommended by OWASP that systems are regularly updated as a countermeasure to securing IoT[2]. The bottleneck in this case is that a SOHO environment might not have the necessary technical teams when compared to a large scale enterprise where necessary updates can be checked and deployed in to the systems[3]. Hence automating the process will be the best possible option under the given circumstances. In this case, even if the process was automated it is necessary to have mechanism to check if the update is a legitimate because an attacker could disguise an old version of software which contains security vulnerability as the version offered as the latest by the vendor[3]. This would revert the system back to a faulty version giving attacker access[3]. A faulty software could leave the hardware

vulnerable involuntarily[4]. Hence by the use of an integrity checker this research aims to overcome the risk of installing a faulty hoax version of software into the user' system without an involvement of the user.

After the update has been done it is essential to check if it has been done successfully which is where blockchain will come into light. Blockchain has 4 main components. [5]

1) Network of Nodes: Nodes that are connected through the internet will maintain all transactions made on the network of blockchain. Once a transaction has taken place it will add it the past transactions ledger. This will be named as "mining"[5]
2) Distributed database system: Every node of the system will contain copies of information of the composed blocks. Each block will contain a timestamp and a list of transactions which will link blocks of the previous chain.[5]
3) Shared ledger: Each time a transaction takes place, the ledger will be updated.[5]
4) Cryptography: Data bound with a strong crypto mechanism. Which will be hard to track by an external unauthorized party.[5]

As shown in figure 1.1 and figure 1.2 the block will contain 2 hash values of 256 bits which will be the hash value of the current block and the previous block[5]. Therefore, the hash values can be used to be compared against each other to determine if the update has been done successfully.

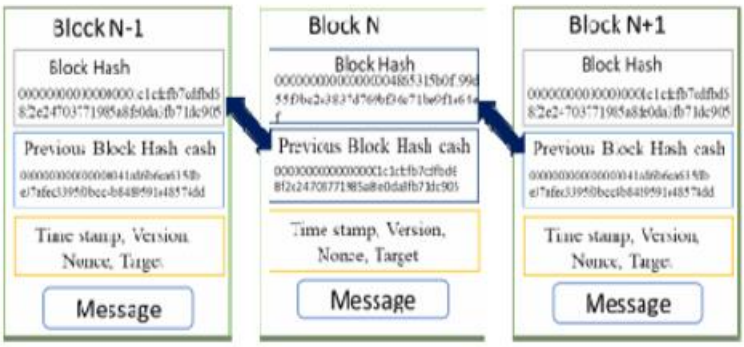

Figure 1.1: Block of a blockchain

Figure 1.2: Hash value generation of a block

Going further into the updating process, on occasion systems could lose its functionalities post update. In order to address such situation, the proposed system will be using a machine learning approach. Machine learning comes under an area of Artificial Intelligence where computer programs will be enabled to learn through examples, analogies and experience.[6] Machine learning is divided into supervised and unsupervised learning [7]. As shown in figure 1.3. these methods will have different techniques to help detect any anomalies within the system [7].

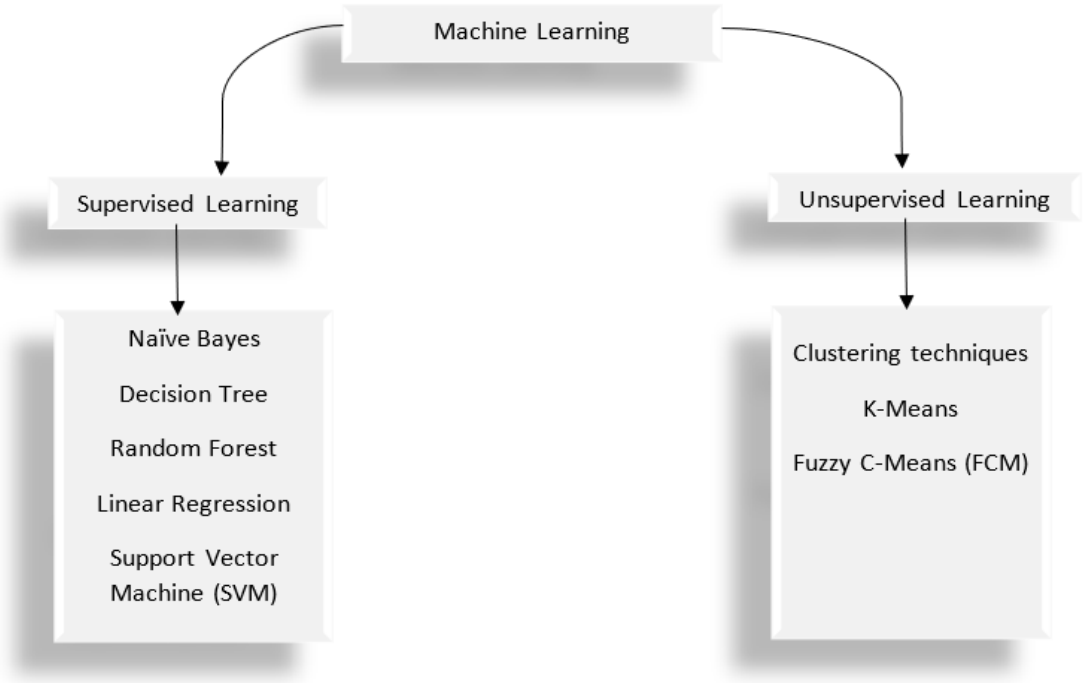

Figure 1.3: Machine Learning techniques and algorithms

According to the paper supervised machine algorithms are suggested to be more suitable for anomaly detection [7]. Hence, the proposed research will be using a supervised machine learning algorithm in order to train the data to detect abnormal behavior within the system as shown below in figure 1.4.

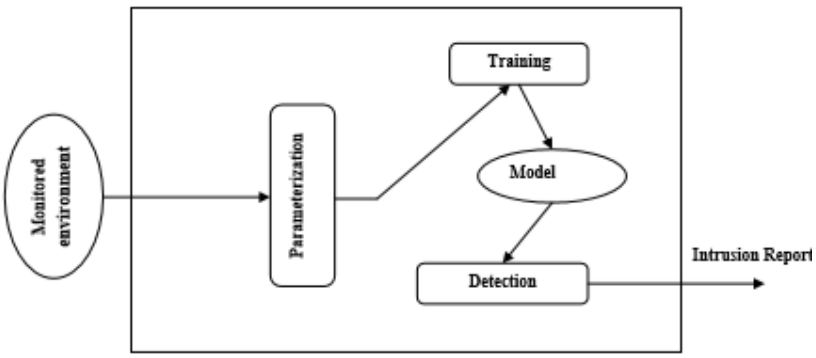

Figure 1.4: Abnormal behavior detection based on Machine Learning

Training the system to detect abnormal behavior will be broken down in to 3 stages [8].

1) Information/ data collection: Data can be collected and separated into the network according to position [8].
2) Analysis engine: Will analyze the data fed in and detect abnormal behavior [8].
3) Response: Actions taken after the analysis stage[8].

According to this research, if in case an anomaly is detected within the system, the best-case scenario would be to rollback the system into the previous version of software to retrieve the lost functionalities of the system because in a SOHO environment, a business cannot afford to have downtime. Therefore, the response will be to rollback the system.

1.3. Research gap

Taking the current context in to consideration, there are many products in the market which aims to secure a network of IoT devices such as Bitdefender BOX and Avira SafeThings[9]. Even though these products are able to detect and mitigate attacks, implement advance parental controls etc.[10] they have not focused on features such as automating update procedures whilst maintaining system integrity and stability as shown in figure 1.5.

	Pi-Hole	IoT Inspector	Fingbox	Google WiFi	Bitdefender Box 2
Authentication	N	N	N	Y	Y
User policy	N	N	Y	Y	Y
Network Intrusion Detection	N	N	Y	N	Y
Filtering	Y	N	N	N	Y
Open-Source	Y	Y	N	N	N

Figure 1.5: Comparison on features of devices currently used for IoT security

Keeping the system up to date is an essential aspect for information security, but in most cases even though the devices that are currently in use do address this problem they do not inspect a few essential areas that should be covered to guarantee maximum level of protection for the devices such as checking the legitimacy of the update file, how to handle a situation where if in case after an update any instability in the system occurs. These can be crucial areas where an attacker would take advantage of. Therefore, these aspects such as checking the legitimacy of an update file before enabling auto update and establishing mechanisms to maintain stability within the system at all times can mitigate attacks that can be putting the user information in jeopardy. Hence this area will be mainly focusing on bringing the gap between the aspects that have been aforementioned in order to bring greater user information security which will in return result in greater user satisfaction and confidence.

1.4. Research problem

The problem to be addressed in this case is how to automate the update mechanism whilst maintaining system stability? Therefore, the research will be predominantly focusing on ways to bridge the aforementioned gap and achieve the desired functionalities by ultimately reaching the objective of coming up with mechanism to automate the updating process and with a method to optimize the resource usage and detect any system instability post-update as undermentioned.

2. Objectives

2.1. Main Objective

The main objective of this area of research is to bridge the above stated gap by coming up with a mechanism to automate the updating process and with a method to optimize the resource usage and detect any system instability post-update. Hence the goal is to mainly mitigate any potential threats that can be occurred to the network of devices by keeping the system up to date by automatically pushing the updates to the devices while maintaining system stability. This way the users will be able obtain maximum use of the device's features with uninterrupted system resource availability. This will also help a user that is not technologically savvy to use the devices as desired with minimum trouble shooting due to the system being able to detect any instability and if the updates have been In this case the main objective will be broken down in to three parts to bring more clarity as to how to achieve the main objective with precision.

2.2. Specific Objectives

Specific objectives and the tasks to be done can be broken down as below.

- Disabling the auto-update option of the device.
- Mechanism to cross check the file size of the received update on the device and cross check the file size with the device's site's update file size.
- If the sizes are the same, push update into the device. If not, establish a mechanism to delete the corrupted file from the system.
- Store the hash value of the device's pre-update block as well as the hash value of the device post-update block by using the blockchain.
- Establish a mechanism to check the values against each other to confirm if the update has been successful.

- Find a suitable Machine Learning algorithm to detect any kind of behavior in the system which deviates from the normal/anticipated behavior.
- Find test data and training the machine with the use of selected algorithms.
- Store backup of the device's previous software version file.
- Enable a mechanism to install the previous version software file if in case any abnormality within the system is detected.

3. Methodology

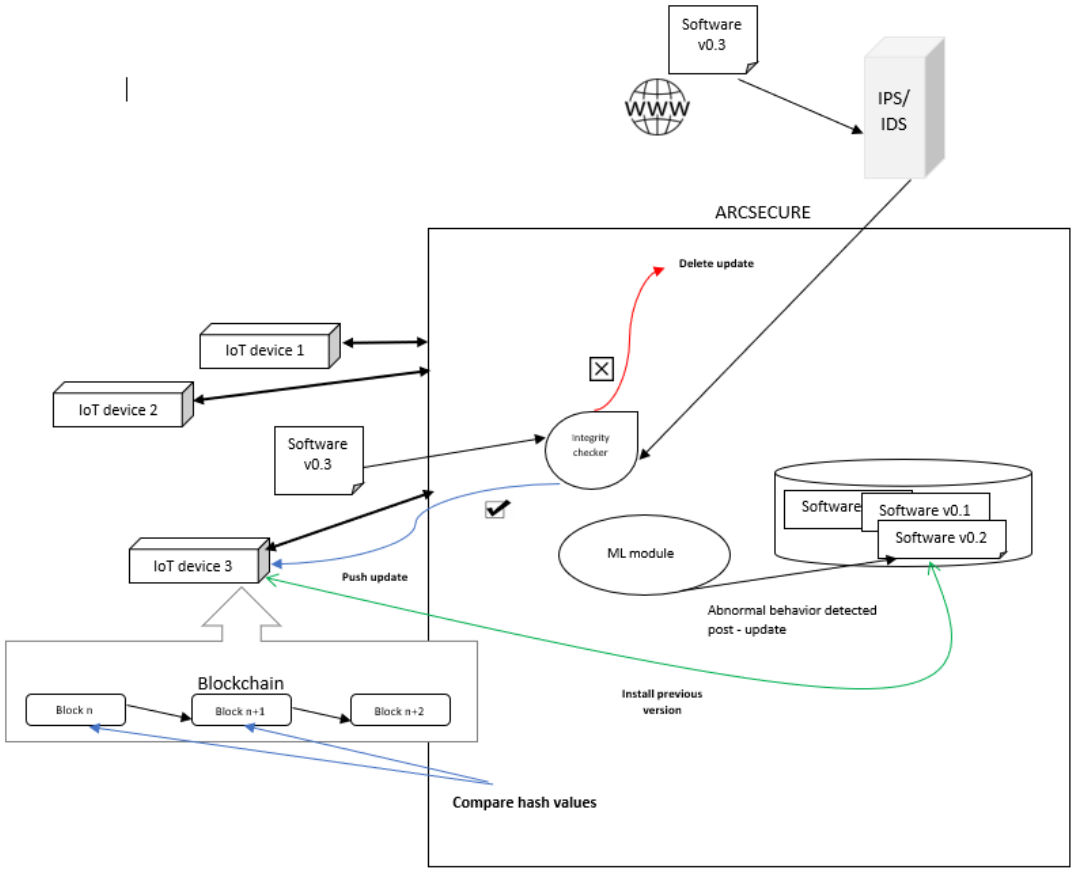

Figure 3.1: Overall system diagram

As the high-level diagram itself presents the hub that will be presented as the final product will be connected as an intermediary between the IoT devices and the IDS/IPS. In this case, as the specific objectives suggest under the main objective of:

1) Automating and optimizing the process of updating of devices
2) Establishing an update verification mechanism
3) Establishing a rollback mechanism in case of detecting system instability

3.1. Automating and optimizing the process of updating devices

Basically, every device within the network will be receiving updates from time to time. But these files received are not being checked for legitimacy.

A file received on the device might be a fabricated by an attacker which might contain a malware code. In this case once the file has been received by the device, the system will cross-check the file's has value against the device's official site's update file using an integrity checker. Initially the device's auto update option will be disabled, and it will only enable the auto-update feature if the two values are the same. If not, the file will be considered as corrupted and will be deleted from the device.

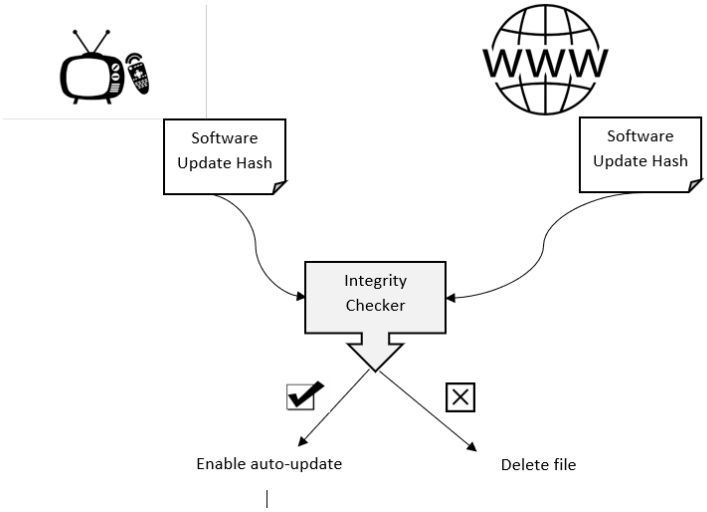

Figure 3.2: Update automation

3.2. Establishing an update verification mechanism

Using blockchain, the block's hash value of the particular device will be always maintained before an update. And another block will be maintained for the device post-update. Post-update block's hash value and the pre-update hash value will be checked against each other to check if the update has been done successfully with the use of blockchain. This process will be done as shown in figure 1.2.

3.3. Establishing a rollback mechanism in case of detecting system instability

In this case, the system will be trained by using machine learning algorithms to detect any deviation in the device from its normal behavior. For this purpose, a supervised machine learning approach will be used such as Naïve Bayes to make sure that a less false positives are generated. This will be the data training module of this particular section. Data sets will be fed into the module and will be trained over time in order to achieve this. If the system detects any deviation within the network post-update from a device, it will reinstall the previous software version file into the device. For this purpose, the system will at all times hold the previous software version file of the device an reinstall it if system reaches an unstable phase.

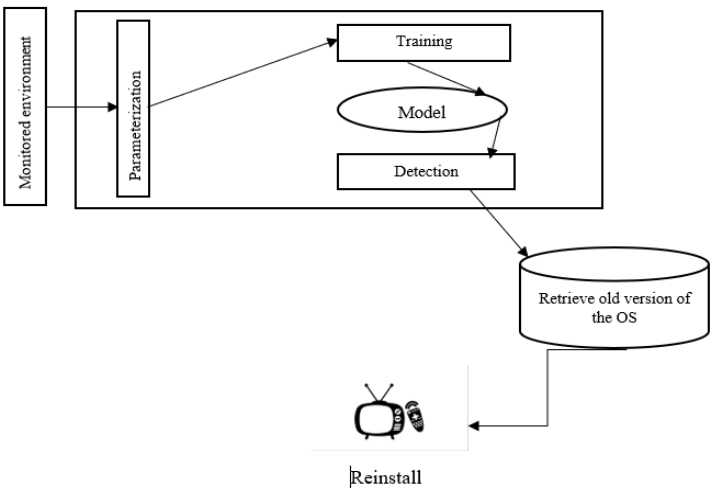

Figure 3.3: Rollback mechanism

3.4. Gantt Chart

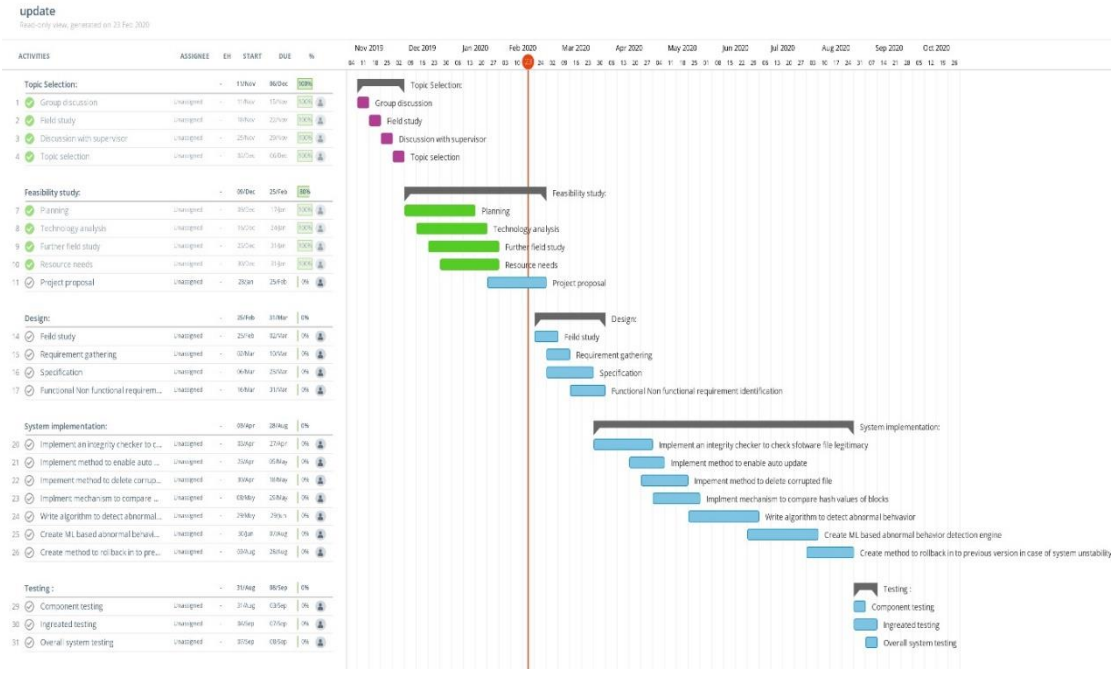

Figure 3.4 Gantt chart

4. Project Requirement

4.1.Functional requirements

System should;

- Be able to detect corrupted files.
- Have the ability to identify a corrupted file from a legitimate file.
- Be able to delete if in case a corrupted file was received.
- Be able to store the hash values of the pre and post update blocks.
- Be able to compare the two values stored and determine update status.
- Be able to store previous OS version file as backup.
- Be able to detect deviation from normal behavior.
- Be able to retrieve and install suitable file from the backup if in case any abnormal behavior is detected.

4.2.Non-functional requirements

- End user should password protect all devices within the network.
- Raspberry pi should be able to handle all devices configured on the network.
- IDS/IPS should perform its normal functionalities.
- Power disruptions should not occur.
- The vendor's software should have been patched for known vulnerabilities.
- The software running on devices should be licensed.

5. Budget and justification

5.1.Total Expenses

Description	Estimated costs
Estimated budget for the devices (Raspberry Pi 4)	Rs. 23500.00
Estimated Budget for other expenses (Stationary – 2000/ printouts – 1500)	Rs. 3500.00
Total estimated cost for the year	Rs. 27000.00

Table 1: Total expenses

6. References

[1] Z. Zhang *et al.*, "IoT Security : Ongoing Challenges and Research Opportunities," 2014.

[2] M. Ahmad and K. Salah, "IoT security : Review , blockchain solutions , and open challenges," *Futur. Gener. Comput. Syst.*, vol. 82, pp. 395–411, 2018.

[3] H. Lin and N. W. Bergmann, "IoT Privacy and Security Challenges for Smart Home Environments," 2016.

[4] J. Wurm, K. Hoang, O. Arias, A. R. Sadeghi, and Y. Jin, "Security analysis on consumer and industrial IoT devices," *Proc. Asia South Pacific Des. Autom. Conf. ASP-DAC*, vol. 25-28-Janu, pp. 519–524, 2016.

[5] M. Singh and S. Kim, "Blockchain : A Game Changer for Securing IoT Data," pp. 51–55.

[6] J. Ca and A. Skjellum, "Using Machine Learning to Secure IoT Systems."

[7] S. Omar, A. Ngadi, and H. H. Jebur, "Machine Learning Techniques for Anomaly Detection: An Overview," *Int. J. Comput. Appl.*, vol. 79, no. 2, pp. 33–41, 2013.

[8] A. Osareh and B. Shadgar, "Intrusion Detection in Computer Networks based on Machine Learning Algorithms," *Ijcsns*, vol. 8, no. 11, p. 15, 2008.

[9] L. Nagy and A. Colesa, "Router-based IoT Security using Raspberry Pi," *Proc. - RoEduNet IEEE Int. Conf.*, vol. 2019-Octob, pp. 1–6, 2019.

[10] S. Wang, R. Claro, and M. L. Pardal, "SPYKE : Security ProxY with Knowledge-based intrusion prEvention," 2018.