Air University, Islamabad
Faculty of Computing and Artificial Intelligence
Department of Cyber Security

Final Year Project

FYP – I: Proposal

Version 1.0

Part A: STUDENT INFORMATION					
Name:	SARDAR SUFIAN AZIZ			**Registration:**	200968
Email:	200968@students.au.edu.pk	**CGPA:**	2.89	**Contact No:**	0316-0912248
Name:	MISHA SOHAIL			**Registration:**	201012
Email:	201012@students.au.edu.pk	**CGPA:**	3.18	**Contact No:**	0330-5474602
Name:	ALI IRTAZA			**Registration:**	200962
Email:	200962@students.au.edu.pk	**CGPA:**	2.97	**Contact No:**	0304-4381829

Part B: PROJECT INFORMATION					
Project Name:	**SECURE UPDATE X**				
Project Title:	A Secure and Centralized Repository for IoT Firmware Management				
Project Area:	IOT Research, Updating, Development and monitoring		**End Users:**	Blue Team Operators (Blue-Team project)	
Project Type:	☐ Research	☑ Development	**Technology:**	☑ IOT Based	☐ Structured
Platform:			☐ Web based		☐ Others

	☑ Desktop	☑ Web App		☑ Distributed	
If Others:	Server and Centralized Repository				

Part C: SUPERVISOR INFORMATION			
Signature:		**Signature:**	
Supervisor:	Dr Ghalib	**Co-Supervisor:**	Dr Naveed Bhatti

Part D: FOR OFFICE USE ONLY				
Decision:	Approved	Approved (Minor)	Approved (Major)	Rejected
Comment:				
Reviewer:			**Reviewer Signature**	**Authorized Signature**

Table of Contents

1. Project Title:

A Secure and Centralized Repository for IoT Firmware Management With SECURE UPDATE X

SECURE UPDATE X

2. Project Overview

IoT technology arises with the advent of industry 4.0, focusing automation in industrial processes. IoT based solutions are deployed from less critical to very critical infrastructures like in industrial control systems and in military applications. These small constrained embedded devices have not been designed by keeping security as priority. The rate of the cyber-attacks on these small devices is also at its peak. In fact, there is a dire need of designing and implementing security solutions for each level of IoT devices ranging from firmware, hardware and communication (network or cloud). The implementation of security solutions at every level has its own challenges and concerns due to heterogeneity of the devices.

A firmware is a software program that provides necessary instructions to an IoT device to work as its manufacturer intended. The firmware is stored in read-only memory of an IoT device which can be erased and updated easily using special software [1]. The vulnerabilities in an IoT device firmware can allow an attacker to code a malware into an IoT device's firmware to perform malicious activities [2].

The firmware of IoT devices has various versions and types which means there is a lot of space for unique vulnerabilities. One of the ways of securing firmware is by auditing it thoroughly before its deployment. Extensive testing techniques are required to check proper configurations and compliance of the firmware. There are several reasons why a firmware update is required. One of the critical reasons is that without an update, a device with no firmware updating capability, is a target waiting to be attacked. Updating a firmware securely on an IoT device is a challenging task. Typical updates bring security patches, new features addition, improved performance, and fixing bugs in existing code. Firmware update and patch management systems are rigorously researched specifically in the domain of IoT [3], [4] because they run autonomously. Patch management is quite mature in conventional systems but it is hard to implement for IoT due to the diversity of devices and lack of the standardizations. The first and foremost challenge is the correct monitoring of the network for IoT devices as device identification sometimes leads to the inaccurate results. Secondly, identification of the vulnerable devices on a network is also challenging and it requires a hybrid of active and passive scanning techniques [5], [6]. Moreover, it's cumbersome to update a large number of devices existing in the user network because they have no user interface. The other major challenge while implementing a patch management solution is to get real-time information about the new or patched release of firm wares. The required information such as correct firmware name and version cannot be acquired just by sniffing the traffic of the device. Since there are numerous IoT devices and most of them are notorious for being insecure, they can become an attractive tool for hackers to be used in their successful attacks. Therefore, an insecure update mechanism in IoT devices can play a role of a promising attack vector in the favor of the hackers.

3. Goals and Objectives

The primary goals and objectives that are to be achieved after the completion of this project are:

1. Improve the security posture of IoT systems by ensuring that firmware updates are applied promptly and securely.
2. Reduce the risk of security threats by providing a platform for secure and authorized firmware updates.
3. Increase the reliability and stability of IoT devices by ensuring that firmware updates are compatible and can be rolled back if necessary.

Objectives:

1. Create a secure platform that includes authentication, authorization, encryption, and integrity capabilities for the storage and distribution of firmware updates.
2. Use version control and compatibility features to make sure that firmware updates work with different IoT device designs and can be undone if necessary.
3. Create a notification system that notifies users when new firmware updates are available and offers guidance on how to install them.
4. Offer logging and auditing tools to record all actions using the firmware store.
5. Make sure the firmware store can scale to accommodate a high number of devices and firmware updates. Test the firmware store for functionality, usability, and security, including identifying and fixing bugs, performance issues, and security vulnerabilities.
6. Deploy the firmware store to a production environment and provide training and support to users.
7. Monitor the firmware store for issues and provide ongoing maintenance and support.

By achieving these goals and objectives, the Secure IoT Firmware Store project can provide an effective solution for managing and distributing firmware updates for IoT devices securely, reducing the risk of security threats and improving the overall security posture of IoT systems.

8. 

4. High-level System Components

- **User Interface**: The user interface is the front-end of the system and provides a user-friendly platform for managing and applying firmware updates. The user interface should be easy to use and navigate, with clear instructions on how to apply firmware updates. There will be two types of users
- **IoT Device users (Devices Management):** Which will register their IoT devices so that portal will automatically mange them.
- **Vendors (Firmware Management):** Which will upload their Firm wares on the portal so that devices can easily download those Firm wares.
- **Generic**: Different Notifications and devices related Information will be uploaded which users can view to check issues ad regular updates.
- **Authentication and Authorization**: The authentication and authorization component provide a secure way for users to log in to the system and access the firmware store. To guarantee that only authorized users may access the system, technologies like multi-factor authentication and role-based access control should be included.
- **Firmware Repository:** The firmware repository is where all firmware updates are kept in one place. With version control and compatibility tools to make sure that firmware upgrades are compatible with different IoT device designs and can be rolled back, if necessary, it should allow secure and authorized access to firmware updates.
- **Notification System**: This feature notifies users when new firmware updates are available and explains how to install them. Users should be able to select which devices and delivery methods for which notifications they want to get from it, making it personalized.
- **Logging and Auditing:** The component that logs and audits all firmware store activity, including who accessed it, when, and what they did, retains a record of everything that happens. The system can be monitored for security concerns using this information, and it can also be utilized to abide by legal requirements.
- **Security Features:** To guarantee the confidentiality, integrity, and accessibility of firmware upgrades, the security features component incorporates all security mechanisms put into place throughout the system, such as encryption and integrity features. In addition, security safeguards against unauthorized access to the firmware store should be included.
- **Integration with IoT Devices:** This component enables connections between IoT devices and the firmware store so that they can receive firmware upgrades. It should offer a safe and permitted means for devices to access the firmware store and be interoperable with a range of IoT device architectures.
- **Scalability and Performance:** This component makes sure the system can manage a lot of devices and firmware updates. It ought to be built with features like load balancing, caching, and optimization in mind in order to be scalable and performant.

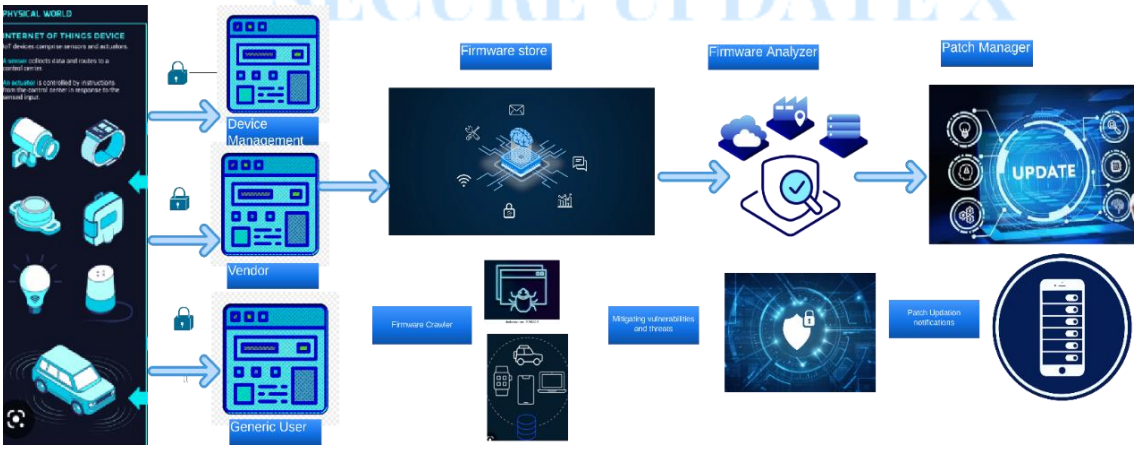


Figure 1High level components

The Secure IoT Firmware Store can offer a safe and efficient method of organizing and delivering firmware upgrades for IoT devices by putting these high-level system components into place.

5. Scope

The goal of the project is to create a system that alerts customers when new firmware patches are available or when their IoT devices are susceptible. The main emphasis is on keeping track of IoT device security status and guaranteeing timely updates to fix any vulnerabilities.

The following significant elements will be a part of the project:

- **Device Monitoring:** The linked IoT devices will be regularly checked by the system for potential security flaws or threats. This can be done in a number of ways, including by examining device logs, scanning for known vulnerabilities, or using threat intelligence feeds.

- **Vulnerability Detection:** The system will check a database of known vulnerabilities against the IoT devices' most recent firmware versions. A warning or notification will be sent out if a device is discovered to be using a vulnerable firmware version.

- **Patch Management**: The system will keep a current repository of firmware updates and patches that have been supplied by device manufacturers. Users will be informed when new versions are available for their devices when the software periodically checks for new patches and updates.

- **Notification System:** The system will have a notification mechanism to let consumers know when new updates are available and how secure their devices are. This can be done through a variety of methods, including push alerts, email, SMS, and mobile apps.

- **User Interface:** The system will include an easy-to-use interface that enables users to check the security standing of their devices, customize their notification settings, and access comprehensive details regarding fixes and vulnerabilities that are readily available.

By promptly notifying users of vulnerabilities and firmware updates, the project's main objective is to increase the security of IoT devices. By doing this, it hopes to lessen security threats and guarantee that gadgets are using the most recent and safe firmware updates.

6. Optional Functional Units

In case our project is finished long before the deadline, following optional features may also be added:

1. **Integration with Device Management Platforms:** The system may integrate with existing device management platforms or APIs provided by device manufacturers to retrieve device information, firmware versions, and apply patches remotely.
2. **Automated Testing**: The automated testing unit could be added to automatically test firmware updates to ensure that they work as intended and do not introduce any new bugs or vulnerabilities.
3. **Predictive Analytics**: The predictive analytics unit could be added to analyze data collected from IoT devices and predict when firmware updates will be needed, improving the efficiency of the firmware update process.

4. **Machine Learning**: The machine learning unit could be added to learn from user behavior and device data to provide personalized firmware update recommendations and to identify potential security threats.
5. **Patch Management:** The patch management unit could be added to manage security patches for the firmware store itself and for the IoT devices that use the firmware store.
6. **Backup and Recovery**: The backup and recovery unit could be added to create regular backups of the firmware store and to enable quick recovery in case of a data loss or system failure.
7. **Compliance and Reporting:** The compliance and reporting unit could be added to generate reports that comply with regulatory requirements and to provide visibility into the status of firmware updates for auditing and compliance purposes.
8. **Device Health Monitoring:** The device health monitoring unit could be added to monitor the health of IoT devices and to provide alerts when devices are not functioning properly, improving the overall reliability and stability of the IoT system.
9. **Separately operating Static and AI based analyzer.**

7. Application Architecture

The proposed solution consists of three major modules as shown in Figure 1. These modules include: firmware store, firmware analyzer, and patch manager. The firmware store is responsible to collect (directly through vendors or online platforms), manage, store and transmit the firmware of IoT devices. Once the firmware is collected, it will be analyzed by the firmware analyzer in order to make sure that the collected firmware is secure, and trustworthy. Finally, when a firmware is validated as secure and trustworthy, the patch manager installed at customer site will pull the firmware/patch from the firmware store and transmit it to the end user IoT devices where the firmware update/patch will be installed to fix the vulnerabilities of IoT devices or to improve the functionalities of an IoT device. These modules are further explained in the following sections.

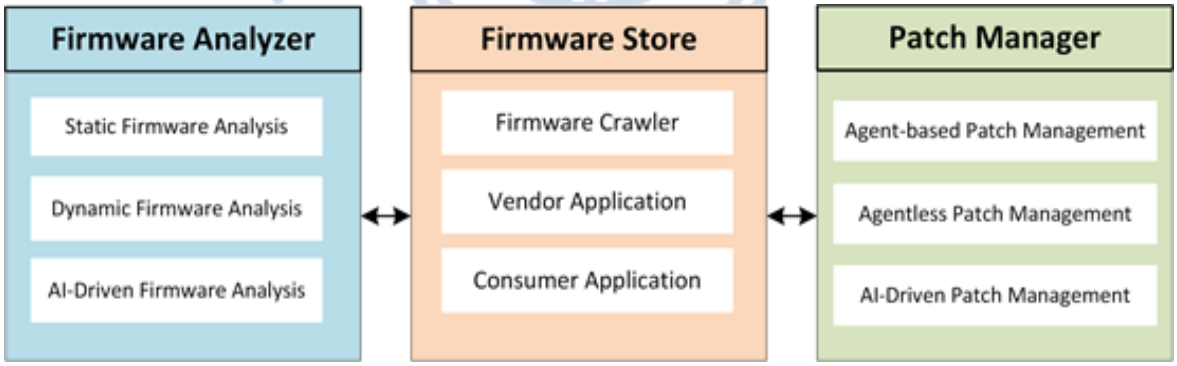


Figure 2Modular View of the Proposed Solution for Secure and Timely Firmware Update Management in IoT devices

The lack of secure firmware update mechanism can lead to vulnerable space, attack risks and asset damage. Each IoT device requires updates to fix bugs, maintain support, and introduce new features. At the latest, the update process, however, is unreliable and insecure with much room for improvements. With no universal update mechanism followed by IoT device vendors, the need for a standard update mechanism is of high value. Moreover, there is no single secure repository for collecting, approving and securing to use IoT firmware like google play store for android applications. Although, it has been observed and presented in extant literature that having such a repository or store can reduce the threat landscape drastically. Researchers have argued that one of the major challenges for implementing such a firmware store is its standardization. Firmware stores in addition to patch management systems can provide full, secure and reliable IoT firmware security solutions. Model of the proposed firmware store is shown in Figure 2.

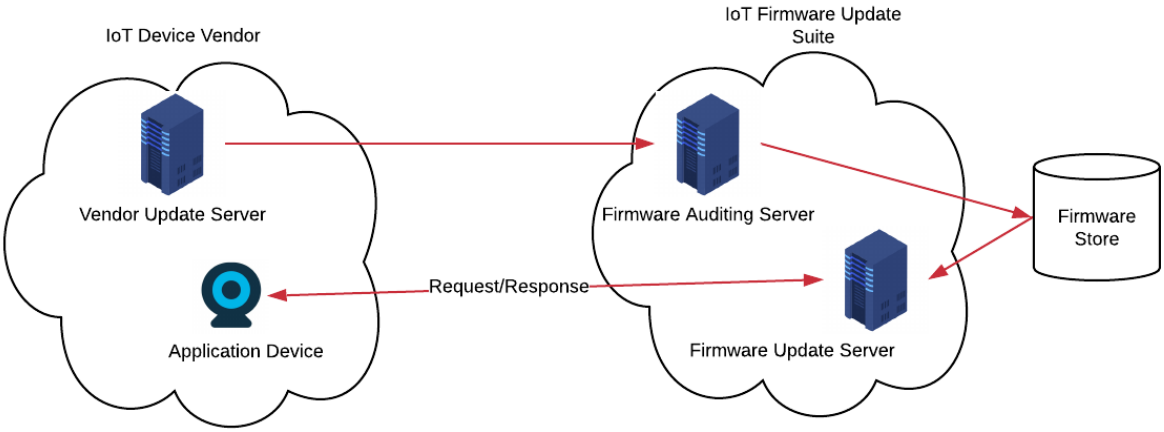


Figure 3: Model of the Proposed Firmware Store

As per the proposed plan, the IoT devices vendors will upload the updated firmware to the proposed IoT firmware update suite. At the suite, comprehensive firmware audit will be performed and upon audit clearance the firmware will be stored into the proposed firmware store. Firmware-update-server will support a two-way firmware update model, 1: an IoT device can request for the update or 2: if an update is available, it may automatically push from server to the device.

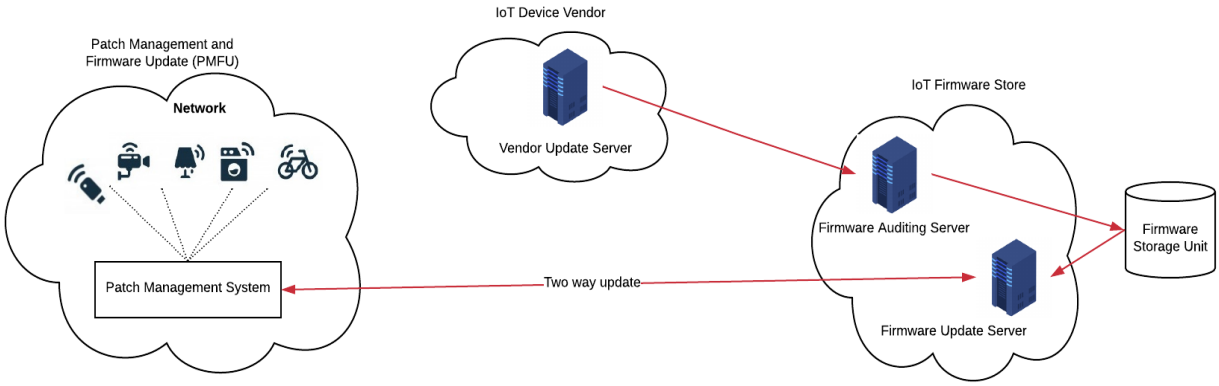


Figure 4:Figure 3: Overview of the Proposed Solution

1.**Authentication and authorization**: Only authorized personnel should be able to access the firmware store. This requires a secure login process and role-based access control.

2.**Encryption**: Firmware updates should be encrypted in transit and at rest to protect against unauthorized access.

3.**Integrity**: The firmware store should verify the integrity of the firmware updates to prevent tampering.

4.**Scalability**: A high number of devices and firmware updates should be supported by the firmware store.

5.**Version control:** The firmware store ought to keep track of several firmware update revisions and enable consumers to revert to an earlier version if necessary.

6.**Compatibility:** The firmware store must provide firmware updates for various IoT device kinds and architectures.

7.**alerts:** Users who have new firmware updates should receive alerts from the firmware store with information on how to install the upgrades.

8.**Logging and auditing:** The firmware store needs to keep track of all activity, including who accessed it and when, which firmware updates were installed, and any failures.

By implementing these features, a secure IoT firmware store can help ensure that IoT devices are updated with the latest firmware in a secure and efficient manner, reducing the risk of security threats.

Firmware crawler

A software program known as a firmware crawler analyzes and indexes firmware pictures from various online sources. Software applications known as firmware images are built into hardware components like switches, routers, and other embedded systems. The instructions necessary for the device to operate properly are contained in these photos.

The following are some uses for firmware crawlers:

1. According to security studies, firmware images frequently have security flaws that can be used by hackers. Firmware crawlers can be used to check for and locate these flaws, assisting security researchers in creating defenses against assaults.

2. Firmware updates: To find and download the most recent firmware updates for devices, use firmware crawlers. This helps to guarantee that equipment is using the most recent software and is less susceptible to assaults.

3. Device identification: By analyzing the firmware images that a device contains; software crawlers can determine the make and model of the device. Asset tracking and inventory management may benefit from this.

4.Reverse engineering: The firmware images themselves can be extracted and examined using firmware crawlers. This can help with device reverse engineering and functionality analysis. Firmware crawlers typically work by scanning the internet for firmware images and downloading them to a database. They may use various techniques to identify firmware images, such as searching for known firmware filenames or analyzing the contents of firmware images to identify their signatures. Once the firmware images have been indexed, they can be searched and analyzed for various purposes.

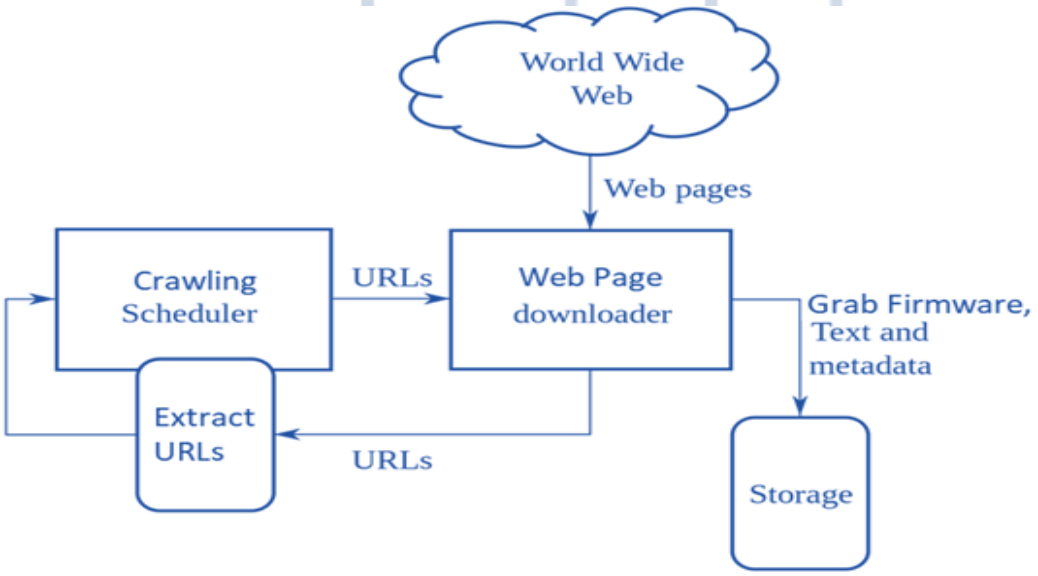
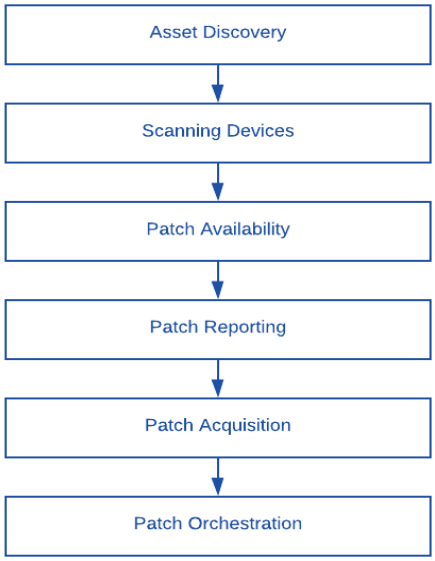


Figure 5: Crawler working mechanism

It's important to note that using firmware crawlers to scan and download firmware images without authorization can potentially be illegal and unethical. Therefore, it is important to ensure that any use of firmware crawlers is done in compliance with applicable laws and regulations



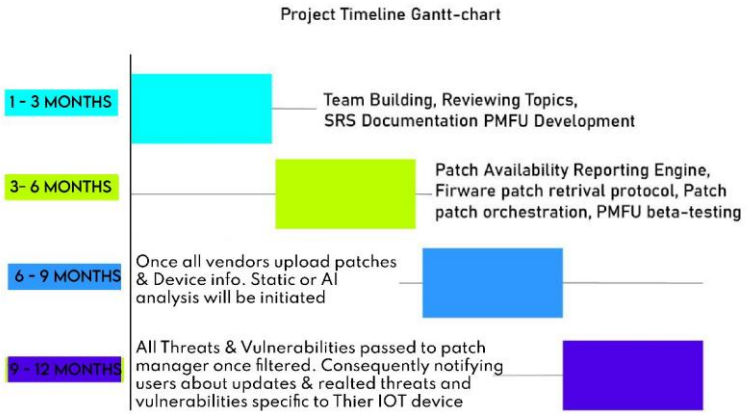
8. Impact of Project/Research on the Society

Cyber security field is still in a growth phase in Pakistan and is not matured enough. Although cyber security is a much demanding field all over the world, lots of automated solutions towards improving cyberspace are proposed and implemented. Cyber security is a big picture of this proposed project, however, specifically the proposed solution is curated for the IoT devices security. The presence of IoT devices in different environments also increases the threat area so that's why it is necessary to secure these embedded solutions. In critical infrastructures security of IoT/automation devices is very vital as upon compromise it can lead to catastrophic events. In the successful attack scenario, loss of human lives, damages to the infrastructure facilities, financial loss and reputational damages of the company can be anticipated. To provide better protection of homes, offices, and industries etc. against cyber-attacks, this proposed research is of immense use.

In the national context this research is very valuable in a number of ways. As this project consists of conference and journal papers in deliverables, it can help in improving research culture at national level. Such a research domain is valuable as IoT security is one of the most discussed topics at the global level. The production of good researchers in the IoT security field is highly appreciated.

In a developing country like Pakistan, the research and development efforts in the field of cyber security are growing at a rapid pace and with a very positive attitude. Understanding the importance of data privacy and security is now considered vital in many public and private sectors. It is now understood that privacy of the user's personal data must be ensured in every way as it is one of the human rights. The protection of personal and critical spaces by this project will support the country socially and economically.

9. Gantt Chart:



Gantt chart is shown in Figure 1.3

Project Phases:

- Planning and Requirements Gathering: This phase involves defining the project scope, identifying the stakeholders, gathering requirements, and designing the architecture of the firmware store.
- Design and Development: This phase involves designing and implementing the firmware store platform, including authentication and authorization, encryption and integrity, scalability, version control and compatibility, notification, and logging and auditing features.
- Testing and quality assurance: In this stage, the functionality, usability, and security of the firmware store are tested. Finding and repairing bugs, performance problems, and security vulnerabilities are all part of it.
- Deployment & Rollout: In this phase, the firmware store is deployed to a production environment and made available to users. Along with monitoring the firmware store for problems, it also entails teaching and supporting users.

Project Deliverables:

1. Functional and technical specifications
2. Design documents, including architecture, data model, and user interface design
3. Source code for the firmware store platform
4. Test cases and test results
5. User documentation and training materials
6. Maintenance and support plan

10. Hardware and Software Requirements

1. **Server**: A physical or virtual machine that meets the minimum system requirements for the operating system and database software being used. The server should have enough storage capacity to store firmware updates and related data.

2. **Network**: A reliable and secure network infrastructure that can handle the traffic and bandwidth requirements of the firmware updates and related data transfers.

Software Requirements:

1. Operating System: A server operating system with the security measures required to assure the safety of sensitive data as well as compatibility with the database software being used.

2. Database: A relational database management system (RDBMS) like MySQL, PostgreSQL, or Oracle that is able to manage and store the firmware changes and associated data.

3. Web Server: A web server like Apache, Nginx, or IIS that can handle serving the web-based interface for administering and dispersing firmware upgrades.

4. Programming Languages and Frameworks: The Secure IoT Firmware Store will probably need to be developed using programming languages and frameworks including Java, Python, Node.js, React, and Angular.

5. Security Software: The Secure IoT Firmware Store should offer security software including firewalls, intrusion detection and prevention systems, and encryption tools to guarantee the security of firmware upgrades and related data.

It's crucial to keep in mind that the precise hardware and software requirements may change depending on the project's scope, the number of users, and other elements. It's crucial to carefully assess the requirements and make sure they're enough to satisfy the project's needs.

11. Tools and Technologies

The Secure IoT Firmware Store may be developed using the following typical tools and technologies:

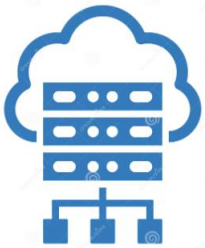
1. **Microservices Architecture**: To build a scalable and modular system, the microservices architecture can be used. Some of the popular frameworks for building microservices include Spring Boot, Micronaut, and Node.js.

2. **Docker:** Docker is a framework for containerization that enables the development of small, portable, and self-contained software containers. The Secure IoT Firmware Store's microservices can be deployed and managed via it.

2. **Kubernetes**: Kubernetes is a container orchestration platform that can be used to manage and scale Docker containers in a production environment. It can help ensure high availability and resilience of the Secure IoT Firmware Store.

3. **Relational Database Management Systems** (RDBMS): RDBMS such as MySQL, PostgreSQL, and Oracle can be used to store and manage the firmware updates and related data.

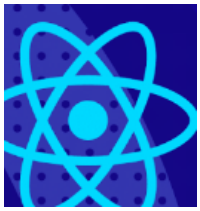
4. **Web Servers**: Web servers such as Apache, Nginx, and IIS can be used to serve the web-based interface for managing and distributing firmware updates.



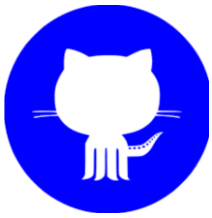
5. **RESTful APIs**: RESTful APIs can be used to expose the functionality of the microservices to external clients and enable integration with other systems.

6. **JSON Web Tokens** (JWT): JWTs can be used for authentication and authorization, ensuring secure access to the Secure IoT Firmware Store.

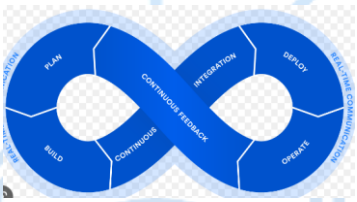
7. **React and Angular**: These are popular front-end JavaScript frameworks that can be used to build the web-based interface for managing and distributing firmware updates.



8. **Git**: Git is a version control system that can be used to manage the source code of the Secure IoT Firmware Store and enable collaboration between developers.



9. **Continuous Integration and Continuous Deployment** (CI/CD) Tools: Tools such as Jenkins, Travis CI, and GitLab CI/CD can be used to automate the testing, building, and deployment of the Secure IoT Firmware Store, ensuring fast and reliable releases.



It's important to note that the specific tools and technologies used in the development of the Secure IoT Firmware Store may vary depending on the requirements and preferences of the development team.

12. Expertise of Team Members:

- **Ali Irtaza:**
  Python
  C++/C
  Backend Development
  IOT device management
  Django/Django Rest Framework
  Docker (In-Progress)
  Server Management (In progress)
  React (In-Progress)
  FIGMA (In-Progress)

- **Sardar Sufian Aziz:**
  C++/C
  Front-end Development
  IOT device management
  UI/UX Design
  Backend Development
  Server Management
  FIGMA (In-Progress)

- **Misha Sohail:**
  Front-end Development (In-Progress)
  IOT device management

UI/UX Design
Graphic Designing and Animations
FIGMA (In-Progress)
JavaScript (In-Progress)
React (In-Progress)

References:

1. **Secure Firmware Update for IoT Devices: A Comprehensive Survey**:
   Paper by S. Ahmed et al. that provides a comprehensive survey of secure firmware update mechanisms for IoT devices, including challenges, protocols, and best practices.
   Link: https://ieeexplore.ieee.org/abstract/document/8121389

2. **OWASP Internet of Things Project**:
   The Open Web Application Security Project (OWASP) provides valuable resources on IoT security, including guidelines, best practices, and tools for securing IoT systems.
   Link: https://owasp.org/www-project-internet-of-things/

3. **NISTIR 8259: IoT Firmware Update Process for Smart Manufacturing Systems:**
   A guide published by the National Institute of Standards and Technology (NIST) that outlines a recommended firmware update process for IoT devices in smart manufacturing systems.
   Link: **https://nvlpubs.nist.gov/nistpubs/ir/2020/NIST.IR.8259.pdf**

4. **Firmware Security Best Practices:**
   A GitHub repository that provides a curated list of resources, tools, and best practices for securing firmware in IoT devices.
   Link: https://github.com/fkie-cad/awesome-firmware-security

5. **IoT Security Foundation:**
   An organization dedicated to promoting security best practices in IoT. Their website offers a range of resources, guidelines, and white papers on IoT security.
   Link: https://www.iotsecurityfoundation.org/

6. **GitHub Documentation:**
   The official documentation provided by GitHub covers various topics, including Git basics, repository management, collaboration, and workflows.
   Link: https://docs.github.com/

These references can provide you with insights into secure firmware update mechanisms, IoT security best practices, and guidance on managing GitHub repositories. They can serve as valuable resources to deepen your understanding and assist you in completing your project successfully.