A Security Vulnerability Scanner Prototype for IoT Devices on Wi-Fi and Bluetooth Networks

By

Njeru, Edwin Muchiri

068423

Submitted in partial fulfillment of the requirements for the award of a Degree of Master of Science in Information Systems Security

School of Computing and Engineering Sciences

Strathmore University

Nairobi, Kenya

June, 2025

This dissertation is available for Library use on the understanding that it is copyright material and that no quotation from the dissertation may be published without proper acknowledgement.

Declaration and Approval

Declaration

I declare that this work has not been previously submitted and approved for the award of a degree by this or any other University. To the best of my knowledge and belief, the

dissertation contains no material previously published or written by another person except where due reference is made in the dissertation itself.

© No part of this dissertation may be reproduced without the permission of the author and Strathmore University

Edwin Muchiri Njeru

21st May 2025

Approval

The dissertation of Njeru, Edwin Muchiri was reviewed and approved by the following:

Dr. Vitalis Ozianyi,

Senior Lecturer, School of Computing Engineering Sciences,

Strathmore University.

Dr. Julius Butime,

Dean, School of Computing Engineering Sciences,

Strathmore University.

Prof. Bernard Shibwabo,

Director of Graduate Studies,

Strathmore University.

Abstract

The Internet of Things (IoT) is significantly changing everyday experiences by connecting various devices that can communicate and exchange information. This interconnectivity improves efficiency and convenience across homes, businesses, and numerous industries. However, with the rapid expansion of IoT devices, security vulnerabilities have emerged as a significant concern. Many IoT devices are susceptible to attacks due to weak passwords, unprotected open ports, and unencrypted data transfers. Current vulnerability scanners for IoT devices, such as the Mirai Scanner and Firmalyzer, address some of these risks but exhibit notable limitations: they are often costly, require additional hardware, and lack support for Bluetooth-connected devices.

This dissertation aims to develop a comprehensive and user-friendly vulnerability scanner prototype for IoT devices that address these gaps. By adopting the Rapid Application Development (RAD) methodology, the study iteratively focused on the design, development, and refinement of the tool that enables security administrators to scan and detect vulnerabilities across multiple connection types. This consisted of key components, including a network scanning engine, a vulnerability assessment module and a report generation system. The network scanning engine identified connected IoT devices and analyzed communication protocols including Wi-Fi and Bluetooth. The vulnerability assessment module cross-references the identified devices against a database of security weaknesses, detecting common threats such as open ports and default credentials. The report generation module provided downloadable reports, providing recommendations to remedy the identified vulnerability.

To validate the effectiveness of the scanner, the study employed a series of controlled experiments. These were to assess the scanner's accuracy, ability to detect a range of vulnerabilities, and ease of use in comparison to existing solutions. The results from the validation phase were to demonstrate the tool's potential to improve IoT security by providing an accessible, cost-effective solution for proactive vulnerability management. The scanner successfully identified vulnerabilities across the IoT devices connected via Wi-Fi and Bluetooth in the test environment, demonstrating detection accuracy, risk assessment, and security reporting.

Keywords: Security, Internet of Things, Rapid Application Development

Table of Contents

Declaration and Approval ..... **ii**

Abstract ..... **iii**

List of Figures ..... **viii**

Dedication ..... **xvi**

Chapter 1 : INTRODUCTION **1**

1.1. Background of Study **1**

1.2. Problem Statement **3**

1.3. Study Objectives **4**

1.4. Study Questions **4**

1.5. Justification of the Study **4**

1.6. Scope and Limitations **5**

Chapter 2 : Literature Review **7**

2.1 Introduction **7**

2.2 Trends of Network Technologies **7**

2.2.1 Traditional Networks 7

2.2.2 Modern Networks 8

2.3 IoT Ecosystem **9**

2.3.1 Sensors and Devices 10

2.3.2 Connectivity 10

2.3.3 Data Processing 10

2.3.4 User Interface 11

2.4 IoT Vulnerabilities **11**

2.4.1 Weak and Guessable Passwords 11

2.4.2 Insecure Network Services 11
2.4.3 Insecure Data Transfer and Storage 11
2.4.4 Unprotected Patches and Upgrades 12
2.5 Vulnerability Management 12
2.5.1 Vulnerability Management Preparation 13
2.5.2 Vulnerability Scan 14
2.5.3 Define Remediation actions 15
2.5.4 Implement Remediation Actions 16
2.5.5 Rescan 17
2.6 General Security Approaches for IoT Devices 18
2.6.1 Change the Default Credentials 18
2.6.2 Secure Your Network 18
2.6.3 Update the Firmware 18
2.6.4 Removing Old Devices 19
2.7 Existing Vulnerability Scanner Solutions for IoT 19
2.7.1 Dojo Intelligent IoT Vulnerability Scanner App 19
2.7.2 Mirai Scanner for IoT 20
2.7.3 Retina IoT Vulnerability Scanner 22
2.7.4 Firmalyzer 23
2.8 Gaps in the Existing Solutions 25
2.9 Conceptual Framework 26
2.9.1 Key Components of the Proposed Vulnerability Scanner 28
Chapter 3 : Methodology 29
3.1 Introduction 29
3.2 Study Design 29

3.2.1 Identification of IoT Vulnerabilities 30
3.2.2 Review of Existing Tools 30
3.2.3 Design and Development 30
3.2.4 System Validation 33
3.2.5 Study Quality 33
3.2.6 Data Analysis 34
3.2.7 Ethical Considerations 34
Chapter 4 : SYSTEM DESIGN AND ARCHITECTURE 36
4.1 Introduction 36
4.2 Requirements Planning 36
4.2.1 Functional Requirements 36
4.2.2 Non-Functional Requirements 37
4.3 System Architecture 37
4.3.1 Input 38
4.3.2 Processes 38
4.3.3 Outputs 39
4.4 System Design Tools 40
4.4.1 Use Case 40
4.4.2 Flow Charts 41
4.4.3 Class Diagram 42
4.4.4 Database Schema 43
4.4.5 Sequence Diagram 44
4.5 Network Design 45
4.6 Security Design 46
4.7 Wireframe 46

4.8 Conclusion 47

Chapter 5 : SYSTEM IMPLEMENTATION, TESTING AND VALIDATION 48

5.1 Introduction 48

5.2 Implementation 48

5.2.1 Hardware Requirements 50

5.2.2 Software Requirements 50

5.2.3 Network Requirements 50

5.2.4 Populating Vulnerability Database 52

5.2.5 Generating Attacks 53

5.3 System Modules/Key Functions 53

5.4 Testing 54

5.4.1 Functional Testing 55

5.4.2 Vulnerability Detection and Accuracy Rate 63

5.5 Prototype Validation 64

5.6 User Feedback 66

5.7 Prototype Verification 68

Chapter 6 : CONCLUSIONS, RECOMMENDATIONS AND FUTURE WORKS 69

6.1 Introduction 69

6.2 Discussion 69

6.3 Conclusion 70

6.4 Recommendations 71

6.5 Future Works 72

REFERENCES 73

APPENDICES 80

List of Figures

Figure 2.1: Vulnerability Management - Preparation Phase 13

Figure 2.2: Vulnerability Management - Initial Scan 14

Figure 2.3: Vulnerability Management - Remediation Phase 15

Figure 2.4: Vulnerability Management - Implementing Remediation Actions 16

Figure 2.5: Vulnerability Management - Rescan Phase 17

Figure 2.6: Dojo Vulnerability Scanner App 19

Figure 2.7: Mirai Scanner Summary 20

Figure 2.8: Unsuccessful Scan response from Mirai Scanner 21

Figure 2.9: Successful scan showing a possible vulnerability to a Miriai botnet infection 21

Figure 2.10: Mirai Vulnerability Scanner 22

Figure 2.11: Retina IoT Vulnerability Scanner 23

Figure 2.12: Firmalyzer Dashboard 24

Figure 2.13: Firmalyzer Scanner 25

Figure 2.14: Conceptual Framework 27

Figure 3.1: Rapid Application Development (RAD) Framework 31

Figure 4.1: System Architecture 38

Figure 4.2: Activity Flow for the IoT Security Vulnerability Scanner 39

Figure 4.3: Use Case Diagram of Connecting to an existing Network and a scan and Analysis of an IoT Device 41

Figure 4.4: Connecting to an Existing Network and a Scan and Analysis Flow Chart 42

Figure 4.5: Class Diagram of the Tool 43

Figure 4.6: Database Schema 44

Figure 4.7: Connecting to an Existing Network and a Scan and Analysis Sequence Diagram 45

Figure 5.1: Devices used in the Design of the solution 49
Figure 5.2: Command with possible arguments towards establishing a connection 51
Figure 5.3: Python Engine to Cracking the WI-FI network..... 52
Figure 5.4: Successful connection to WI-FI Network through command line 55
Figure 5.5: Unsuccessful attempt to connect to a WI-FI Network 56
Figure 5.6: Use of a dictionary to gain access to a WI-FI Network 56
Figure 5.7: Successful connection to a Bluetooth Connection 57
Figure 5.8 : Unsuccessful attempt to connect to a Bluetooth Connection 57
Figure 5.9: Attempt to use a dictionary to gain access to a Bluetooth Connection 57
Figure 5.10: A scan that found no open ports in a WI-FI Connection..... 58
Figure 5.11: Successful scan to an open port on a WI-FI connection 59
Figure 5.12: Captured data of a Temperature IoT Sensor 59
Figure 5.13: Results of a scan conducted on a Bluetooth connection 60
Figure 5.14: Report for a successful connection to a WI-FI Network..... 60
Figure 5.15: Report on a successful attempt to a Bluetooth Connection..... 61
Figure 5.16: Report generated for a Successful scan with an open port..... 61
Figure 5.17: Report shows No encryption on data captured from Temperature Sensor..... 62
Figure 5.18: Report of Scan due to an open port in the Bluetooth Connection..... 62
Figure 5.19: Report generated due to unencrypted data from the Barometer sensor 63
Figure 5.20: Results of scan conducted by IoT Seeker scanner tool 65
Figure 5.21: Report generated by IVS Tool..... 66

List of Tables

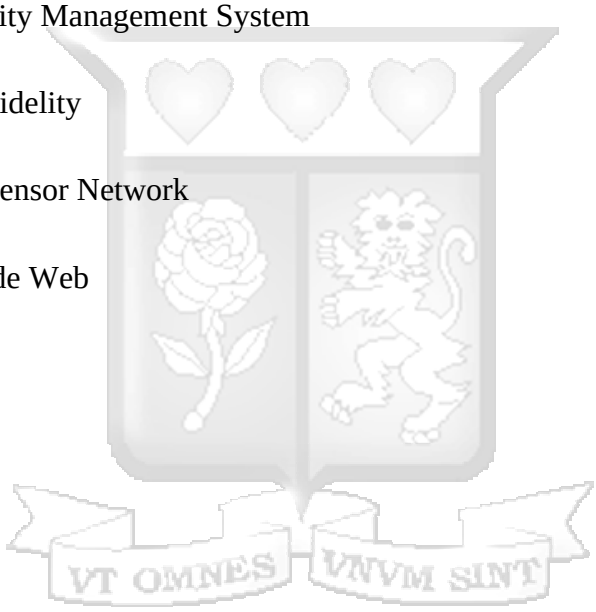
Table 2.1: Summary of Comparison of Existing Vulnerability Scanning Tools 25

Table 5.1: Summary of whether Functional Requirements were achieved 67

List of Abbreviations/Acronyms

API	Application Programming Interface
CLI	Command Line Interface
CDR	Computing Device Recognition
CVE	Common Vulnerabilities and Exposures
CVSS	Common Vulnerability Scoring System
DDoS	Distributed Denial of Service
DoS	Denial of Service
ERD	Entity Relationship Diagram
IdM	Identity Management
IoT	Internet of Things
IT	Information Technology
IVS	IoT Vulnerability Scanner
LAN	Local Area Network
LPWAN	Low Power Wide Area Networks
M2M	Machine to Machine
NVD	National Vulnerability Database
PDF	Portable Document Format
PII	Personally Identifiable Information

RAD	Rapid Application Development
RBAC	Role-Based Access Control
SDLC	Software Development Life Cycle
SSO	Single Sign-On
TLS	Transport Layer Security
UCBerkeley	University of California, Berkeley
VMS	Vulnerability Management System
Wi-Fi	Wireless Fidelity
WSN	Wireless Sensor Network
WWW	World Wide Web



Definition of Terms

IoT (Internet of Things) - It means the system of all the connected devices and the technology that helps them talk to each other and to the cloud (AWS, What is IoT (Internet of Things)?, 2025).

Vulnerability - A problem or weak spot in how an asset is built, used, or managed that someone could take advantage of (Puzder, 2023).

Vulnerability Scanner - Identifies computers and their features (like what operating systems they use, what programs are on them, and what ports are open). They identify exposed devices, open ports, weak/default credentials, and N-day vulnerabilities (Zhao B. , et al., 2022).

CVE (Common Vulnerabilities and Exposures) – This is a list of entries, and each entry includes a unique ID number, a description, and at least one public reference. This list covers cyber-security vulnerabilities that have been found in certain software and reported to **https://cve.mitre.org** (Dempsey, Eavy, Moore, & Takamura, 2020).

Bluetooth Security Risk - A security weakness in Bluetooth communication that allows attackers to exploit devices through unauthorized access, interception, or manipulation of data (Stouffer, 2022).

Wi-Fi Security Risk - Potential threats associated with Wi-Fi networks, including weak encryption, open ports, or misconfigurations that allow unauthorized access (Meil, 2025).

Brute Force Attack – This is a way to break into accounts by trying out lots of different possible passwords and codes. It's a straightforward and effective method for getting into both personal accounts and the systems of organizations without permission (Fortinet, What Is A Brute Force Attack?, 2025).

Port Scanning – Process that involves finding weak spots in a computer network. This technique helps them identify open ports and determine whether data is being sent or received through those ports (Fortinet, What Is A Port Scan? How To Prevent Port Scan Attacks?, 2025).

Mitigation Strategies - Recommended actions or security measures taken to reduce or eliminate the impact of identified vulnerabilities in a system (Gargan, 2024).

Acknowledgement

In recognition of the long journey taken in pursuing my Post Graduate Dissertation, I extend my heartfelt gratitude to everyone who contributed to its realization.

First and foremost, I sincerely thank my supervisors, Dr. Vitalis G. Ozianyi, for their exceptional guidance, support and patience throughout this journey. Their valuable insights and encouragement, which was constant encouragement, played a crucial role in the development of this work.

I am truly thankful to my family and friends for their unwavering support, motivation, and understanding throughout this process. Their belief in me kept me going even in the most challenging moments.

Lastly, I appreciate all the individuals, institutions, and resources that contributed to my study. This dissertation would not have been possible without their input and collaboration.

Dedication

I first and foremost dedicate this dissertation to the almighty God for the wisdom and grace towards this this, my family, whose unwavering support, encouragement, and belief in me have been my greatest source of strength.

To my supervisor, Dr. Vitalis G. Ozianyi, for his guidance and support, mentors and friends, who have inspired and motivated me throughout this journey.

Chapter 1 : INTRODUCTION

1.1. Background of Study

The Internet is an exceptional interconnected system that enables devices to communicate far and wide using set protocols connected through heterogeneous networks that include businesses, government institutions and academic institutions. While in the beginning the internet presented only websites and email communication, different forms of internet application can be seen everywhere providing various applications and services to meet the user needs (Berners-Lee, Hendler, & Lassila, 2023). In contemporary organizations, residences, governmental bodies, and educational institutions, there appears to be a notable rise in the implementation of the Internet of Things (IoT). This technology holds significant potential for enhancing service delivery across these sectors. Technically speaking, the IoT presence provides a large number of globally connected objects that include devices, sensors or actuators that provide different services.

IoT devices depend on cost-effective wireless communication systems and sensors that allow them to communicate with one another and send important data to a central system (Anik, Gao, Meng, Agee, & McCoy, 2022). Information and communications technologies (ICT) have become a big part of society that has changed how work, learning, interactions, and even the sense of individual identity (Bentley et al, 2024).

Unfortunately, a large number of IoT Devices are not built with robust security measures as part of the process, which leads to numerous vulnerabilities within IoT networks. The lack of robust security design in IoT systems leads to vulnerabilities in confidentiality, authentication, data integrity, access control, and data secrecy. This necessitates the integration of multiple security measures to protect against these issues effectively (Wakili & Bakkali, 2025). Every single day, IoT devices are the points of attack and intrusion by malicious users. A recent review reveals that 70% of IoT devices are highly vulnerable to attacks, highlighting the urgent need for robust mechanisms to securely protect these internet-connected devices from hackers and intruders (Vardakis et al, 2024).

Existing vulnerability scanners, like the Dojo Intelligent IoT Vulnerability Scanner App, Mirai Scanner, Firmalyzer, and Retina IoT Scanner, aim to identify and address security gaps that include

weak passwords, open ports, and unencrypted data transfer. Additionally, Denial of Service (DoS) attacks are common, where attackers flood IoT devices or IoT networks with traffic, disrupting processes (Ali & Awad, 2018); (Gelgi et al, 2024). However, most existing vulnerability scanners for IoT devices face significant limitations that hinder their effectiveness in real-world environments. Many of these tools are commercial solutions with high costs, which limit their accessibility, particularly for smaller organizations and individual users. In addition, they are often built to scan only specific types of networks typically supporting either Wi-Fi or LAN, but not both. This narrow compatibility, combined with their expense, restricts their usability in diverse IoT environments and results in some vulnerabilities remaining undetected. There is, therefore, a critical need for a more affordable, flexible, and comprehensive scanning solution to address these shortcomings and enhance IoT security on a broader scale.

The urgency of this need is underscored by IoT botnets, like Mirai and its newer versions, that have been a big problem. They take advantage of devices that still use the default usernames and passwords, infecting hundreds of thousands. This leads to huge DDoS attacks, which can disrupt online services (Gelgi et al, 2024). The persistence of such threats has only intensified with the continued expansion of the IoT paradigm. Researchers are looking into ways to monitor IoT devices by tracking their energy use. They point out that by keeping an eye on how much power these devices consume; we can spot unusual activities or potential security threats. This method, called the "energy-based approach," helps in identifying when something odd is going on (Valentino & Dario, 2024). These examples illustrate how overlooked vulnerabilities in IoT networks can be exploited in unexpected and intrusive ways, reinforcing the need for accessible and adaptable security solutions.

A vulnerability management system (VMS) is like a security guard for connected devices. Its job is to find weak spots, or "vulnerabilities," in devices like smart home gadgets, computers, and IoT Devices (Microsoft, 2025). Once a vulnerability is found, it is intended to provide steps for one to follow to fix the problem.

Vulnerability scanners certainly help identify known flaws, but they have important shortcomings. For one thing, they only show a point-in-time snapshot of the network; because new vulnerabilities and exploits appear constantly, systems must be scanned repeatedly (or continuously) to stay up

to date. Moreover, automated scanners often generate many incorrect alerts. It was noted that such tools are “prone to generating false positives” (benign issues flagged as problems) and false negatives (real vulnerabilities missed), which can “overwhelm analysts with unnecessary alerts” and leave gaps in coverage (Bennouk, et al., 2024).

The purpose of this study was to develop a solution that identifies the common vulnerabilities present in communicating IoT devices, communicate this to an administrator as indicators of compromise to take the necessary steps to mitigate any possible attack and eventually reduce the attack surface.

1.2. Problem Statement

The use of IoT devices brings significant security challenges due to inherent vulnerabilities in the design and deployment of these devices. They often suffer from factors such as weak or default passwords, open ports, unencrypted data transfers and inadequate firmware updates. When exploited, these vulnerabilities present severe risks, as unauthorized individuals can gain entry to the devices, disrupt services, and compromise sensitive information. For example, weak passwords and open ports make IoT devices prime targets for botnets that execute attacks such as Distributed Denial of Service (DDoS). Such can render critical services unavailable to users.

Moreover, unencrypted data transfer means that attackers can intercept and manipulate data, leading to breaches in privacy and possible financial losses, especially if sensitive information like health data or financial transactions is compromised (Bakhshi, Ghita, & Kuzminykh, 2024).

The existing literature indicates that the current solutions are successful in detecting some vulnerabilities, however they exhibit some gaps. For instance, the Mirai Scanner only scans for vulnerabilities related to the Mirai botnet that exploit default credentials (Gelgi at al, 2024). Other solutions, such as Dojo IoT and Firmalyzer Vulnerability Scanners, are often costly and vary in cost based on their specific applications (Mukhtar et al, 2023). These solutions have notable limitations: they typically only support limited network types, either Wireless or LAN networks restricting their applicability across diverse IoT environments (Zhao B. , et al., 2023). This can result in incomplete vulnerability assessments, as they may not comprehensively cover all device types or network configurations.

This study seeks to address these gaps by developing a solution that supports Wi-Fi and Bluetooth networks, is easy and free to use and importantly requires no extra hardware to scan existing IoT Devices in a network.

1.3. Study Objectives

The specific objectives of the study are:

i. To enhance IoT security across multiple connection types by identifying key vulnerabilities and common attack vectors.

ii. To design and develop a prototype IoT vulnerability scanning tool tailored to address the identified security gaps.

iii. To test and evaluate the performance and accuracy of the prototype IoT vulnerability scanning tool under varying conditions.

iv. To validate the effectiveness of the proposed vulnerability scanning tool through real-world simulations and comparative analysis with existing solutions.

1.4. Study Questions

The study seeks to answer the following study questions:

i. What are the typical vulnerabilities found in IoT devices?

ii. What are the advantages and limitations of current vulnerability scanning methods and tools for IoT devices?

iii. How should the tool be designed, developed, and tested effectively?

iv. How can we assess the tool's effectiveness?

1.5. Justification of the Study

One of the most notable IoT security incidents was in 2016, where the Mirai botnet attack used thousands of IoT devices to launch a large-scale DDoS attack. This demonstrated the potential for

IoT vulnerabilities to affect the broader internet ecosystem, causing outages for services and data exposure (Yu, Zhuge, Cao, Shi, & Jiang, 2020).

Modern security strategies emphasize proactive monitoring and management tools that enable administrators to detect and remediate threats before they cause harm, thereby reducing downtime and supporting continuous operation. At the same time, all security efforts are grounded on the well-known CIA triad. Enterprise security "lies in the Confidentiality, Integrity, and Availability (CIA) triad," which together guide policies for protecting data from unauthorized access, alteration, or loss (Owobu, et al., 2024). Around these principles, administrators can more effectively implement the measures needed to preserve data confidentiality and availability while guarding integrity.

The need for such measures is highlighted by the explosive growth of IoT. Tens of billions of IoT devices are now deployed worldwide as both a boon and a source of privacy and security challenges (Tawalbeh et al, 2020). As IoT networks expand, so too do the potential attack surfaces and vectors available to malicious actors who increasingly employ sophisticated attack tools and extortion techniques to exploit vulnerable systems (Allianz Global Corporate & Specialty, 2022). Because the full potential of IoT depends on trust in these systems, researchers warn that connected devices must be safeguarded by robust security frameworks. In sum, ensuring the CIA properties through proactive platforms and up-to-date defenses is critical to support administrators and fully realize the benefits of IoT.

1.6. Scope and Limitations

To demonstrate the achievable objectives of the study, the scope is limited to examining protocol-based vulnerabilities within IoT devices, specifically those tied to individual ports and protocols that could be exploited to compromise security. The study focuses on identifying vulnerabilities through a set of connected IoT devices, including weak passwords, open ports, and unencrypted data. These vulnerabilities will be determined by scanning an existing IoT network, identifying all connected devices, and generating a report that outlines the discovered vulnerabilities along with recommended remediation steps.

Some of the specific processes that the solution is not able to accomplish are giving or providing instant alerts upon a possible exploit that could have occurred, to guarantee the immediate response to the exploit.

Chapter 2 : Literature Review

2.1 Introduction

This chapter reviews past literature including existing studies, models, tools, and methodologies related to IoT security and vulnerability scanning. Given the rapid proliferation of IoT devices across industries, from smart homes to industrial IoT systems, the need for effective security measures has become paramount. The chapter explores common IoT Vulnerabilities, existing vulnerability scanners and their limitations while identifying gaps that necessitate the development of a vulnerability scanning tool.

2.2 Trends of Network Technologies

2.2.1 Traditional Networks

Traditional networking involves fixed-function network devices, such as switches and routers, each designed to perform specific tasks that collectively support the network. When these functions are implemented in hardware, network performance is typically enhanced, resulting in higher speeds (IBMServices, 2019). Traditional networks often face challenges with flexibility. They expose a few Application Programming Interfaces (APIs) for provisioning, and most switching hardware and software are proprietary to the vendor. While traditional networks function well with proprietary provisioning software, they cannot be modified when necessary. Key characteristics of traditional networking include network functions are primarily implemented using dedicated devices, such as switches, routers, and controllers and most of its functionality is embedded in specialized hardware.

A major drawback of this hardware-centric approach is its inherent limitations, restricting adaptability and scalability. The limitations to this type of network may range from hardware being vendor specific, very costly, difficult to add features as this can only be added by a vendor and devices are function specific meaning they can only behave according to their set behavior. The worst of them all is that in a large setting with multiple devices, each must be configured individually that is very time consuming (Gupta, 2020).

2.2.2 Modern Networks

With age of Modern networking and provision of services, this seeks to eliminate the limitations brought about by traditional networks. This involved moving away from hardware and vendor specific solutions to software defined networks, Internet of Things Services and other cloud-based services.

2.2.2.1 Cloud Networking

This type of network provides on-demand capabilities, including networking and resource provisioning, hosted on a cloud platform through a third-party service provider. Resources such as virtual routers, firewalls, bandwidth management, and network management are available, with additional tools and functions accessible as needed. (VMware, 2020). Companies can either use such services to manage an in-house network or operate entirely in the cloud. Additionally, they offer an isolated and highly secure cloud environment for connecting virtual machines and applications. They also help balance inbound and outbound traffic to application or service endpoints, enhancing uptime and performance (IBM, 2020).

Cloud computing offers several key advantages. Firstly, it eliminates the capital costs associated with purchasing hardware and software, as well as the expenses of establishing and maintaining on-site data centers. Additionally, it features self-service portals, allowing users to access services on demand. This means that large quantities of computing resources can be provisioned in just minutes, streamlining the process and removing the need for tasks like hardware setup, software updates, and other time-intensive IT management activities. It is run on a worldwide network of secure datacenters that are regularly upgraded/updated to fast and efficient computing hardware at the biggest cloud computing services. This offers several benefits including reduced network latency for applications and greater economies of scale (Microsoft, Top benefits of cloud computing, 2020).

2.2.2.2 IoT Network

In today's world, there are various methods for data transition between IoT devices and mobile applications or web via an IoT platform. Each method comes with its own set of benefits and drawbacks, and there is no one-size-fits-all solution (IoTFactory, 2020).

These network options that can be employed for IoT Networks include;

i. Wired and Short-Range wireless networks
ii. Machine to Machine (M2M) – 2G, 3G, 4G and 5G networks
iii. Low Power Wide Area Networks (LPWAN)

Due to the fact that IoT deployments consist of thousands of devices connected within a series of networks, it is crucial to track, monitor, and manage connected devices to ensure they function properly and remain secure after deployment (AWS, 2019). It facilitates the remote configuration and management, provisioning and authentication of devices, monitoring and diagnostics and firmware/software updates and maintenance. IoT device management providers often offer advanced features that enable the smooth management of various IoT devices and sensors within a single platform. They utilize Smart Layer technology to support communication across different protocols and effectively manage all kinds of IoT devices (Anisenko, 2019).

2.3 IoT Ecosystem

The Internet of Things (IoT) refers to a global network of interlinked devices that transmit data through the internet, significantly transforming how people live and work. It is regarded as the third wave of the World Wide Web (WWW), following static web pages and social networking-based web. Thanks to advances in technology, just about any item we use daily can now connect to the internet. This is all driven by the need for better automation and efficiency (Chataut, Phoummalayvane, & Akl, 2023).

Various industries, such as healthcare, farming, smart cities and homes, and advanced manufacturing (often called Industry 4.0), are quickly embracing the Internet of Things (IoT). This means a lot more everyday objects and machines are now connected online. Overall, the trend

shows that the use of IoT is growing rapidly in many different areas (Chataut, Phoummalayvane, & Akl, 2023).A process flow is essential for creating a solid framework on which an IoT solution can be developed (Bharani, 2018). The IoT architecture typically consists of four key components:

2.3.1 Sensors and Devices

Sensors capture information from the external environment and convert it into data for analysis, while actuators take it a step further by interacting with the physical world. For instance, they can turn off lights or adjust room temperature as needed (Stokes, 2018). A thing in the context of IoT should be equipped with sensors and actuators thus giving the ability to emit, accept and process signals (Bharani, 2018).

2.3.2 Connectivity

This is a very important aspect in the IoT environment. IoT sensors, end devices, and computing components must be interconnected to interpret data and trigger actions. Without smooth connectivity, this process cannot occur (EduCBA, 2020).

Internet gateways link to the sensor network, collecting data transmitted via Wi-Fi or wired LANs, and handle additional processing (Stokes, 2018). To summarize, the data originating from the sensors is in analog format and must be aggregated and converted into digital streams before further processing (Bharani, 2018).

2.3.3 Data Processing

Data has significant power and can greatly impact any business. Analytics is employed to interpret the large volumes of analog data (Khan, 2020). For instance, this can comprise the determination of changes in weather temperatures to further indicate the likelihood of a certain weather event.

In modern IoT deployments, sensors and devices generate massive, continuous data streams that contain valuable information for business operations and environmental management (Merenda, Porcaro, & Iero, 2020). Advanced analytics and machine learning are applied to these high-volume datasets to extract meaningful patterns and predictions (Andriulo et al, 2024).

2.3.4 User Interface

Devices such as smartphones, tablets, laptops and desktop machines serve as the end components of an IoT ecosystem. This user interface serves as a visible and easily accessible component, allowing the IoT user to maintain control and can be used to set their preferences. User interaction increases the user-friendly this component is (Khan, 2020).

2.4 IoT Vulnerabilities

2.4.1 Weak and Guessable Passwords

The use of credentials that easily brute forced, publicly available, or hardcoded such as default passwords including backdoors in firmware or client software poses a risk of unauthorized access to deployed systems (OWASP, 2020). Through exploitation of this vulnerability, the impact from this kind of attack is usually denial of service and can also lead to compromise of device.

2.4.2 Insecure Network Services

The network services running on the device itself may be unneeded or insecure, especially those exposed to the internet that compromise the confidentiality, integrity/authenticity, or availability of information or allow unauthorized remote control (OWASP, 2020). Diagnostic and testing tools, along with services like debugging, are typically enabled on these devices. However, they may not have been thoroughly tested, making them prone to vulnerabilities and potentially containing exploitable source code.

2.4.3 Insecure Data Transfer and Storage

The absence of encryption or access control for sensitive data across the ecosystem, whether at rest, in transit, or during processing poses a significant security risk (OWASP, 2020). A few IoT vendors pay attention to secure storage, making sure data remains secure during transfer but many more IoT vendors often ignore this. (Paul, 2020).

2.4.4 Unprotected Patches and Upgrades

During upgrades and updates of patches, it is not guaranteed that this was to improve the security of a device but rather expose it more to spyware or malware. This adds more questions than answers when attempting to address security vulnerabilities.

2.5 Vulnerability Management

The Vulnerability Management process entails the identification, assessment, mitigation, and documentation of security vulnerabilities within systems and the software they operate. When integrated with other security strategies, it is essential for organizations to focus on potential threats and minimize their "attack surface" (Vulnerability Management and Scanning , 2020). In the global world, there are a vast number of vendors supplying IoT devices and the last thing on the minds is to properly and effectively seal all the security gaps that may be present before releasing the product to the public. To summarize this, focus is on functionality over security. This poses challenges because intentionally or unintentionally, IoT can directly collect sensitive personal information such as financial account numbers, geo-location, and health information and in terms of managing this personal information collected and ensuring that this information is safe and cannot be uncovered intentionally becomes a very hefty task.

Consumers are not aware that devices already in their homes can capture and process their information. As speakers pointed out, from personal health information on wearables to cameras on baby monitors, consumers have an expectation of privacy that is currently not met (Megas, Piccarreta, & O'Rourke, 2017). These security gaps are classified as vulnerabilities as they can be exploited to gain access to unauthorized information. According to the Symatec official blog (Park, 2015), "Some of the attacks conducted are for espionage or to just expose the lack of security in such networks," explains Shankar Somasundaram, Senior Director, Internet of Things, Symantec. "For example, a hacker hacked into the major city water treatment plant just to expose the lack of security and posted part of the plant's schematic on the Internet." A vulnerability management process is key towards minimizing an attack surface on an organization or home network.

2.5.1 Vulnerability Management Preparation

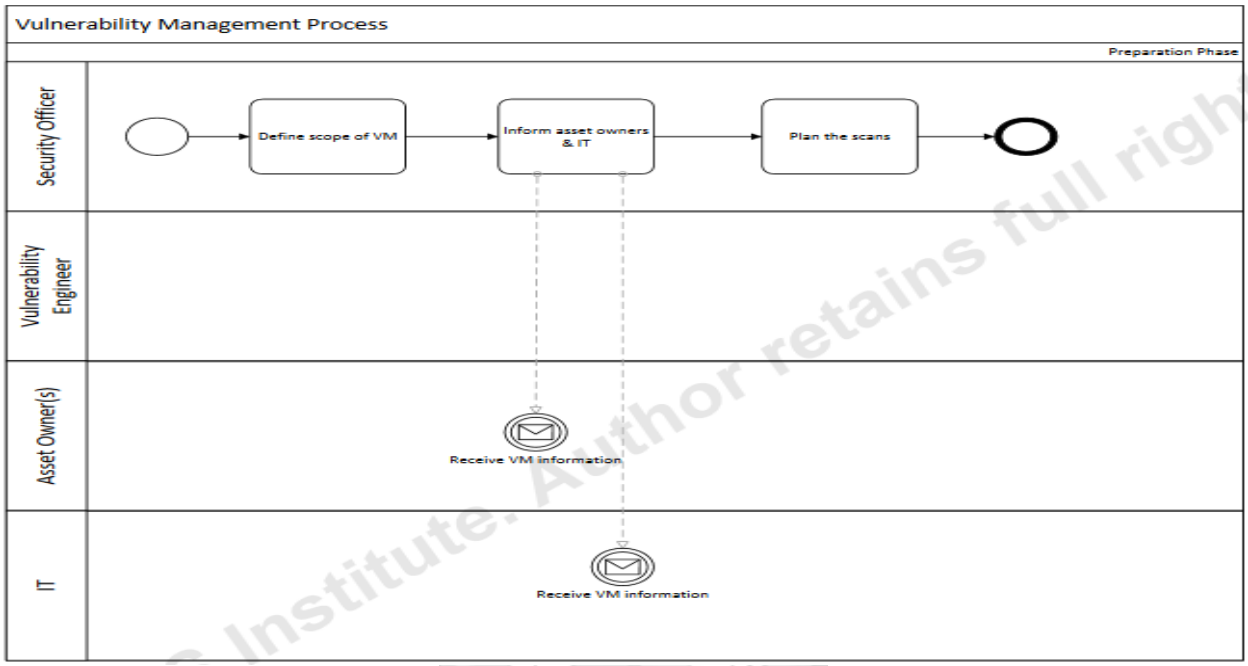


Figure 2.1: Vulnerability Management - Preparation Phase

(Palmaers, 2020)

Figure 2.1 shows the initial phase of the vulnerability management process that involves narrowing the scope to avoid being overwhelmed by the large number of vulnerabilities detected in the initial scans. It is recommended to focus only on vulnerabilities with known exploits that grant remote access. The preparation phase is mainly the responsibility of the Security Office in an organization (Palmaers, 2020).

2.5.2 Vulnerability Scan

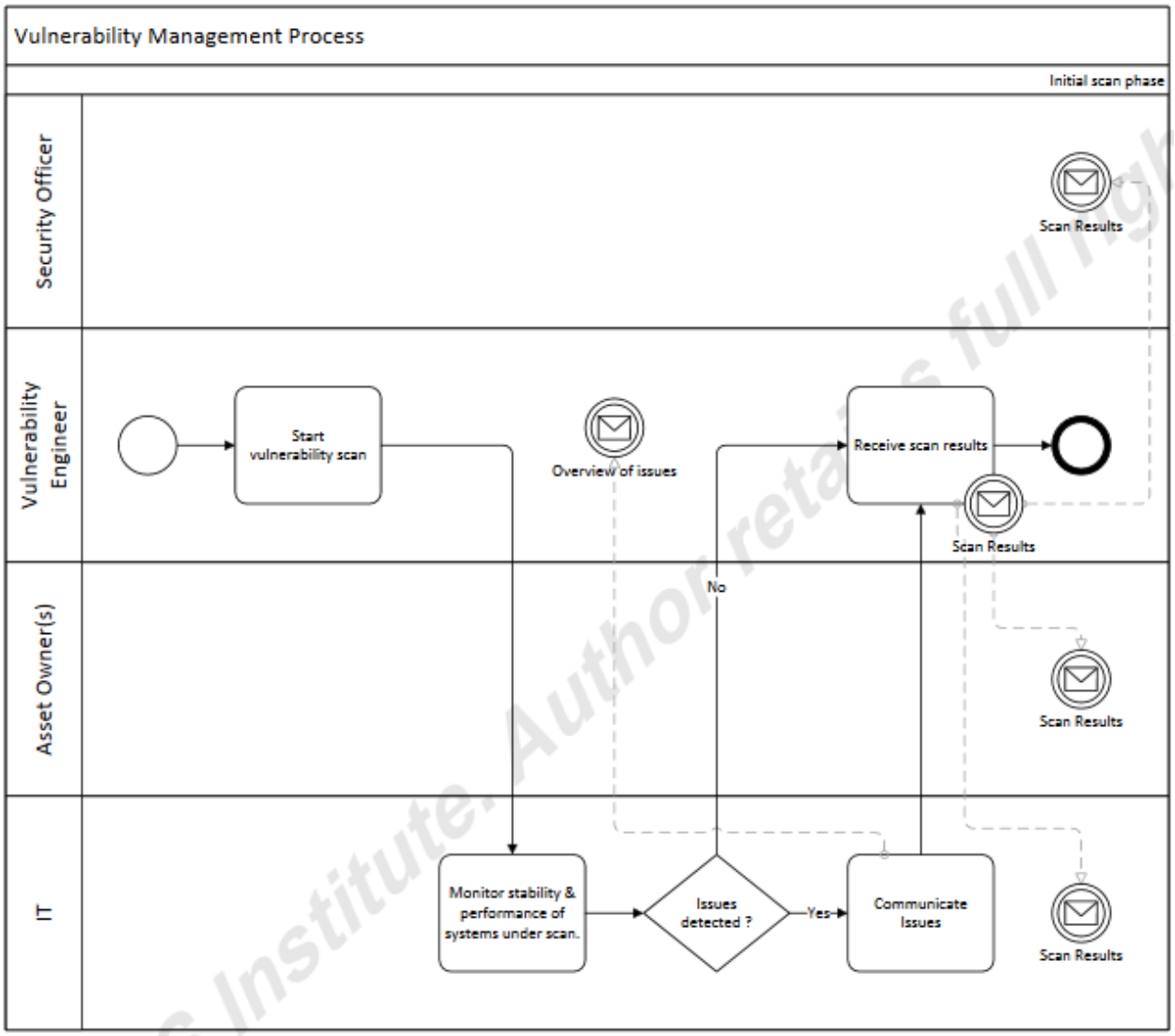


Figure 2.2: Vulnerability Management - Initial Scan

(Palmaers, 2020)

As seen in Figure 2.2, any issues that arise during the scans, such as system unavailability or poor application performance, should be documented since this may happen again in the future. Appropriate actions may be defined to minimize the impact of future scans on the stability or performance of the target systems. (Palmaers, 2020).

2.5.3 Define Remediation actions

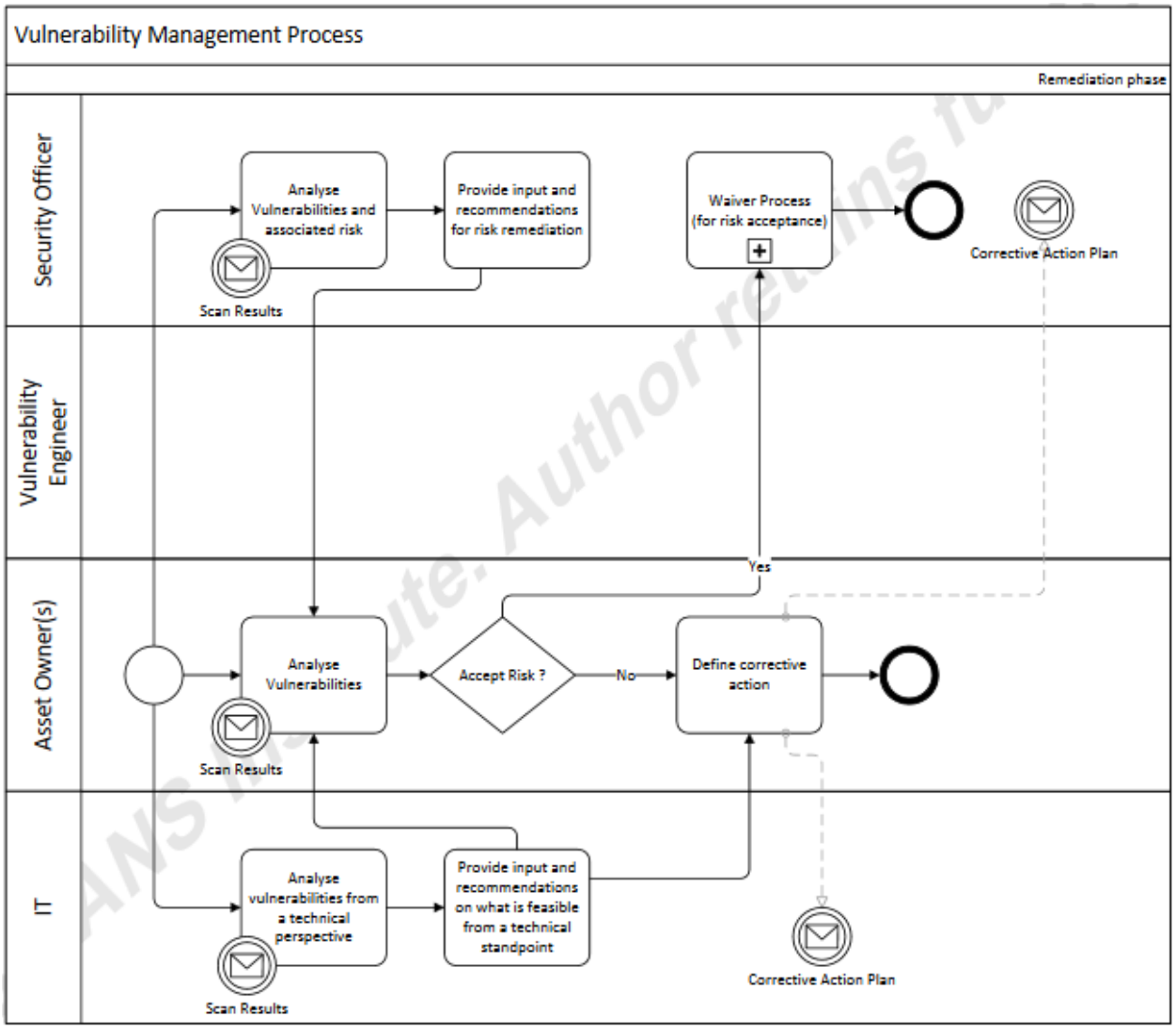


Figure 2.3: Vulnerability Management - Remediation Phase

(Palmaers, 2020)

In Figure 2.3, the owners, in collaboration with the IT department and the security officer, outline the necessary remediation actions. The security officer evaluated the vulnerabilities, assessed the related risks, and provided guidance on how to address these risks. Meanwhile, the IT department examined the vulnerabilities from a technical perspective, identifying available patches and options for strengthening configurations. Their recommendation also includes the feasibility of

the possible remediation actions such as whether applying a specific patch might render an application unsupported by the vendor. (Palmaers, 2020).

2.5.4 Implement Remediation Actions

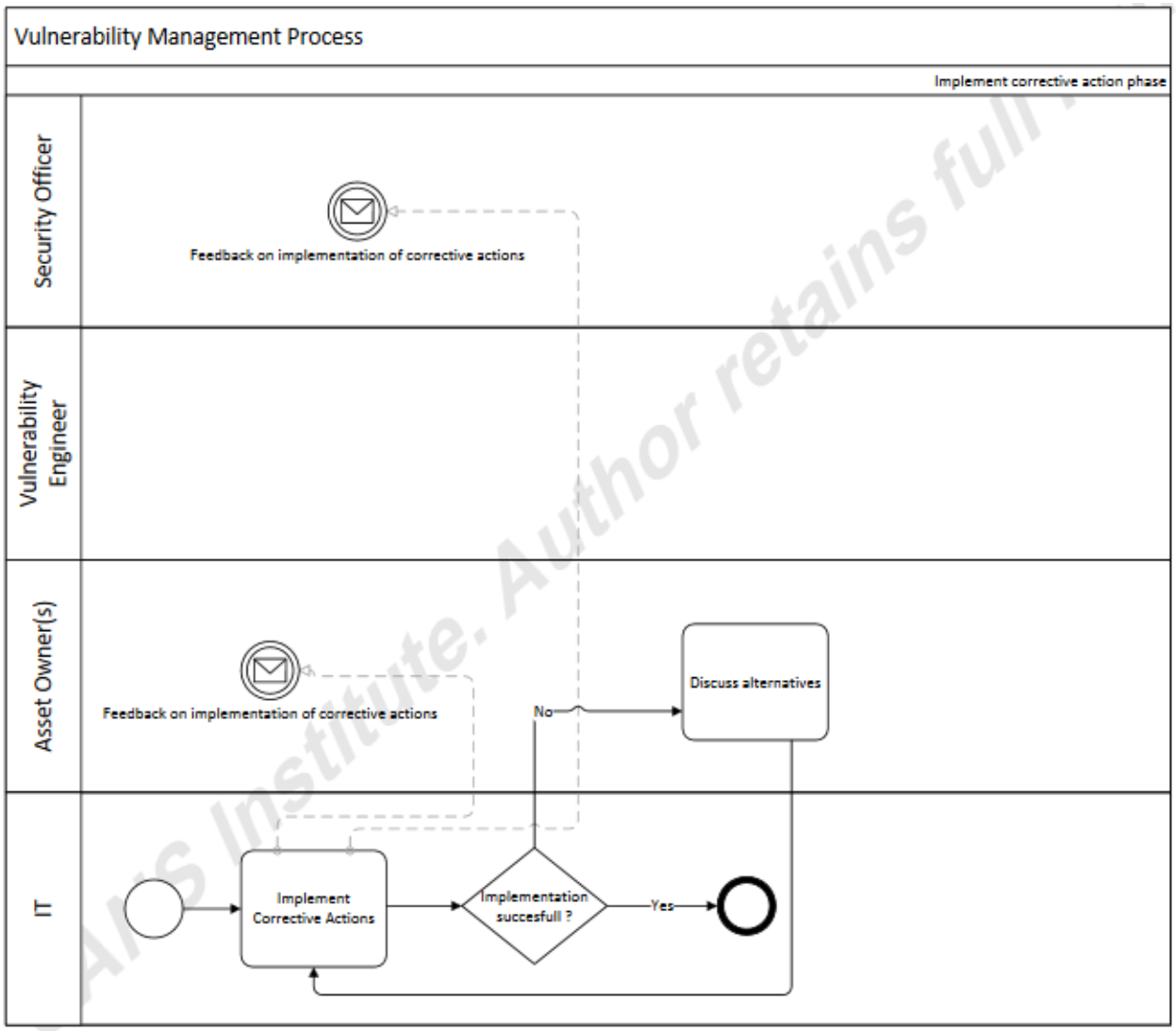


Figure 2.4: Vulnerability Management - Implementing Remediation Actions

(Palmaers, 2020)

In Figure 2.4, the planned remediation actions must be carried out within the specified timeframes. If any issues arise during implementation, they should be documented. The owner, with input from

the security officer and IT department, should define alternative actions to address the problem. (Palmaers, 2020).

2.5.5 Rescan

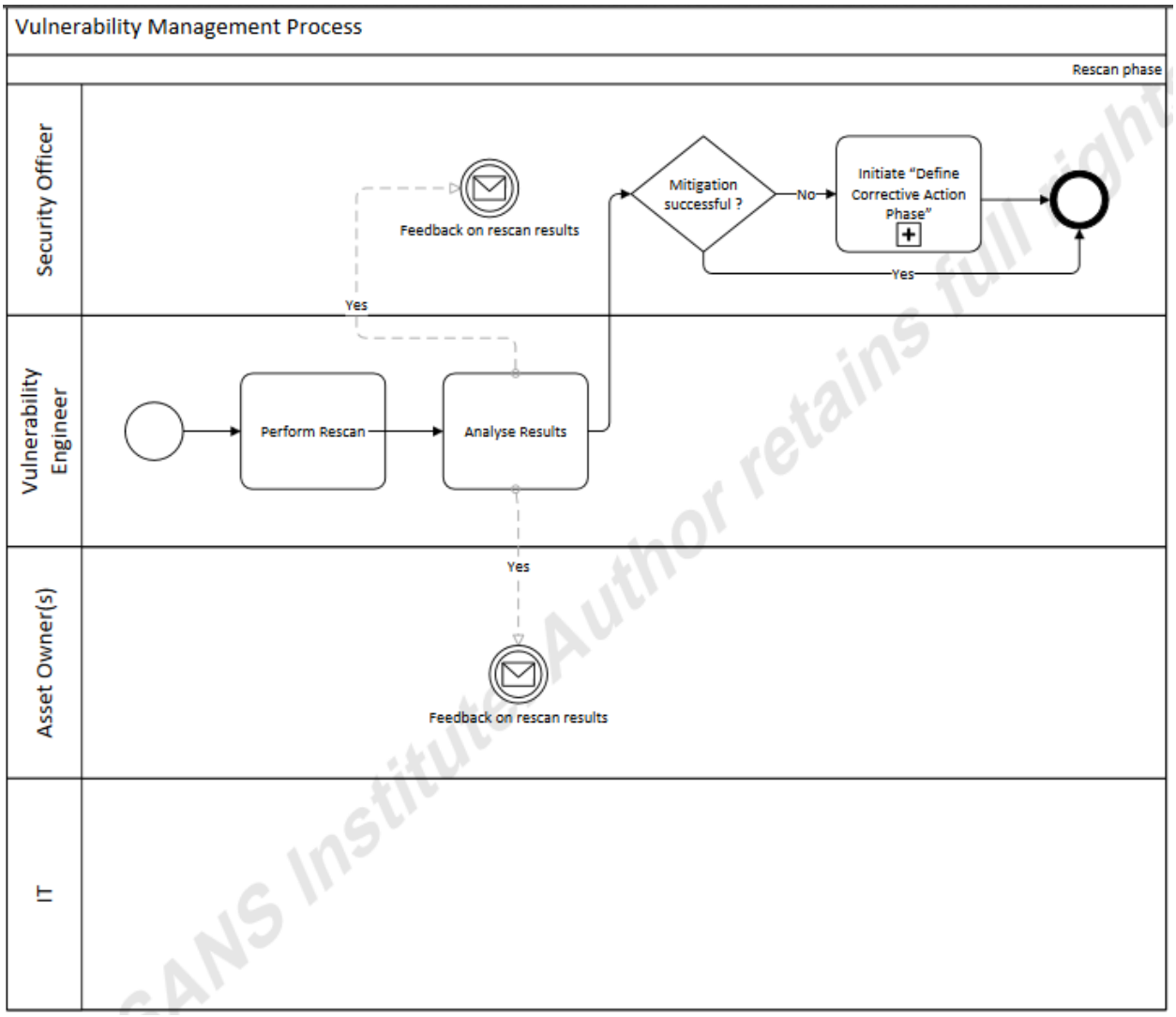


Figure 2.5: Vulnerability Management - Rescan Phase

(Palmaers, 2020)

As seen in Figure 2.5, after the vulnerability is remediated, a rescan should be scheduled to confirm that the corrective actions have been successfully implemented. This scan must use the same

vulnerability scanning tools and identical configuration settings as the initial scan to ensure accuracy and avoid errors caused by configuration inconsistencies. (Palmaers, 2020).

2.6 General Security Approaches for IoT Devices

While attacks on IoT are not a new phenomenon, DDoS and other malicious attacks have become an emerging threat. Millions of IoT devices are connected to the internet, and this number continues to grow at a rapid pace (James, 2019). The DDoS attacks on October 21st, 2016, (O'Brien, 2016) were anticipated by many nearly a year before they occurred. However, the lingering question then, as it is now, remains: How do you patch a 'thing'?

2.6.1 Change the Default Credentials

Changing the IoT device(s) from the default password to a password that only you know and can remember is recommended, immediately during the configuration stage. In the event one forgets the password, most devices have a reset switch that can be used to restore to the thing to its factory-default settings (HaddadPajouh et al, 2021).

2.6.2 Secure Your Network

Second defense measures consist of using passwords to safeguard your devices. Secure IoT devices set the main network as your primary entry path since proper security protocols strengthen them. A standalone firewall represents one of the most efficient methods to protect networks while several other approaches exist. The electronic barrier serves as a specific defense system which prevents hackers from reaching sensitive information (Sears, 2018) .

2.6.3 Update the Firmware

IoT device vendors often release security updates for the software that runs on their consumer devices, known as "firmware." It's recommended to visit the vendor's website to check for any available firmware updates before using your IoT devices and to regularly check for new updates as they become available (HaddadPajouh et al, 2021).

2.6.4 Removing Old Devices

Various new IoT devices are surfacing and substitutes for such devices are being introduced to the marketplace than ever before. It is more and more easy to fail to recall every connected device on an existing network but, some if not all the old IoT devices may be carrying forgotten passwords, old security protocols, and a list of potential threats to the existing networks. All connected IoT devices in a network are considered a potential weak point that needs to be secured. Old and no longer in use IoT devices must be disconnected from the network is an activity that one has to undertake (Sears, 2018).

2.7 Existing Vulnerability Scanner Solutions for IoT

2.7.1 Dojo Intelligent IoT Vulnerability Scanner App

The Dojo Intelligent IoT Vulnerability Scanner for Communication Service Providers (CSPs) represents a friendly application available free-of-charge for customer use. The first and only CSP-grade solution enables active home network scanning to detect all vulnerabilities of connected Wi-Fi devices (Gibson & Hull, 2020).

Figure 2.6 shows a screenshot of the application in action from Dojo by BullGuard.

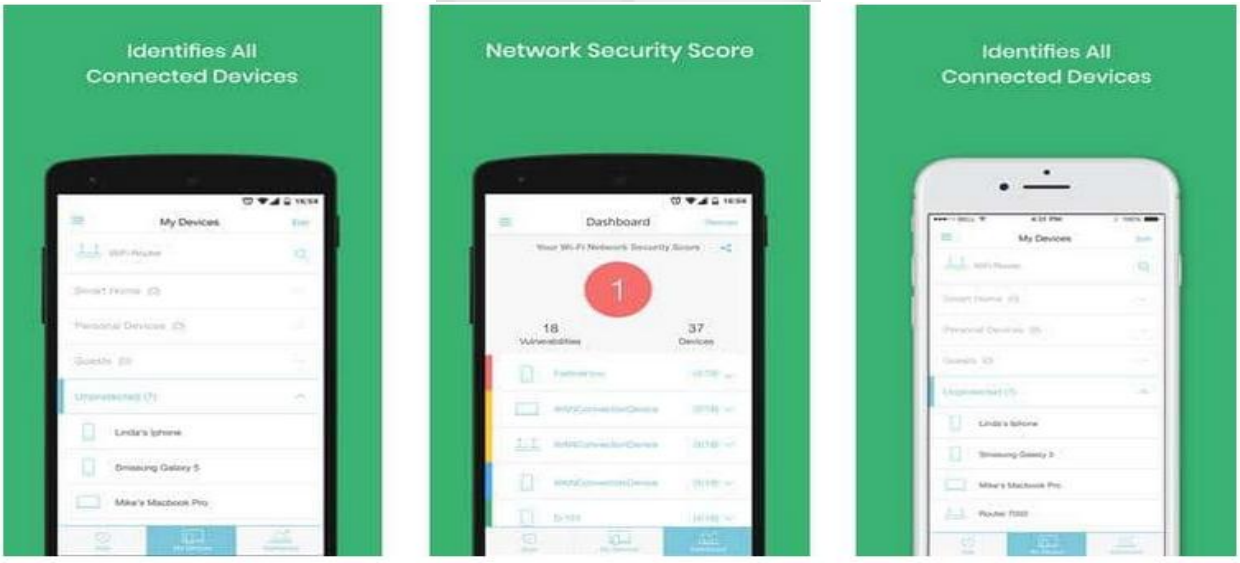


Figure 2.6: Dojo Vulnerability Scanner App

(Sharma, 2020)

The Dojo Intelligent IoT Vulnerability Scanner app performs an Automatic Device Discovery that scans a customer's Wi-Fi network without having to install any extra software and uses combination of both local and cloud-based intelligent detection engines for fast and accurate device discovery. For each device discovered, the tool utilizes the cloud-based security risk assessment platform to analyze vulnerabilities, and after each full network scan, the system reports all identified vulnerabilities and assigns an overall score on a scale from 1 (worst) to 10 (best) for the network.

2.7.2 Mirai Scanner for IoT

A botnet called Mirai, which uses malware, was behind the DDoS attack. This malware constantly looks online for unprotected IoT devices, which it then takes over to launch attacks. Mirai uses a small list of 62 common usernames and passwords to find these vulnerable devices. Since many IoT devices are not secured very well, this simple method helps the bot connect to hundreds of thousands of them (America's Cyber Defense Agency, 2017).

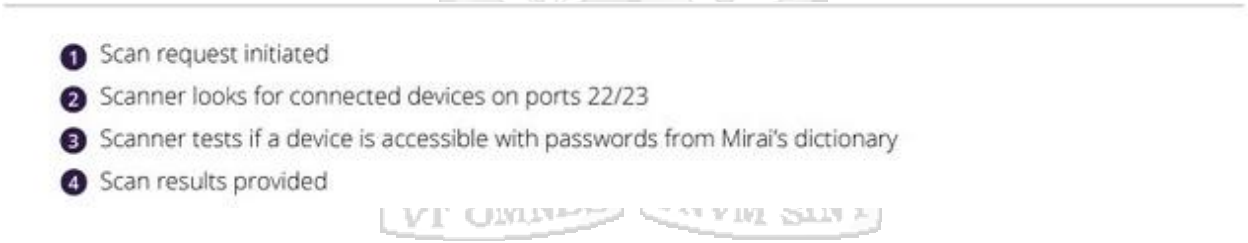


Figure 2.7: Mirai Scanner Summary

(Hamilton, 2016)

As seen in Figure 2.8, when one first runs a scan, the message may appear because a device being scanned is either infected with Mirai or has no vulnerable ports—most likely the latter.

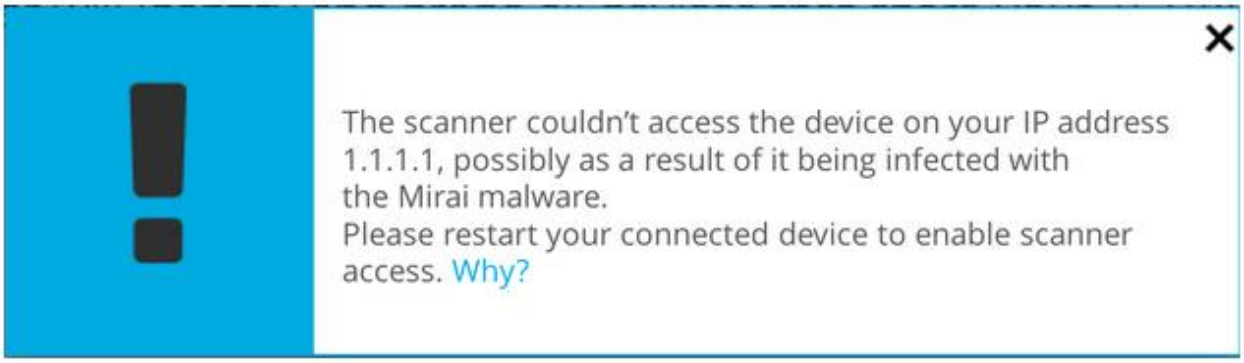


Figure 2.8: Unsuccessful Scan response from Mirai Scanner

(Hamilton, 2016)

To confirm that a device is not infected with Mirai or it does not have vulnerable ports, restart any IoT devices on the network to clear Mirai's ability to block ports on an infected device, preventing scans. After the devices reboot, rerun the scan. If the same message appears again, it indicates that remote access ports are closed, preventing Mirai from infiltrating your devices. When the scanner examines your network, it checks whether any devices can be remotely accessed using passwords from Mirai's dictionary (Hamilton, 2016). Figure 2.9 shows the message when the scanner finds a vulnerability.

Figure 2.9: Successful scan showing a possible vulnerability to a Miriai botnet infection

(Hamilton, 2016)

This message being generated indicates that one or more devices have been detected by the scanner on the network that are vulnerable to the Mirai malware, though this does not necessarily mean an infection is present.

If a device with an existing vulnerability is found, one would take the following steps:

i. Log in to each IoT device on your network and set the password to a strong, secure one.
ii. Rescan your network to verify that the vulnerability has been addressed (Sheridan, 2020).

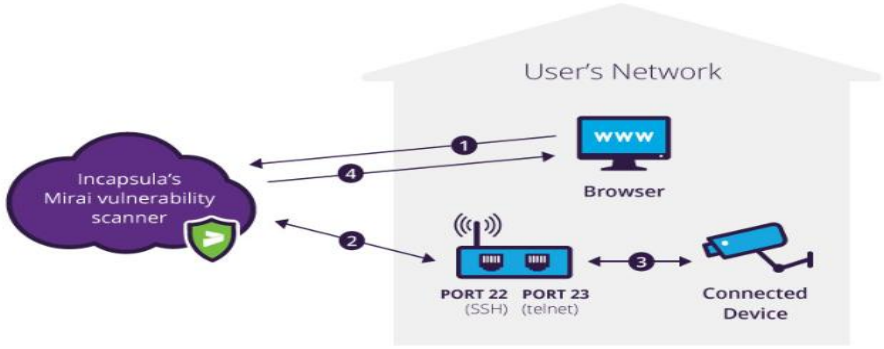


Figure 2.10: Mirai Vulnerability Scanner

(Hamilton, 2016)

2.7.3 Retina IoT Vulnerability Scanner

The scanner provides you with complete visibility into your IoT devices where it identifies systems using either basic default passwords or insecure passwords. Most users do not modify default passwords in their devices thus creating security hazards because their devices remain exposed to threats (Dean, 2017). This scanner is free for a trial basis for corporate/business use only and no software or hardware to install. The shortfall of this solution is that it has to be paid for a business or individual to access the full functionality of the solution. It also does not support the download of reports that can be used by administrators at a later date.

Figure 2.11: Retina IoT Vulnerability Scanner

(Dean, 2017)

2.7.4 Firmalyzer

This scanner helps different involved parties to perform security compliance checks as well as risk assessment on the IoT connected devices. The only thing required to make this assessment is the device itself connected to the platform and existing firmware is analyzed and vulnerabilities reported (Firmalyzer, 2019). It focuses on device-level detection, control and visibility that is completely ignored by other IoT security solutions. These try to address the security issues at a network-level or on device level and are just protective solutions at the hardware level which are not enough (Firmalyzer, 2019).

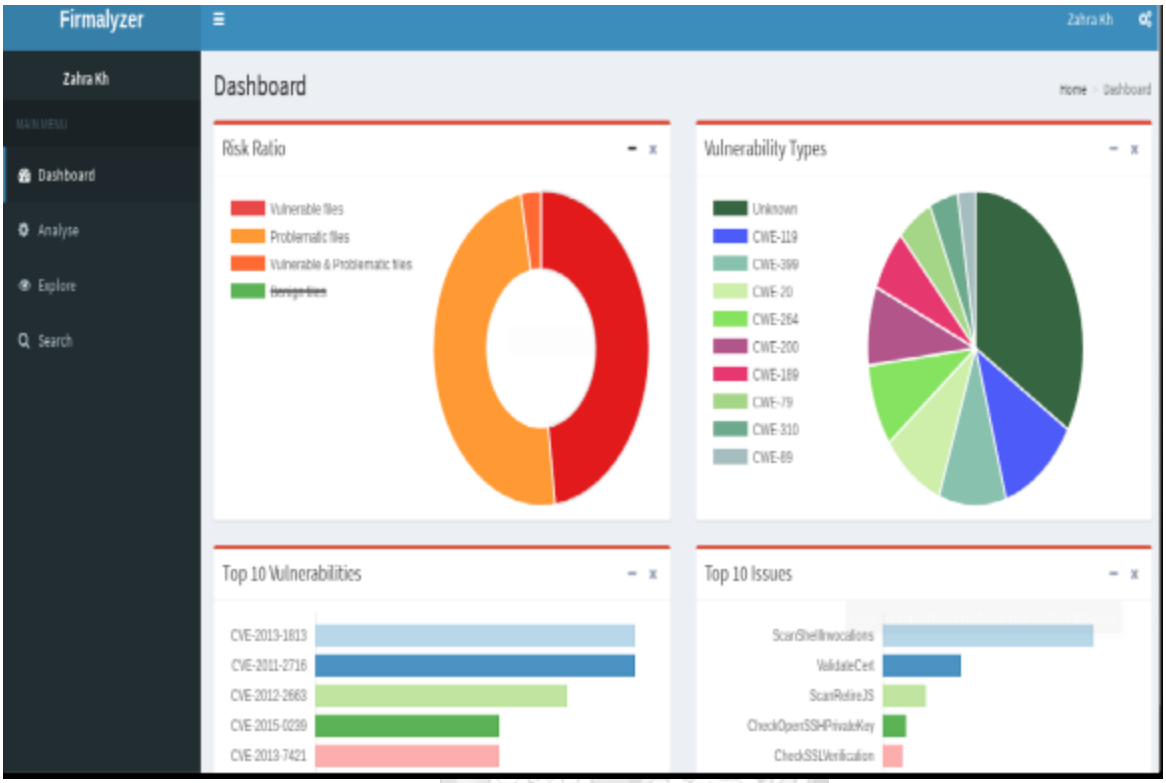


Figure 2.12: Firmalyzer Dashboard

(Firmalyzer, 2019)

Figure 2.12 shows the dashboard of a firmalyzer setup that shows the types and top 10 vulnerabilities present in connected system/network. Using the Common Vulnerabilities and Exposures (CVE) database, you can get a detailed description of the vulnerabilities present using the unique coding system.

Vendors that provide IoT can easily add Firmalyzer enterprise to their security software development lifecycle in the test phase. For each instance they want to release a new firmware image, upon completion of development and generating the binary firmware image, upload to Firmalyzer would follow which sequentially and automatically analyzes the binary firmware and any security issues are reported. Users can redistribute the firmware through the platform after resolving identified issues and reanalyzing it. Users should start the updated release firmware process after fixing all issues (Firmalyzer, 2019).

The shortfall of this solution is that it has to be paid for a business or individual to access the full functionality of the solution.

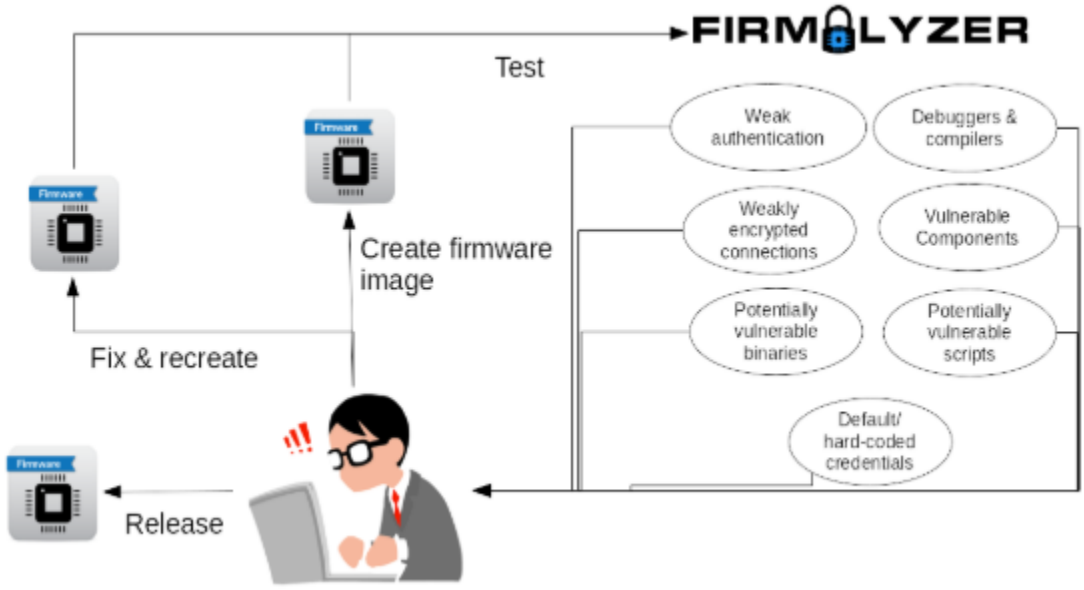


Figure 2.13: Firmalyzer Scanner

(Firmalyzer, 2019)

2.8 Gaps in the Existing Solutions

The existing IoT vulnerability scanning solutions have made significant strides in identifying security weaknesses in IoT devices. A review of commonly used scanners reveals several key limitations that hinder their effectiveness in diverse IoT environments. A solution needs to be provided that provides additional options according to the table data. In addition to scanning IoT devices, the tool developed in this study through LAN or WI-FI can also scan connected devices through Bluetooth and analyze data from a live scan. The tool is open-source and free for administrators to conduct analysis of vulnerabilities.

Table 2.1: Summary of Comparison of Existing Vulnerability Scanning Tools

Tool	Strengths	Limitations
Dojo IoT Vulnerability Scanner	Provides overall network security and detects vulnerabilities	Supports on Wi-Fi networks, lacks detailed remedy reports.
Mirai Scanner for IoT	Identifies IoT devices vulnerable to Mirai botnet attacks.	Limited to Mirai-specific vulnerabilities.
Retina IoT Vulnerability Scanner	Detailed insights to network administrators	Requires a paid license and does not support Bluetooth-connected devices.
Firmalyzer	Analyzes IoT firmware	Requires additional hardware and the commercial tool is expensive.

In this study, the tool covered three gaps that are present in the existing solutions. First it can scan devices connected to a Bluetooth Connection. Existing solutions do not have a scan feature for Bluetooth connections. Secondly, where majority of the existing solutions are for commercial use, the tool covered in this study is free for use. Lastly, the tool provides downloadable reports that the existing solutions may not be freely made available. All reporting features for existing solutions are present within the application rather than providing a downloadable report.

2.9 Conceptual Framework

This study intends to show that IoT devices connected can be scanned and analyzed using data from an existing database to validate and show the vulnerabilities in the connected IoT devices. The IoT Vulnerability Scanner is designed to help find security flaws in Internet of Things (IoT) devices. It has three key steps: scanning IoT systems, analyzing and validating the data, and identifying vulnerabilities.

The process starts by locating connected IoT devices using Wi-Fi or Bluetooth. Once we gather this information, we check it to find possible security issues like weak passwords, open ports, and unencrypted data and compare the discovered vulnerabilities with a large database of known attacks. As the last step, a report is generated indicating the identified vulnerabilities as well as the steps to mitigate them. This clear approach makes it easier to spot problems early and take steps to protect connected devices.

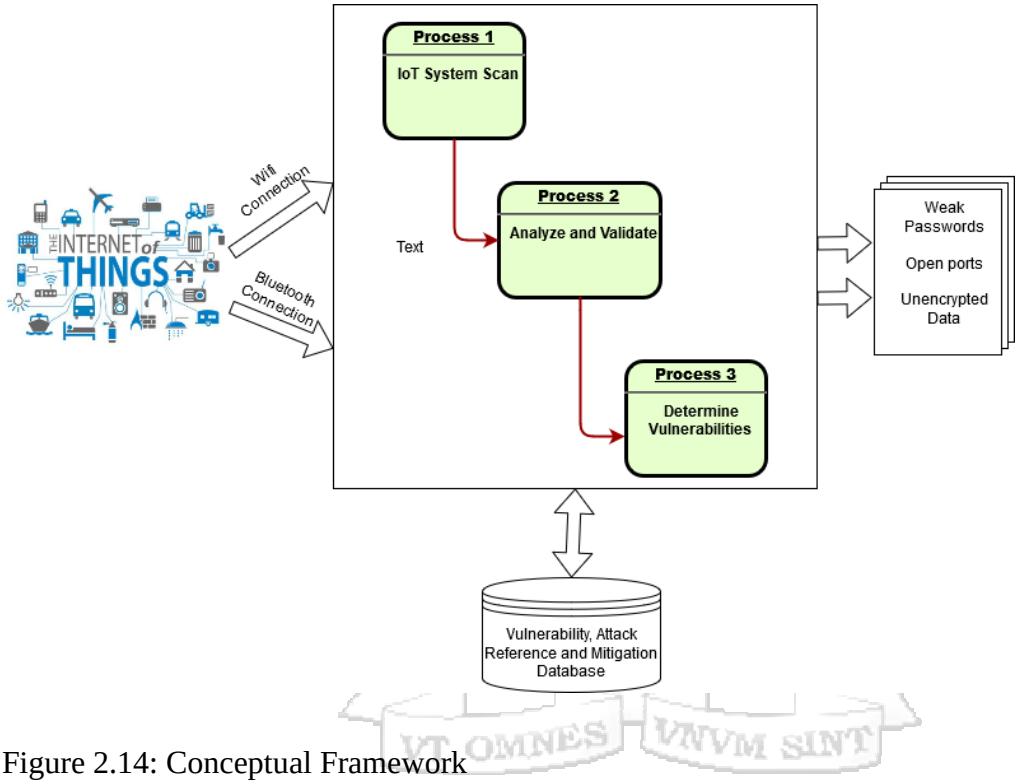


Figure 2.14: Conceptual Framework

In Figure 2.14, the illustration shows the Data captured from the IoT devices is can and cross-referenced from the Vulnerability Database and show the vulnerabilities present in the IoT Devices. In comparison to the existing solutions that primarily focus on Wi-Fi networks, the proposed tool supports both Wi-Fi and Bluetooth connections. The Rapid Application Development (RAD) methodology used in the development of the tool allows for prototyping and continuous improvements.

2.9.1 Key Components of the Proposed Vulnerability Scanner

The IoT Vulnerability Scanner has three main parts, the Network Scanning Engine, the Vulnerability Assessment Module, and the Reporting Module. The Network Scanning Engine is the heart of the system. It finds all the IoT devices connected through Wi-Fi and Bluetooth. It gathers important details like IP addresses, MAC addresses, and which ports are active, giving you a clear picture of every device connected to your network.

Next, the Vulnerability Assessment Module checks these devices against an up-to-date database of known issues. It looks for common problems like weak passwords, and open ports that could be security risks. Lastly, the Reporting Module creates detailed reports that you can download, showing you, the vulnerabilities found and offering suggestions on how to fix them. This setup not only helps you spot potential threats but also gives you practical tips to strengthen your IoT security.

VT OMNES VNVM SINT

Chapter 3 : Methodology

3.1 Introduction

Chapter three reviews the approaches used to address the study objectives. The chapter focuses on the study design and methodology for developing a security vulnerability scanner prototype for IoT devices. The study further outlines the process that was used in identifying IoT vulnerabilities through setting up a controlled testing environment and using a comprehensive attack database. The study also covers the methodologies that were used to review existing vulnerability scanning tools to determine their strengths and limitations, which informs the development of the prototype. The chapter also covers system validation to ensure the tool meets the study objectives and includes the ethical considerations related to data protection and user consent.

3.2 Study Design

This study utilized an experimental study design to assess current IoT vulnerability scanning tools and create a prototype that fills the gaps identified in these existing solutions. The experimental framework was set up in multiple phases, beginning with the evaluation of existing tools. A variety of widely used IoT vulnerability scanners were tested in a controlled environment with two different sets of IoT devices. Key performance metrics such as detection accuracy, false positive/negative rates, platform compatibility, and scanning efficiency were analyzed to establish a baseline for comparison. The results from this phase informed the design and development of a new prototype tool that incorporates features to overcome the limitations of existing tools.

Following prototype development, it underwent extensive testing in both controlled lab environments and real-world IoT networks to assess its effectiveness. The prototype was deployed in smart homes, industrial systems, and other practical IoT environments to validate its vulnerability detection capabilities. Data collected from these tests included both quantitative metrics and qualitative feedback from users regarding the tool's usability and real-world performance. By comparing the prototype's performance with existing solutions, the study aims to demonstrate its effectiveness in addressing gaps and improving IoT security.

3.2.1 Identification of IoT Vulnerabilities

To identify vulnerabilities within IoT devices, a controlled environment was set up where one or two IoT modules were connected to an existing network. This setup enables a structured approach to testing weaknesses. A comprehensive database containing various known attack types and exploitable vulnerabilities was used as a reference. By simulating different attacks from this database, the IoT modules were assessed to determine whether the system could accurately identify the exploited vulnerability and the specific attack type used. This method allowed for vulnerability mapping and validation of the device's response to potential threats.

3.2.2 Review of Existing Tools

To assess the current capabilities of vulnerability scanning tools, an in-depth review of these tools and their sample output data was conducted. This evaluation directly addresses the first and second study questions: identifying common vulnerabilities currently affecting IoT devices and determining the strengths and limitations of existing IoT vulnerability scanning tools. The review process involved examining each tool's input and output requirements, such as supported file types, as well as the execution platform, operating system compatibility, scanning scope, and whether the tool is available in free or paid versions.

In addition, each tool was assessed for its ability to detect specific IoT vulnerabilities, providing insight into their utility in real-world applications. This comprehensive evaluation will allow the identification of tools that will meet the needs of the study and highlight areas where existing tools may fall short, such as limited scanning capabilities and the type of network configurations they support. This analysis informed the development of a prototype vulnerability scanner by revealing gaps in current tools and identifying features essential for effective IoT security.

3.2.3 Design and Development

To achieve the objective of design and development, the study employed Rapid Application Development (RAD) methodology. The methodology was selected because it heavily emphasizes rapid prototyping and iterative delivery with its biggest advantage being the adaptability of changing and updating the requirements quickly and easily during development (Yen & Davis,

2019). This methodology is a people-centered and incremental development approach with active user involvement, collaboration and co-operation between all stakeholders. The RAD model has four phases; Requirements planning, User Design, Rapid Construction and Cut Over (Yen & Davis, 2019).

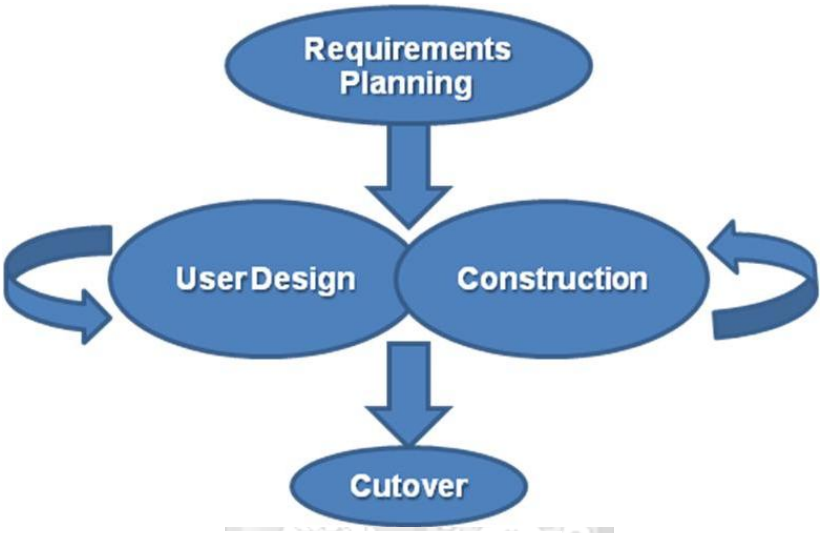


Figure 3.1: Rapid Application Development (RAD) Framework

3.2.3.1 Requirements Planning

The study includes looking at everything related to the proposed prototype. This helps us come up with clear requirements of what the prototype needs to do in terms of its functions (The Government of the Hong Kong Special Administrative Region of the People's Republic of China, 2024). At this phase, it involved making rough sketches of the scope to be covered and the necessary application requirements that included some necessary hardware elements to make this work.

3.2.3.2 Design

A system requires its design phase to be treated as a fundamental development step. Throughout this phase the SDLC process transforms its attention from answering "what" questions in analysis to developing solutions that answer "how" questions about implementation and development (NIOS, 2019).

At this step, potential user feedback was taken and then utilized several developer tools of the project under development. Where the requirements were relative at this stage some distinct features were employed towards further customization.

System Development Tools and Techniques

i. Sublime text - A text and source code editor programming and scripting languages. Used in the development of the tool
ii. Microsoft Visio - Used for drawing database ERDs, Use case and Sequence Diagrams.
iii. XAMPP – To be used in Database management
iv. Microsoft word 2010 – To be used in the documentation of the system.

3.2.3.3 Rapid Construction

The actual coding and testing took place at this stage and was repeated often to accommodate alterations to meet the requirements of the tool.

Testing

This is the process of establishing the functionality of a solution and determining if it meets the requirements set during the requirements and planning phase. This phase answered the third question on how the tool should be designed, developed, and tested effectively. This involved evaluating whether the tool successfully met its intended objectives. This included;

i. Scanning the connected IoT Device to determine the open port, the type of data being captured and how the data is transmitted
ii. Analysis the findings and generate a report showing the possible attacks, the vulnerability present, possible outcomes and possible remediation factors.

3.2.3.4 Cut Over

The current phase involves the development team moving the components into the live production environment. This step allows for comprehensive testing on a larger scale and provides an opportunity for team training to occur (Powell-Morse, 2018). The current phase involves the rolling out the new system that means showing users how to use the prototype, and providing

support to fix any issues that come up right after the prototype goes into production (The Government of the Hong Kong Special Administrative Region of the People's Republic of China, 2024)

This took place upon improvements made to the tool that allowed ease to use as this was designed for the consumption by the administrators.

3.2.4 System Validation

System validation establishes through verified evidence that the built solution fulfills its purpose along with its designed set of user requirements (UCBerkeley, 2020). The system was validated by ensuring that the tool conforms to the study objectives. This was achieved by putting it in a set of controlled experiments. This included gaining unauthorized access to a network, a live scan of connected IoT devices through the WI-FI and Bluetooth Connections, analysis of findings with information captured in a database, and finally generating a report that can be used by an administrator as remediation actions. Validation was performed by comparing the outcomes from the developed tool against those obtained from various other vulnerability scanning tools.

3.2.5 Study Quality

The quality of this study was ensured through rigorous attention to validity, reliability, rigor, generalizability, and ethical considerations. Validity was addressed through a controlled experimental design and real-world testing to ensure accurate and applicable results. Reliability was maintained by standardizing testing procedures and using automated scripts to minimize human error, ensuring consistent results across repeated tests. The study was conducted with rigor, using a systematic approach to evaluate existing tools and develop the prototype, with comprehensive feedback from real-world environments to refine the tool's design. Generalizability was ensured by testing the prototype across a diverse range of IoT devices and environments, ensuring that findings apply to a large variety of IoT configurations.

In this study, the ethical considerations were prioritized by focusing on the privacy and security of Internet of Things (IoT) devices. It was ensured that no systems are compromised during the testing process. Any collected data was anonymized compliant with data protection regulations. The use

of secure, controlled environments for testing was further safeguarded against unintended breaches. By maintaining high standards of quality in validity, reliability, and ethical conduct, this study aims to generate robust and applicable findings that give way to the advancement of IoT security practices.

3.2.6 Data Analysis

Data analysis for this study combined both quantitative and qualitative methods to evaluate the performance and effectiveness of existing IoT vulnerability scanning tools and the newly developed prototype. Quantitative data focused on performance metrics such as detection accuracy, false positives/negatives, scan time, and usability scores. To compare performance metrics between the prototype with existing tools, a mixed-methods evaluation using both descriptive and inferential statistics established significant differences (Barbosa & Mattos, 2025). This analysis provided a clear, objective measure of how well the tools perform in detecting vulnerabilities and how efficiently they operate.

In addition to the quantitative data, qualitative feedback from users and security professionals was collected to assess the usability and real-world applicability of the tools. Thematic analysis was used to identify key themes in the feedback, such as usability challenges, tool limitations, and the effectiveness of the scanner in different IoT environments. By integrating quantitative and qualitative data (mixed-methods research) it produces evaluation of complex problems. This aligns with the original assertion that combining both data types "provides valuable insights" into a system's strengths and weaknesses (Creswell & Creswell, 2022). This approach ensured a well-rounded understanding of the prototype's effectiveness in addressing IoT security vulnerabilities.

3.2.7 Ethical Considerations

This study was conducted with strict adherence to ethical guidelines to ensure integrity, security, and responsible handling of study data. All IoT devices used for testing were not obtained from external device owners or administrators, eliminating concerns regarding informed consent and data privacy. The devices selected were under the full control of the study team, ensuring that no third-party individuals or organizations were exposed to unintended risks. The study did not interfere with live operational systems or compromise the functionality of any devices beyond the

controlled testing environment. Additionally, no personally identifiable information (PII) or sensitive data was collected during the vulnerability scans. The study findings are intended solely for academic and cyber-security improvement purposes, with no commercial exploitation or misuse. Clearance from the Strathmore Ethics Committee was sought as seen in Appendix B.

Chapter 4 : SYSTEM DESIGN AND ARCHITECTURE

4.1 Introduction

This chapter outlines the design and architecture of the proposed IoT Vulnerability Scanner (IVS) tool. It covers both functional and non-functional requirements, presents the overall system architecture, illustrates data flow, and highlights important design components. The goal is to develop a scalable, efficient, and user-friendly security tool that can detect vulnerabilities in Wi-Fi and Bluetooth-connected IoT devices.

4.2 Requirements Planning

The requirement specification served as a crucial tool during the testing phase to ensure that the objectives are achieved. Through the analysis of case studies and various secondary materials, we have identified the following requirements. These encompass both functional and non-functional aspects.

4.2.1 Functional Requirements

In the development of the IVS tool, this is a major requirement. The tool is capable of scanning connected IoT devices within an IoT Network that utilize Wi-Fi and/or Bluetooth to determine the vulnerabilities present and generate a report show casing the steps to mitigate such a vulnerability.

What the tool is capable of:

i. Scan an IoT Device(s) connected through a WI-FI connection

ii. View data from connected IoT device(s) through WI-FI connection

iii. Ability to identify vulnerabilities in IoT Devices connected through WI-FI connection

iv. Scan an IoT Device(s) connected through a Bluetooth connection

v. View data from connected IoT device(s) through Bluetooth connection

vi. Ability to identify vulnerabilities in IoT Devices connected through Bluetooth connection

vii. Generate reports for analysis/scan conducted

4.2.2 Non-Functional Requirements

The Non-Functional requirements included in the tool are:

i. The performance in report generation is quite fast

ii. A scan is performed in a relatively short time dependent on the IoT devices connected

iii. The tool is scalable to incorporate a large number of IoT devices during a scan

iv. The tool is quite reliable when a small number of IoT devices are connected

v. The tool is easily accessible and at any given time.

vi. Practical and easy to use by System Administrators.

4.3 System Architecture

The system is designed using a two-tier architecture, which includes a frontend client terminal interface and a backend that is hosted remotely. The client interface is built with Python and enables users to connect to, scan, and analyze IoT devices. Additionally, it generates reports that present information in an easy-to-understand format.

The MySQL database is part of the backend that contains a list of possible vulnerabilities, possible attacks and possible mitigation procedures. During analysis of a scan, the results of the scan are crosschecked against this database for a possible vulnerability match. Once a match is found, a report is generated for the consumption of an administrator for further action. The system architecture is seen in Figure 4.1.

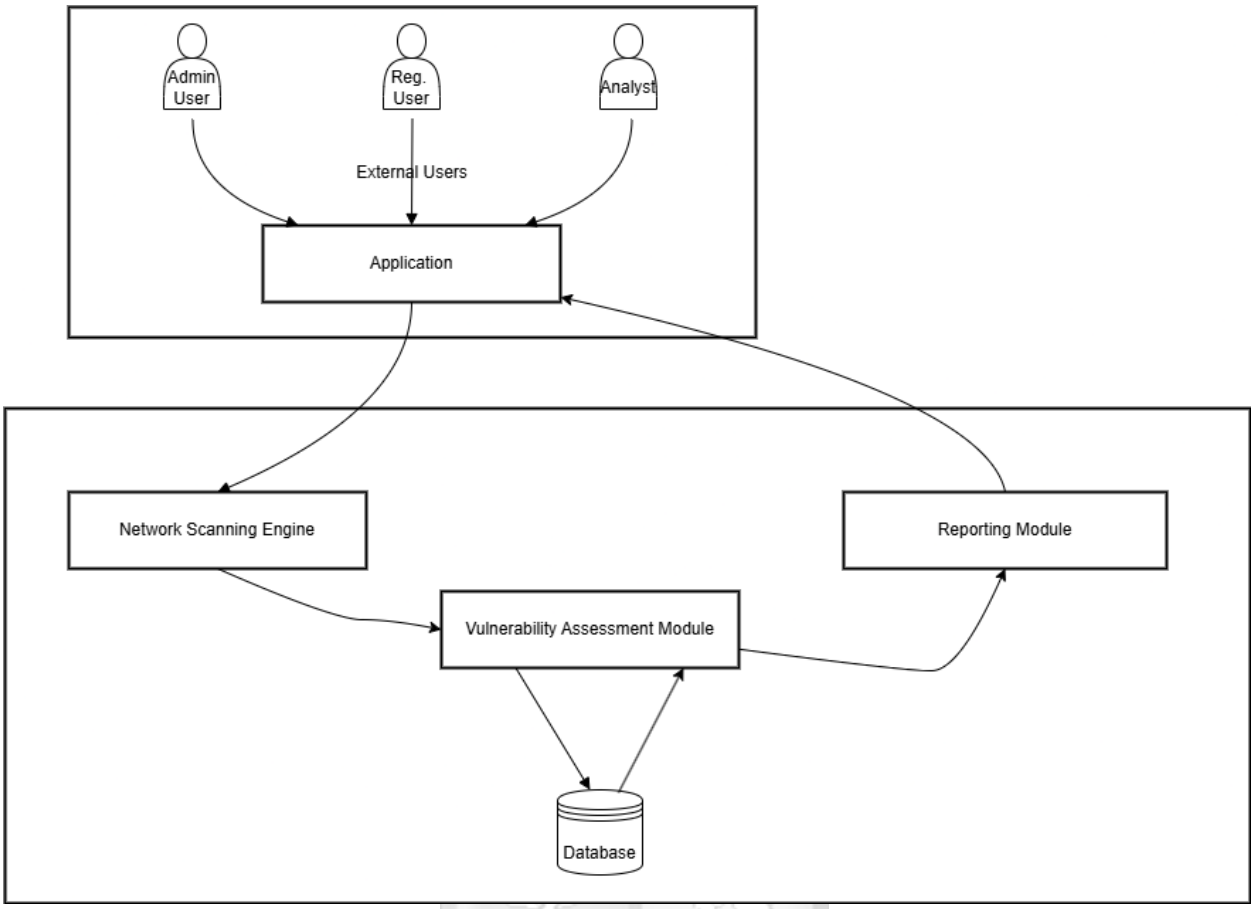


Figure 4.1: System Architecture

4.3.1 Input

Upon a successful connection to a WI-FI or Bluetooth network, a scan is initiated that scans all the IoT Devices connected. The scan is done for each individual connection.

4.3.2 Processes

When the IoT vulnerability scanner is initiated, it goes through a clear process to find and analyze any security issues in connected devices. It starts with understanding what devices are on the network, a step called Device Discovery and Network Mapping. During this phase, it spots all the IoT devices using the active scanning method. It gathers important information like IP addresses and open ports, creating a comprehensive picture of the network.

Next, in the Vulnerability Identification and Cross-Checking phase, the scanner checks the devices for common security flaws, such as open ports and easy-to-guess passwords. It matches its findings against a database of known vulnerabilities, using both detection and analysis techniques.

Backing this up is a strong Vulnerability Database that's updated from sources like the CVE database. Once vulnerabilities are found, as seen in Figure 4.2, the system creates a detailed report that lists any affected devices, the vulnerabilities found, and suggestions for fixing them.

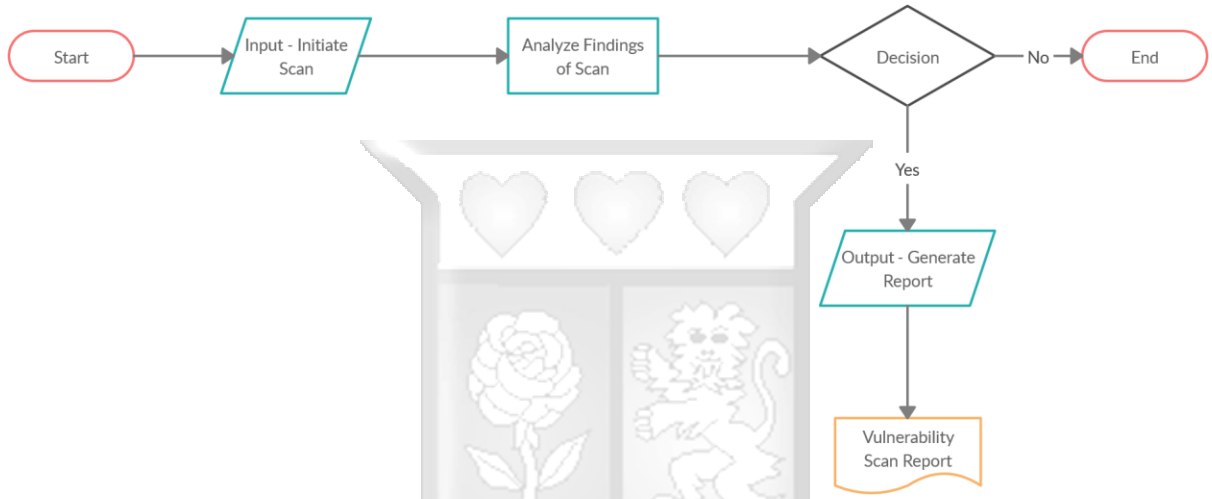


Figure 4.2: Activity Flow for the IoT Security Vulnerability Scanner

4.3.3 Outputs

Once the vulnerability scan is completed, the system generates a comprehensive security report that consolidates all identified vulnerabilities across the scanned IoT devices. This report is structured to provide an overview of the security status of each connected device and includes detailed information for each detected vulnerability. Instead of generating separate reports for individual vulnerabilities, the system compiles all findings into a single, structured report.

Each report contains the following key components:

1. Summary Overview – A high-level analysis of the scan results, including the attack type, any credentials determined, and the vulnerabilities detected.

2. Device-Specific Findings – A breakdown of vulnerabilities per device, listing the device name, IP address, open ports, and the protocols in use. This section helps system administrators quickly identify at-risk devices.
3. Report Storage and Accessibility – The generated report is automatically stored on the system administrator's computer for further analysis. The report can be exported in a PDF format.

4.4 System Design Tools

In a development process, the design phase of a system is crucial, as it lays the groundwork for what was ultimately built. During this phase, the primary focus is to create a model representing the object. The quality of the system is heavily influenced by decisions made at this stage. A well-thought-out design leads to a stable system, facilitating effective testing later on.

The tool proposed is an IoT Vulnerability Scanner Tool with the goal of scanning, analyzing and reporting. It contains three main modules: Connect, Scan and analyze and Report.

The Connect phase is where the administrator checks to make sure that both Wi-Fi and Bluetooth connections are working properly. It also looks for any weaknesses in these connections that could make them vulnerable to password attacks. Once the connections are confirmed to be good, we move on to the Scan and Analyze stage. Here, the system verifies that IoT devices are linked through either Wi-Fi or Bluetooth. Then, a scan is performed to gather information being sent through the devices' ports and communication protocols. This information is analyzed to determine its format and to spot any vulnerabilities. Finally, during the Report stage, the system puts together everything we found into a detailed report. This report highlights the vulnerabilities we identified and gives clear advice on how to fix them.

4.4.1 Use Case

A use-case diagram illustrates how the system operates, making it easier to visualize and understand both the system itself and its components. This tool is crucial for specifying features and simplifying the approach to systems and subsystems. The implemented tool is dubbed IVS

Tool (IoT Vulnerability Scanner Tool) and has two functionalities, to connect to an existing network and perform a live scan on connected IoT devices. This is illustrated in Fig 4.3.

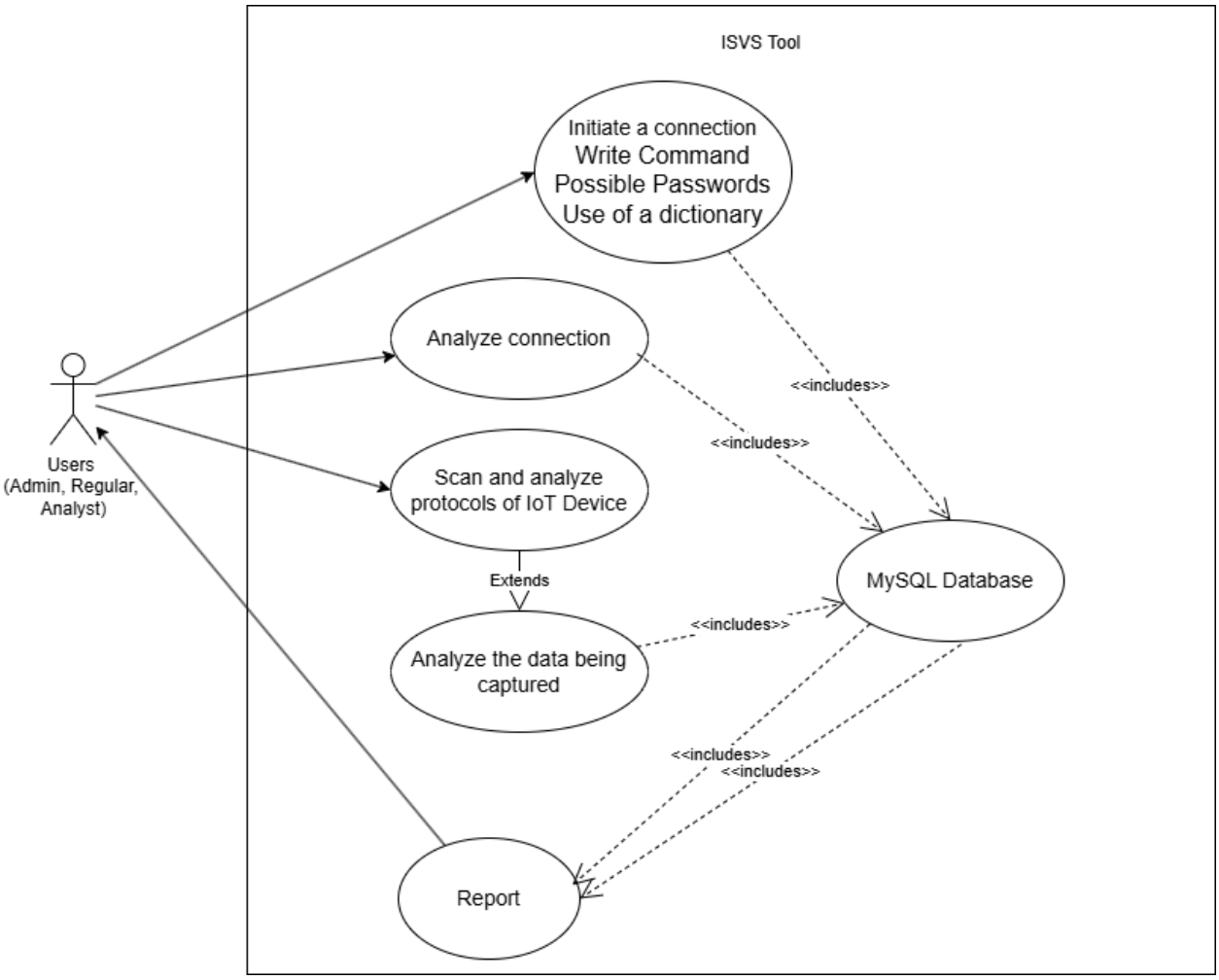


Figure 4.3: Use Case Diagram of Connecting to an existing Network and a scan and Analysis of an IoT Device

4.4.2 Flow Charts

A flowchart serves as a visual representation of a workflow or process, illustrating each step and the sequence in which they occur by linking them with arrows. The tool offers two key features: the ability to connect to an existing network and conduct a live scan of connected IoT devices. This is illustrated in figures 4.4.

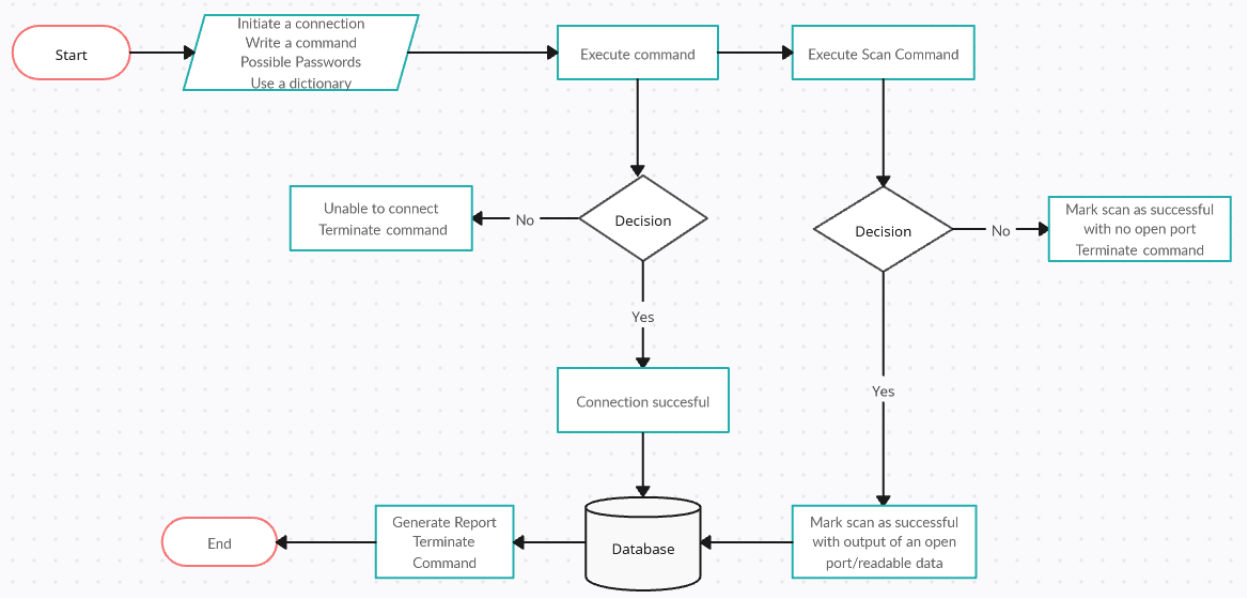


Figure 4.4: Connecting to an Existing Network and a Scan and Analysis Flow Chart

4.4.3 Class Diagram

This section outlines the system's structure by illustrating its classes, including their attributes, operations, and the relationships between different objects. In Figure 4.7, you can see the various classes along with their respective attributes and relationships clearly represented.

VT OMNES VNVM SINT

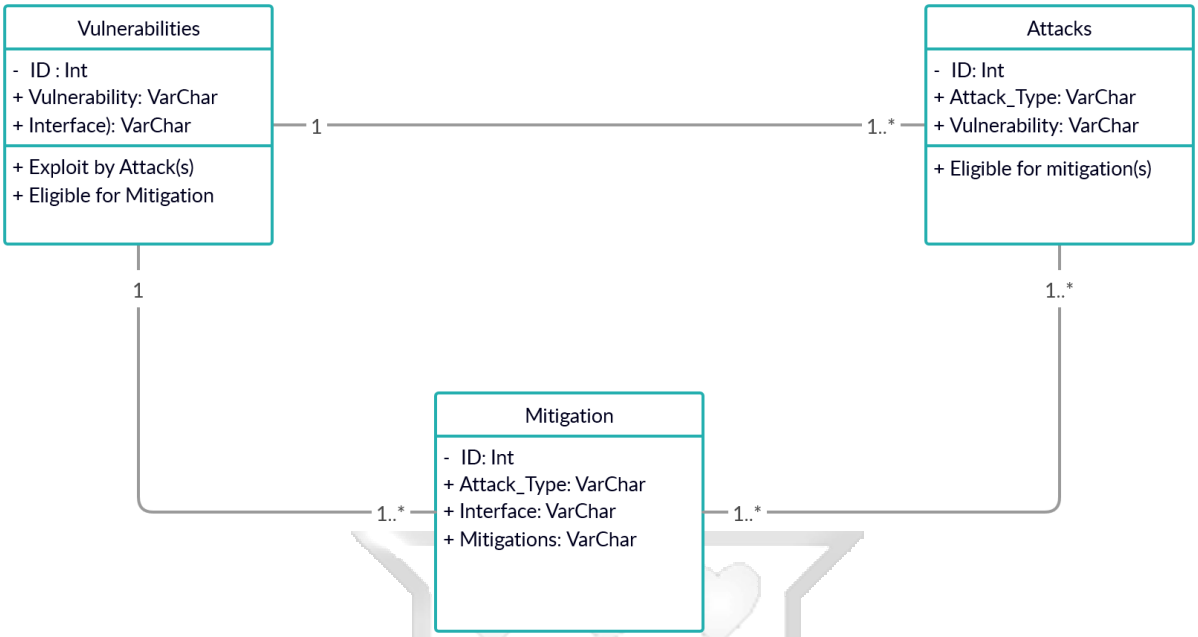


Figure 4.5: Class Diagram of the Tool

4.4.4 Database Schema

This is an abstraction used to characterize the storage of data in a database and the relationship between various tables in a database. Figure 4.8 shows the tables in the database and the respective relationships.

VT OMNES VNVM SINT

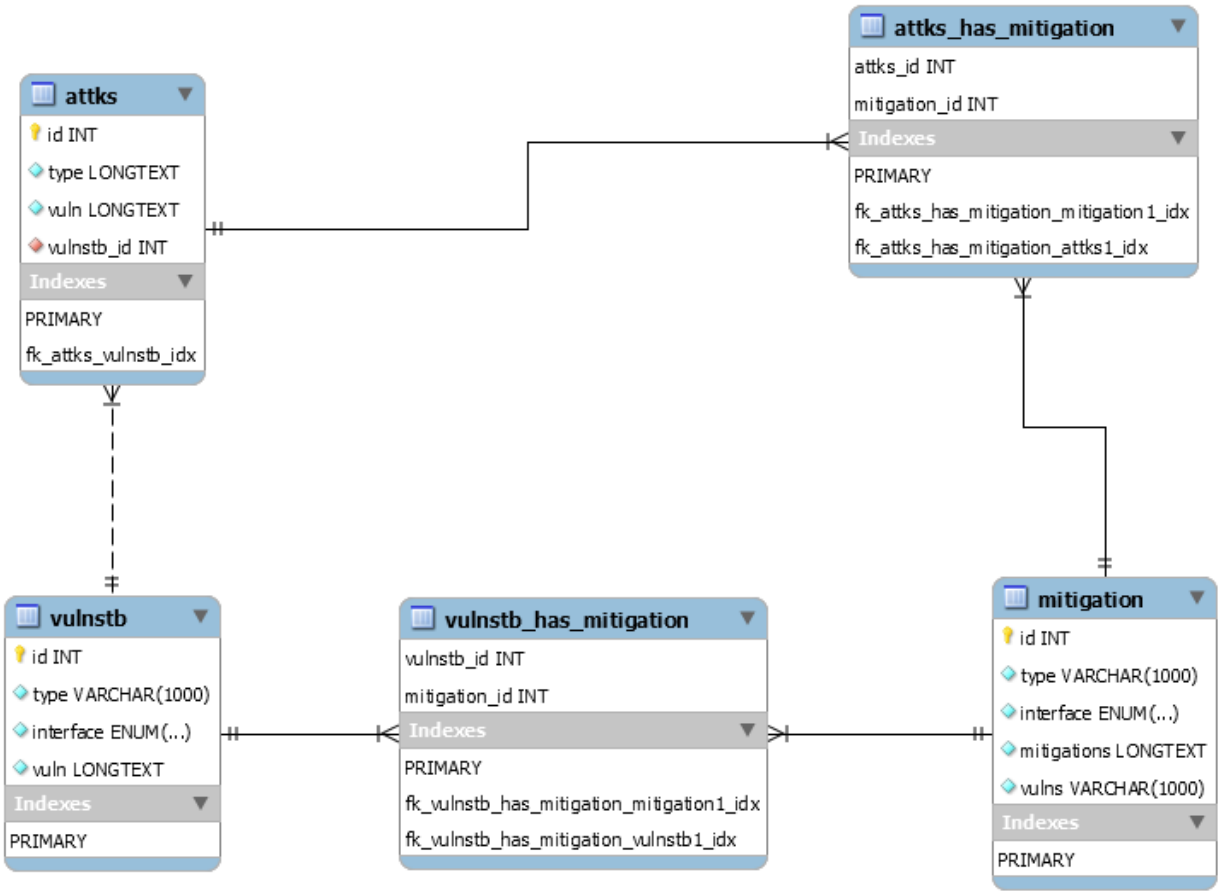


Figure 4.6: Database Schema

4.4.5 Sequence Diagram

A Sequence Diagram shows how a user inputs data into the system and how that information is then presented back to them. In Figures 4.7, you can see the sequence of actions involved in using the tool's functionality.

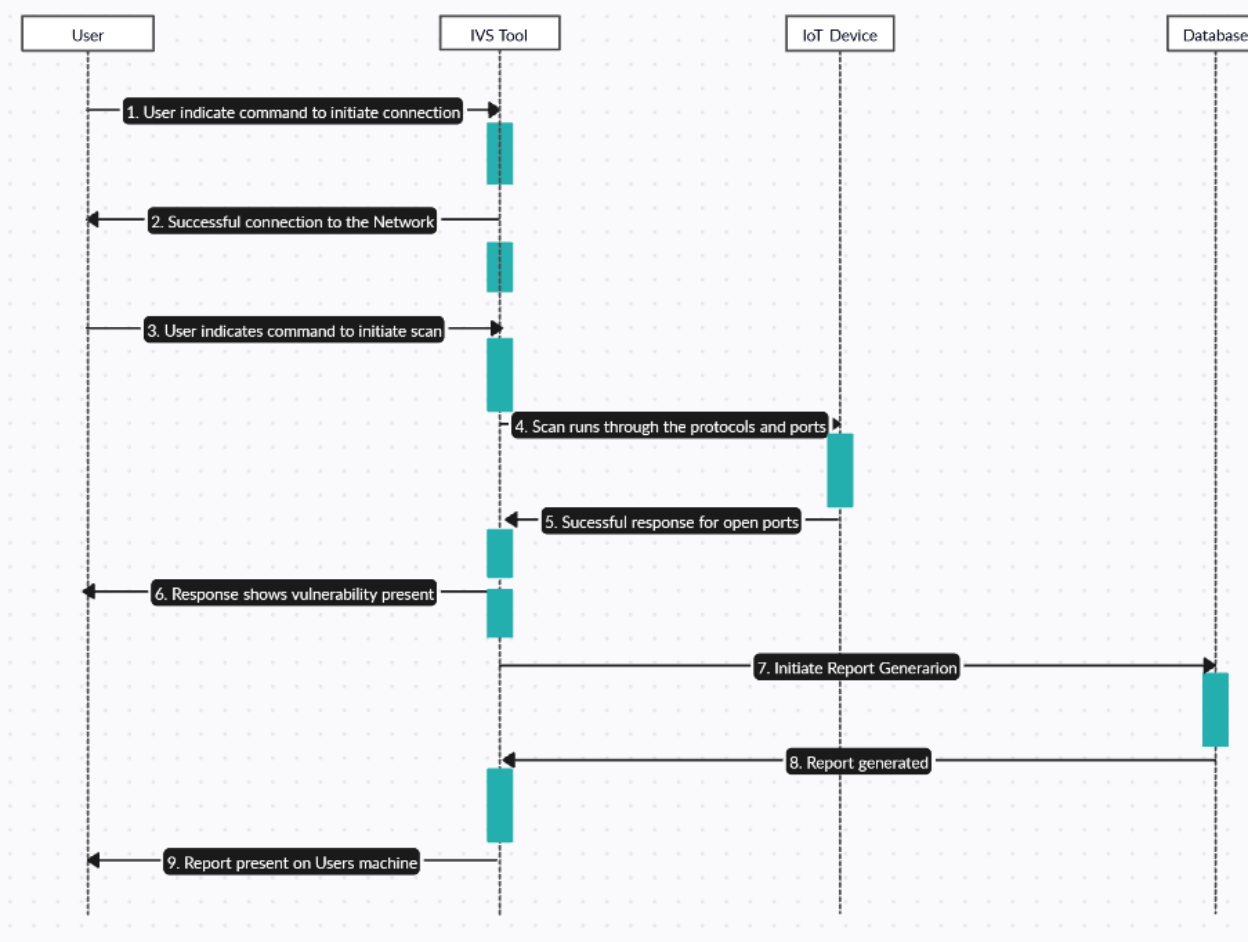


Figure 4.7: Connecting to an Existing Network and a Scan and Analysis Sequence Diagram

4.5 Network Design

In the development of this tool, having a running network is key. The network used was carefully analyzed and evaluated. It was important to ascertain whether a new network would be used or an existing one during the design phase. A new network was used for the experimentation stage.

The following were considered when looking at the design

i. Was the tool used for enterprises or private institutions?

ii. Was the tool incorporated in the use of remote site installations?

iii. Was it used in private homes?

Using the above as the scope, the design of the tool supports the use in private homes and some aspects of remote site installations making it usable in multiple places spanning of network connections that support both WI-FI and Bluetooth if an experienced and well-versed system administrator is present.

4.6 Security Design

The tool's primary goal is to provide adequate information that system administrators can act on upon a successful vulnerability scan. To achieve this, several security measures are implemented to protect both the tool's integrity and the data it processes.

A key aspect of the security design is ensuring the vulnerability database remains intact and cannot be accessed for unauthorized modifications, preventing misleading information. To achieve this, role-based access control (RBAC) is enforced, allowing only authorized personnel, such as system administrators, to manage and update the database.

To maintain secure connections, all communications between the scanning tool, IoT devices, and the system database are encrypted using TLS (Transport Layer Security). This prevents interception of sensitive data, such as device configurations or scan results.

Necessary configuration activities are conducted to ensure the consistency and security of the tool, ensuring that each scan produces reliable results. This includes documenting any changes made to the system.

By implementing these security controls, the tool ensures that both vulnerability data and scanning processes remain secure, reliable, and tamper-proof.

4.7 Wireframe

The developed tool uses a Command Line Interface (CLI) where no User interactive interface was present. This is a single step where upon a system administrator connecting to a network with the presence of IoT devices, a command is run that initiates a scan for the IoT devices connected in both the WI-FI and Bluetooth Networks that consequently generate a report that is automatically stored in the administrator’s machine.

There is an extra feature that shows an engine where the administrator can run a command to connect to a network without having knowledge of the password. If the connection is successful, then the administrator knows that this is a weak password that must be changed.

4.8 Conclusion

This chapter has discussed system design and architecture, outlining the necessary requirements for the tool to function effectively. It also demonstrated how the tool processes data flows and the resulting outcomes. It further showed how the tool was able to achieve the third study question of how the tool was designed and developed.

Chapter 5 : SYSTEM IMPLEMENTATION, TESTING AND VALIDATION

5.1 Introduction

This chapter describes how to implement the IoT Vulnerability Scanner prototype specifically designed for IoT devices. It also covers the testing methods used to ensure the scanner's functionality, performance, and security. The implementation process includes setting up the system components, integrating the vulnerability database, and confirming that the tool successfully identifies and reports security weaknesses in IoT devices.

Testing plays a vital role in the development process, ensuring that the scanner operates accurately, efficiently, and reliably. We employ a range of testing methods, particularly functional testing, to confirm that the system aligns with the expected requirements.

5.2 Implementation

The IoT Vulnerability Scanner (IVS) tool integrates the requirements and specifications identified during the system analysis phase to realize the system's design and architecture. The implementation process involved configuring the scanner’s components, integrating the vulnerability database, and ensuring compatibility with various IoT communication protocols.

To validate the tool’s functionality, four different IoT devices were used in the experiment:

i. **Wi-Fi Connector Sensor** – This device was used to simulate IoT devices that rely on Wi-Fi networks for communication. It helped test the scanner’s ability to detect open ports, weak encryption, and unsecured network transmissions within Wi-Fi-enabled IoT devices.

ii. **Bluetooth Connector Sensor** – This device allowed for testing the scanner’s capability in identifying vulnerabilities in Bluetooth-connected IoT devices. The experiment focused on detecting unauthorized access risks, and outdated Bluetooth security protocols.

iii. **Barometer Sensor** – This sensor was integrated into the test environment to analyze how the scanner interacts with industrial IoT (IIoT) devices. The scanner was tested for its ability to identify firmware vulnerabilities and insecure data transmissions in environmental monitoring sensors.

iv. **Temperature Sensor** – This IoT device was included to assess how the scanner handles low-power, embedded IoT systems. It helped evaluate whether the scanner could detect vulnerabilities such as default credentials and weak data encryption methods.

By utilizing these four devices, the IVS tool was tested across the different IoT communication protocols (Wi-Fi and Bluetooth). This ensured a comprehensive evaluation of the scanner's effectiveness in identifying security weaknesses across multiple IoT environments. Figure 5.1 shows the devices used in the experiment.

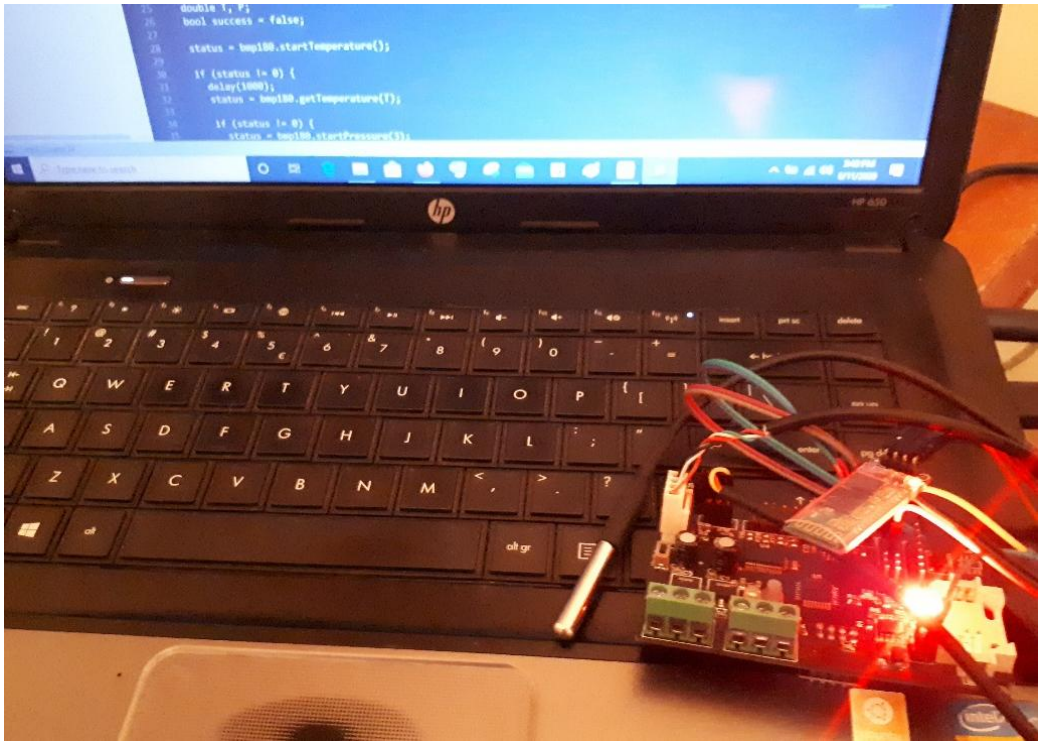


Figure 5.1: Devices used in the Design of the solution

The vulnerability database was generated and stored locally in a MySQL database through study while determining the common vulnerabilities in comparison with the online common vulnerabilities exposures (CVE) database, the attacks that could be used and the mitigation factors. As an additional component, the consequences upon exploitation of vulnerabilities were also added.

5.2.1 Hardware Requirements

i. Machine: Intel Dual Core or higher

ii. System Memory: 2 GB or higher

iii. Microcontroller

iv. Barometer Sensor

v. Temperature sensor

vi. Wi-Fi and Bluetooth Modules

vii. WI-FI Routers or Access Points

5.2.2 Software Requirements

i. Operating Systems: Windows 11/10

ii. MySQL Database

iii. Text Editor i.e. Sublime text

iv. Back-end: Python

v. Bluetooth Command Line Tools

5.2.3 Network Requirements

i. WI-FI Networks

ii. Bluetooth Network

iii. Routing capability

The tool has three modules; Connect, Scan and analyze and Report modules. Figure 5.2 shows the command line interface with the possible arguments when initiating a command.

```
C:\Users\Edwin\Documents\Dissertation\Disser Python Project\Project-Python 3.7\proj>python grave_walker.py
C:\Users\Edwin\AppData\Roaming\Python\Python37\site-packages\pyhanko_certvalidator\errors.py:6: CryptographyDe
eprecated in cryptography. A future release of cryptography will remove support for Python 3.7.
  from cryptography.exceptions import InvalidSignature
usage: grave_walker.py [-h] -i INTERFACE [-swf SWF] [-saw SAW] [-sbt SBT]
                       [-sab SAB] [-p PWDS] [-cf CRACKFILE] [-gp GPL]
                       [-pt PSTYP] [-wfn WIFINAME] [-btn BLUETOOTHNAME]

optional arguments:
  -h, --help            show this help message and exit
  -i INTERFACE, --interface INTERFACE
                        interface to crack : bt -> bluetooth or wf -> wi-fi
  -swf SWF, --show-all-wifi SWF
                        show all wifi ever connected, used as [-swf *]
  -saw SAW, --show-active-wifi SAW
                        show available wifi [-saw *]
  -sbt SBT, --show-all-paired SBT
                        show all bluetooth paired devices [-sbt *]
  -sab SAB, --show-active-bluetooh SAB
                        show available bluetooth devices [-sab *]
  -p PWDS, --pwd PWDS   password to crack current interface
  -cf CRACKFILE, --crack-file CRACKFILE
                        file with passwords
  -gp GPL, --generate-passwords GPL
                        length of passwords to be generated
  -pt PSTYP, --passwords-type PSTYP
                        password to be generated : alpha, digits or * for both
                        alpha and digits
  -wfn WIFINAME, --wifi-name WIFINAME
                        name of wifi or * for current available unconnected
                        wireless network to crack
  -btn BLUETOOTHNAME, --bluetooth-name BLUETOOTHNAME
                        name of bluetooth or * for all currently unpaired
                        available devices to crack
```

Figure 5.2: Command with possible arguments towards establishing a connection

Towards the discovery of the vulnerabilities in the connected IoT Devices, the detection component of the tool was configured in such a way that it had to make sure that it has established a connection to the two networks, that are WI-FI and Bluetooth. This was to ensure that where the tool is being run from, it belonged to the same network that the IoT devices are connected to before proceeding to initiate the scan that in turn discovers the vulnerabilities.

In the discovery of these vulnerabilities present in the connected IoT devices, the tool performed relatively well and gave at least one result upon completion of the scan in the connected devices. Upon this successful scan that gave the result as a positive identification of the vulnerability upon a further cross-reference to the vulnerability database, the report is generated in pdf format and automatically stored in the machine the scan was initiated from.

Figure 5.3 shows the engine that was used for an attacker to potentially gain access to WI-FI or Bluetooth network.

```
if interface=='wf' and wifi_name:
    print '\n\t'
    if wifi_name=='*':
        cracked_wfs=[]
        if len(all_passwords)==0:
            print 'provide password(s) eg -p asdf -p ccccc or\n -cp aaa.txt -cp bbb.c --> load passwords from fi
            sys.exit(0)
        cw=show_active_networks()
        aln=list_connected_wifi()
        to_crk=[]
        for i in cw:
            if i not in aln:
                to_crk.append(i)
            else:
                pass
        if len(to_crk)==0:
            print 'You have logged in all the current networks Fix ->: go to settings;wi-fi;Manage known network
        for wifi2crk in to_crk:
            for password in all_passwords:
                print '\n>>Preparing User and password'
                create_profile(wifi2crk,password)
                add_profile(wifi_profile_xml)
                connect_to_wifi(wifi2crk)
                print '>>Cracking %s with password %s'%(wifi2crk,password)
                if check_new_wifi_interface():
                    print 'wifi_cracked  :-)'
                    print 'password : %s'%(password)
                    cracked_wfs.append(password)
```

Figure 5.3: Python Engine to Cracking the WI-FI network

5.2.4 Populating Vulnerability Database

To make the IoT Vulnerability Scanner (IVS) tool effective, it needs a comprehensive and up-to-date database of known security weaknesses in IoT devices. This database acts as a reference point for identifying vulnerabilities during scans. The database was built using a trusted cyber security source that is Common Vulnerabilities and Exposures (CVE) database which is a public repository that tracks reported security flaws in IoT devices. Once the vulnerabilities were collected, they were organized and categorized based on factors such as:

i. Vulnerability type (e.g., weak passwords, open ports, outdated firmware).

ii. Potential exploits that hackers could use to take advantage of the vulnerability.

iii. Recommended fixes, such as software patches or security settings adjustments.

To keep the database current, this is currently possible through manually adding new vulnerabilities when needed.

5.2.5 Generating Attacks

To test the effectiveness of the IoT Vulnerability Scanner (IVS) tool, different types of cyber-attacks were simulated. These controlled tests helped determine whether the scanner could detect, analyze, and report security weaknesses in IoT devices. The following attack techniques were used:

i. Brute Force Attacks – This method attempts to guess passwords by systematically trying different combinations until the correct one is found. The test involved attacking IoT devices with default or weak passwords to check if the scanner could identify poor authentication practices and recommend stronger security measures.

ii. Encoding/Decoding Attacks – Some IoT devices store or transmit sensitive information in an encoded format rather than encrypting it. This attack tested whether the scanner could detect data that was poorly protected, making it possible for attackers to decode and read sensitive information. The tool was expected to flag weak or improperly implemented encryption methods that could expose private data.

iii. Port Scanning Attacks – This technique scans IoT devices to identify open network ports that could be exploited by attackers. The test involved mapping which ports were active, analyzing whether they were properly secured, and determining if any unnecessary or high-risk ports were left exposed. The scanner needed to correctly detect these open ports and provide recommendations on closing or securing them to prevent unauthorized access.

By simulating these attacks in a controlled environment, the IVS tool was able to demonstrate its ability to identify security risks and recommend mitigation strategies.

5.3 System Modules/Key Functions

The IoT Vulnerability Scanner (IVS) tool is made up of different modules, each responsible for a specific function. These modules work together to detect security weaknesses in IoT devices, analyze them, and provide security recommendations. Below are the key modules and their functions:

i. Network and device scanning module for identifying all IoT devices connected to the network via Wi-Fi or Bluetooth and utilizing port scanning to find open or exposed communication channels that could be vulnerable to attacks.
ii. Vulnerability detection module to cross-check IoT devices against a database of known security flaws for example, weak passwords or open ports and uses pattern matching to detect security risks.
iii. Threat analysis module to maps vulnerabilities to potential cyber threats, helping users understand how an attacker might exploit them.
iv. Report generation module to generate a detailed security report listing all detected vulnerabilities, their risks, and step-by-step mitigation strategies. Additionally, it allows users to export reports in PDF format for further review.
v. Database management module to store known vulnerabilities and the respective mitigation strategies.

5.4 Testing

The tool was assessed to evaluate its functionality and identify any potential bugs. This testing was conducted systematically, covering every feature of the tool to confirm that it operates as intended. Below are the specific functionalities that were tested.

i. Establish a WI-FI and Bluetooth Connection
ii. Scan and analyze the Barometer and/or the Temperature sensors connected through a WI-FI connection
iii. Scan and analyze the Barometer and/or the Temperature sensors connected through a Bluetooth connection
iv. Report Generation of scans conducted.

The above was addressed in the functional testing section. Additionally, each device was pre-configured with known vulnerabilities to create a controlled environment for testing.

5.4.1 Functional Testing

To verify that the scanner effectively identifies vulnerabilities and produces reports, functional testing was carried out.

5.4.1.1 Establishing Wi-Fi and Bluetooth Connections

The respective users were able to run a command that was able to show case how an attacker can attempt to connect to a network. As shown in Figure 5.4, a successful connection attempt by an attacker through running a command on the terminal to a WI-FI Connection. Successful and unsuccessful connection attempts were recorded and analyzed and brute-force password attempts using dictionary attacks were performed to check if the scanner could detect repeated failed authentication attempts.

Figure 5.4: Successful connection to WI-FI Network through command line

The illustration on Figure 5.5 below shows the result of an unsuccessful attempt to connect to a WI-FI connection.

Figure 5.5: Unsuccessful attempt to connect to a WI-FI Network

An illustration on Figure 5.6 shows the use of a dictionary to attempt to gain access to a WI-FI connection. Due to the number of passwords in the dictionary, this takes a very long time.

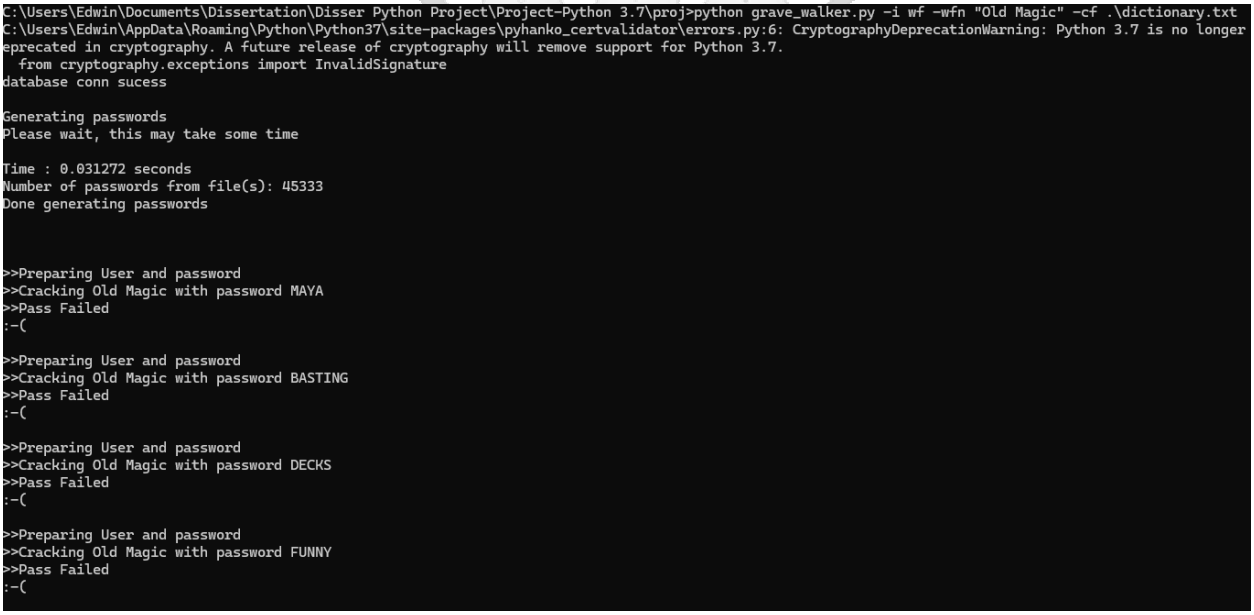


Figure 5.6: Use of a dictionary to gain access to a WI-FI Network

The illustration on Figure 5.7 below shows a successful connection attempt by an attacker through running a command on the terminal to a Bluetooth Connection

```
C:\Users\Edwin\Documents\Dissertation\Disser Python Project\Project-Python 3.7\proj>python grave_walker.py -i bt -btn "HC-06" -p asdfghfdsd -p gdsdfgf -p 1234
C:\Users\Edwin\AppData\Roaming\Python\Python37\site-packages\pyhanko_certvalidator\errors.py:6: CryptographyDeprecationWarning: Python 3.7 is no longer supported
eprecated in cryptography. A future release of cryptography will remove support for Python 3.7.
  from cryptography.exceptions import InvalidSignature
database conn sucess
>>Adding passwords provided in the terminal
>>Done adding passwords provided in the terminal


>>cracking dev: HC-06 using : asdfghfdsd
>>Pass Failed
:-(
>>cracking dev: HC-06 using : gdsdfgf
>>Pass Failed
:-(
>>cracking dev: HC-06 using : 1234
bluetooth cracked  :-)
password : 1234
Reparing Report init
pdf name: HC-06_2025-03-3121_17_02_pair_report.pdf
Reparing Report Complete
```

Figure 5.7: Successful connection to a Bluetooth Connection

The illustration on Figure 5.8 below shows an unsuccessful attempt to connect to a Bluetooth connection

```
C:\Users\Edwin\Documents\Dissertation\Disser Python Project\Project-Python 3.7\proj>python grave_walker.py -i bt -btn "HC-06" -p asdfghfdsd -p gdsdfgf -p 1234
C:\Users\Edwin\AppData\Roaming\Python\Python37\site-packages\pyhanko_certvalidator\errors.py:6: CryptographyDeprecationWarning: Python 3.7 is no longer supported
eprecated in cryptography. A future release of cryptography will remove support for Python 3.7.
  from cryptography.exceptions import InvalidSignature
database conn sucess
>>Adding passwords provided in the terminal
>>Done adding passwords provided in the terminal


>>cracking dev: HC-06 using : asdfghfdsd
>>Pass Failed
:-(
>>cracking dev: HC-06 using : gdsdfgf
>>Pass Failed
:-(
```

Figure 5.8 : Unsuccessful attempt to connect to a Bluetooth Connection

The illustration on Figure 5.9 below shows an attempt to connect using the password in a dictionary to a Bluetooth connection. Due to the large number of passwords, this would take a considerable amount of time.

```
C:\Users\Edwin\Documents\Dissertation\Disser Python Project\Project-Python 3.7\proj>python grave_walker.py -i bt -btn "HC-06" -cf .\dictionary.txt
C:\Users\Edwin\AppData\Roaming\Python\Python37\site-packages\pyhanko_certvalidator\errors.py:6: CryptographyDeprecationWarning: Python 3.7 is no longer su
eprecated in cryptography. A future release of cryptography will remove support for Python 3.7.
  from cryptography.exceptions import InvalidSignature
database conn sucess

Generating passwords
Please wait, this may take some time

Time : 0.015996 seconds
Number of passwords from file(s): 45333
Done generating passwords


>>cracking dev: HC-06 using : BLOCKADES
>>Pass Failed
:-(
>>cracking dev: HC-06 using : POINTERS
>>Pass Failed
:-(
>>cracking dev: HC-06 using : ORR
>>Pass Failed
:-(
>>cracking dev: HC-06 using : BLANTON
>>Pass Failed
:-(
```

Figure 5.9: Attempt to use a dictionary to gain access to a Bluetooth Connection

5.4.1.2 Scanning and Analyzing IoT Devices

The scanner was used to identify open ports and weak encryption on IoT devices and data transmissions from connected devices were inspected for unencrypted data that could be intercepted. As shown in Figure 5.10, a successful scan for no open ports that can be utilized for an attack

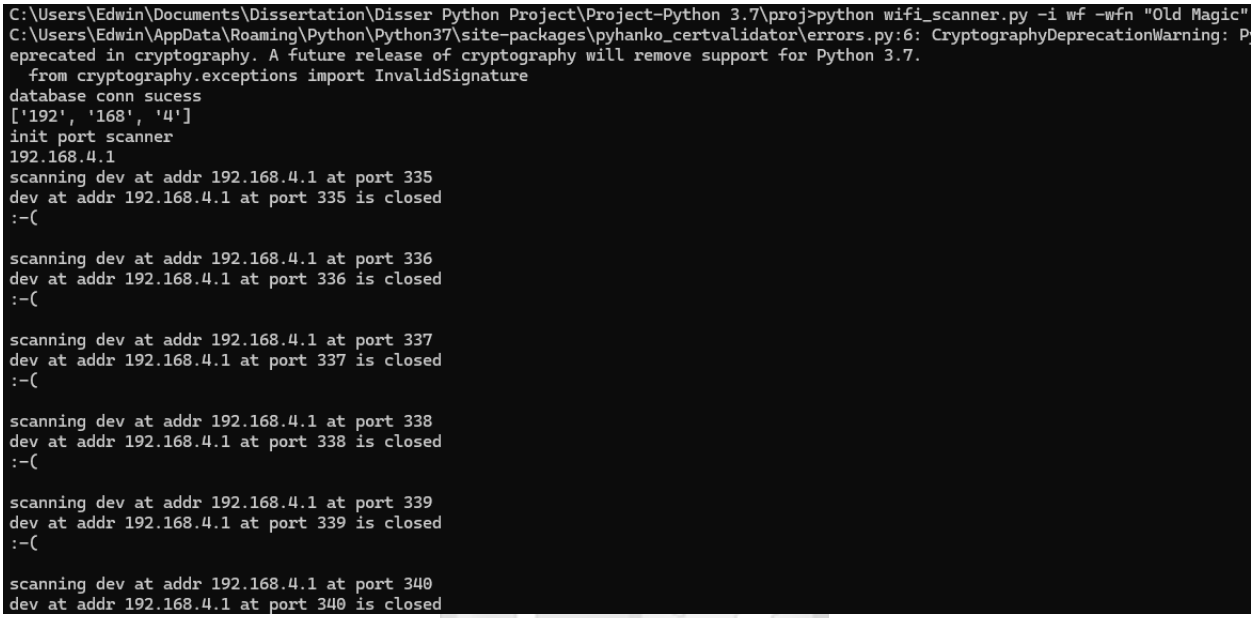


Figure 5.10: A scan that found no open ports in a WI-FI Connection

The illustration on Figure 5.11 below shows a successful scan for an open port in the WI-FI Network is currently connected to an IoT Device.

```
C:\Users\Edwin\Documents\Dissertation\Disser Python Project\Project-Python 3.7\proj>python wifi_scanner.py -i wf -wfn "Old Magic"
C:\Users\Edwin\AppData\Roaming\Python\Python37\site-packages\pyhanko_certvalidator\errors.py:6: CryptographyDeprecationWarning: Python 3.7 is
eprecated in cryptography. A future release of cryptography will remove support for Python 3.7.
  from cryptography.exceptions import InvalidSignature
database conn sucess
['192', '168', '4']
init port scanner
192.168.4.1
scanning dev at addr 192.168.4.1 at port 330
dev at addr 192.168.4.1 at port 330 is closed
:-(

scanning dev at addr 192.168.4.1 at port 331
dev at addr 192.168.4.1 at port 331 is closed
:-(

scanning dev at addr 192.168.4.1 at port 332
dev at addr 192.168.4.1 at port 332 is closed
:-(

scanning dev at addr 192.168.4.1 at port 333
>>connected: 192.168.4.1 port 333
open port found
pdf name: open_port_333_192.168.4.1_2025-03-31_21_25_30_report.pdf
Reparing Report Complete
```

Figure 5.11: Successful scan to an open port on a WI-FI connection

The illustration on Figure 5.12 below shows the connected IoT Device that is a temperature sensor that shows the data being received is unencrypted. Two attempts are made to determine if the data being received is in readable format.

```
The temperature is : 23.63
gold
unencrypted data
pdf name: 192.168.4.1_2025-03-31_21_25_43_wifi_encdec_report.pdf
Reparing Report Complete
The temperature is : 23.56
```

Figure 5.12: Captured data of a Temperature IoT Sensor

The illustration on Figure 5.13 below shows an unsuccessful scan to determine any ports that can be used for an attack through a Bluetooth connection.

```
C:\Users\Edwin\Documents\Dissertation\Disser Python Project\Project-Python 3.7\proj>python bt_scanner.py
C:\Users\Edwin\AppData\Roaming\Python\Python37\site-packages\pyhanko_certvalidator\errors.py:6: CryptographyDep
eprecated in cryptography. A future release of cryptography will remove support for Python 3.7.
  from cryptography.exceptions import InvalidSignature
database conn sucess
scanner init
Discovered Devices: ('98:D3:91:FD:84:A0', 'HC-06')
bt dev HC-06 addr 98:D3:91:FD:84:A0 port 1 closed
bt dev HC-06 addr 98:D3:91:FD:84:A0 port 2 closed
bt dev HC-06 addr 98:D3:91:FD:84:A0 port 3 closed
bt dev HC-06 addr 98:D3:91:FD:84:A0 port 4 closed
bt dev HC-06 addr 98:D3:91:FD:84:A0 port 5 closed
bt dev HC-06 addr 98:D3:91:FD:84:A0 port 6 closed
bt dev HC-06 addr 98:D3:91:FD:84:A0 port 7 closed
bt dev HC-06 addr 98:D3:91:FD:84:A0 port 8 closed
bt dev HC-06 addr 98:D3:91:FD:84:A0 port 9 closed
bt dev HC-06 addr 98:D3:91:FD:84:A0 port 10 closed
bt dev HC-06 addr 98:D3:91:FD:84:A0 port 11 closed
bt dev HC-06 addr 98:D3:91:FD:84:A0 port 12 closed
bt dev HC-06 addr 98:D3:91:FD:84:A0 port 13 closed
```

Figure 5.13: Results of a scan conducted on a Bluetooth connection

5.4.1.3 Report Generation

After each scan, a detailed security report was generated in PDF format. Reports included identified vulnerabilities, their severity, potential risks, and recommended mitigation strategies. Once a successful connection is initiated through the terminal, a pdf report is generated as highlighted in Figure 5.14.

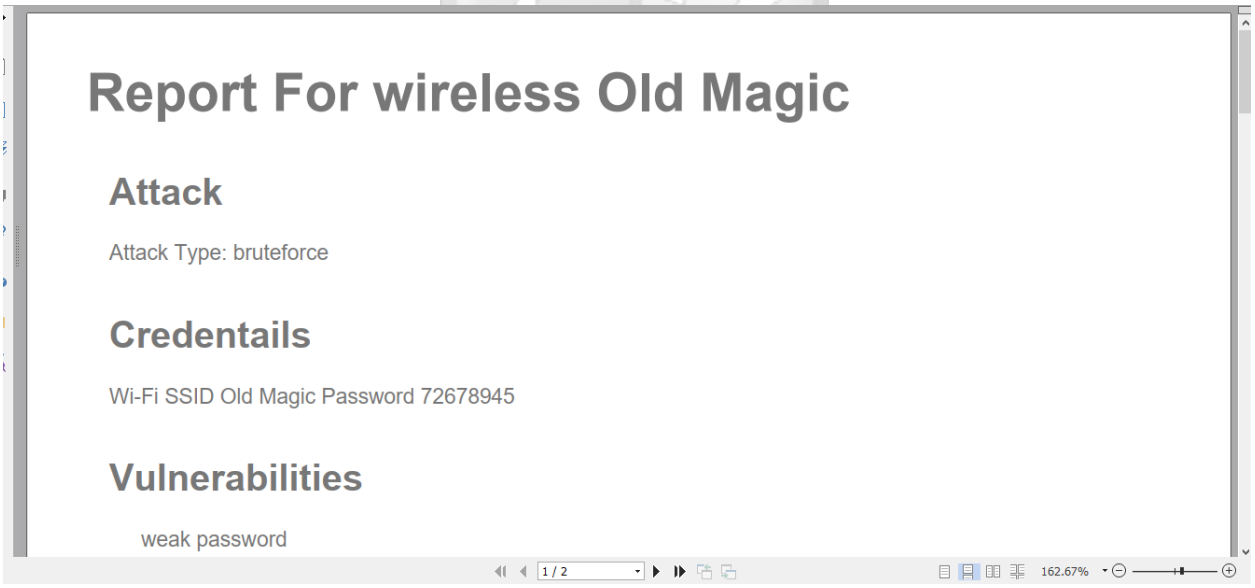


Figure 5.14: Report on a successful connection to a WI-FI Network

The illustration in Figure 5.15 below shows the report generated in pdf on the successful connection attempt using the command line interface for a Bluetooth connection

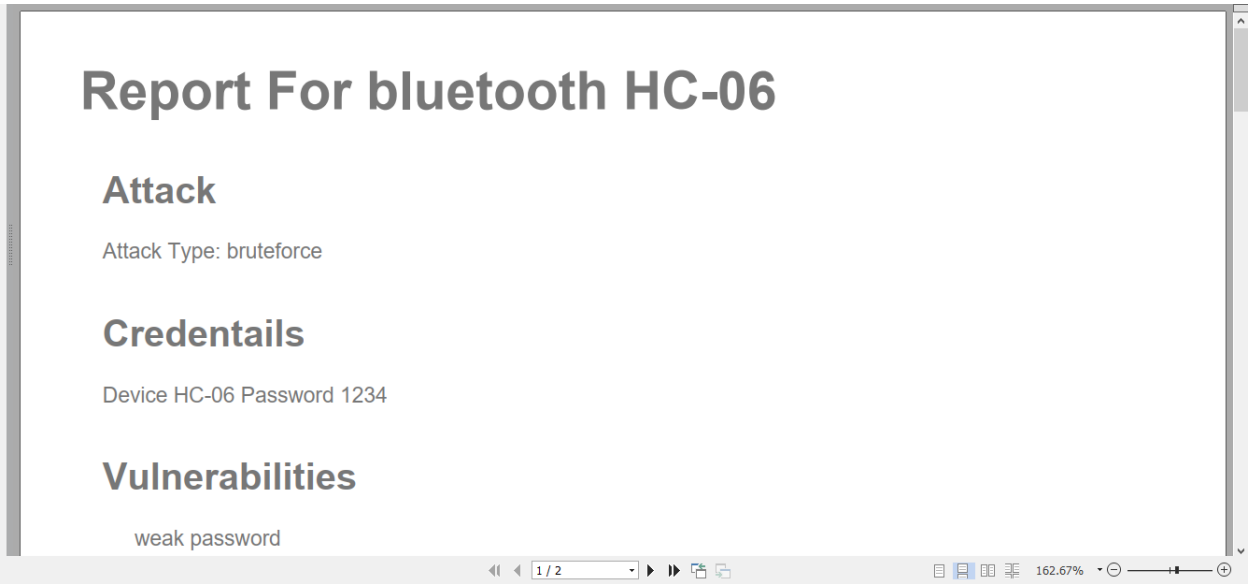


Figure 5.15: Report on a successful attempt to a Bluetooth Connection

The illustration on Figure 5.16 below shows the report generated based on the scan conducted

Figure 5.16: Report generated for a Successful scan with an open port

The illustration on Figure 5.17 below shows the report generated for the IoT Device scanned. It is connected through port 333 through the IP 192.168.4.1

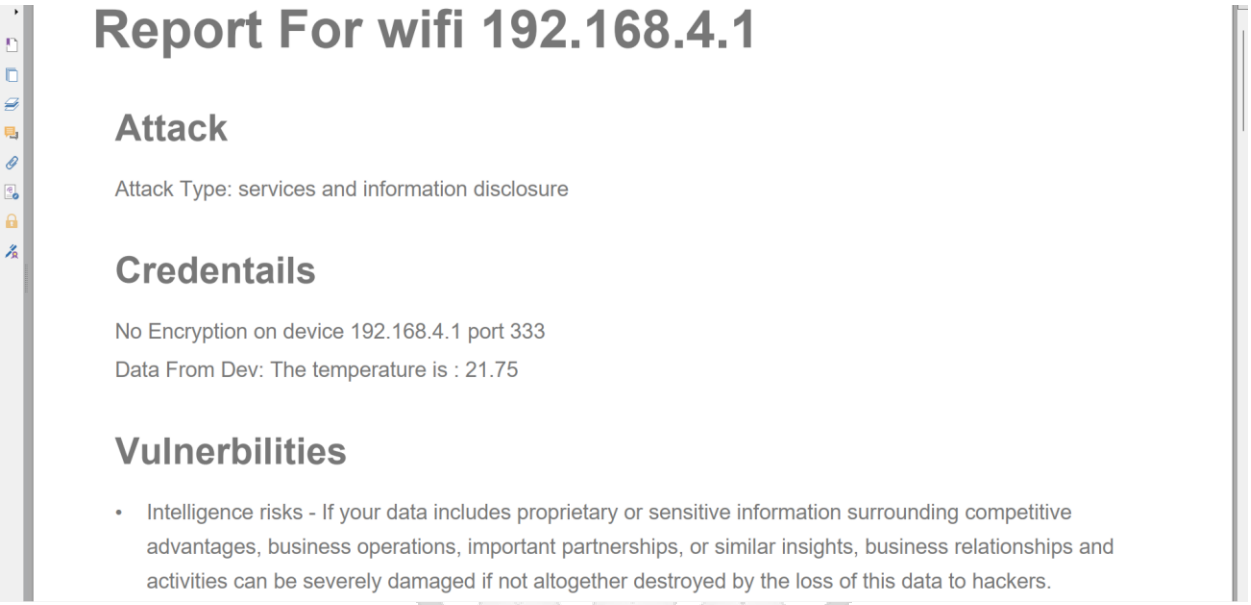


Figure 5.17: Report shows No encryption on data captured from Temperature Sensor

The illustration on Figure 5.18 below shows the report generated due to an open port in the Bluetooth connection

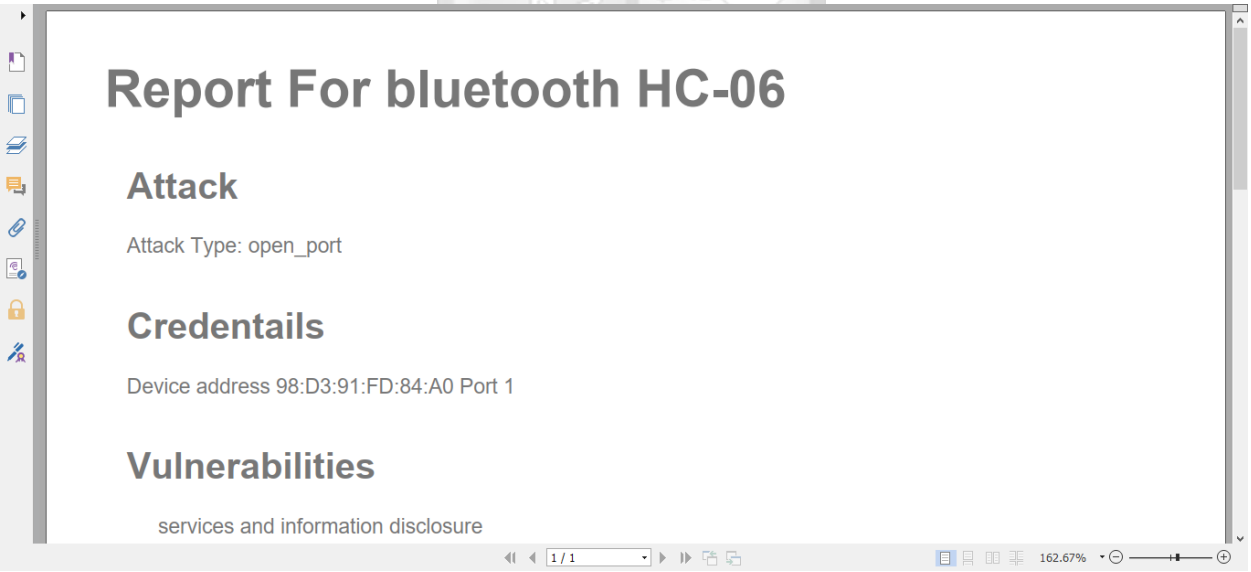


Figure 5.18: Report on Scan due to an open port in the Bluetooth Connection

The illustration on Figure 5.19 below shows the report generated due to the unencrypted data from the Barometer

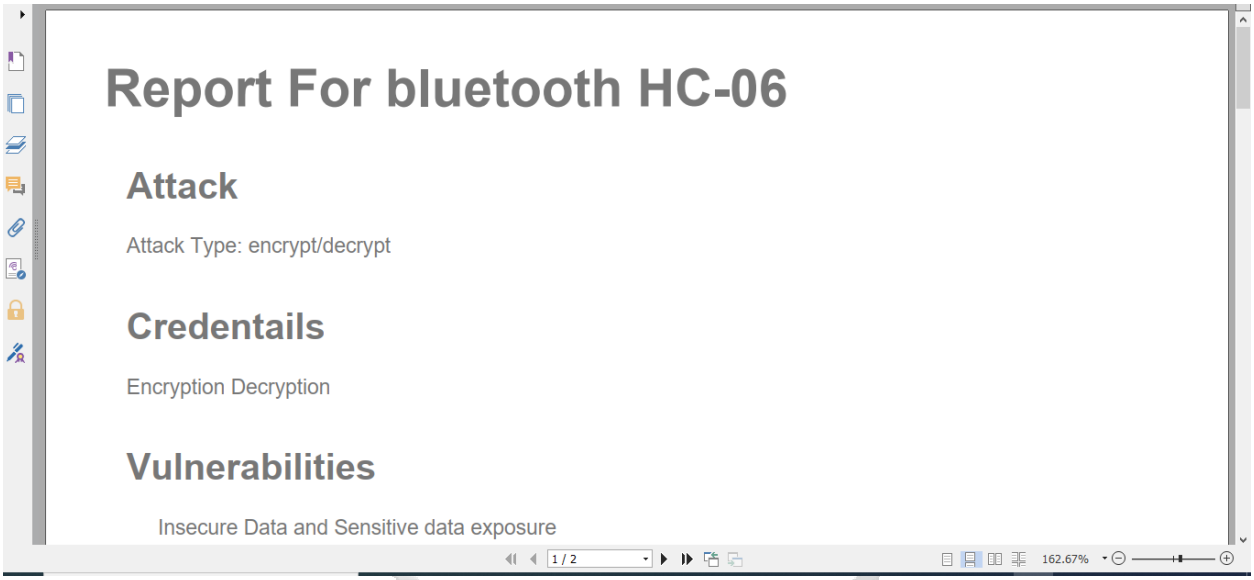


Figure 5.19: Report generated due to unencrypted data from the Barometer sensor

5.4.2 Vulnerability Detection and Accuracy Rate

Each of the four IoT devices (Wi-Fi, Bluetooth, Barometer, Temperature sensors) was manually configured with known vulnerabilities. These included:

i. Open network ports that could be exploited.
ii. Default credentials that attackers could use for unauthorized access.
iii. Unencrypted data transmissions.
iv. Weak Bluetooth pairing mechanisms.

After the scan was conducted, the results were compared through manual verification of the known vulnerabilities in the devices. A total of 4 distinct vulnerabilities were intentionally introduced across the test devices. After performing the scans using the IVS tool, the results were manually verified against the known, pre-configured vulnerabilities. The scanner successfully detected all 4 vulnerabilities, achieving an accuracy rate of 100%.

The accuracy was calculated using the following formula:

i. Accuracy Rate = Number of Correctly Detected Vulnerabilities / Total Number of Known Vulnerabilities × 100

ii. Accuracy Rate = 4/4 × 100 = 100%

This demonstrates that, within the controlled environment, the IVS tool was fully effective at identifying the targeted vulnerabilities. While this result is promising, further testing in a more diverse and real-world IoT environments is recommended to assess performance under varying conditions and with unknown vulnerabilities.

5.5 Prototype Validation

This refers to the collection of processes and activities designed to ensure that models are functioning as intended, aligning with the prototype design goals and business applications. The primary aim of this initiative was to create a tool for scanning IoT vulnerabilities. The tool was able to identify the three most common vulnerabilities that are present in most IoT devices relating to ports and protocols. These include open ports, weak passwords and unencrypted data.

By Connecting to a WI-FI or Bluetooth connection the scan analyzes the ports and protocols of the connected devices. Analysis is performed by comparison of the discovered vulnerabilities with a Vulnerability Database. Lastly the tool generates reports in pdf format for analysis done.

Key to a vulnerability management program, the results must be those that identify vulnerabilities, evaluate vulnerabilities, treat vulnerabilities and report vulnerabilities. (Rapid7, 2020)

To verify that the developed tool was free of bugs and errors, comprehensive testing was conducted. This involved performing multiple tests repeatedly to ensure consistent results were achieved.

i. Vulnerability identification

ii. To identify the type of attack that can be used to exploit a vulnerability.

iii. To achieve consistent results when applying the same method to identical test items across various testing environments, it's essential to install the tool on different machines.

iv. To provide remediation and mitigation to resolving issues related to the identified vulnerabilities.

An IoT Scanner tool called IoTSeeker was used to identify vulnerabilities in connected devices within the network. The results of this scan were visually compared, as illustrated in figures 5.20 and 5.21.

Figure 5.20: Results of scan conducted by IoT Seeker scanner tool

Figure 5.20 shows the tool IoT Seeker Scanner Tool that looked at 20 IoT devices to see if any were still using their original passwords, which is a big security issue. Out of those 20 devices, 3 were still on the default settings:

i. 1 Stardot device of IP 1.23.123.431
ii. 1 W-Box device of IP 1.23.123.477
iii. 1 American Dynamics device of IP 1.23.123.481

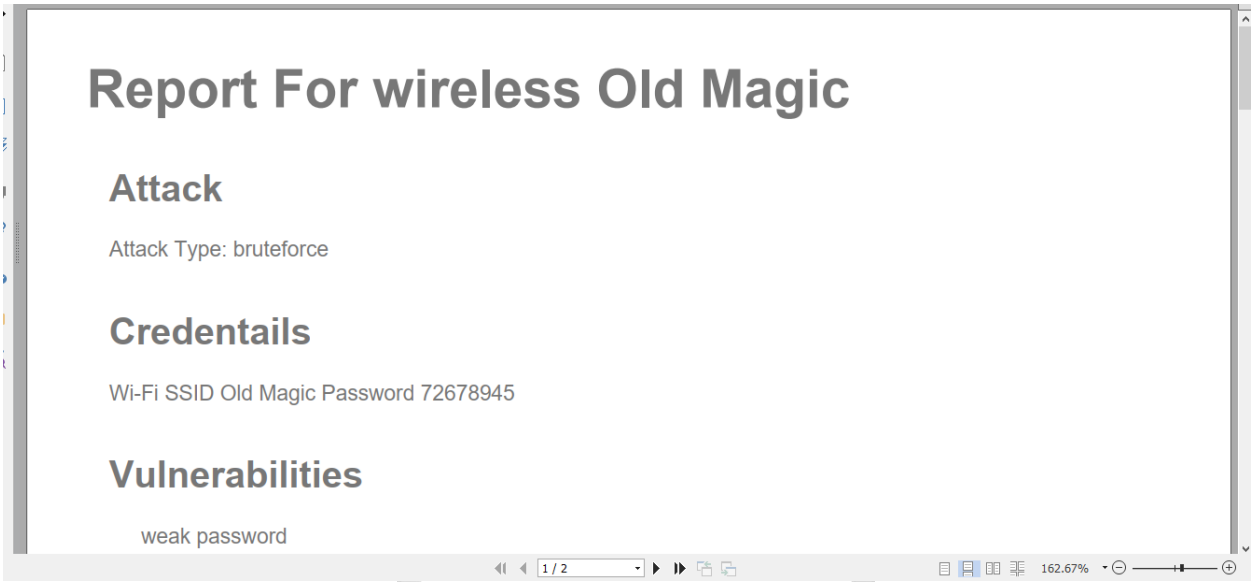


Figure 5.21: Report generated by IVS Tool

Figure 5.21 shows the Prototype IVS Tool that generated a downloadable report showing the attack and the vulnerability present. In comparison to the results in Figure 5.20, the IoT Seeker Scanner Tool did not have the capability to download a report with the results of the scan.

5.6 User Feedback

User feedback plays a crucial role in assessing whether the developed tool aligns with user requirements and effectively addresses user needs. This assessment was carried out through a peer review process, where three experts in system development evaluated the solution by interacting with the IoT devices on the isolated environment utilizing their individual machines. The feedback provided by the experts includes:

i. The core functionality of the tool in the discovery of vulnerabilities was achieved.
ii. When compared to other tools, most were provided by major companies that have focus on discovery and recommendations to solve the vulnerability concerns and where IVS tool does not 100% match with what is out there, its capabilities support more connections, and the reporting is done in a way it is readily available to the administrator.
iii. The tool contained a few bugs where a few errors were discovered upon initiating a scan but not readily available for review when the scan was initiated once more.

iv. All three experts were able to verify that with this isolated environment, on the Wi-Fi connection, the weak password was accessed through brute-force attack and the unencrypted data from the temperature could be seen.

v. Two out of the three experts were unable to ascertain the transmission of unencrypted data from the barometer through the Bluetooth connection.

vi. Improvements can be made to the tool, to make it more customer intuitive, adding to the present capabilities seen.

Table 5.1: Summary of whether Functional Requirements were achieved

Functional Requirements	**Expert 1**	**Expert 2**	**Expert 3**	**Achieved?**
Scan an IoT Device(s) connected through a WI-FI connection	Yes	Yes	Yes	Yes
View data from connected IoT device(s) through WI-FI connection	Yes	Yes	Yes	Yes
Ability to identify vulnerabilities in IoT Devices connected through WI-FI connection	Yes	Yes	Yes	Yes
Scan an IoT Device(s) connected through a Bluetooth connection	Yes	No	No	Yes, with intermittent connection
View data from connected IoT device(s) through Bluetooth connection	Yes	No	No	Yes, with intermittent connection

Ability to identify vulnerabilities in IoT Devices connected through Bluetooth connection	Yes	No	No	Yes, with intermittent connection
Generate reports for analysis/scan conducted	Yes	Yes	Yes	Yes

5.7 Prototype Verification

Verification is intended to ensure that the model does what it is intended to do. More so, it is like debugging (Model Validation and Verification, 2020)

The tool covered the following areas in this phase:

i. Discovery of the vulnerabilities was relatively accurate, but the detection speed was not as fast, especially when a large number of ports needed to be scanned.

ii. Review from an expert indicates that the tool delivers the intended objective of identification of vulnerabilities. Though fast when a small number of IoT Devices are connected, the detection speeds decrease when more devices are connected.

iii. Expected results of the tool in the identification of vulnerabilities, and providing a report that further shows mitigation steps were achieved. The tool showed the vulnerability discovered and a pdf report was generated with the mitigation steps that are used by an administrator.

iv. The tool was designed such that upon a successful connection of the tool to a network that IoT devices are connected to, the scan was run successfully. The tool when run in network that has no IoT devices connected, gives no results when the scan is initiated.

To assess that the problems mentioned were resolved, the tool was developed for the use by skilled and knowledgeable administrator. The administrators needed expertise on the Command Line Interface limiting the number of individuals who can perform scans and use generated reports to mitigate future attacks. Secondly, the tool is non-commercial for the use by system administrators.

Chapter 6 : CONCLUSIONS, RECOMMENDATIONS AND FUTURE WORKS

6.1 Introduction

This chapter discusses how the study successfully met its study objectives through the design, testing, and validation of the Prototype IoT Vulnerability Scanner (IVS) tool. Each study objective is addressed in detail, demonstrating how the tool's implementation contributed to enhancing IoT security across different connection types.

6.2 Discussion

i. The first objective of this study was to design, develop, and test an IoT Vulnerability Scanning Tool. This study met this objective by designing and developing an IoT vulnerability scanner capable of detecting security weaknesses in devices connected via Wi-Fi and Bluetooth. The development process included integrating a vulnerability database and generating detailed security reports. To ensure the tool's functionality, multiple controlled experiments were conducted using IoT devices with known vulnerabilities. These tests validated the scanner’s ability to identify security vulnerabilities and provide actionable recommendations to administrators.

ii. The second objective of this study was to enhance IoT Security across Multiple Connection Types. This study met this objective albeit in a controlled environment by ensuring the scanner was capable of detecting vulnerabilities in the selected IoT devices connected through both Wi-Fi and Bluetooth networks. The scanner successfully detected vulnerabilities such as unencrypted data transmission and open network ports highlighting security risks that attackers could exploit across multiple connection types.

iii. The third objective of this study was to evaluate the IoT Vulnerability Scanning Tool's Performance and Accuracy. This study met this objective by conducting a series of controlled experiments to assess the scanner's performance in real-world scenarios. The tool was tested on four IoT devices, each with predefined vulnerabilities, to measure its detection accuracy. The scanner successfully identified 6 vulnerabilities, with a 50% accuracy rate. Additionally, the tool was evaluated based on its ability to generate security reports demonstrating high efficiency and reliability in identifying IoT security threats.

iv. The fourth objective of this study was to validate the effectiveness of the IoT Vulnerability Scanning tool for IoT devices. This study met this objective by testing the scanner in different environments with varying Wi-Fi and Bluetooth network configurations. The tool successfully detected and analyzed connected IoT devices, scanning them for vulnerabilities. The validation process confirmed that the tool could effectively identify security threats, recommend mitigation strategies, and simulate potential attacks such as brute force password cracking. Additionally, the study compared the scanner's capabilities with existing commercial tools, demonstrating its effectiveness as a free, accessible alternative for system administrators managing IoT security.

Through this tool, this study has shown the ability to scan, analyze and provide reports for consumption on live scan. The tool has also proven the capability of an attacker to break in a network by cracking a password. The tool is also free for all system administrators.

6.3 Conclusion

A vulnerability scanner is a tool that administrators may choose to forgo; however, incorporating it into your toolkit can be beneficial. Running it regularly could uncover vulnerabilities, which should be viewed as a positive find. Based on the insights gathered from the literature review and the conducted tests, the research looks at how effective and complicated it is to find vulnerabilities by using a mix of different methods, tools, and measurements (Reyes, Fuertes, & Macas, 2022). To ensure the appropriate level of confidentiality, integrity, and availability of information, conducting routine vulnerability scans is crucial for organizations. These scans help identify

potential security risks and critical areas that require attention. They provide valuable insights into how security is managed and enhanced, although it's important to remember that these tools do have their limitations.

The developed tool was able to scan and analyze the IoT devices connected through two connection types and was also able to give an overview of the security status of each of the devices. The tool provides system administrators and any user with terminal-use knowledge with a no-cost option, distinguishing it from other IoT Vulnerability Scanners discussed in Chapter 2 of this study. They can utilize this tool to attempt password cracking on a network, while simultaneously scanning and analyzing devices, all without the necessity of upgrading to a paid version. In conclusion, this tool has accomplished the following objectives:

i. Identify the common vulnerabilities that are present in connected IoT Devices

ii. Review other connection platforms that IoT Devices can connect to identify their vulnerabilities

iii. To Perform a live scan and analyze connected IoT devices to both WI-FI and Bluetooth Networks

iv. To generate individual reports for each scan showing the vulnerabilities discovered and the mitigations required.

6.4 Recommendations

Based on the findings of this study, it is advisable to incorporate the developed tool for the large-scale analysis of connected IoT devices within a network. As the world continues to gain more knowledge through information captured by a vast number of connected IoT devices, it would be important for school institutions to take advantage and employ their students with skills to improve on the security of IoT devices. A legal framework governing the utilization of IoT devices, especially where public data is involved to mitigate unauthorized access to information without proper scanning and security procedures being adhered to. Another important characteristic is the validation of IoT vulnerability scanning tools developed to be used by various entities. While IoT

vulnerability scanner tools have been around for some time, significant efforts are still required to enhance the security of various IoT devices. Implementing thorough and expedient validation processes would greatly improve the functionality and reliability of these devices

6.5 Future Works

Further research is needed to explore effective methods for conducting and automating vulnerability scans in IoT systems. This tool currently provides a manual approach for analyzing vulnerabilities and suggesting mitigation strategies to address them. The tool can be explored further improve on the interact ability of the tool from a command line to a user interface that is more appealing to users including an automated way of analyzing and proposing vulnerability mitigation. To extend its usability from only designated system administrators to individual given the mandate of administration. To enhance integration with other tools for more efficient analysis, it is important to include additional report formats beyond PDF, such as CSV and Excel. Looking ahead, the tool should be designed to facilitate large-scale scans, enabling the analysis of a greater number of IoT devices connected to a network—an objective that was unfortunately not achievable within the specified timeframe.

REFERENCES

Ali, B., & Awad, A. I. (2018). Cyber and Physical Security Vulnerability Assessment for IoT-Based Smart Homes. *Sensors*.

Allianz Global Corporate & Specialty. (2022). *Cyber: The changing threat landscape.*

America's Cyber Defense Agency. (2017, October 17). *Heightened DDoS Threat Posed by Mirai and Other Botnets*. Retrieved from America's Cyber Defense Agency: Heightened DDoS Threat Posed by Mirai and Other Botnets

Andriulo, F. C., Fiore, M., Mongiello, M., Traversa, E., & Zizzo, V. (2024). Edge Computing and Cloud Computing for Internet of Things: A Review. *Informatics*, 71. Retrieved from https://doi.org/10.3390/informatics11040071

Anik, S. M., Gao, X., Meng, N., Agee, P. R., & McCoy, A. P. (2022). A cost-effective, scalable, and portable IoT data infrastructure for indoor environment sensing. *Journal of Building Engineering, 49, 104027*.

Anisenko, L. (2019, November 18). *IoT Platforms vs. IoT Device Management*. Retrieved from https://www.friendly-tech.com/blog/iot-platform-vs-iot-device-management

AWS. (2019, November 18). *AWS IoT Device Management*. Retrieved from AWS: https://aws.amazon.com/iot-device-management/

AWS. (2025, March 24). *What is IoT (Internet of Things)?* Retrieved from Amazon Web Services: https://aws.amazon.com/what-is/iot/

Bakhshi, T., Ghita, B., & Kuzminykh, L. (2024). A Review of IoT Firmware Vulnerabilities and Auditing Techniques. *sensors*.

Barbosa, G. N., & Mattos, D. M. (2025). A Practical Evaluation of a Federated Learning Application in. *Journal of the Brazilian Computer Society, 31*, 1-10. Retrieved from https://doi.org/10.5753/jbcs.2025.4324

Bennouk, K., Aali, N. A., Idrissi, Y. E., Sebai, B., Faroukhi, A. Z., & Mahouachi, D. (2024). A Comprehensive Review and Assessment of Cybersecurity Vulnerability Detection Methodologies. *CyberSecurity*.

Bentley, S. V., Naughtin, C. K., McGrath, M. J., Irons, J. L., & Cooper, P. S. (2024). The digital divide in action: how experiences of digital technology shape future relationships with artificial intelligence. *AI Ethics*, 901-915.

Berners-Lee, T., Hendler, J., & Lassila, O. (2023). In Linking the World's Information: Essays on Tim Berners-Lee's Invention of the World Wide Web. *The Semantic Web: A new form of Web content*, 91-103.

Bharani. (2018, July 19). *What is IoT (Internet of Things)? IoT Architecture Explained*. Retrieved February 7, 2019, from https://www.edureka.co/blog/what-is-iot/

Chataut, R., Phoummalayvane, A., & Akl, R. (2023). Unleashing the Power of IoT: A Comprehensive Review of IoT Applications and Future Prospects in Healthcare, Agriculture, Smart Homes, Smart Cities, and Industry 4.0. *Sensors*, 7194.

Creswell, J. W., & Creswell, J. D. (2022). *Research design: Qualitative, quantitative, and mixed methods approaches*. London: SAGE Publications, Inc.

Dean, M. (2017, March 27). *The best 3 Internet of Things scanners to use*. Retrieved from https://windowsreport.com/internet-of-things-scanner/

Dempsey, K., Eavy, P., Moore, G., & Takamura, E. (2020). Automation Support for Security Control Assessments: Software Vulnerability Management. *NISTIR 8011 Volume 4*, 7.

EduCBA. (2020, May 05). *Introduction to IoT Ecosystem*. Retrieved from https://www.educba.com/iot-ecosystem/

Firmalyzer. (2019). Retrieved from Firmalyzer: https://firmalyzer.com

Fortinet. (2025). *What Is A Brute Force Attack?* Retrieved from https://www.fortinet.com/resources/cyberglossary/brute-force-

attack#:~:text=A%20hybrid%20brute%20force%20attack%20is%20when,methods%20to%20discover%20an%20account%20login%20combination.

Fortinet. (2025). *What Is A Port Scan? How To Prevent Port Scan Attacks?* Retrieved from https://www.fortinet.com/resources/cyberglossary/what-is-port-scan

Gargan, R. (2024, June 12). *IoT Security Threats & How to Mitigate Them*. Retrieved from https://www.netmaker.io/resources/iot-security

Gelgi, M., Guan, Y., Arunachala, S., Rao, M. S., & Dragoni, N. (2024). Systematic Literature Review of IoT Botnet DDOS Attacks and Evaluation of Detection Techniques. *Sensors*.

Gibson, N., & Hull, M. M. (2020, April 14). *Dojo by BullGuard Introduces First-of-Its-Kind Intelligent IoT Vulnerability Scanner for Communication Service Providers*. Retrieved from BullGuard: https://www.bullguard.com/press/press-releases/2018/dojo-by-bullguard-introduces-first-of-its-kind-int.aspx

Gupta, V. (2020, April 14). *Traditional networks vs SDN* . Retrieved from https://www.slideshare.net/vipin2411/traditional-networks-vs-sdn

HaddadPajouh, H., Dehghantanha, A., Parizi, R. M., Aledhari, M., & Karimipour, H. (2021). A survey on internet of things security: Requirements, challenges, and solutions. . *Internet of Things, 14, 100129*.

Hamilton, R. (2016, November 3). *Mirai Scanner: Are You an Unwitting Mirai Botnet Recruit?* Retrieved from https://www.imperva.com/blog/mirai-scanner-unwitting-mirai-botnet-recruit/

IBM. (2020, April 16). *Cloud Network*. Retrieved from https://www.ibm.com/cloud/network

IBMServices. (2019, August 9). *SDN Versus Traditional Networking Explained*. Retrieved from IBM: https://www.ibm.com/services/network/sdn-versus-traditional-networking

IoTFactory. (2020, April 16). *Overview of IOT Networks*. Retrieved from https://iotfactory.eu/iot-knowledge-center/overview-of-iot-networks/

James, H. (2019, July 16). *IoT Security: 42 Top Internet of Things Security Solutions*. Retrieved from https://haydenjames.io/iot-internet-of-things-security-solutions/

Khan, Y. (2020, May 05). *5 Essential Components of an IoT Ecosystem*. Retrieved from https://learn.g2.com/iot-ecosystem

Megas, K., Piccarreta, B., & O'Rourke, D. G. (2017). Internet of Things (IoT) Cybersecurity Colloquium., (pp. 3-5).

Meil, D. (2025, February 07). *A Pragmatic Overview of Wi-Fi Security Risks*. Retrieved from Communications of the ACM: https://cacm.acm.org/blogcacm/a-pragmatic-overview-of-wi-fi-security-risks/

Merenda, M., Porcaro, C., & Iero, D. (2020). Edge Machine Learning for AI-Enabled IoT Devices: A Review. *Sensors*. Retrieved from https://doi.org/10.3390/s20092533

Microsoft. (2020, April 16). *Top benefits of cloud computing*. Retrieved from https://azure.microsoft.com/en-us/overview/what-is-cloud-computing/#benefits

Microsoft. (2025). *What is vulnerability management?* Retrieved May 2025, from Microsoft: https://www.microsoft.com/en-us/security/business/security-101/what-is-vulnerability-management

Model Validation and Verification. (2020, June 07). Retrieved from http://www.inf.ed.ac.uk/teaching/courses/ms/notes/note14.pdf

Mukhtar, B. I., Elsayed, M. S., Jurcut, A. D., & Azer, M. A. (2023). IoT vulnerabilities and attacks: SILEX malware case study. *Symmetry, 15(11), 1978*.

NIOS. (2019). *Phases of System Development Life Cycle*. Retrieved from National Institute of Open Schooling: http://oer.nios.ac.in/wiki/index.php/Phases_of_System_Development_Life_Cycle

O'Brien, S. A. (2016, October 22). *Unprecedented' cyberattack involved tens of millions of IP addresses*. Retrieved from https://money.cnn.com/2016/10/22/technology/dyn-cyberattack/index.html

OWASP. (2020, May 06). *OWASP Internet of Things Top 10* . Retrieved from https://owasp.org/www-pdf-archive/OWASP-IoT-Top-10-2018-final.pdf

Owobu, W. O., Abieba, O. A., Gbenle, P., Onoja, J. P., Daraojimba, A. I., Adepoju, A. H., & Chibunna, U. B. (2024). Review of Enterprise Communication Security Architectures for Improving Confidentiality Integrity and Availability in Digital Workflows. *International Journal of Advanced Multidisciplinary Research and Studies, 4*(6), 1417-1426.

Palmaers, T. (2020). *Implementing a vulnerability management process.* SANS Institute Reading Room.

Park, R. (2015, April 17). *The Internet of Things (IoT) and Security Risks*.

Paul, F. (2020, June 06). *Top 10 IoT vulnerabilities*. Retrieved from Network World : https://www.networkworld.com/article/3332032/top-10-iot-vulnerabilities.html

Powell-Morse, A. (2018, January 30). *Waterfall Model: What is it and When Should you use it?* Retrieved from https://airbrake.io/blog/sdlc/waterfall-model

Puzder, D. (2023, April 27). *Vulnerabilities, Threats, and Risks Explained*. Retrieved from Office of Information Security: https://informationsecurity.wustl.edu/vulnerabilities-threats-and-risks-explained/

Rapid7. (2020, April 16). *Vulnerability Management and Scanning* . Retrieved from https://www.rapid7.com/fundamentals/vulnerability-management-and-scanning/

Rapid7. (2020, April 16). *Vulnerability Management Program Framework* . Retrieved from RAPID7: https://www.rapid7.com/fundamentals/vulnerability-management-program-framework/

Reyes, J., Fuertes, W., & Macas, M. (2022). Development Processes of Vulnerability Detection Systems: A Systematic Review, Approaches, Challenges, and Future Directions. *Communications in Computer and Information Science*, 335-350.

Sears, A. (2018, August 14). *A Beginner's Guide to Securing Your IoT Devices*. Retrieved from https://www.iotforall.com/how-to-secure-iot-devices/

Sharma, R. (2020, April 16). *Dojo by BullGuard Intros AI/ML-powered IoT Vulnerability Scanner* . Retrieved from https://www.thefastmode.com/technology-solutions/12748-dojo-by-bullguard-intros-ai-ml-powered-iot-vulnerability-scanner

Sheridan, K. (2020, April 14). *New Free Mirai Scanner Tools Spot Infected, Vulnerable IoT Devices* . Retrieved from DARKReading: https://www.darkreading.com/perimeter/new-free-mirai-scanner-tools-spot-infected-vulnerable-iot-devices-/d/d-id/1327436

Stokes, P. (2018, December 5). *4 Stages of IoT architecture explained in simple words*. Retrieved February 10, 2019, from https://medium.com/datadriveninvestor/4-stages-of-iot-architecture-explained-in-simple-words-b2ea8b4f777f

Stouffer, C. (2022, October 02). *Bluetooth security risks to know (and how to avoid them)*. Retrieved from Norton: https://us.norton.com/blog/mobile/bluetooth-security#:~:text=Bluesnarfing,activities%20such%20as%20identity%20theft.

Tawalbeh, L., Muheidat, F., Tawalbeh, M., & Quwaider, M. (2020). IoT Privacy and Security: Challenges and Solutions. *Applied Sciences*.

The Government of the Hong Kong Special Administrative Region of the People's Republic of China. (2024). *AN INTRODUCTION TO RAPID APPLICATION DEVELOPMENT*. Hong Kong.

UCBerkeley. (2020, April 20). *System Validation and Verification Plans*. Retrieved from https://connected-corridors.berkeley.edu/developing-system/system-validation-and-verification-plans

Valentino, M., & Dario, A. (2024). Energy-based approach for attack detection in IoT devices: A survey. *Internet of Things*.

Vardakis, G., Hatzivasilis, G., Koutsaki, E., & Papadakis, N. (2024). Review of Smart-Home Security Using the Internet of Things. *Special Issue New Challenges in Information Security and Privacy and Cyber Resilience)*.

VMware. (2020, April 16). *What is cloud networking?* Retrieved from https://www.vmware.com/topics/glossary/content/cloud-networking

Wakili, A., & Bakkali, S. (2025). Privacy-preserving security of IoT networks: A comparative analysis of methods and applications. *Cyber Security and Applications, 3*. Retrieved from https://doi.org/10.1016/j.csa.2025.100084

Yen, D. C., & Davis, W. S. (2019). Rapid application development (RAD). *The Information System Consultant's Handbook*, 247-252.

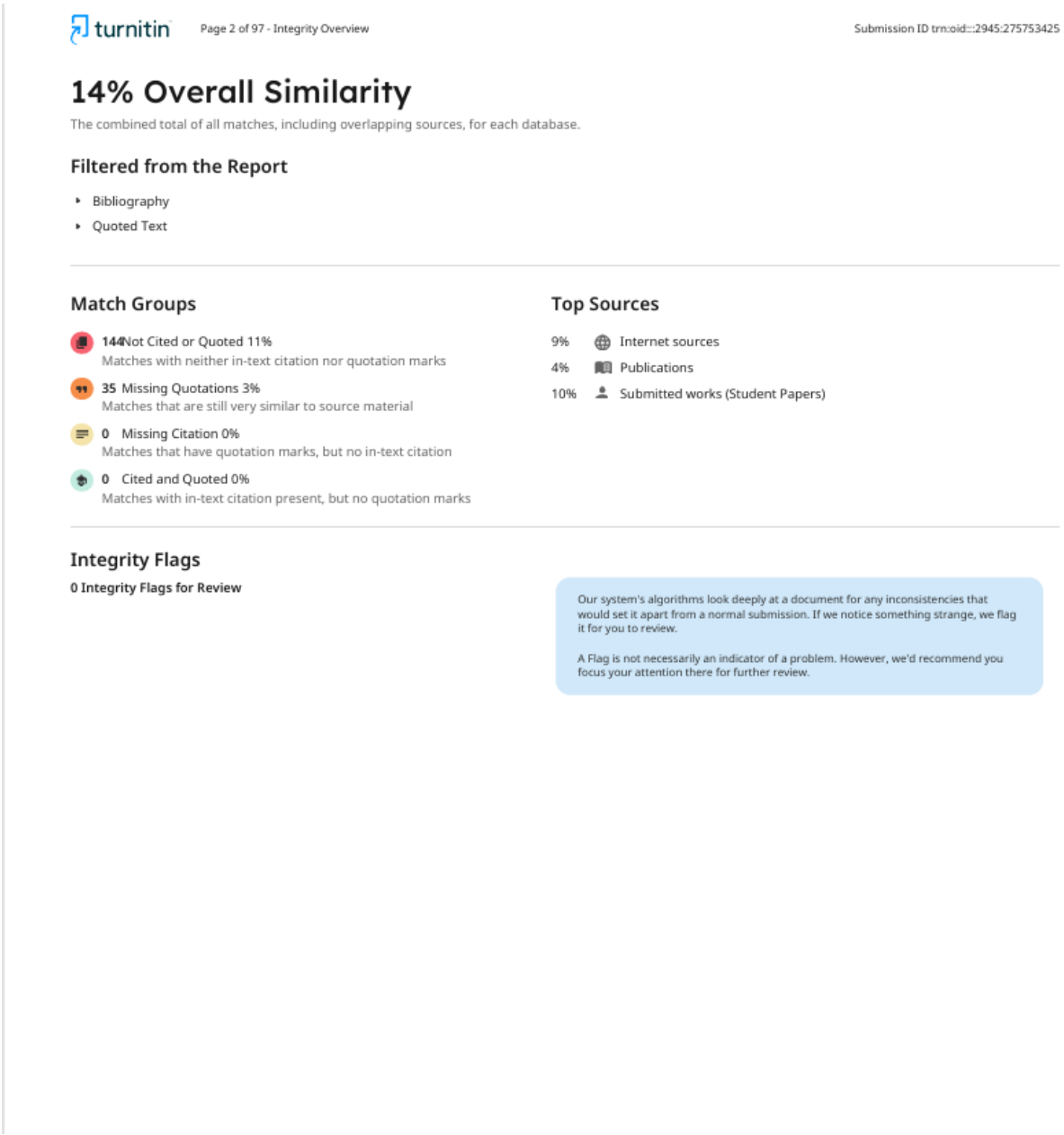
Yu, M., Zhuge, J., Cao, M., Shi, Z., & Jiang, L. (2020). A Survey of Security Vulnerability Analysis, Discovery, Detection, and Mitigation on IoT Devices. *future internet*.

Zhao, B., Ji, S., Lee, W.-H., Lin, C., Weng, H., Wu, J., . . . Beyah, R. (2022). A Large-scale Empirical Study on the Vulnerability of Deployed IoT Devices. *IEEE Transactions on Dependable and Secure Computing, 19*(3), 1826-1840.

Zhao, B., Ji, S., Zhang, X., Tian, Y., Wang, Q., Pu, Y., & Beyah, R. (2023). UVSCAN: Detecting third-party component usage violations in IoT firmware. *32nd USENIX Security Symposium (USENIX Security 23)*, 3421-3438.

APPENDICES

Appendix A: Similarity Report



turnitin Page 2 of 97 - Integrity Overview Submission ID trn:oid:::2945:275753425

14% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Filtered from the Report

- Bibliography
- Quoted Text

Match Groups

144 Not Cited or Quoted 11%
Matches with neither in-text citation nor quotation marks

35 Missing Quotations 3%
Matches that are still very similar to source material

0 Missing Citation 0%
Matches that have quotation marks, but no in-text citation

0 Cited and Quoted 0%
Matches with in-text citation present, but no quotation marks

Top Sources

9% Internet sources
4% Publications
10% Submitted works (Student Papers)

Integrity Flags

0 Integrity Flags for Review

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

Appendix B: Ethical Clearance Confirmation

Strathmore UNIVERSITY

28th March 2025

Mr Njeru Edwin,
edwin.njeru@strathmore.edu

Dear Mr Njeru,

RE: A Security Vulnerability Scanner Prototype for IoT Devices on Wi-Fi and Bluetooth Networks

This is to inform you that SU-ISERC has reviewed and **approved** your above **SU-masters** proposal. Your application reference number is **SU-ISERC2804/25.** The approval period is from **28th March 2025 to 27th March 2026.**

This approval is subject to compliance with the following requirements:

i. Only approved documents including (informed consents, study instruments, MTA) will be used.
ii. All changes including (amendments, deviations, and violations) are submitted for review and approval by SU-ISERC.
iii. Death and life-threatening problems and serious adverse events or unexpected adverse events whether related or unrelated to the study must be reported to SU-ISERC within 72 hours of notification.
iv. Any changes anticipated or otherwise that may increase the risks or affected safety or welfare of study participants and others or affect the integrity of the research must be reported to SU-ISERC within 72 hours.
v. Clearance for the export of biological specimens must be obtained from relevant institutions.
vi. Submission of a request for renewal of approval at least 60 days prior to the expiry of the approval period. Attach a comprehensive progress report to support the renewal.
vii. Submission of an executive summary report within 90 days of completion of the study to SU-ISERC.

Before commencing your study, you will be expected to obtain a research license from National Commission for Science, Technology, and Innovation (NACOSTI) https://research-portal.nacosti.go.ke/ and obtain other clearances needed.

Yours sincerely,

Mr Ambrose Rachier,
Chairperson; SU-ISERC