Security evaluation of existing open-source Arduino projects

[1]Karthik Perumal, 40258997, [2]Richard Rathinaraj, 40280830, [3]Dharaneesh Kuppusamy Ganesan,40268161, [4]Lingeshwaran Rameshkumar, 40277718, [5]Sree Lakshme Srinivasan, 40273589, [6]Sai Ashwin Dhayalan, 40260758, [7]Chris Regy Vallikunnathu, 40232485, [8]Harshitha Raja, 40270598, [9]Harleen Kaur, 40232489, [10]Savithri Balasubramanian, 40273594.

***Abstract*— In this report, we conduct an in-depth examination of the security frameworks of various Arduino-based Internet of Things (IoT) projects. The initial phase of our study analyses the implementation details of these projects to establish an understanding of their technical and functional domains. Subsequently, a rigorous security analysis is carried out, revealing a series of vulnerabilities and errors that could potentially be exploited, compromising the integrity and confidentiality of the data managed by these IoT systems. These findings underline the critical need for enhanced security measures within such projects. Based on these insights, the report proposes a set of targeted improvements aimed at enhancing the security mechanisms of these projects. These strategies are presented with an emphasis on practical implementation, so that they are not only theoretically sound but also viable for real-world application, thus enhancing the resilience of these systems against evolving security threats. This comprehensive approach not only highlights the existing security flaws in current IoT implementations but also sets forth a path toward more secure and robust IoT solutions in the future.**

Index Terms—Security frameworks Implementation analysis, Vulnerability assessment, Data integrity, Confidentiality, Security measures, Practical implementation, Resilience, Evolving security threats, IoT security enhancements, Robust IoT solutions

I. INTRODUCTION

In today's interconnected landscape, the Internet of Things (IoT) stands as a revolutionary force, offering unprecedented connectivity and efficiency across a spectrum of applications. Arduino, an open-source electronics platform, has been instrumental in democratizing IoT development, allowing enthusiasts and professionals alike to innovate in various domains.

However, this expansion of IoT through Arduino brings forth pressing concerns surrounding cybersecurity and data privacy. As IoT devices become ubiquitous, spanning from smart homes to industrial setups, the need to address security vulnerabilities becomes increasingly urgent.

This report seeks to assess the security framework of IoT projects built upon the Arduino platform. Through an in-depth examination of common security threats and vulnerabilities, alongside an exploration of best practices for secure development and deployment, this study aims to empower developers and stakeholders with the knowledge to enhance the security of their IoT projects.

Key objectives of this report include:

1. Identification of prevalent security risks and vulnerabilities inherent in Arduino-driven IoT projects.
2. Evaluation of the efficacy of current security mechanisms and protocols in mitigating these risks.
3. Provision of practical recommendations and guidelines for the creation, implementation, and management of secure IoT solutions utilizing Arduino.
4. Utilization of real-world case studies and examples to elucidate security challenges and exemplify best practices in IoT development.

Through these objectives, we aim to contribute meaningfully to the ongoing discourse on IoT security, facilitating the development of resilient and trustworthy IoT ecosystems. By nurturing collaboration and innovation, we can harness the potential of IoT technologies while safeguarding against emergent threats and challenges.

Subsequent sections of this report will delve deeper into the intricacies of IoT security, examining various attack vectors, security mechanisms, and risk mitigation strategies within the realm of Arduino-driven IoT initiatives. Through empirical analysis, theoretical discourse, and pragmatic insights, we endeavour to provide valuable perspectives and actionable recommendations for fortifying the IoT landscape of tomorrow.

II. MATERIALS AND METHODS

In this section, we detail how we conducted the security analysis. We begin by explaining the reference classification used to categorize the security issues found in the projects. Following that, we discuss the criteria we used to select the projects. We then describe the methodology we employed to carry out the analysis. Additionally, we highlight the novel aspects and contributions put forth by the paper. Finally, we provide an overview of the projects, accompanied by brief descriptions of each.

A. *Classification*

1. *Network Attacks*
   IoT devices utilize the network layer to transmit data to servers or other devices for processing, following reception from the physical layer. Network attacks are conducted to disrupt IoT network systems by manipulating them. Unlike attacks requiring physical proximity, network attacks can be executed remotely with relative ease. These attacks represent a subset of cyberattacks aimed at compromising the communication infrastructure of IoT devices. They exploit vulnerabilities in communication channels, network protocols, and device connections to undermine the security, availability, and integrity of the IoT ecosystem. Consequently, IoT network attacks can lead to severe consequences, including unauthorized access, data breaches, service disruptions, and in some cases, physical harm. The following are the classified network attacks:

   a. *Traffic Analysis Attack*
   b. *Routing Attacks/Routing Information Attacks*
   c. *Man-in-the-Middle Attack (MiTM)*
   d. *Replay Attack*
   e. *Denial/Distributed Denial of Service (DoS/DDoS) Attack*

2. *Payload Attacks*
   Payload based attacks in the realm of IoT (Internet of Things) entail exploiting vulnerabilities present in IoT devices, systems, or network software components to compromise security, steal data, disrupt operations, or gain unauthorized access. Such attacks target the software layer of IoT devices, encompassing their operating systems, applications, firmware, and any software interfaces involved in communication. The following are the classified software attacks:

   a. Malware Attack
   b. Code Injection
   c. Command Injection
   d. Remote Code Execution
   e. Phishing Attacks

3. *Physical Attacks:*
   Physical attacks can occur when the attacker maintains close physical proximity to the network or devices within the system. Through physical attacks, new vulnerabilities in IoT systems are often discovered. The attacker seeks to physically interact with the device prior to launching an attack, possibly by acquiring a duplicate of the targeted IoT device from the market. Subsequently, they conduct a simulated attack "test" through reverse engineering to assess potential outcomes. These physical attacks serve to expose vulnerabilities within the system. The following are the classified physical attacks:

   a. *Tampering*
   b. *RF Spoofing*
   c. *Social Engineering*
   d. *Fault Injection*

4. *Side Channel Attacks*
Side channel attacks constitute a category of attacks exploiting "side channel information," which encompasses data obtainable from encryption devices beyond the plaintext or ciphertext generated during encryption. Encryption devices yield measurable time data, power consumption statistics, and various other metrics. These attacks exploit diverse forms of information to extract cryptographic keys used by the targeted device. This assertion stems from the recognition that logical operations exhibit physical characteristics dependent on their input data.

   a. *Tampering*
   b. *RF Spoofing*
   c. *Social Engineering*
   d. *Fault Injection*

III. METHODOLOGIES

In our study, we used a systematic approach to assess the security of IoT-based Arduino projects. Initially, our 10-person team selected repositories for analysis based on predefined research criteria such as popularity, relevance to IoT applications, source code availability and other factors. Each team member independently reviewed approximately 2-3 Arduino-related codes downloaded from GitHub to ensure a diverse and representative sample of projects.

To ensure consistency and quality in our analysis process, we developed a set of guidelines outlining specific criteria for identifying security issues such as active or passive attacks, data confidentiality concerns, and spreading resource attacks. These guidelines contributed to a well-structured evaluation process.

Following that, each team member performed a systematic manual analysis of the selected repositories, meticulously reviewing the Arduino projects' source code to identify vulnerabilities and weaknesses that could be exploited by attackers. The identified security issues were then classified into predefined categories as mentioned in Section 3.1, such as threats to authentication, authorization, data confidentiality, integrity, and availability, allowing for an organized and structured categorization of the results.

Following independent analyses, the research team held extensive discussions to discuss their findings and reach an agreement on the identified security issues. Any discrepancies or disagreements were resolved through additional discussion and analysis, ensuring the accuracy and dependability of the results.

While our analysis was limited to Arduino-like projects, we recognized the complexities of cybersecurity and the possibility of missing certain vulnerabilities. Furthermore, our evaluation did not assess the severity of potential attacks, but rather determined the viability of each identified security issue by examining values like Usernames, passwords, or access tokens that allow unauthorized access to the implementation making the vulnerability not just restricted to accessing Arduino board.

Overall, our research aimed to contribute to a better understanding of security challenges in IoT-based Arduino projects by providing valuable insights for scholars, developers, and users looking to improve the security of Arduino-based IoT solutions.

A. Analyzed Projects

In this section, the study describes the repositories that were found. Each project is accompanied by the following information: a brief description of its purpose, the number of stars the repository had at the time of download, the estimated number of Source Lines of Code (SLoc) written for the Arduino-like board, the number of developers involved. Table outlines the major specifications for all the projects concerned.

Table 1: Summary of projects involved in the security analysis.

ID	GitHub Project Name	Stars	SLoc	Contributors
01	Pixel Cube	9	~140	2
02	Wi-Fi Jammer with nRF24L01	108	~250	1
03	Send SMTP email with Arduino and ESP8266	-	~470	1
04	Twitter Mood Light	9	~320	1
05	Multipurpose Smart Truck	2	~230	2
06	Capacitive-soil-moisture-sensor	47	~160	1
07	The IKEA PS 2014 Lamp	11	~190	1
08	DIY energy meter	7	~240	1
09	Control of TV Using Alexa and Arduino IoT Cloud	-	~150	4
10	Arduino Self-Driving Cars with Ultrasonic Sensors	6	~115	1
11	Arduino Combustion Gas IOT Monitor	-	~72	1

12	Battery Node	41	~280	2
13	Smart Indoor Hydroponic Farming System	1	~500	1
14	Reginald: a UDP Surveillance Bot; Controlled Via the Internet	-	~40	1
15	Overview of Smart Outlet-IOT	9	~280	1
16	Industrial IoT Pressure Monitoring	-	~290	1
17	ChatGPT-Arduino-Cloud	10	~220	1
18	Smart Energy Consumption Meter	-	~105	2
19	Smart Home Monitor	-	~216	1
20	Arduino Light Controller Using MKR IoT Carrier	1	~230	1
21	Temperature Monitoring with Arduino IoT Cloud using DHT22	-	~140	1
22	Stratum-1 - A GNSS Time Server	107	~1595	2

1. Pixel Cube [16]

The project aims to create a DIY time-tracking cube inspired by devices like TimeFlip or Timeular. It utilizes an Arduino Nano microcontroller, Bluetooth module, gyroscope, accelerometer, RGB LEDs, and Electron/Vue.js for software. Users flip the cube to track tasks, with the cube logging time spent on the active face. Steps involve assembling electronics, writing firmware, developing a desktop app, integrating firmware with the app, testing, refining, documenting, and iterating for improvements.

2. Wi-Fi Jammer with nRF24L01[17]

The project utilizes an Arduino UNO along with the SparkFun Transceiver Breakout - nRF24L01+ to create a 2.4GHz network scanner. The nRF24L01+ module is a transceiver capable of both sending and receiving data. It operates on the 2.4GHz band and can communicate over a range of up to 1000 meters outdoors with an antenna. The project aims to detect and display activity in the 2.4GHz range, including interference from devices such as telephones, Bluetooth, WiFi, car alarms, and microwaves.

3. Send SMTP email with Arduino and ESP8266[18]

The project aims to demonstrate how to send an SMTP email using an Arduino Mega 2560 and an ESP8266 ESP-01 module. The project is based on a web server example by Sebastiaan Ebeltjes, with modifications to include SMTP email functionality. The goal is to send an email without using WiFi libraries, as the user faced compilation errors with existing libraries.

4.Twitter Mood Light [4]

The "Twitter Mood Light" project merges social media and physical computing by using Arduino to translate real-time Twitter sentiments into visual displays with connected lights.

Sentiment analysis algorithms assess the emotional tone of tweets accurately, offering a unique and interactive way to engage with online conversations.

5.Multipurpose Smart Truck [6]

The Multipurpose Smart Truck project revolutionizes goods delivery with advanced automation, employing RFID technology for secure access and a decentralized payment system for efficiency. Its adaptability extends beyond the TCB, bolstered by robust security measures and environmental monitoring. Automated alerts and data analytics enhance operational efficiency and decision-making.

6.Capacitive-soil-moisture-sensor[15]

The document details the development of a capacitive soil moisture sensor with wireless communication capabilities. Powered by an ATmega328P microcontroller and NRF24L01 radio module, it achieves a remarkable ten-year battery life using two AAA batteries. Secure message signing ensures data integrity during transmission, and the sensor provides extensive moisture readings across the spectrum. Temperature stability is maintained with an external 8MHz resonator, and a 3D printed enclosure enhances durability.

7. The IKEA PS 2014 Lamp [8]

The IKEA PS 2014 lamp is a manually expandable pendant light known for its distinctive design and versatility. Inspired by its unique movement controlled by a string, a Computer Science Engineer sought to enhance its functionality by adding remote control features. The lamp offers a novel approach to lighting solutions, drawing inspiration from science fiction and video games to captivate and impress. Its adjustable brightness and space-saving design make it ideal for various settings, while its aesthetic appeal adds a touch of style to any space.

8.DIY energy meter [5]

This project focuses on creating a DIY energy meter using Arduino, Modbus communication, and RS485. The primary goal is to monitor current, power, and energy consumption using a Finder energy meter. The setup involves Arduino Cloud for data storage and visualization. The project emphasizes accessibility and customization, enabling users to build their energy monitoring system at home.

9. Control of TV Using Alexa and Arduino IoT Cloud [10]
The Arduino sketch facilitates TV remote control through infrared signals, utilizing the Arduino IoT Cloud for command execution. It defines IR patterns for functions like channel switching and volume adjustment, with global variables managing previous TV states. The main loop updates cloud connections for commands, and the onTvChange () function translates state changes into IR signals. Debugging information is communicated via serial, enabling remote TV control via the IoT Cloud.

10. Arduino Self-Driving Cars with ultrasonic Sensors [7]
The project aims to create a self-driving car prototype using ultrasonic sensors and Arduino technology. Key objectives include implementing autonomous navigation algorithms, motion control with four motors for movement and a servo motor for steering, user interaction via a push button, obstacle detection and response capabilities, and ensuring safety and reliability. By achieving these goals, the project aims to demonstrate the feasibility of constructing a functional self-driving car prototype with the specified technologies.

11. Arduino Combustion Gas IOT Monitor [11]
The Arduino Combustion Gas IoT Monitor project is designed to employ sensor technology alongside the Arduino IoT cloud for monitoring combustion gas by focusing on carbon monoxide (CO) and carbon dioxide (CO2) levels within the environment.

12. Battery Node [9]
BatteryNode is a DIY framework designed for creating low-power, cost-effective, standalone, or web-connected IoT networks without the need for cloud services or advanced programming skills. It focuses on simplicity, affordability, and customization, making it ideal for various monitoring and control applications.

13. Smart Indoor Hydroponic Farming system [24]
The Smart Indoor Hydroponic Farming system is an automated solution for indoor farming. It features a Web Admin Dashboard providing a user-friendly interface for system management and real-time data monitoring. The system's Arduino Control Unit monitors environmental parameters and controls various aspects of the hydroponic setup, such as water pumps, nutrient delivery, and lighting. Communication between the control unit and the web dashboard is facilitated through the ESP8266 Wi-Fi Module.

14. Reginald: a UDP Surveillance Bot; Controlled Via the Internet [25]
Reginald" is an internet-controllable surveillance bot that utilizes UDP for communication, allowing for live video streaming and remote control. The project started with a simple idea but grew into a sophisticated system. Unlike traditional surveillance systems using TCP, Reginald's use of UDP introduces unique advantages and risks. This report evaluates Reginald's communication methods and proposes security improvements.

15. Overview of Smart Outlet-IOT [12]
The home automation POC system allows remote control of home devices via three main components: the outlet device, gateway, and client. The outlet device executes commands, the gateway enables communication between client and outlet, and the client sends control requests. Users can manage devices remotely over the local Wi-Fi network using any connected client. Flask, a lightweight web app framework, facilitates efficient API communication among system components.

16. Industrial IoT Pressure Monitoring [13]
The project aimed to create a cost-effective prototype for remotely monitoring pressure on industrial equipment using cellular data transmission. Its main goal was to convert pressure readings from a transducer into actionable data, facilitating timely notifications for abnormal pressure conditions. By combining Arduino devices, cellular communication modules, and cloud-based services, the project sought to establish an efficient monitoring and notification system suitable for industrial applications.

17. ChatGPT-Arduino-Cloud [14]
The ChatGPT-Arduino-Cloud project merges Arduino IoT Cloud-compatible devices with OpenAI's GPT-3.5 language model, allowing physical IoT devices to communicate with advanced natural language processing capabilities. Users can interact with the language model through physical prompts, facilitating seamless communication between IoT devices and AI-driven applications. With clear setup instructions provided, users can easily configure OpenAI accounts, obtain API keys, and set up IoT Cloud dashboards to experiment with AI-powered interactions in their IoT projects.

18. Smart Energy Consumption Meter [19]
The Smart Energy Consumption Meter project utilizes IoT technology to monitor energy usage efficiently. It employs various hardware components to measure key parameters like voltage, current, and power. The collected data is sent to a server for analysis, allowing users to track their energy consumption patterns and optimize usage. Overall, this project facilitates more effective energy management practices, contributing to greater energy efficiency.

19.Smart Home Monitor [20]
The Smart Home Monitor project employs Arduino MKR WiFi 1010 and MKR IoT Carrier to monitor environmental conditions in homes. It utilizes sensors to track various parameters and manages data analysis through the Arduino IoT Cloud. Implementation involves configuring the hardware and programming it via Arduino IDE or Web Editor. Real-time monitoring capabilities offer adaptability for diverse home monitoring tasks.

20. Arduino Light Controller Using MKR IoT Carrier [21]
The project is a smart light controller using Arduino MKR IoT Carrier, designed for easy and interactive light management. It allows the user to control the lights from anywhere in the house. More details about the project can be found at Arduino Light

Controller Using MKR IoT Carrier | Arduino Project Hub.

21. Temperature Monitoring with Arduino IoT Cloud using DHT22[22]

The "Temperature Monitoring with Arduino IoT Cloud using DHT22" project employs a DHT22 sensor and an MKR WiFi 1010 to monitor temperature and humidity, integrated with the Arduino IoT Cloud. It enables sensor data acquisition, transmission to the cloud, and real-time variable updates. Configuration involves setting up the IoT Cloud for the device and variables and writing the sketch to handle sensor data and cloud communication. Users can conveniently monitor room or outdoor temperature and humidity via the Arduino IoT Cloud platform.

22. Stratum-1 - A GNSS Time Server [23]

The GNSSTimeServer project combines IoT and Global Navigation Satellite Systems (GNSS) to create a precise time server. Utilizing ESP8266/ESP32 and Arduino platforms, it offers NTP/RDATE services synchronized with satellites like GPS, BeiDou, GLONASS, and Galileo. Featuring a web dashboard, it displays metrics such as uptime, temperature, IP address, and GNSS data, with configuration options. Additionally, OTA firmware updates are supported for efficient device management, while a security analysis aims to identify potential vulnerabilities.

IV. RESULTS

Considering how widely used Arduino is, a lot of developers join its ecosystem without having a solid grasp of security principles. Due of this, many open-source Arduino projects are created and published on public repositories like GitHub. As a result, anyone would be able to access the source codes and find ways to exploit the vulnerabilities in the projects. In this section, we will see detailed analysis of various attack categories for a range of projects. For this purpose, we have defined each attack as Feasible or Not Feasible. We shall begin with Network attacks, the basic attack concepts based on which many forms of sophisticated exploits detailed in this section are designed.

1. **Network attacks:**
   Network attacks on IoT devices can be executed from a distance and can disrupt the IoT network systems by exploiting vulnerabilities in communication channels, network protocols, and device connections. These attacks can lead to unauthorized access, data breaches, service disruptions, and even physical harm. They include Traffic Analysis Attacks, Routing Attacks/Routing Information Attacks, Man-in-the-Middle Attacks (MiTM), Replay Attacks, and Denial/Distributed Denial of Service (DoS/DDoS) Attacks.

1.1 **Traffic Analysis Attack**: This type of attack involves an attacker accessing the same network as the victim to capture and analyze all network traffic. The attacker can learn valuable information about the victim or their organization from this data. This is considered a passive form of attack.

1.2 **Routing Attacks/Routing Information Attacks**: These are cyberattacking that target an Internet service provider with the goal of reducing uptime or preventing users from accessing a web-enabled system like a blockchain. The attacker can divide a network into separate parts, preventing communication between nodes that are part of one chain and those that are not.

1.3 **Man-in-the-Middle Attack (MiTM)**: In a man-in-the-middle (MiTM) attack, an attacker intercepts and potentially alters the communications between two parties who believe they are directly communicating with each other. The attacker can listen in on the conversation, alter the messages being exchanged, or impersonate one of the parties to gain access to sensitive information.

1.4 **Replay Attack**: A replay attack is when a cybercriminal listens in on a secure network communication, intercepts it, and then fraudulently delays or resends it to trick the receiver into doing what the hacker wants. The main goal is to trick the system into accepting the retransmission of the data as a legitimate one.

1.5 **Denial/Distributed Denial of Service (DoS/DDoS) Attack**: A distributed denial-of-service (DDoS) attack is a malicious attempt to disrupt the normal traffic of a targeted server, service, or network by overwhelming the target or its surrounding infrastructure with a flood of Internet traffic. DDoS attacks achieve effectiveness by utilizing multiple compromised computer systems as sources of attack traffic.

Table 2: Network attacks issues inside the analyzed projects

Project Name	Traffic Analysis Attack	Routing Attacks/Routing Information Attacks	Man-in-the-Middle Attack (MiTM)	Replay Attacks	Denial/DDoS Attack
TwitterMoodLight	**Not Feasible** - Limited processing operations make it	**Not Feasible** - Lack of involvement in	**Feasible:** Lack of data encryption	**Not Feasible** - No sensitive data exchange	**Not Feasible** - Not a critical

	an unlikely target for traffic analysis.	network routing operations.	during transmission	susceptible to replay.	service subject to DoS attacks.
Energy meter	**Feasible** - Can analyze energy usage patterns to infer behavioral data.	**Feasible** - Vulnerable to routing manipulation affecting energy data.	**Not Feasible** - Energy meters typically do not facilitate direct communication between parties.	**Not Feasible**: **e.g.,** Malicious retransmission of valid data.	**Not Feasible**: **e.g.,** No network connectivity for attack.
Multipurpose Smart Truck	**Feasible** - Can monitor vehicle movements and communication patterns.	**Feasible** - Routing manipulation can disrupt truck's routing and communication.	**Not Feasible**: No data transmission within the system.	**Not Feasible**: No data transmission within the system.	**Not Feasible**: No network connection for external attackers to flood the system with traffic.
Arduino self-driving car	**Feasible** - Can monitor traffic patterns and communication with other vehicles.	**Feasible** - Routing manipulation can disrupt car's routing and communication.	**Feasible:** **e.g.,** Intercepting and altering sensor data.	**Feasible:** **e.g.,** Replaying old sensor readings.	**Feasible:** **e.g.,** Overwhelming the system with excessive traffic.
IKEA PS 2014 DIY Lamp	**Not Feasible** - No involvement in network communication or data exchange.	**Not Feasible** - Lack of involvement in network routing operations.	**Feasible:** Absence of secure communication protocols.	**Feasible:** Lack of authentication mechanisms.	**Not Feasible**: No network connection for attackers.
Battery node	**Not Feasible** - Limited involvement in network communication or data exchange.	**Not Feasible** - Lack of involvement in network routing operations.	**Feasible**: HTTP update without encryption or authentication.	**Feasible**: No input validation for commands from gateway.	**Not Feasible**: No Network connection.
Control of your TV using Alexa and Arduino IoT cloud	**Feasible** - Can monitor user interaction patterns and communication with TV.	**Feasible** - Routing manipulation can disrupt communication between devices.	**Feasible**: **e.g.,** Manipulating TV control commands during transmission.	**Not Feasible**: No replay potential as commands are executed immediately.	**Feasible** - Disruption of TV's communication can lead to inconvenience and service disruption.
Combustion gas IoT monitor	**Feasible** - Can analyze gas usage patterns to infer user behavior.	**Feasible** - Routing manipulation can disrupt monitor's routing and communication.	**Feasible** - Vulnerable to interception and alteration of communication with monitoring systems.	**Feasible**: **e.g.,** Repetition of sensor readings	**Feasible**: **e.g.,** Overloading the IoT cloud service
SmartOutlet	**Feasible** - Can monitor power consumption patterns to infer user behavior.	**Feasible** - Routing manipulation can disrupt outlet's routing and communication.	**Feasible**: Lack of data encryption during transmission	**Feasible**: Lack of data encryption during transmission	**Not Feasible** - Not a critical service subject to DoS attacks.
IoT Pressure sensor	**Feasible** - Can monitor pressure data transmission patterns.	**Feasible** - Routing manipulation can disrupt sensor's routing and communication.	**Feasible:** **E.g.,** Intercepting communication between Arduino	**Feasible:** **E.g.,** Replicating pressure data transmissions.	**Not Feasible** - Not a critical service subject to DoS attacks.

			devices and cloud services.		
Chat with ChatGPT through Arduino IoT Cloud	**Feasible** - Can monitor chat interaction patterns and user behavior.	**Feasible** - Routing manipulation can disrupt chat communication.	**Feasible:** **e.g.,** Intercepting and altering data	**Feasible** - Replay of chat interactions could lead to unauthorized access or manipulation of messages.	**Not Feasible:** No Network connection
Capacitive soil moisture sensor	**Feasible** - Can monitor soil moisture data transmission patterns.	**Feasible** - Routing manipulation can disrupt sensor's routing and communication.	**Feasible:** Absence of secure communication protocols.	**Not Feasible** - Limited application as replayed data would not significantly impact soil moisture.	**Feasible:** No declared protection against DoS attacks.
Pixel cube	**Feasible** - Can monitor pixel data transmission patterns.	**Feasible** - Routing manipulation can disrupt cube's routing and communication.	**Feasible:** Manipulating data between the cube and devices.	**Feasible:** Attackers could replay previous commands	**Not Feasible:** No floodable connection for attack.
Wifi Jammer	**Feasible** - Can monitor Wi-Fi traffic patterns.	**Feasible** - Routing manipulation can disrupt Wi-Fi network routing.	**Not Feasible:** No data transmission.	**Not Feasible:** No data transmission.	**Not Feasible:** No Network connection
Send SMTP email with Arduino	**Feasible** - Attackers could analyze patterns in email traffic to infer sensitive information about communication patterns.	**Not Feasible** - Typically not directly involved in routing traffic.	**Not Feasible:** No data transmission.	**Not Feasible:** No replay-able data transmission.	**Not Feasible:** No floodable network connection
Smart energy consumption meter	**Feasible** - Analysis of energy usage patterns could reveal insights about user behavior and occupancy patterns.	**Not Feasible** - Energy meters typically do not participate in routing decisions.	**Feasible:** Data sent over unencrypted HTTP connection	**Not Feasible:** Lack of network connection prevents the replay of captured data.	**Not Feasible** - Unlikely target for DoS/DDoS attacks.
Smart home monitor	**Feasible** - Monitoring patterns in sensor data transmission could reveal information about user behavior.	**Not Feasible** - Generally not involved in routing network traffic.	**Not Feasible:** Not applicable since the scenario lacks data transmissions susceptible to MitM attacks.	**Not Feasible:** Not applicable since the scenario lacks replay-able data transmissions.	**Feasible:** **e.g.,** Blocking delays in critical functions make the system vulnerable to DoS attacks.
Arduino light controller	**Feasible** - Analysis of patterns in light control commands could reveal insights about user behavior.	**Not Feasible** - Typically not involved in routing network traffic.	**Feasible:** **e.g.,** Unauthorized replay of light commands.	**Not Feasible:** Data sent over unencrypted HTTP connection	**Not Feasible:** No network connection for DoS exploitation.
Temperature Monitoring with Arduino IoT	**Not Feasible** - Limited processing operations make it	**Not Feasible** - Lack of involvement in	**Not Feasible** - Not directly	**Not Feasible** - No sensitive data exchange	**Not Feasible** - Not a critical

Cloud using DHT22	an unlikely target for traffic analysis.	network routing operations.	involved in network	susceptible to replay.	service subject to DoS attacks.
Stratum-1 GNSS Time server	**Feasible** - Analysis of timing data could reveal patterns in time synchronization requests.	**Not Feasible** - Time servers typically do not participate in routing decisions.	**Feasible:** Data sent over unencrypted HTTP connection	**Not Feasible:** No replay-able data transmission	**Feasible:** Lack of rate limiting during data transmission.
Smart Indoor Hydroponic Farming	**Feasible** - Monitoring patterns in sensor data transmission could reveal insights about crop growth or environmental conditions.	**Not Feasible** - Typically not involved in routing network traffic.	**Feasible:** Insecure WebSocket implementation allows for interception of traffic.	**Feasible:** Weak password handling allows replaying commands.	**Feasible:** Lack of proper error handling leave the system vulnerable to flooding attacks.
Surveillance Bot	**Feasible** - Analyzing traffic patterns related to camera feeds could reveal information about monitored areas.	**Not Feasible** - Typically not involved in routing network traffic.	**Feasible:** Lack of encryption exposes data to MitM interception.	**Feasible:** Weak password handling allows replaying commands.	**Feasible:** Vulnerabilities allow attackers to flood the network.

2. **Payload Attacks:**
   In the presented table, various IoT projects are assessed for vulnerabilities to different types of payload-based attacks, including malware, code injection, phishing, command injection and remote code execution.

   Projects such as TwitterMoodLight, Energy Meter, and Multipurpose Smart Truck are susceptible to malware attacks due to their potential to manipulate data and systems. Additionally, they may be vulnerable to command injection if they accept commands from untrusted sources. However, these projects are less likely targets for phishing attacks due to their limited user interaction.

On the other hand, projects like WiFi Jammer, Send SMTP Email with Arduino, have functionalities that can make them appealing targets for various attacks. For instance, they can potentially fall victim to code injection and remote code execution if they accept and execute commands unsafely. Moreover, their capability to overwhelm network or email server resources makes them prone to denial-of-service attacks. However, projects such as IKEA PS 2014 DIY Lamp and Smart Indoor Hydroponic Farming exhibit fewer vulnerabilities due to their limited attack surface and lack of direct interaction with external commands or data, rendering them less susceptible to cyber threats.

Table 3: Issues exploitable using a payload inside the analyzed projects.

Project Name	Malware attack	Code injection	Phishing attack	Command Injection	Remote Code execution
TwitterMoodLight	**Not Feasible** - Limited processing operations, unlikely target for malware.	**Feasible** - Possible if system accepts external commands unsafely.	**Feasible** - Can be tricked into accessing malicious links or content.	**Feasible -** If commands are processed unsafely.	**Not Feasible -** Limited attack surface, not interactive with external data.
Energy meter	**Feasible** - Can infect or manipulate energy data.	**Feasible** - Vulnerable if accepting and executing external code.	**Not Feasible** - Unlikely target for phishing, not directly user-facing..	**Feasible** - If accepting commands from untrusted sources.	**Not Feasible** - Limited attack surface, not interactive with external data.

Multipurpose Smart Truck	**Feasible** - Can infect vehicle systems or data.	**Feasible** - Vulnerable if accepting and executing external code.	**Not Feasible** - Unlikely target for phishing, not directly user-facing.	**Feasible -** If accepting commands from untrusted sources.	**Not Feasible** - Limited attack surface, not interactive with external data.
Arduino self driving car	**Not Feasible** - Can infect vehicle systems or data.	**Not Feasible** - Vulnerable if accepting and executing external code.	**Feasible** - Unlikely target for phishing, not directly user-facing.	**Feasible -** If accepting commands from untrusted sources**.**	**Not Feasible** - Limited attack surface, not interactive with external data.
IKEA PS 2014 DIY Lamp	**Not Feasible** - Limited attack surface, not interactive with external data.	**Not Feasible** - No direct external code execution or user input.	**Not Feasible** - Can be tricked into accessing malicious links or content.	**Not Feasible -** No direct interaction with external commands or data.	**Not Feasible** - Limited attack surface, not interactive with external data.
Battery node	**Feasible** - Can infect or manipulate battery data.	**Feasible** - Vulnerable if accepting and executing external code.	**Not Feasible** - Unlikely target for phishing, not directly user-facing.	**Feasible -** If accepting commands from untrusted sources.	**Not Feasible -** Limited attack surface, not interactive with external data.
Control of your TV using Alexa and Arduino IoT cloud	**Feasible** - Can infect or manipulate device controls.	**Feasible** - Vulnerable if accepting and executing external code.	**Feasible** - Can be tricked into accessing malicious links or content.	**Feasible -** If accepting commands from untrusted sources.	**Feasible -** Potential if device accepts and executes commands unsafely.
Combustion gas IoT monitor	**Feasible** - Can infect or manipulate gas readings.	**Feasible** - Vulnerable if accepting and executing external code.	**Not Feasible** - Unlikely target for phishing, not directly user-facing.	**Feasible -** If accepting commands from untrusted sources.	**Not Feasible -** Limited attack surface, not interactive with external data.
SmartOutlet	**Feasible** - Can infect or manipulate device controls.	**Feasible** - Vulnerable if accepting and executing external code.	**Not Feasible** - Unlikely target for phishing, not directly user-facing.	**Feasible -** If accepting commands from untrusted sources.	**Feasible -** Potential if device accepts and executes commands unsafely.
IoT Pressure sensor	**Feasible** - Can infect or manipulate pressure readings.	**Feasible** - Vulnerable if accepting and executing external code.	**Not Feasible** - Unlikely target for phishing, not directly user-facing.	**Feasible -** If accepting commands from untrusted sources.	**Not Feasible -** Limited attack surface, not interactive with external data.
Chat with ChatGPT through Arduino IoT Cloud	**Feasible** - Can infect or manipulate chat interactions.	**Feasible** - Vulnerable if accepting and executing external code.	**Feasible** - Can be tricked into accessing malicious links or content.	**Feasible -** If accepting commands from untrusted sources.	**Feasible -** Potential if device accepts and executes commands unsafely**.**
Capacitive soil moisture sensor	**Feasible** - Can infect or manipulate soil moisture readings.	**Feasible** - Vulnerable if accepting and executing external code.	**Not Feasible** - Unlikely target for phishing, not directly user-facing.	**Feasible -** If accepting commands from untrusted sources.	**Not Feasible -** Limited attack surface, not interactive with external data.
Pixel cube	**Feasible** - Can infect or manipulate pixel data.	**Feasible** - Vulnerable if accepting and executing external code.	**Not Feasible** - Unlikely target for phishing, not directly user-facing.	**Feasible -** If accepting commands from untrusted sources.	**Not Feasible -** Limited attack surface, not interactive with external data.

Wifi Jammer	**Feasible** - Can infect or manipulate Wi-Fi functionality.	**Feasible** - Vulnerable if accepting and executing external code.	**Feasible** - Can be tricked into accessing malicious links or content.	**Feasible -** If accepting commands from untrusted sources.	**Feasible -** Potential if device accepts and executes commands unsafely.
Send SMTP email with Arduiono	**Feasible** - Can infect or manipulate email functionality.	**Feasible** - Vulnerable if accepting and executing external code.	**Feasible** - Can be tricked into accessing malicious links or content.	**Feasible -** If accepting commands from untrusted sources.	**Feasible -** Potential if device accepts and executes commands unsafely.
Smart energy consumption meter	**Feasible** - Can infect or manipulate energy data.	**Feasible** - Vulnerable if accepting and executing external code..	**Not Feasible** - Unlikely target for phishing, not directly user-facing.	**Feasible -** If accepting commands from untrusted sources.	**Not Feasible -** Limited attack surface, not interactive with external data.
Smart home monitor	**Feasible** - Can infect or manipulate sensor readings.	**Feasible** - Vulnerable if accepting and executing external code.	**Not Feasible** - Unlikely target for phishing, not directly user-facing.	**Feasible -** If accepting commands from untrusted sources.	**Not Feasible -** Limited attack surface, not interactive with external data.
Arduino light controller	**Feasible** - Can infect or manipulate light controls.	**Feasible** - Vulnerable if accepting and executing external code.	**Not Feasible** - Unlikely target for phishing, not directly user-facing.	**Feasible -** If accepting commands from untrusted sources.	**Not Feasible -** Limited attack surface, not interactive with external data.
Temperature Monitoring with Arduino IoT Cloud using DHT22	**Not Feasible** - Unlikely to be targeted by malware attacks.	**Not Feasible** - Limited attack surface for injecting code.	**Not Feasible** - Not typically associated with user interactions.	**Not Feasible** - Limited interaction points for command injection.	**Not Feasible** - No known vulnerabilities for remote code execution.
Stratum-1 GNSS Time server	**Feasible** - Can infect or manipulate time synchronization.	**Feasible** - Vulnerable if accepting and executing external code.	**Not Feasible** - Unlikely target for phishing, not directly user-facing.	**Feasible -** If accepting commands from untrusted sources.	**Not Feasible -** Limited attack surface, not interactive with external data.
Smart Indoor Hydroponic Farming	**Feasible** - Can infect or manipulate sensor readings.	**Feasible** - Vulnerable if accepting and executing external code.	**Not Feasible** - Unlikely target for phishing, not directly user-facing.	**Feasible -** If accepting commands from untrusted sources.	**Not Feasible -** Limited attack surface, not interactive with external data.
Surveillance Bot	**Feasible** - Can infect or manipulate camera feeds..	**Feasible** - Vulnerable if accepting and executing external code.	**Feasible** - Can be tricked into accessing malicious links or content.	**Feasible -** If accepting commands from untrusted sources.	**Feasible -** Potential if device accepts and executes commands unsafely.

3. **Physical Attacks:**
Physical attacks leverage direct access to hardware components or proximity to networked devices to compromise system integrity and confidentiality. Despite their potential impact, the exploration of physical attacks on IoT infrastructure, particularly within the context of Arduino Feasible-based projects, remains limited. Arduino Feasible platforms serve as popular tools for prototyping and deploying IoT solutions due to their accessibility and flexibility. This paper aims to address this gap by analyzing the implications of physical attacks on Arduino IoT projects. Through this analysis, we seek to enhance our understanding of IoT security challenges and contribute to the development of effective countermeasures to mitigate these risks.

3.1 **Tampering:** Tampering refers to the unauthorized alteration or modification of system components, data, or processes. This attack involves manipulating hardware components or software configurations without permission, aiming to compromise the integrity and security of the system. Tampering attacks can lead to various consequences such as data breaches, system malfunctions, or unauthorized access, posing significant risks to the overall security posture of the system.

3.2 **RF Spoofing:** RF (Radio Frequency) Spoofing is a type of attack where attackers manipulate radio signals to deceive target devices or systems. By generating counterfeit RF signals, attackers can impersonate legitimate devices or networks, tricking the target into accepting false information or commands. RF Spoofing attacks are particularly concerning in wireless communication systems like Wi-Fi, Bluetooth, RFID, and IoT devices, as they can bypass security measures, gain unauthorized access, or manipulate communication protocols.

3.3 **Social Engineering:** Social Engineering involves psychological manipulation techniques used by attackers to deceive individuals or employees into divulging confidential information, performing actions, or granting access to restricted resources. Attackers exploit human psychology and trust to trick victims into disclosing sensitive data, clicking on malicious links, or executing harmful commands. Social Engineering attacks encompass tactics such as pretexting, phishing, baiting, or impersonation, leading to significant security breaches, data theft, or unauthorized access to systems.

3.4 **Fault Injection:** Fault Injection is a technique where attackers intentionally introduce faults or errors into a system to disrupt its normal operation, manipulate its behavior, or exploit vulnerabilities. This attack involves injecting faults at various levels of the system, including hardware, software, or communication channels. Fault Injection attacks can trigger unexpected behaviors, bypass security mechanisms, or extract sensitive information. Methods of fault injection include voltage glitching, clock glitching, laser fault injection, electromagnetic interference, and software fault injection techniques. These attacks are often used in security testing, penetration testing, or adversarial attacks to assess and exploit system vulnerabilities.

Table 4: Physical Attacks issues inside the analyzed project

Project Name	**Tampering**	**RF Spoofing**	**Social Engineering**	**Fault Injection**
TwitterMoodLight	**Not Feasible** - Limited attack surface, not interactive with external data.	**Not Feasible** - No direct interaction with RF signals.	**Feasible** - Can be tricked into accessing malicious links or content.	**Not Feasible** - Unlikely target for fault injection, no user input.
Energy Meter	**Feasible** - Can be tampered with physically or remotely to manipulate energy data.	**Feasible** - Vulnerable to RF signal manipulation to falsify energy readings.	**Feasible** - Social engineering attacks could trick users into divulging energy usage data.	**Feasible** - Fault injection could manipulate energy readings or disrupt functionality.
Multipurpose Smart Truck	**Feasible** - Vehicle systems or data could be tampered with remotely or physically.	**Feasible** - Vulnerable to RF spoofing for unauthorized vehicle control.	**Feasible** - Social engineering could trick drivers into compromising vehicle security.	**Feasible** - Fault injection could disrupt vehicle operations or manipulate data.

Arduino Self Driving Car	**Not Feasible** - Highly secure vehicle systems, limited external access.	**Not Feasible** - Secure RF communication protocols, minimal exposure.	**Not Feasible** - Limited interaction with external users, highly automated.	**Not Feasible** - Highly controlled software environment, limited external input.
IKEA PS 2014 DIY Lamp	**Not Feasible** - Limited attack surface, minimal interaction with external data.	**Not Feasible** - No RF interaction, standalone functionality.	**Not Feasible** - No user-facing components, no interaction with external users.	**Not Feasible** - Simple functionality, no external input.
Battery Node	**Feasible** - Battery data could be tampered with remotely or physically.	**Feasible** - Vulnerable to RF signal manipulation to alter battery data.	**Feasible** - Social engineering could trick users into compromising battery data.	**Feasible** - Fault injection could disrupt battery functionality or manipulate data.
Control of Your TV Using Alexa and Arduino IoT Cloud	**Feasible** - Device controls could be tampered with remotely.	**Feasible** - Vulnerable to RF signal manipulation for unauthorized control.	**Feasible** - Social engineering attacks could trick users into compromising device controls.	**Feasible** - Fault injection could disrupt device functionality or manipulate controls.
Combustion Gas IoT Monitor	**Feasible** - Gas readings could be tampered with remotely or physically.	**Feasible** - Vulnerable to RF spoofing for falsifying gas readings.	**Feasible** - Social engineering attacks could trick users into divulging gas data.	**Feasible** - Fault injection could manipulate gas readings or disrupt functionality.
SmartOutlet	**Feasible** - Device controls could be tampered with remotely.	**Feasible** - Vulnerable to RF signal manipulation for unauthorized control.	**Feasible** - Social engineering attacks could trick users into compromising device controls.	**Feasible** - Fault injection could disrupt device functionality or manipulate controls.
IoT Pressure Sensor	**Feasible** - Pressure readings could be tampered with remotely or physically.	**Feasible** - Vulnerable to RF spoofing for falsifying pressure readings.	**Feasible** - Social engineering attacks could trick users into divulging pressure data.	**Feasible** - Fault injection could manipulate pressure readings or disrupt functionality.
Chat with ChatGPT through Arduino IoT Cloud	**Feasible** - Chat interactions could be tampered with remotely.	**Feasible** - Vulnerable to RF signal manipulation for unauthorized control.	**Feasible** - Social engineering attacks could trick users into compromising chat interactions.	**Feasible** - Fault injection could disrupt chat functionality or manipulate messages.
Capacitive Soil Moisture Sensor	**Feasible** - Soil moisture readings could be tampered with remotely or physically.	**Not Applicable** - No RF interaction, standalone functionality.	**Feasible** - Social engineering attacks could trick users into divulging soil moisture data.	**Feasible** - Fault injection could manipulate soil moisture readings or disrupt functionality.
Pixel Cube	**Feasible** - Pixel data could be tampered with remotely.	**Not Applicable** - No RF interaction, standalone functionality.	**Feasible** - Social engineering attacks could trick users into divulging pixel data.	**Feasible** - Fault injection could manipulate pixel data or disrupt functionality.

WiFi Jammer	**Feasible** - Wi-Fi functionality could be tampered with remotely.	**Feasible** - Vulnerable to RF signal manipulation for disrupting Wi-Fi signals.	**Feasible** - Social engineering attacks could trick users into compromising Wi-Fi networks.	**Feasible** - Fault injection could disrupt Wi-Fi functionality or manipulate signals.
Send SMTP Email with Arduino	**Feasible** - Email functionality could be tampered with remotely.	**Feasible** - Vulnerable to RF signal manipulation for unauthorized email access.	**Feasible** - Social engineering attacks could trick users into compromising email accounts.	**Feasible** - Fault injection could disrupt email functionality or manipulate messages.
Smart Energy Consumption Meter	**Feasible** - Energy data could be tampered with remotely or physically.	**Feasible** - Vulnerable to RF signal manipulation to falsify energy readings.	**Feasible** - Social engineering attacks could trick users into divulging energy usage data.	**Feasible** - Fault injection could disrupt chat functionality or manipulate messages
Smart Home Monitor	**Feasible** - Sensor readings could be tampered with remotely.	**Feasible** - Vulnerable to RF signal manipulation for unauthorized control.	**Feasible** - Social engineering attacks could trick users into compromising sensor data.	**Feasible** - Fault injection could disrupt sensor functionality or manipulate data.
Arduino Light Controller	**Feasible** - Light controls could be tampered with remotely.	**Feasible** - Vulnerable to RF signal manipulation for unauthorized control.	**Feasible** - Social engineering attacks could trick users into compromising light controls.	**Feasible** - Fault injection could disrupt light functionality or manipulate controls.
Temperature Monitoring with Arduino IoT Cloud using DHT22	**Feasible** - An attacker physically manipulates the DHT22 sensor to provide false temperature readings.	**Not Feasible** - This project does not involve RF communication.	**Not Feasible** - Social engineering attacks such as phishing emails are not applicable to this project.	**Not Feasible** - Fault injection attacks require specific vulnerabilities in the system, which are not present in this project.
Stratum-1 GNSS Time Server	**Feasible** - Time synchronization could be tampered with remotely or physically.	**Feasible** - Vulnerable to RF spoofing for falsifying time data.	**Feasible** - Social engineering attacks could trick users into divulging time.	**Feasible** - Fault injection could manipulate time data or disrupt functionality.
Smart Indoor Hydroponic Farming	**Feasible** - Sensor readings could be tampered with remotely.	**Feasible** - Vulnerable to RF signal manipulation for unauthorized control.	**Feasible** - Social engineering attacks could trick users into compromising sensor data.	**Feasible** - Fault injection could disrupt sensor functionality or manipulate data.
Surveillance Bot	**Feasible** - Camera feeds could be tampered with remotely.	**Feasible** - Vulnerable to RF signal manipulation for unauthorized control.	**Feasible** - Social engineering attacks could trick users into compromising surveillance data.	**Feasible** - Fault injection could disrupt camera functionality or manipulate feeds.

4. Side Channel Attacks:

Side channel attacks are a class of attacks that exploit information leaked from the physical implementation of a system rather than directly targeting the algorithms or protocols. These attacks rely on unintended side effects or "side channels," such as power consumption, electromagnetic emanations, timing variations, or even sound emissions, to infer sensitive information about the system's cryptographic operations or internal state.

4.1. Power Analysis Attacks: Power analysis attacks involve analyzing the power consumption patterns of a device during cryptographic operations. By observing fluctuations in power consumption, an attacker can infer information about the cryptographic keys or data being processed, allowing them to recover sensitive information.

4.2. Electromagnetic Attacks: Electromagnetic attacks exploit electromagnetic radiation emitted by electronic devices during cryptographic operations. By monitoring and analyzing these emissions, attackers can extract information about the cryptographic keys or operations, compromising the security of the system.

4.3. Timing Attacks: Timing attacks involve exploiting variations in the execution time of cryptographic algorithms or operations. By measuring the time taken to perform certain computations, an attacker can infer sensitive information about the cryptographic keys or data, potentially leading to the recovery of secret information.**.**

4.4. Environmental Attacks: Environmental attacks exploit environmental factors such as temperature, humidity, or radiation to compromise the security of a system. These attacks can affect the physical properties of electronic devices, potentially leaking sensitive information through unintended side channels. However, environmental attacks are often more challenging to execute and may require specialized equipment or controlled conditions.

Side channel attacks pose a significant threat to the security of cryptographic systems, as they can bypass traditional security measures based on algorithmic complexity or key length. Mitigating side channel attacks requires implementing countermeasures such as secure hardware design, noise injection, or algorithmic masking to reduce the leakage of sensitive information through unintended channels.

Table 5: Side Channel Attacks issues inside the analyzed projects

Project Name	**Power Analysis Attacks**	**Electromagnetic Attacks**	**Timing Attacks**	**Environmental Attacks**
TwitterMoodLight	**Feasible** - Analyze power consumption patterns to extract cryptographic keys.	**Feasible** - Analyze electromagnetic emissions to extract cryptographic keys.	**Feasible** - Analyze timing data to extract cryptographic keys.	**Not Feasible** - Limited electronic components and environmental factors make it an unlikely target.
Energy Meter	**Feasible** - Analyze power consumption patterns to infer encryption key usage.	**Feasible** - Measure electromagnetic emissions to potentially extract cryptographic keys.	**Not Feasible** - Energy meters typically lack precise timing data for exploitation.	**Not Feasible** - Environmental factors may not significantly affect the encryption process.
Multipurpose Smart Truck	**Feasible** - Monitor power consumption to extract keys.	**Feasible** - Analyze electromagnetic emissions for potential key extraction.	**Feasible** - Analyze timing data to infer key usage patterns.	**Not Feasible** - Environmental factors may not significantly affect.
Arduino self driving car	**Not Feasible** - Limited electronic components and lack of encryption make it an unlikely target.	**Not Feasible** - Minimal electronic components reduce electromagnetic emissions.	**Not Feasible** - The self-driving car lacks precise timing data for exploitation.	**Not Feasible** - Limited environmental impact on the encryption process.
IKEA PS 2014 DIY Lamp	**Not Feasible** - Limited electronic components make it less susceptible to power analysis.	**Not Feasible** - Minimal electronic components reduce electromagnetic emissions.	**Not Feasible** - The lamp lacks precise timing data for exploitation.	**Not Feasible** - Limited environmental impact on the encryption process.

Battery node	**Feasible** - Analyze power consumption patterns to potentially extract cryptographic keys.	**Not Feasible** - Batteries may not emit significant electromagnetic signals.	**Not Feasible** - Limited timing data available for exploitation.	**Not Feasible** - Environmental factors may not significantly affect the encryption process.
Control of your TV using Alexa and Arduino IoT cloud	**Feasible** - Analyze power consumption patterns to potentially extract cryptographic keys.	**Not Feasible** - Minimal electronic components reduce electromagnetic emissions.	**Not Feasible** - Limited timing data available for exploitation.	**Not Feasible** - Limited environmental impact on the encryption process.
Combustion gas IoT monitor	**Feasible** - Analyze power consumption patterns to potentially extract cryptographic keys.	**Not Feasible** - Limited electronic components and environmental factors make it an unlikely target.	**Not Feasible** - Limited timing data available for exploitation.	**Not Feasible** - Environmental factors may not significantly affect the encryption process.
SmartOutlet	**Feasible** - Analyze power consumption patterns to potentially extract cryptographic keys.	**Not Feasible** - Minimal electronic components reduce electromagnetic emissions.	**Not Feasible** - Limited timing data available for exploitation.	**Not Feasible** - Limited environmental impact on the encryption process.
IoT Pressure sensor	**Feasible** - Analyze power consumption patterns to potentially extract cryptographic keys.	**Not Feasible** - Minimal electronic components reduce electromagnetic emissions.	**Not Feasible** - Limited timing data available for exploitation.	**Not Feasible** - Environmental factors may not significantly affect the encryption process.
Chat with ChatGPT through Arduino IoT Cloud	**Feasible** - Analyze power consumption patterns to potentially extract cryptographic keys.	**Not Feasible** - Minimal electronic components reduce electromagnetic emissions.	**Not Feasible** - Limited timing data available for exploitation.	**Not Feasible** - Limited environmental impact on the encryption process.
Capacitive soil moisture sensor	**Feasible** - Analyze power consumption patterns to potentially extract cryptographic keys.	**Not Feasible** - Limited electronic components and environmental factors make it an unlikely target.	**Not Feasible** - Limited timing data available for exploitation.	**Not Feasible** - Environmental factors may not significantly affect the encryption process.
Pixel cube	**Feasible** - Analyze power consumption patterns to potentially extract cryptographic keys.	**Not Feasible** - Minimal electronic components reduce electromagnetic emissions.	**Not Feasible** - Limited timing data available for exploitation.	**Not Feasible** - Limited environmental impact on the encryption process.
Wifi Jammer	**Feasible** - Analyze power consumption patterns to potentially extract cryptographic keys.	**Not Feasible** - Minimal electronic components reduce electromagnetic emissions.	**Not Feasible** - Limited timing data available for exploitation.	**Not Feasible** - Limited environmental impact on the encryption process.

Send SMTP email with Arduiono	**Feasible** - Analyze power consumption patterns to potentially extract cryptographic keys.	**Not Feasible** - Minimal electronic components reduce electromagnetic emissions.	**Not Feasible** - Limited timing data available for exploitation.	**Not Feasible** - Limited environmental impact on the encryption process.
Smart energy consumption meter	**Feasible** - Analyze power consumption patterns to potentially extract cryptographic keys.	**Not Feasible** - Minimal electronic components reduce electromagnetic emissions.	**Not Feasible** - Limited timing data available for exploitation.	**Not Feasible** - Environmental factors may not significantly affect the encryption process.
Smart home monitor	**Feasible** - Analyze power consumption patterns to potentially extract cryptographic keys.	**Not Feasible** - Minimal electronic components reduce electromagnetic emissions.	**Not Feasible** - Limited timing data available for exploitation.	**Not Feasible** - Environmental factors may not significantly affect the encryption process.
Arduino light controller	**Feasible** - Analyze power consumption patterns to potentially extract cryptographic keys.	**Not Feasible** - Minimal electronic components reduce electromagnetic emissions.	**Not Feasible** - Limited timing data available for exploitation.	**Not Feasible** - Limited environmental impact on the encryption process.
Temperature Monitoring with Arduino IoT Cloud using DHT22	**Not Feasible** - Unlikely to be vulnerable to power analysis attacks.	**Not Feasible** - Unlikely to be susceptible to electromagnetic attacks.	**Not Feasible** - Unlikely to be vulnerable to timing attacks.	**Not Feasible** - Unlikely to be affected by environmental factors in this context.
Stratum-1 GNSS Time server	**Feasible** - Analyze power consumption patterns to potentially extract cryptographic keys.	**Not Feasible** - Limited electronic components and environmental factors make it an unlikely target.	**Not Feasible** - Limited timing data available for exploitation.	**Not Feasible** - Environmental factors may not significantly affect the encryption process.
Smart Indoor Hydroponic Farming	**Feasible** - Analyze power consumption patterns to potentially extract cryptographic keys.	**Not Feasible** - Limited electronic components and environmental factors make it an unlikely target.	**Not Feasible** - Limited timing data available for exploitation.	**Not Feasible** - Environmental factors may not significantly affect the encryption process.
Surveillance Bot	**Feasible** - Analyze power consumption patterns to potentially extract cryptographic keys.	**Not Feasible** - Limited electronic components and environmental factors make it an unlikely target.	**Not Feasible** - Limited timing data available for exploitation.	**Not Feasible** - Environmental factors may not significantly affect the encryption process.

V. DISCUSSION

Our attack analysis model is based on five security attacks: Network. Payload, Physical, and Side Channel. This section describes some of our findings from the analysis on the behavior of hobbyist developers.

Table 6: Reports a quantitative summary of the observed security issues in the analyzed repositories.

Attack Category	Overall Feasibility	Security Issue	Feasibility for Each Category	Feasible Projects	Infeasible Projects
Network	57%	Traffic Analysis Attack	82%	18	4
		Routing Attacks/Routing Information Attacks	50%	11	11
		Man-in-the-Middle Attack (MiTM)	73%	16	6
		Replay Attacks	45%	10	12
		Denial/DDoS Attack	36%	8	14
Payload	63%	Malware attack	86%	19	3
		Code injection	91%	20	2
		Phishing attack	32%	7	15
		Command Injection	95%	21	1
		Remote Code execution	32%	7	15
Physical	82%	Tampering	86%	19	3
		RF Spoofing	73%	17	3 + 2 Not Applicable
		Social Engineering	86%	20	2
		Fault Injection	83%	19	3
Side Channel	27%	Power Analysis Attacks	86%	19	3
		Electromagnetic Attacks	14%	3	19
		Timing Attacks	9%	2	20
		Environmental Attacks	0	0	22

From the above data, we are able to observe that almost all the projects we have analysed are vulnerable to either an active or a passive attack. Interestingly, even though not all the projects are susceptible to active attacks, the same cannot be said for passive attacks. In general, there wasn't any form of defense against passive attacks. Methods like using unique nonce and timestamps during packet transmission, at least for critical data, could considerably improve resistance against such attacks. On the other hand, encryption is a key method to safeguard against active attacks. This becomes much more evident when we look at our findings from rest of the categories.

In our analysis, the average frequency of issues falling into the *Network Attacks* category is 57%. When we examined the Arduino-like IoT projects, we found that most of the time, the developers did not encrypt the data in their self-made Internet of things applications. Encryption aside, poor authentication mechanisms and lack of integrity checks was also prevalent throughout all the projects inspected. Nonetheless, encryption and secure authentication practices are especially helpful in preventing (or lessening) attacks that fall into this group. In fact, encryption can even lessen the effects of both Passive and Active attacks as well as some forms of Routing attacks. The built-in defense against DDoS/DoS or flooding attacks on the Arduino projects is basically nonexistent. Though we cannot completely rule out the potential that some developers set up their routers to safeguard the Arduino, we did not uncover any further protection based on what we read in the repositories.

Payload based attacks, as seen earlier, is a popular way attackers employ to exploit all kinds of devices connected to the internet, and of course IoT are no exception. Several projects we have analysed have an interface through which input or certain data is taken and passed to the devices. Leveraging this, attackers could pass malicious data via their entries or manipulate the devices which would allow them to steal information e.g., *Remote Code Execution*, *Command Injection*. Unfortunately, all those projects have absolutely zero defense mechanisms to such prevent malicious data injections. In order to mitigate such issues, developers should enforce secure code practices in their programs like input validation, sanitization of user inputs, data verification (e.g., firmware updates) and bounds checking for buffers.

We have already discussed the potential impact of *Physical Attacks* on Arduino devices. They are, in fact, simple to override and violate which is evident by their average feasibility rate of 82%. For this reason, developers need to be aware that safeguarding their devices physically is crucial to preventing a number of these kinds of problems, which might arise from physical access to the board by hostile users. Furthermore, since these boards are resource-constrained, they may be a simple target for DoS attacks. We also observed a few battery-powered boards in a few instances, especially one that had explicitly warned about a weakness in the battery used. Due to such components, a battery-exhaustion or battery tampering assault may result in damage for those projects. Even though it might be difficult to address all these issues, encryption may be able to lessen some problems even for this attack category. Encrypting sensitive information can improve protection against these attacks. One such glaring problem that drew our attention was credentials hardcoded as plaintext within the code, increasing the exploitability even further. Therefore, encrypting or obfuscating sensitive information, such as network SSID and password, is a best practice that was neglected in the investigated repositories.

At last, we have the *Side Channel* attack category. Although this category has the lowest feasibility rate comparatively, looking deeper into the security issues under it will give us a better understanding. As we can see, 86% of the projects are vulnerable to *Power Analysis* attacks but on the contrary none of them are susceptible to *Environmental Attacks.* This drastic difference is due to the simple factor that all projects need power to run but they are not necessarily exposed to external environments. Nevertheless, the threat cannot be ruled out since the projects we've evaluated are done on a smaller scale and hence environmental attacks don't have much effect on them. But let's say a larger scale system, like smart farming is implemented over acres of land, inspired from one of the several open-source Arduino projects, these kinds of attacks can cause severe damage. Hence, it is important for developers to consider and safeguard against side channel attacks using through the code itself using previously suggested measures like power supply modulation, introducing noise etc. There are also Arduino libraries which developers can take advantage of such as *protectedAES*[*27],* which offer AES encryption with countermeasures against side-channel attacks, such as first-order masking and random interrupts.

VI. FUTURE ENHANCEMENTS

The Internet of Things has continued to and will continue to evolve rapidly moving forward. Due to such growth and widespread adoption, new IoT projects and devices are published every day. Hence, we would like to evaluate more types of devices and technologies used in DIY projects, to discover new kinds of threats that might cause dire consequences. Additionally, with references from various researchers and MITRE ATT&CK [28] tactics, we also plan to create a database mapped specifically to IoT threats and vulnerabilities, which hopefully could serve useful to all IoT developers. To conclude, we hope to expand our analysis to more DIY technologies and make useful contributions to this community.

VII. CONCLUSION

In this study, we have conducted a source code level analysis of open source IoT projects from multiple repositories based on Arduino like devices. Upon investigation, we discovered serious security concerns leaving the projects vulnerable to issues which may potentially lead to exfiltration of sensitive data or manipulation of the IoT devices. This has shown how most developers neglect even basic cybersecurity concepts.

We have detailed a variety of attacks that can be carried out in the projects' source codes that have been reviewed. Several users might download and use these codes on their own boards without thinking about the possible security risks, because they are publicly accessible over the internet. Not to mention, inexperienced developers could use these projects as reference. and build upon them to create their own. These security flaws could therefore be carried over onto the freshly created projects, especially ones with network connection would automatically become vulnerable. Hence, it is crucial that developers take secure code practices and security concepts into consideration, like the suggestions made in this paper for example, while designing their projects to at least safeguard against easily exploitable and critical vulnerabilities, if not all In conclusion, identification of several security flaws prevalent in Arduino-like devices was possible through this analysis. We hope that developers will find the information in this study helpful in assessing and strengthening the security of their projects.

In conclusion, identification of several security flaws prevalent in Arduino-like devices was possible through this analysis. We hope that developers will find the information in this study helpful in assessing and strengthening the security of their projects.

REFERENCES

[1] GitHub. Available online: https://github.com/ (accessed on 3 February 2024).

[2] Arduino Project Hub. Available online: https://projecthub.arduino.cc/ (accessed on 3 February 2024).

[3] Autodesk Instructables. Available online: https://www.instructables.com/.

[4] TwitterMoodLight @ GitHub. Available online: https://github.com/HanYangZhao/MoodLight (accessed on 11 February 2024).

[5] Arduino IoT based Energy Meter @ Arduino Project Hub. Available online: https://projecthub.arduino.cc/Arduino_Genuino/arduino-iot-based-energy-meter-39e01b (accessed on 13 February 2024).

[6] Multipurpose Smart Truck @ GitHub. Available online: https://github.com/FahimFBA/multipurpose-smart-truck (accessed on 15 February 2024).

[7] Arduino Self-Driving Cars with ultrasonic sensors @ GitHub. Available online: https://github.com/abhineetraj1/arduino-self-driving-car (accessed on 16 February 2024).

[8] IKEA PS 2014 Lamp @ GitHub. Available online: https://github.com/biagiobotticelli/IKEA_DIY_Lamp (accessed on 23 February 2024).

[9] Battery Node @ GitHub. Available online: https://github.com/happytm/BatteryNode (accessed on 5 February 2024).

[10] Full Control of your TV using Alexa and Arduino IoT Cloud @ Arduino Documents. Available online: https://docs.arduino.cc/tutorials/projects/full-control-of-your-tv-using-alexa-and-arduino-iot-cloud/ (accessed on 15 February 2024).

[11] Arduino Combustion Gas IoT Monitor @ Arduino Project Hub. Available online: https://projecthub.arduino.cc/strawbob/arduino-combustion-gas-iot-monitor-421286 (accessed on 1 March 2024).

[12] SmartOutlet-IOT @ GitHub. Available online: https://github.com/ManolescuSebastian/SmartOutlet-IOT (accessed on 23 February 2024).

[13] IoT Pressure Sensor @ Arduino Project Hub. Available online: https://projecthub.arduino.cc/wahltharvey/iot-pressure-sensor-mkr-gsm-arduino-cloud-google-sheets-4f507f (accessed on 8 March 2024).

[14] Chat with ChatGPT through Arduino IoT Cloud @ Arduino Project Hub. Available online: https://projecthub.arduino.cc/dbeamonte_arduino/chat-with-chatgpt-through-arduino-iot-cloud-6b4ef0 (accessed on 2 March 2024).

[15] Capacitive Soil Moisture Sensor @ GitHub. Available online: https://github.com/RonMcKay/capacitive-soil-moisture-sensor (accessed on 14 February 2024).

[16] Pixel Cube @ GitHub. Available online: https://github.com/mstrlaw/pixel_cube (accessed on 8 February 2024).

[17] Wi-Fi Jammer with nRF24L01 @ Arduino Project Hub. Available online: https://projecthub.arduino.cc/CiferTech/how-to-make-wifi-jammer-but-with-nrf24l01-2c6ea1 (accessed on 8 February 2024).

[18] Send SMTP Email with Arduino and ESP8266 @ Arduino Project Hub. Available online: https://projecthub.arduino.cc/Hans63/send-smtp-email-with-arduino-and-esp8266-dbcaf5 (accessed on 11 February 2024).

[19] Smart Energy Consumption Meter @ Arduino Project Hub. Available online: https://projecthub.arduino.cc/yasirutishan/smart-energy-consumption-meter-iot-1f7d52.

[20] Smart Home Monitor @ Arduino Project Hub. Available online: https://projecthub.arduino.cc/cospis/smart-home-monitor-cloudgames2022-8e3eb7.

[21] Arduino Light Controller Using MKR IoT Carrier @ Arduino Project Hub. Available online: https://projecthub.arduino.cc/ratack0/arduino-light-controller-using-mkr-iot-carrier-2e1c56.

[22] Temperature Monitoring with Arduino IoT Cloud using DHT22 @ Arduino Project Hub. Available online: https://projecthub.arduino.cc/attari/temperature-monitoring-with-arduino-iot-cloud-using-dht22-cd8e34.

[23] GNSSTimeServer @ GitHub. Available online: https://github.com/Montecri/GNSSTimeServer/.

[24] IETP SIHFarming @ GitHub. Available online: https://github.com/Nebyat19/IETP-SIHFarming/.

[25] Reginald: a UDP Surveillance Bot @ Autodesk Instructables. Available online: https://github.com/Nebyat19/IETP-SIHFarming/.

[26] T. Sasi, A. H. Lashkari, R. Lu, P. Xiong, S. Iqbal, "A comprehensive survey on IoT attacks: Taxonomy, detection mechanisms and challenges", in Journal of Information and Intelligence, Dec. 2023.

[27] ProtectedAES Library @ Arduino Reference. Available online: https://reference.arduino.cc/reference/en/libraries/protectedaes/

[28] ATT&CK Matrix @ MITRE ATT&CK. Available online: https://attack.mitre.org/

Contribution Summary:

Student Name	Student ID	Contributions
Karthik Perumal	40258997	Projects Evaluated: 1. Pixel cube 2. How to make Wi-Fi Jammer with nRF24L01 For Final Report: Made the Classification under Materials & Methods.
Richard Rathinaraj	40280830	Projects Evaluated: 1. Smart truck 2. Twitter mood light For Final Report: Made Payload Attacks and Network attacks sections under Results.
Dharaneesh Kuppusamy Ganesan	40268161	Projects Evaluated: 1. TV remote control through Alexa and Arduino 2. Energy meter For Final Report: Made Physical attacks and Network attacks sections under Results. Formatted the Final report in IEEE format.
Lingeshwaran Rameshkumar	40277718	Projects Evaluated: 1. Reginald: UDP Surveillance Bot 2. Smart Indoor Hydroponic Farming 3. Stratum 1 - GNSSTimeServer For Final Report: Made Discussion, Future Work and Conclusion sections for the Final Report. Reviewed the report and made corrections.
Sree Lakshme Srinivasan	40273589	Projects Evaluated: 1. IKEA PS 2014 DIY lamp 2. Arduino self-driving cars with ultrasonic sensors For Final Report: Made the Methodologies section. Formatted the Final report to IEEE Format.
Sai Ashwin Dhayalan	40260758	Projects Evaluated: 4. Smart outlet 5. ChatGPT through Arduino 6. Industrial pressure using IOT For Final Report: Made Side-Channel and Network attacks sections under Results.
Chris Regy Vallikunnathu	40232485	1. Arduino light controller using MKR IoT carrier 2. Temperature monitoring with Arduino IoT cloud using DHT22
Harshitha Raja	40270598	Projects Evaluated: 1. Capacitive soil moisture sensor 2. Send SMTP email with Arduino and ESP8266 For Final Report:

		Made the Analyzed Projects part under Methodologies.
Harleen Kaur	40232489	Projects evaluated: 1. Smart energy consumption meter 2. Smart home monitor
Savithri Balasubramanian	40273594	Projects Evaluated: 1. Battery Node 2. Arduino Combustion Gas IoT monitor For Final Report: Created the Abstract, Introduction and References sections.