# Simulating and Evaluating Security Vulnerabilities in Smart Home IoT Devices Using Node-RED

MSc Research Project
Programme Name

Shivaprasad Hegde
Student ID: 22224670

School of Computing
National College of Ireland

Supervisor: Dr Raza UI Mustafa

**National College of Ireland**

**MSc Project Submission Sheet**

**School of Computing**

| | | | |
|---|---|---|---|
| **Student Name:** | Shivaprasad Hegde | | |
| **Student ID:** | 22224670 | | |
| **Programme:** | MSc Cybersecurity | **Year:** | 2023 |
| **Module:** | Practicum | | |
| **Supervisor:** | Dr Raza UI Mustafa | | |
| **Submission Due Date:** | 02/09/2024 | | |
| **Project Title:** | Simulating and Evaluating Security Vulnerabilities in Smart Home IoT Devices Using Node-RED | | |
| **Word Count:** | 6482 | **Page Count 18** | |

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

**Signature:** Shivaprasad M Hegde

**Date:** 02/09/2024

**PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST**

| | |
|---|---|
| Attach a completed copy of this sheet to each project (including multiple copies) | □ |
| **Attach a Moodle submission receipt of the online project submission,** to each project (including multiple copies). | □ |
| **You must ensure that you retain a HARD COPY of the project**, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer. | □ |

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

| **Office Use Only** | |
|---|---|
| Signature: | |
| Date: | |
| Penalty Applied (if applicable): | |

# Simulating and Evaluating Security Vulnerabilities in Smart Home IoT Devices Using Node-RED

Shivaprasad Hegde
22224670

**Abstract**

This study investigates the security vulnerabilities in smart home Internet of Things (IoT) devices using Node-RED, a flow-based development tool for visual programming. As the number of IoT devices are increasing every day it is important to verify that these devices are safe to use and do not possess any security flaw which may impact the end user. This research focusses on simulating and analysing the vulnerabilities within a controlled smart home virtual environment to understand the threats and develop mitigation strategies. The devices are configured with real-world vulnerabilities to simulate the exact real-world scenario. By configuring these vulnerabilities this study demonstrates the risk of unpatched systems and weak authentication methods. The findings reveal that vulnerable devices are prone to cyber-attacks, while secure systems are safe. This research contributes to enhancing IoT security by proposing targeted strategies to safeguard smart home ecosystems from cyber threats.

## 1 Introduction

The use of Internet of Things has revolutionized modern industry. These devices have ensured that everyday devices are connected to streamline daily activities, optimize industrial processes and create smart home environments. The rapid growth in the industry has been proven with almost every home having one or the other IoT device. From thermostats to smart refrigerators IoT devices are everywhere. But, unlike conventional computers these devices are resource-constrained, with limited processing power, memory and battery life. This means that these devices are more susceptible to cyber-attacks.

Attacks on IoT devices have risen by 107% year-over-year in the first half of 2024. The reason for the drastic increase in the number of attacks on IoT devices are very sub-par security standards of the IoT devices. Attacks prefer easier targets to try and exploit and IoT devices are among the easiest targets. One of the biggest factors in this increase of attacks is the CVE-2023-1389, which is a TP-Link command injection vulnerability. This vulnerability has impacted over 20% of the SMBs by itself. The 2024 cyberattack on Roku compromised over 576,000 accounts and marked a significant escalation in cybersecurity threats to IoT devices in homes. This incident highlighted the vulnerabilities of IoT devices and raised important questions about consumer trust, regulatory compliance, and the integration of cybersecurity measures across technological ecosystems. (Riddles, 2024) Roku was compromised by credential sniffing, where hackers employed stolen usernames and passwords from previous breaches to access Roku accounts. The target of this attack was to capitalize on reusing passwords across multiple platforms, revealing several vital vulnerabilities. The increasing number of IoT devices, such as smart TVs and home security systems, enlarged the attack surface, providing more targets for cybercriminals. Widespread reuse of passwords across different services exposed user accounts to greater risk. The final domino fell due to the lack of security features in many IoT devices, like two-factor authentication, making them particularly susceptible to these cyberattacks. (Media, 2024)

Manufacturers often prioritize speed and cost of the device rather than focusing on the security of the device. Inadequate testing and the lack of user awareness on the implementation of best practices make these devices more vulnerable. IoT devices face numerous vulnerabilities like lack of encryption, weak authentication mechanisms and delayed patching of security flaws. According to the IoT security landscape report more than 60% of the users still make use of default passwords given to them by the IoT provider. This study focuses on simulating these vulnerabilities within a smart home environment using Node-RED to better understand their impact and develop strategies for mitigating potential cyber-attacks. The research aims to contribute to a safer future for smart home ecosystems by identifying and addressing the most prevalent security challenges facing IoT devices today.

### 1.1. Background

The Internet of Things has emerged as a revolution in technology that promises a world of devices that are interconnected to streamline daily life as well as optimize the process of industries and usher in an era of infrastructure that is smart (Ramakrishnan and Gaur, 2019). It is everywhere in everyone's life from thermosets that are learning the preferences to variables that are monitoring health. The potential in the application of the Internet of Things seems boundless. However, this growth is rapid and has outpaced the considerations of security which has left a vast network of devices of IoT vulnerable to exploitation (Daugherty and Wilson, 2022). The very nature of the devices of IoT is the root of this insecurity. These devices are often resource-constrained, unlike conventional computers. They pose limited power for processing as well as memory and battery life (Novo, 2021). This necessitates lightweight systems of operating as well as protocols of security which can be less robust than their counterparts on the devices that are traditional. Additionally, the number and variety of the devices of the Internet of Things poses a challenge. The manufacturers often prioritize speed to market as well as cost-effectiveness which leads to inadequate testing of security as well as implementation of the best practices (Swamy and Kota, 2020).

### 1.2. Research question

How can Node-RED be used to simulate specific vulnerabilities, such as buffer overflows and weak authentication in smart home IoT devices, and how can the effectiveness of mitigation strategies like patching, encryption, and strong password policies be measured in terms of exploit success rate, response time, and attack surface reduction?

### 1.3. Research Aim and Objectives

<u>Aim</u>

This dissertation aims to contribute to the development of a secure future for smart home ecosystems and identify as well as analyze the vulnerabilities in the interconnected environment of the Internet of Things.

Objectives

- To conduct a Thorough investigation to identify as well as categorize the vulnerabilities that are most prevalent within the smart home devices as well as protocols and the channels of communication.
- To evaluate the consequences that are the potential of these vulnerabilities.
- To explore as well as propose robust strategies of mitigation to address the vulnerabilities that were identified.
- To analyze the evolving landscape of smart home technology as well as identify the challenges of security that are emerging.

# 2 Related Work

## 2.1 Introduction

Homes have become networked centers of efficiency and convenience due to the vast use of Internet of Things (IoT) gadgets. However, there are critical cybersecurity problems raised through the short incorporation of clever technology. IoT tool vulnerabilities put human beings and larger social infrastructures at risk of the whole lot from privacy breaches to wider community invasion. Despite those risks, people still do not know enough about IoT protection, which is made worse via a loss of consumer education and standardized safety protocols. This assessment of the literature looks at current research on IoT vulnerabilities, how they affect the security of smart homes and a way to grow consumer understanding and decrease dangers within the hastily converting subject of the smart home era.

## 2.2 IoT vulnerabilities and their risks

With their exceptional connection and convenience, Internet of Things (IoT) devices have revolutionized industries from production to healthcare. But this connectivity also leaves those gadgets open to a huge range of cybersecurity risks, which puts both consumers and agencies in critical danger (Ghazal *et al.* 2020). Inadequate authentication processes are a notable weak spot.

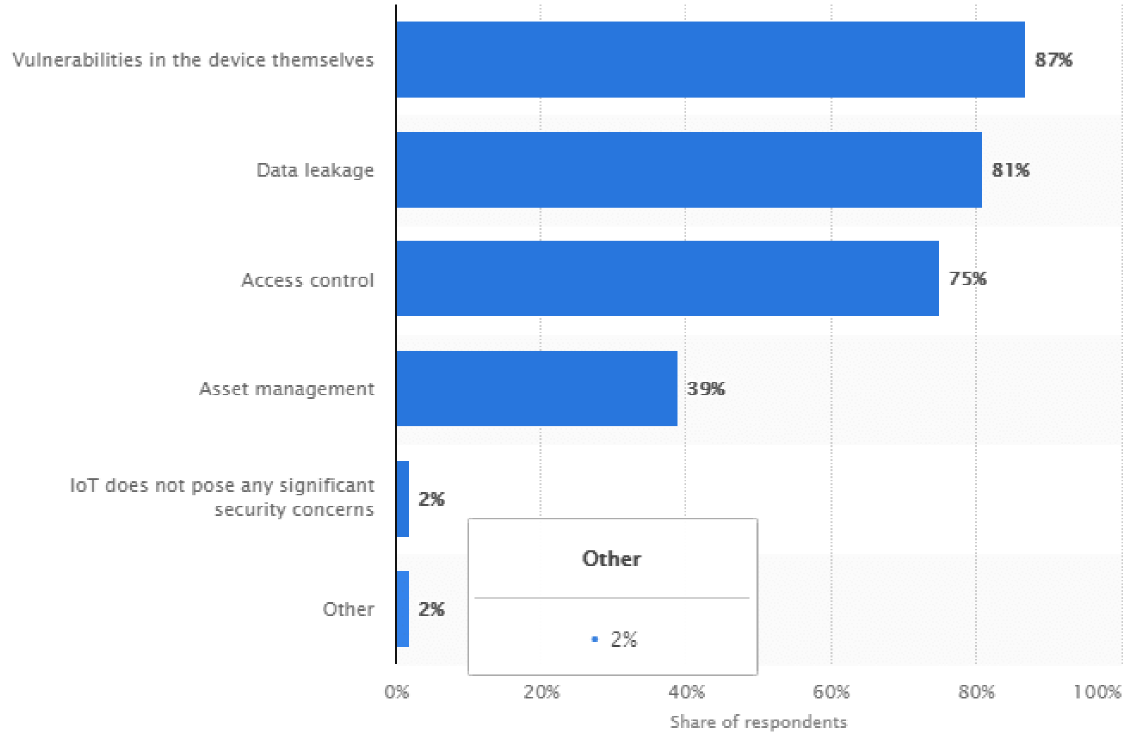


**Figure 2.1: IoT vulnerabilities and risks**

(Source: Statista, 2023)

Because many IoT devices even use default or easy-to-guess passwords, they're prone to brute-pressure attacks. Kandasamy *et al*. (2020) said that in 2016 the Mirai botnet took gain of prone passwords in the Internet of Things cameras and routers, inflicting massive disruptions in offerings supplied through using Dyn, a DNS enterprise, impacting famous sites inclusive of Twitter and Netflix. The lack of encryption in the statistics switch between Internet of Things devices and related networks is each different immoderate trouble. Srivastava *et al*. (2020) said that touchy statistics, such as personal facts or surveillance photos, can be intercepted and changed through unauthorized occasions in the absence of encryption. On the other hand, Ferrara *et al*. (2021) stated that keep in mind the safety holes determined in Ring smart doorbells, whereby weaknesses in the video streaming protocol also can moreover make it feasible for attackers to view stay feeds without requiring authentication, consequently jeopardizing man or woman privacy. Anand *et al*. (2020) stated that these dangers are further extended using the patching of vulnerabilities in IoT device firmware, that is both nonexistent or executed so slowly. It changed in 2019 when there were security holes in Philips Hue's clever bulbs that would let hackers into home networks. Firmware updates were speedily launched by way of Philips in response, highlighting the significance of making use of protection fixes on time to save viable vulnerabilities. In contrast, industries that include healthcare are extra prone because of the important nature of the IoT system they use. On the contrary, Hammi *et al*. (2022) stated that concerns concerning affected persons' safety and statistics privacy have been highlighted in recent years because of vulnerabilities in scientific IoT gadgets. For instance, the FDA has alerted sufferers to vulnerabilities in Medtronic insulin pumps that may be used to remotely modify insulin dosage settings, putting their health in extreme danger. Rytel *et al*. (2020) stated that the Stuxnet bug uncovered the weaknesses of industrial control structures (ICS) connected through the Internet of Things (IoT) in business settings, and the 2010 cyberattack specifically centered on Siemens structures utilized in nuclear electricity flora (Figueroa-Lorenzo *et al*. 2020). This assault highlighted the necessity for strong cybersecurity safeguards in industrial IoT deployments with the aid of demonstrating the possibility of bodily harm through cyberspace.

## 2.3 Education and user awareness

Successful education and consciousness campaigns are important for lowering those dangers and inspiring more secure IoT use in a selection of industries. Lee (2020) stated that the absence of standardized safety standards and customer-pleasant facts is an enormous subject. Many clients depart their gadgets open to exploitation because they're unaware of easy protection precautions like robotically updating firmware or changing default passwords. Even while IoT gadgets are becoming increasingly commonplace in day-by-day life, there may nevertheless be a big information and education hole among customers about the cybersecurity threats that come with them.

Businesses are crucial in informing customers approximately IoT security, practices, and risks. Kandasamy *et al*. (2020) stated that Google's "Be Internet Awesome" marketing campaign gives materials to teach youngsters and households about online safety, which also serves to raise attention to IoT protection. Similarly to this, Amazon offers commands and films on a way to ease devices that may be managed by using Alexa, stressing the significance of using robust passwords and turning on two-component authentication (Ghazal *et al*. 2020). Comparatively, due to the fact touchy facts are concerned, industries like finance and healthcare regularly undertake greater stringent instructional tasks. Regularly instructing its clients about phishing schemes and online fraud, banks and different monetary institutions also target IoT devices that are connected to their banking apps or clever domestic systems. To defend the confidentiality and integrity of patient information, healthcare practitioners teach their patients approximately the safety of scientific IoT devices (Srivastava *et al*. 2020). Nevertheless, problems exist in accomplishing each consumer. Only 31% of respondents to a Statista study from 2021 expressed confidence in their abilities to shield IoT gadgets, underscoring the continuing want for big schooling tasks. Furthermore, users must get ongoing education to live up to cutting-edge safety updates and rising dangers because of the complexity of IoT ecosystems and the short growth of gadgets. In conclusion, cooperation among tech firms, authorities' agencies, and educational institutions is necessary to enhance personal education and understanding concerning IoT safety. Organizations can efficiently lessen the risks associated with IoT vulnerabilities and sell safer digital surroundings by imparting customers with resources and understanding.

## 2.4 Theoretical framework

User behavior, risk management, and cybersecurity resilience are key to the theoretical basis for comprehending IoT safety. Strong safety tactics and activated assault recovery are key components of cybersecurity resilience. To guard networks, for instance, Cisco's IoT Threat Defense device combines segmentation, device visibility, and hazard detection. The impact of user cognizance and schooling on relaxed IoT utilization is examined through the user behavior concept, inclusive of the Technology Acceptance Model (TAM) (Ferrara *et al*. 2021). Just 40% of customers trade their default passwords, in keeping with an IBM survey, which highlights the need for extra person education. Proactive danger assessment and mitigation measures are emphasized with the aid of danger management philosophies. To keep away from and respond to cyberattacks, Siemens uses a radical danger management approach in its business IoT structures, combining chance intelligence and ongoing monitoring of the threats.

### 2.5 Literature gap

Notable gaps remain in our understanding of IoT vulnerabilities and user recognition in spite of sizable study. The absence of uniform safety protocols between IoT devices from one-of-a-kind producers is a substantial gap. Due to this discrepancy, there are differences in the safety stages of numerous gadgets, leaving them vulnerable to cyberattacks. Although Cisco's IoT Threat Defense framework gives large security safeguards, a few smaller manufacturers neglect to comprise comparable protocols, thereby rendering their products susceptible to vulnerabilities (Anand *et al*. 2020). Furthermore, the literature currently in the book shows a persistent lack of personal education and recognition of IoT security. Even though corporations like Google and Amazon provide getting-to-know substances, the simplest 31% of users experience relaxed safeguarding their IoT gadgets, according to a 2021 Statista ballot. This emphasizes the need for extra giants and a successful instructional program. To create and verify instructional tasks that can substantially enhance users' cognizance of and behavior related to IoT safety, greater study is needed. These gaps highlight the need for sustained research to create not unusual security guidelines and enhance consumer attention.

### 2.6 Summary

Significant protection gaps in IoT are proven by way of the literature, primarily because of the shortage of enterprise-wide security policies for diverse producers and gadgets. Numerous IoT gadgets are left open to cyberattacks because of this inconsistency, as evidenced by the way of the disparate protection protocols utilized by various businesses. Furthermore, there's nonetheless a loss of user education and awareness concerning IoT protection. Just 31% of customers experience self-belief in safeguarding their IoT devices, consistent with a 2021 Statista ballot, regardless of efforts by companies like Google and Amazon to offer training gear. These gaps display how urgently greater studies are needed to create common protection requirements and green schooling projects to enhance personal behavior and expertise in IoT security. For the IoT surroundings to be safer and greater relaxed, these troubles need to be resolved.

## 3 Research Methodology

The methodology aims to develop a virtual smart home which mimics the real-world devices. The smart home is hosted on Node-Red which is a flow-based, low-code development tool for visual programming developed by IBM. Node-Red provides a web-based flow editor and it can be used for wiring together hardware devices, APIs and online services as a part of the

Internet of Things. This setup will be used to create virtual smart devices which are capable of accepting commands from a recognised user agent similar to real world implementation of how smart devices can be controlled via a recognised application.

### Environment Setup

To install Node-red on our system, first we need to install the latest LTS version of Node.js from the official Node.js website. We then run the downloaded MSI file which requires local administrator rights. Once the installation is completed, we then add the path of the installed file to the environment variables so that it can be accessed via the command line interface. We then continue to install the global module of the Node-Red. Installing the global module of Node-Red is essential as it adds the command node-red to the system path. Once it is installed it can be run by typing node-red in the command line interface. The environment is communicated with the use of a tool called postman app. Postman is used to send requests to the smart home environment to interact with the devices.

### Device Simulation

Once the setup of Node-Red is complete we then proceed to simulate the devices. Two vulnerable devices and one secure device is setup to highlight the difference between the unpatched and patched devices in a smart home. The vulnerable devices are then tried to exploit to show the risks of an unpatched device in a smart home environment.

### Reason for selection of vulnerabilities for simulation

The decision to focus only on two vulnerabilities is justified by the need to create a controlled and manageable environment and observe the behaviour of the IoT devices. This also comes within the scope of a single research project. Testing two devices allowed for a detailed simulation of the vulnerabilities which provided accurate results.

### Rationale for Specific Vulnerabilities like CVE-2020-6007

The CVE-2020-6007 vulnerability represents a broader trend in IoT security where unpatched software and poor authentication methods are common in IoT products used by consumers. Many IoT devices lack timely updates and this is one such vulnerability which shows the risks of the unpatched software. The vulnerability of default password also depicts that various users make of the default password provided by the supplier and never change it. These vulnerabilities are easier to exploit when the IoT devices are managed poorly.

# 4 Design Specification

To create a smart home environment, we start the Node-Red from the command line interface. It is hosted on the localhost/127.0.0.1:8080. On going to that site, we see the Node-Red dashboard where we continue the setup of the smart home. A total of three devices are configured. Two smart lights and one smart lock. The smart light is configured with the CVE-2020-6007 vulnerability. The smart lock is configured with the default password. One more device is configured which is a smart light with fully updated firmware. There is also a motion sensed smart light to simulate the real-world behaviour of smart devices.

### System Architecture

Given below is the architecture of the developed smart home.

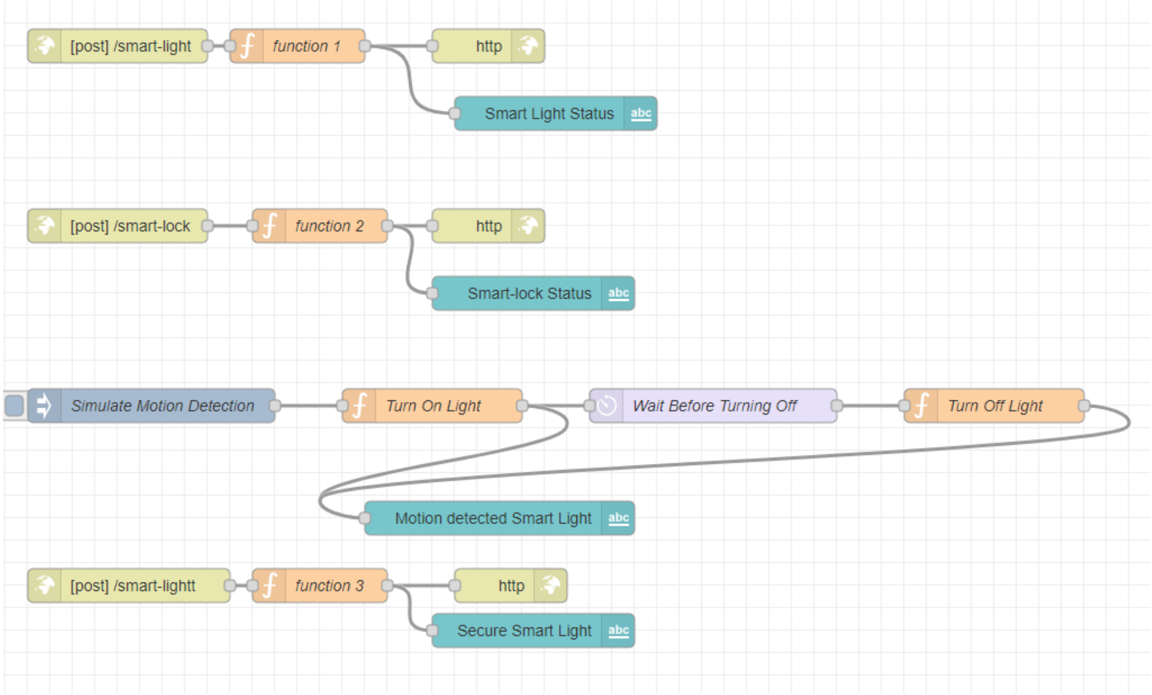


Figure 4.1: Architecture of the virtual smart-home environment.

As depicted in the above figure the environment is hosted on the localhost/127.0.0.1:8080. The dashboard shows the configuration of 4 devices. In all of the above flow diagrams there is a HTTP request which is sent from a recognized user agent. This mimicks the ability of the smart light to be controlled via an app. The http post request is sent to the function node. This is the node where the smart device is configured. The function node is connected to the HTTP out node which shows the status of the smart device.

### Node Configuration

The first flow diagram shows the configuration of the vulnerable smart light. The second flow diagram depicts a vulnerable smart lock. It is configured with a default password vulnerability which is then exploited by using different wordlists. The third flow diagram shows the motion sensed smart light. This light turns on if there is any motion detected and turn off if there is no motion detected for a duration of 5 seconds. The last flow diagram shows the configured secure smart light. This device is fully patched and updated to the latest firmware.

### User Interaction

The users can interact with the smart devices via a recognised user agent. In our case we have made use of Postman. Postman is an API platform for building and using APIs. Through postman we can also send HTTP requests to the configured node to operate the smart device. For instance the smart devices configured can be controlled via the Postman app. The smart devices are configured to accept requests only from the Postman app as the recognised user agent. This is done to mimick the real world scenario where the smart devices are controlled via the app.

# 5 Implementation

## 5.1 Device Configuration

Now the configuration of the devices are explained in detail:

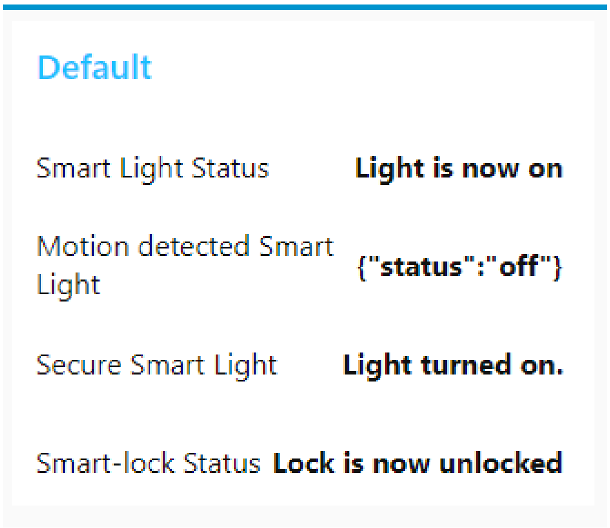
1. **Vulnerable Smart light:** The first virtual device is configured with the CVE-2020-6007 vulnerability. The CVE-2020-6007 vulnerability affected the Philips Hue Bridge model 2.X prior to and including version 1935144020. This contains a Heap-Based overflow when handling a long ZCL(Zigbee Cluster Library) string during the commissioning phase, resulting in a remote code execution. This vulnerability contained a severity score of 7.9(high). The function node is configured with this vulnerability. This does not affect the normal functioning of the light as the light can still be controlled via an app. The user cannot make out any difference in the light as it behaves normally but it also allows for remote code execution. The function node of the light simulates a buffer overflow vulnerability.

2. **Vulnerable Smart lock:** The second device is configured with the default password vulnerability. A weak pasword is used to authenticate the lock. This is done to show that weak passwords allow for the brute force attacks resulting in an unauthorised individual gaining access to the smart device.

3. **Motion sensed Smart light:** This is configured to show that the environment actually depitcs that of a real-world smart home. Here the smart light turns on when motion is detected and then turns off when there is no motion detected for a duration of 5 seconds.

4. **Secure Smart light:** The last device configured is that of the secure smart light which is fully patched and updated. This is configured to highlight the differences of the unpatched and patched systems. This device is also tried to be exploited with the help of remote code execution but it fails because the software is updated to the latest version which many users fail to do so.

5. **Text UI display:** I also configured a page in Node-Red where the status of all the devices can be seen. The picutre of that is given below.



Default

| | |
|---|---|
| Smart Light Status | Light is now on |
| Motion detected Smart Light | {"status":"off"} |
| Secure Smart Light | Light turned on. |
| Smart-lock Status | Lock is now unlocked |

Figure 5.1: UI page displaying the status of the smart devices.

## 5.2 Exploitation of Vulnerabilities

After the configuration of the devices, I then try to exploit it by running scripts against the device. The Heap-Based Buffer Overflow is a type of vulnerability where an application allows more data to be written to a buffer located on the heap than it can handle, causing data to overflow into adjacent memory. When this happens, it can overwrite valid data and control structures, leading to unpredictable behaviour, including the possibility of executing arbitrary code.

### Exploit for the smart light

The exploit script is created and it works as follows:

- It is designed to simulate exploiting a buffer overflow by sending an oversized payload.
- When the script is run it prompts the attacker to enter a command like on or off to control the light.
- Once the command is entered the script constructs an oversized payload by sending additional data beyond the predefined buffer size to simulate the effect of the buffer overflow.
- This oversized payload is sent to the Node-Red server via a HTTP POST request using the axios library.
- The device since it is configured with the buffer overflow vulnerability processes the request and allows the attacker remote access.

### Exploit for the smart lock

The exploit is created for the smart lock and it works as follows:

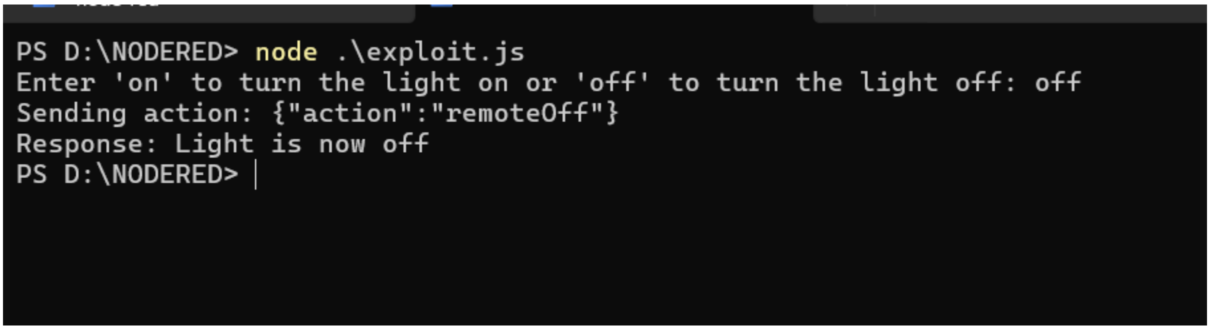
- The script performs a brute-force attack on the smart lock endpoint by testing for passwords to unlock the system.
- It uses the axios library to send POST requests to the smart lock endpoint.
- The script reads passwords from a file wordlist.txt which has all the default passwords.

- If a response indicating success is received the lock is unlocked and the password with which it is unlocked is displayed.

# 6 Evaluation

## 6.1 Experiment / Case Study 1: Exploiting the Vulnerable Smart Light

This case study demonstrates the use of scripts to exploit the vulnerable smart light. A script is developed in JavaScript which aims to exploit the Buffer overflow vulnerability of the smart light. A script to exploit the vulnerability is saved in a notepad file as a .js extension and it is then run on the windows PowerShell. The file is then run using the command node ./exploit.js. After this is run a prompt is shown on the screen to enter the command to control the light remotely. Once the command is typed on the screen it then sends a remote command to the virtual smart home and executes the command successfully.



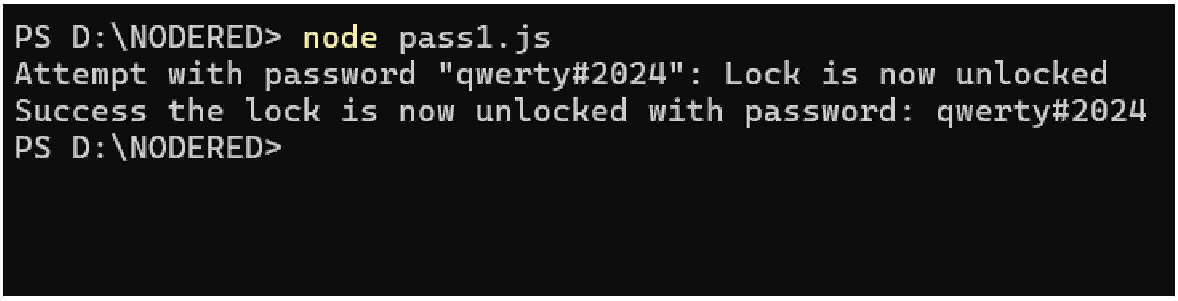
```
PS D:\NODERED> node .\exploit.js
Enter 'on' to turn the light on or 'off' to turn the light off: off
Sending action: {"action":"remoteOff"}
Response: Light is now off
PS D:\NODERED> |
```

Figure 6.1: Command to exploit the vulnerability

The above figure shows that the light is controlled remotely via an attacker who is able to turn the light off and on remotely.

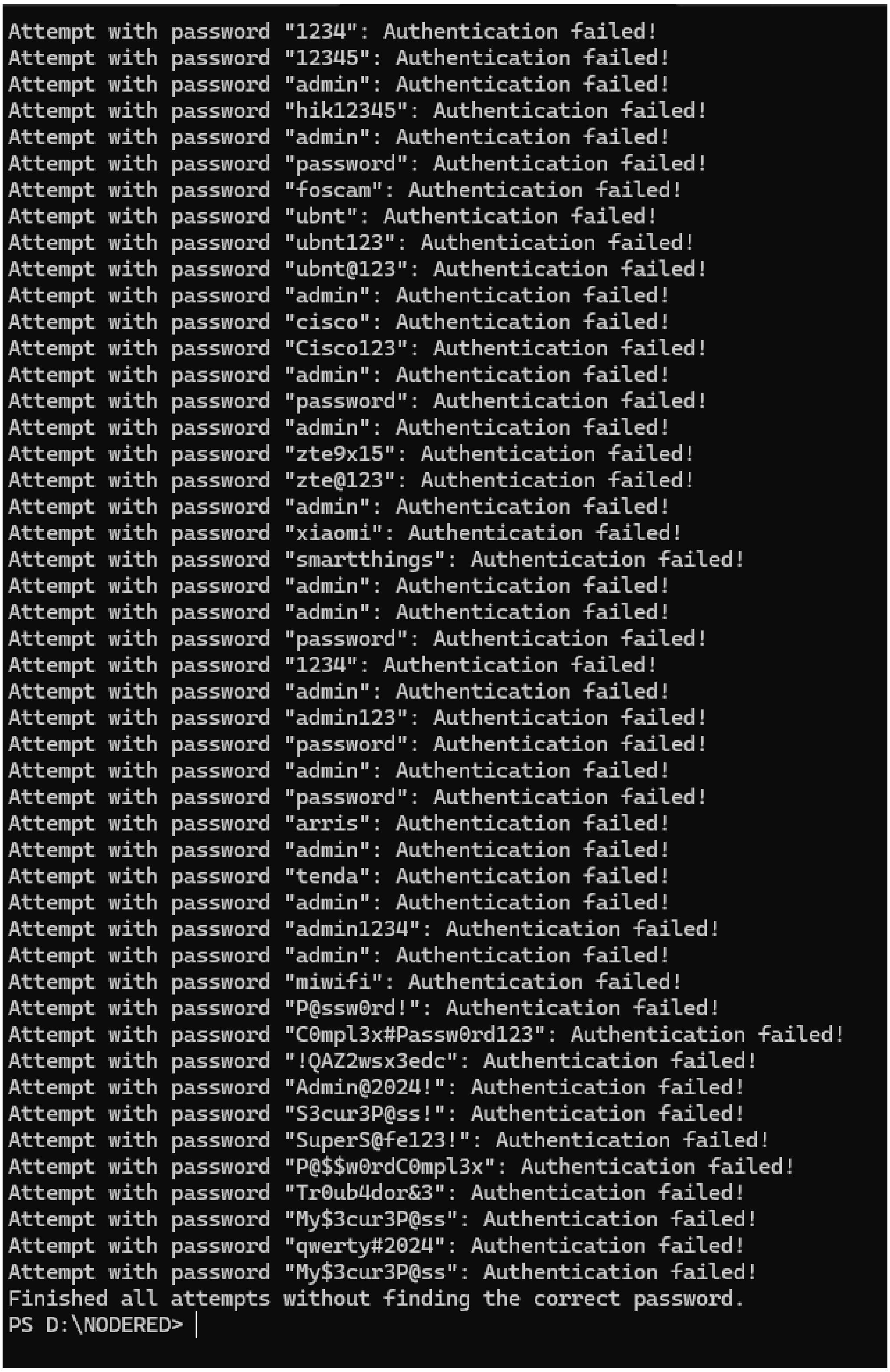
## 6.2 Experiment / Case Study 2: Exploiting the Vulnerable Smart Lock

This case study demonstrates the exploitation of the vulnerable smart lock which uses the default password as a method of authentication. The exploit is developed and saved in the notepad file as a .js extension and then the file is executed with the command node ./pass.js. The code runs a brute force method to try and unlock the smart lock with the given wordlist. The script iterates over each password in the wordlist. For each password it sends a POST request to the smart lock API with the current password in the ‘data’ object. The script includes error handling with a try-catch block to manage any network issues or other errors that might occur during the brute-force attempts.



```
PS D:\NODERED> node pass1.js
Attempt with password "qwerty#2024": Lock is now unlocked
Success the lock is now unlocked with password: qwerty#2024
PS D:\NODERED>
```

Figure 6.2 : Successful attempt in brute-forcing the lock.



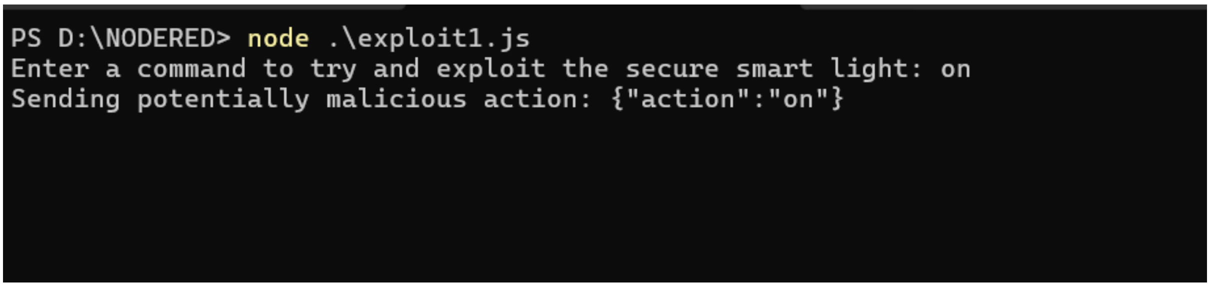
```
Attempt with password "1234": Authentication failed!
Attempt with password "12345": Authentication failed!
Attempt with password "admin": Authentication failed!
Attempt with password "hik12345": Authentication failed!
Attempt with password "admin": Authentication failed!
Attempt with password "password": Authentication failed!
Attempt with password "foscam": Authentication failed!
Attempt with password "ubnt": Authentication failed!
Attempt with password "ubnt123": Authentication failed!
Attempt with password "ubnt@123": Authentication failed!
Attempt with password "admin": Authentication failed!
Attempt with password "cisco": Authentication failed!
Attempt with password "Cisco123": Authentication failed!
Attempt with password "admin": Authentication failed!
Attempt with password "password": Authentication failed!
Attempt with password "admin": Authentication failed!
Attempt with password "zte9x15": Authentication failed!
Attempt with password "zte@123": Authentication failed!
Attempt with password "admin": Authentication failed!
Attempt with password "xiaomi": Authentication failed!
Attempt with password "smartthings": Authentication failed!
Attempt with password "admin": Authentication failed!
Attempt with password "admin": Authentication failed!
Attempt with password "password": Authentication failed!
Attempt with password "1234": Authentication failed!
Attempt with password "admin": Authentication failed!
Attempt with password "admin123": Authentication failed!
Attempt with password "password": Authentication failed!
Attempt with password "admin": Authentication failed!
Attempt with password "password": Authentication failed!
Attempt with password "arris": Authentication failed!
Attempt with password "admin": Authentication failed!
Attempt with password "tenda": Authentication failed!
Attempt with password "admin": Authentication failed!
Attempt with password "admin1234": Authentication failed!
Attempt with password "admin": Authentication failed!
Attempt with password "miwifi": Authentication failed!
Attempt with password "P@ssw0rd!": Authentication failed!
Attempt with password "C0mpl3x#Passw0rd123": Authentication failed!
Attempt with password "!QAZ2wsx3edc": Authentication failed!
Attempt with password "Admin@2024!": Authentication failed!
Attempt with password "S3cur3P@ss!": Authentication failed!
Attempt with password "SuperS@fe123!": Authentication failed!
Attempt with password "P@$$w0rdC0mpl3x": Authentication failed!
Attempt with password "Tr0ub4dor&3": Authentication failed!
Attempt with password "My$3cur3P@ss": Authentication failed!
Attempt with password "qwerty#2024": Authentication failed!
Attempt with password "My$3cur3P@ss": Authentication failed!
Finished all attempts without finding the correct password.
PS D:\NODERED> |
```

Figure 6.3: Failed attempts to Brute force the lock.

The above figures show the attempt to brute force the smart lock. In the first attempt when the default or weak password is used the lock can be unlocked remotely with the script. But in the second instance where the password is much stronger it is not able to find the password even with the provided extensive wordlist. This highlights the importance of using strong passwords as methods of authentication.

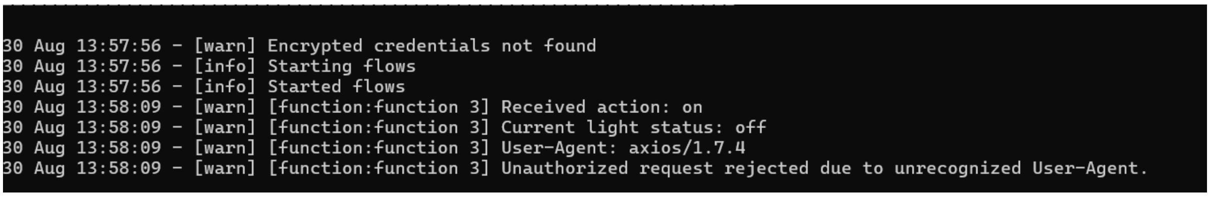
## 6.3 Experiment / Case Study 3: Trying to exploit the Secure Smart Light

This case study shows the method to try and exploit the secure smart light with the use of scripts same like how the vulnerable smart light was exploited. This light is fully patched with the latest firmware.



```
PS D:\NODERED> node .\exploit1.js
Enter a command to try and exploit the secure smart light: on
Sending potentially malicious action: {"action":"on"}
```

Figure 6.4: Trying to exploit the Secure Smart light



```
30 Aug 13:57:56 - [warn] Encrypted credentials not found
30 Aug 13:57:56 - [info] Starting flows
30 Aug 13:57:56 - [info] Started flows
30 Aug 13:58:09 - [warn] [function:function 3] Received action: on
30 Aug 13:58:09 - [warn] [function:function 3] Current light status: off
30 Aug 13:58:09 - [warn] [function:function 3] User-Agent: axios/1.7.4
30 Aug 13:58:09 - [warn] [function:function 3] Unauthorized request rejected due to unrecognized User-Agent.
```

Figure 6.5 : Failing to gain remote access as the request is unauthorised.

The above figures show that the attacker fails to gain the remote access to the secure smart light as it accepts commands only from a recognised user agent.

## 6.4 Discussion

This section provides a comprehensive analysis of the findings from the conducted experiments and case studies on simulating IOT devices with vulnerabilities and trying to exploit them. These experiments focussed on simulating vulnerable and secure smart devices and trying to exploit them by the use of scripts.

### Detailed Analysis of the Findings

Experiment / Case Study 1: Exploiting the Vulnerable Smart Light

- The vulnerable smart light was successfully exploited by the use of a script which exploits the buffer overflow vulnerability. On running the script, it prompts the attacker to enter a command to control the smart light remotely. On entering the command, the script constructs an oversized payload by sending additional data beyond the predefined buffer size to simulate the effect of the buffer overflow. This oversized payload is sent to the Node-Red server via a HTTP POST request using the axios library. The device since it is configured with the buffer overflow vulnerability processes the request and allows the attacker remote access.

Experiment/ Case Study 2: Exploiting the Vulnerable Smart Lock

- The vulnerable smart lock was also successfully exploited by the use of a script which performs a brute-force attack on the smart lock endpoint by testing for passwords to unlock the system. It uses the axios library to send POST requests to the smart lock endpoint. The script reads passwords from a file wordlist.txt which has all the default passwords. If a response indicating success is received the lock is unlocked and the password with which it is unlocked is displayed.
- In the next experiment, the password used for the smart lock was much more secure and strong. As the lock made use of a strong password the script was unable to brute-force the lock and unlock the smart lock. This shows the importance of the use of strong passwords as a method of authentication.

Experiment/ Case Study 3: Exploiting the Secure Smart Light

- The secure smart light is also tried to exploit with a script which exploits the buffer overflow vulnerability. The script is also modified to try and exploit the smart light with the higher severity vulnerabilities. The script fails to execute the remote command as the light is fully patched and the firmware is updated.

From the above case studies, it is evident that the vulnerable devices are easily exploited with the use of scripts and the secure smart devices are much harder to exploit with the updated firmware and patches adding an extra layer of security. This highlights the importance of updating the patches of the smart devices and keeping the firmware up to date. The smart lock is also vulnerable when it makes use of a default or weak password. On the use of a stronger password, it is difficult to brute-force the lock even with the use of a comprehensive wordlist.

**Evaluation against existing models**

There are studies which have made use of node-red to simulate attacks like DDoS and brute-forcing. However, these studies do not make use of any real-world vulnerabilities. The method followed in this research to simulate a virtual environment, configure the device with real world vulnerabilities and then exploit them gives a very realistic result of the risks involved by the use of unpatched systems.

1. This research shows the use of Node-RED to simulate the industrial control systems. They used Node-RED to handle programmable logic controllers' tasks and update and host a Modbus TCP/IP server. (Steven Day, 2021)
2. This paper shows Detection and Prevention of DDoS Attacks on the IoT. They make use of the trained CNN model for real experiment verification. (Shu-Hung Lee, 2022)

Comparing to these research papers the method followed in this research provides a novelty on simulating and testing the vulnerable IoT devices.

**Mitigation Strategies**

From the above case studies, it is very evident that not updating the firmware of the device regularly and not using strong passwords as methods of authentication can prove to be very dangerous. These are some of the mitigation strategies proposed:

1. **Intrusion Detection Systems:** Implement lightweight, multilayer IDS to monitor and detect unusual activity within IoT networks. This can help prevent attacks like denial of service and phishing.
2. **Hardware-Based Security:** Use hardware-based security mechanisms such as Secure Return Address Stack (SRAS) and Field Programmable Gate Arrays (FPGA) to protect devices from booting attacks and code injection
3. **Secure communication protocols:** All the data transmitted through the IoT networks should ensure that it uses secure communication protocols like TLS and SSL. These protocols provide encryption and protect against man-in-the-middle attacks.
4. **Authentication Mechanisms:** Strong authentication mechanisms like MFA needs to be used. This reduces the risk of unauthorized access and device hijacking.
5. **Secure Firmware updates:** Ensure that all the devices are updated regularly.

**Existing lightweight methods successfully implemented in IoT devices**

1. **Lightweight Authenticated encryption using LED and PHOTON:** This study proposes an architecture which combines the LED block cipher with the PHOTON hash function. This approach optimizes the performance of the IoT device by reducing power consumption which makes it ideal for IoT devices. The system was implemented on Cyclone FPGA devices and achieved a 46% reduction in power usage, showcasing its efficiency for IoT environments. (Mohammed Al-Shatri, 2023)
2. **Comparison of Lightweight Cryptographic Algorithms (AES-128, SPECK, and ASCON):** This study compares the performance of the two different algorithms when applied to IoT devices. The comparison metrics are execution time, memory usage and the overall efficiency on resource limited IoT devices. The SPECK algorithm was better than the AES-128 when compared using the above metrics. This comparison helps to identify which algorithm is best suited for IoT devices. (Indu Radhakrishnan, 2024)
3. **Lightweight protocols in smart cities:** This paper describes the implementation of lightweight protocols in smart cities. It highlights protocols such as CLEFIA and PRESENT lightweight block ciphers, which have been successful in maintaining robust security while consuming minimal resources, making them well-suited for IoT applications in urban infrastructure. (Mahesh Joshi, 2021)

# 7 Conclusion and Future Work

The main objective of this research was to answer the question: How can Node-RED be utilized to simulate and analyse the security vulnerabilities in smart home IoT devices, and what are the most effective mitigation strategies to prevent potential cyber-attacks? The objectives were to identify the consequences of these vulnerabilities and provide mitigation strategies.

To achieve this a virtual smart home environment was created. In the environment both vulnerable and secure devices were hosted. The environment was configured in such a way that it could receive commands from the recognised user-agent app to mimic the real-world scenario of smart devices being controlled via an app. A UI page was also created which showed the status of each device. After the configuration of the devices an attempt was made to exploit the vulnerabilities using scripts.

The vulnerabilities were successfully exploited which resulted in the attacker having remote access to all the configured vulnerable devices. The attacker is able to control the devices using remote commands and this does not affect the normal behaviour of the device, i.e. the user is not able to find out if the smart device has been compromised. The similar exploit script was tried on the configured secure device as well but was not successful. This showed the difference between the vulnerable and secure smart devices.

This research contributes to the field of cybersecurity by providing a practical example on the risks of unpatched IoT devices and the use of poor authentication methods to control the devices. The mitigation strategies are also provided hoping that the future of IoT smart homes is a much better and safer one.

**Key Findings**

- The vulnerable devices were exploited successfully with the help of scripts.
- The smart devices were not exploited even with the use of sophisticated scripts.
- The use of strong passwords made it difficult to brute-force.

**Future Work**

While this study demonstrated the effectiveness of Node-RED in simulating and evaluating security vulnerabilities in smart home devices, there remain several areas for further research. Future work could include exploring more complex and different types of vulnerabilities across different devices. In my research I have made use of just two devices to test the vulnerabilities. This can be extended and more devices with different vulnerabilities can be included. The simulation environment can be expanded to include more sophisticated attack vectors, such as ransomware or advanced persistent threats (APTs). This would provide a deeper understanding of the threats.

Further research can also be done on implementing lightweight security controls for the IoT devices with limited sources. This can be implemented by collaborating with manufacturers to significantly reduce the risks of the IoT environment.

# 8 References

Indu Radhakrishnan, S. J. (2024). Efficiency and Security Evaluation of Lightweight Cryptographic Algorithms for Resource-Constrained IoT Devices. *sensors*.

Mahesh Joshi, B. M. (2021). Lightweight Security Protocols for Securing IoT Devices in Smart Cities. *Springer Link*, 108.

Media, T. (2024, May 14). *Internet of Things (IoT) Cyberattacks in 2024 – Causes and Implications*. Retrieved from ThriveDX: https://thrivedx.com/resources/blog/internet-of-things-iot-cyberattacks-in-2024-causes-and-implications

Mohammed Al-Shatri, F. A. (2023). IoT Edge Device Security: An Efficient Lightweight Authenticated Encryption Scheme Based on LED and PHOTON. *applied sciences*.

Riddles, J. (2024, July 23). *SonicWall 2024 Mid-Year Cyber Threat Report: IoT Madness, PowerShell Problems and More*. Retrieved from SONICWALL: https://blog.sonicwall.com/en-us/2024/07/sonicwall-2024-mid-year-cyber-threat-report-iot-madness-powershell-problems-and-more/

Shu-Hung Lee, Y.-L. S.-H.-H.-F. (2022). Detection and Prevention of DDoS Attacks on the IoT. *applied sciences*.

Steven Day, W. ". (2021). *Simulating Industrial Control Systems Using Node-RED and Unreal Engine 4.* National Cyber Summit Research Track 2021.

Daugherty, P. and Wilson, H.J. (2022). *Radically Human: How New Technology Is Transforming Business and Shaping Our Future*. [online] *Google Books*. Harvard Business Press. Available at: https://books.google.com/books?hl=en&lr=&id=4bAsEAAAQBAJ&oi=fnd&pg=PT8&dq=+The+potential+in+the+application+of+the+Internet+of+Things+seems+boundless.+However [Accessed 19 Jun. 2024].

Ramakrishnan, R. and Gaur, L. (2019). Internet of Things. doi: https://doi.org/10.1201/9780429486593.

Swamy, S.N. and Kota, S.R. (2020). An Empirical Study on System Level Aspects of Internet of Things (IoT). *IEEE Access*, 8, pp.188082–188134. doi: https://doi.org/10.1109/access.2020.3029847.

Novo, B. (2021). *Study and implementation of security mechanisms in resource-constrained IoT devices*. [online] Utad.pt. Available at: https://repositorio.utad.pt/items/bab51c02-39ef-4570-b0b6-6972353186bb [Accessed 19 Jun. 2024].

Anand, P., Singh, Y., Selwal, A., Alazab, M., Tanwar, S. and Kumar, N., 2020. IoT vulnerability assessment for sustainable computing: threats, current solutions, and open challenges. *IEEE Access*, *8*, pp.168825-168853.https://ieeexplore.ieee.org/abstract/document/9189773/

Ferrara, P., Mandal, A.K., Cortesi, A. and Spoto, F., 2021. Static analysis for discovering IoT vulnerabilities. *International Journal on Software Tools for Technology Transfer*, *23*, pp.71-88.https://link.springer.com/article/10.1007/s10009-020-00592-x

Ghazal, T.M., Afifi, M.A.M. and Kalra, D., 2020. Security vulnerabilities, attacks, threats and the proposed countermeasures for the Internet of Things applications. *Solid State Technology*, *63*(1s).https://research.skylineuniversity.ac.ae/id/eprint/63/

Rytel, M., Felkner, A. and Janiszewski, M., 2020. Towards a safer Internet of things—a survey of IoT vulnerability data sources. *Sensors*, *20*(21), p.5969.https://www.mdpi.com/1424-8220/20/21/5969

Figueroa-Lorenzo, S., Añorga, J. and Arrizabalaga, S., 2020. A survey of IIoT protocols: A measure of vulnerability risk analysis based on CVSS. *ACM Computing Surveys (CSUR)*, *53*(2), pp.1-53.https://dl.acm.org/doi/abs/10.1145/3381038

Statista.com (2023): IoT security issues experienced by users: https://www.statista.com/statistics/1377569/worldwide-annual-internet-of-things-attacks/

Hammi, B., Zeadally, S., Khatoun, R. and Nebhen, J., 2022. Survey on smart homes: Vulnerabilities, risks, and countermeasures. *Computers & Security*, *117*, p.102677.https://www.sciencedirect.com/science/article/pii/S016740482200075X