

Matthew Cutajar

Student Number: 150389703

Consolidating IoT Hardening through a Qualitative and
Experimental Approach



Royal Holloway University of London

Information Security Group

Egham, Surrey, TW20 0EX

United Kingdom

Supervisor: Professor Konstantinos Markantonakis

Submitted as part of the requirements for the award of the MSc in Information Security
of the University of London.

March 2022

Anti-Plagiarism Declaration

I declare that this assignment is all my own work and that I have acknowledged all quotations from published or unpublished work of other people. I also declare that I have read the statements on plagiarism in Section 1 of the Regulations Governing Examination and Assessment Offences, and in accordance with these regulations I submit this project report as my own work.

Signature: 

Date: 31st March 2022

Acknowledgements

Steve Jobs once said that ‘Great things are not done by one person. They’re done by a team of people’. I consider this project as the epitome of a two-year academic journey which will hopefully lead to the completion of my Masters degree. Surely, these two years would not have been possible without the support of a barrage of people who in their own manner have helped pave the way. While I take the opportunity to thank everyone for their support, without a doubt I would like to thank:

My parents - For the last 24 years they had no greater goal than to see me flourish both academically and also personally. Without their moral (and financial) support, love, patience, and dedication towards me this project surely would not have been possible.

My fiancé - As soon as I approached her with the thought of pursuing a Masters Degree she supported and encouraged me to further my studies. I thank her for the motivation, love and support she provides on a daily basis.

My dissertation supervisor, Professor. Konstantinos Markantonakis, who through his knowledge provided insightful guidance and advice, thus facilitating the process.

All my family, friends, current and previous work mates, especially my previous manager, Nicholas, who encouraged me to pursue this course and to always challenge myself. My gratitude is extended to my current manager, Chantelle, who constantly motivates me to improve.

I would like to thank all these people who helped me through this journey.

Abstract

Introduction – It is evident that as the IoT sector is developing so is the IoT threat landscape. The continuous increase of risks and vulnerabilities does not bode well for the future of IoT given the risks these systems present. Evidence indicates that these devices are predisposed to vulnerabilities, threats and security risks. Guaranteeing that these smart devices are secure must be the common obligation and responsibility of all stakeholders involved.

Purpose - The main goal of this study is to provide IoT system owners with a comparison of the chosen hardening procedures published by Industry recognised organisations.

Design/methodology/approach – A literature review was performed to gain insight on hardening procedures. Three hardening procedures were identified as applicable for the methodology and case scenario. A methodology was proposed in order to verify the applicability of hardening controls within a smart home environment. Through the methodology a recommended hardening procedure was proposed. Case scenarios were identified to test the three hardening procedures and the recommended hardening procedure within a simulated environment consisting of a Raspberry Pi, a vulnerability assessment tool and tools to perform manual attacks.

Findings – This research identified that no hardening procedure is targeted at providing controls exclusively to smart home owners. Hardening procedures exist with different controls at times overlapping and which target different audiences. The proposed hardening procedure was as effective in protecting smart home owners as the three identified hardening procedures.

Research limitations – The limitations of this research along with the recommendations will be discussed.

Practical implications – It is imperative that IoT hardening procedures published by respected cybersecurity organisation such as ENISA are implemented to mitigate the substantial amount of threats and vulnerabilities present within the IoT field.

Conclusion - Despite the limitations of the project, the author attests that the results of this study provide relevant information to smart home owners. The author notes that further research can be developed based on the outcomes and recommendations set forth within the project

Keywords: Internet of Things, Smart Homes, Information Security, Hardening Procedures, Smart devices, Security Testing

Table of Contents

1.Introduction	1
1.1 Background	1
1.2 Motivation to conduct the Study	3
1.3 Statement of the Problem	3
1.4 Statement of Purpose	5
1.5 Research Question	6
1.6 Aims and Objectives of Study	7
1.7 Description of Study Organisation	8
2. The Internet of Things	9
2.1 Introduction to the IoT and IoT Devices	9
2.2 IoT Capabilities	12
2.3 IoT Architecture	12
2.4 IoT Protocols	15
3. Security Risks in IoT	17
3.1 OWASP IoT Top 10	17
3.2 Security Risks in IoT	21
4. IoT Pen Testing Tools	28
4.1 PENIOT	28
4.2 Tenable Nessus	29
4.3 OpenVAS	29
4.4 ZAP	30
4.5 BURP Suite	30
4.6 Nmap	31
4.7 Kali Linux	31
4.8 Hydra	32
5. IoT Hardening Procedures	33
5.1 Introduction	33
5.2 Ethical Issues	35
5.3 Literature Review on Hardening Procedures	36
5.4 Methodology	42

5.4.1 Identifying which controls are applicable and implementable within a smart home environment	43
5.4.2 Score each applicable control according to which OWASP IoT Top 10 vulnerability it aims to mitigate	43
5.5 OWASP IoT Security Verification Standard (ISVS)	45
5.6 CIS Controls Internet of Things Companion Guide	47
5.7 ENISA Baseline Security Recommendations for IoT in the context of Critical Information Infrastructures	48
5.8 Hardening Procedures Analysis	50
5.9 Recommended Hardening Procedure	53
6. Case Scenarios	58
6.1 Setup	58
6.2 Research Setup	60
6.3 Case Scenarios Initial Configuration	64
6.4 Case Scenario 1	66
6.4.1 Case Scenario 1 - Port and Service Enumeration	66
6.4.2 Case Scenario 1 - Vulnerability Assessment	67
6.4.3 Case Scenario 1 - Manual Attacks	67
6.4.4 Case Scenario 1 - Risk Evaluation	68
6.5 Case Scenario 2 - ISVS	70
6.6 Case Scenario 3 - CIS	73
6.7 Case Scenario 4 - ENISA	76
6.8 Case Scenario 5 - Recommended Hardening Procedure	79
7. Security Analysis	82
7.1 Introduction	82
7.2 Discussion of results	82
7.2.1 RQ 1: Which options do smart home owners have when they are concerned with hardening their IoT devices?	82
7.2.2a RQ 2a: Which hardening procedures identified as a reply to Question 1 originate from established organisations within the cybersecurity field?	84
7.2.2b RQ 2b: Which three hardening procedures established in Question 2a can be considered as safe to be utilised for the experimental and theoretical part of this study?	84
7.2.3a RQ 3a: Which controls from the three established hardening procedures can be implemented within a smart home and within this study?	85

<i>7.2.3b RQ 3b: Which OWASP IoT Top 10 vulnerabilities are mitigated through the controls established in Question 3a?</i>	85
<i>7.2.4 RQ 4: Can the controls identified within Question 3a be merged into one hardening procedure?</i>	87
<i>7.2.5a RQ 5a: If question 4 is in the affirmative, how does the proposed hardening procedure compare to the other hardening procedures within Question 2?</i>	87
<i>7.2.5b RQ 5b: What additional value do end users gain from the hardening procedure identified in Question 4?</i>	87
7.3 Limitations of this study	89
8.Conclusions and Recommendations	92
Bibliography	94
Appendices	105
Appendix A - IoT Protocols	106
Appendix B - OWASP IoT Security Verification Standard (ISVS) Applicable Controls	112
Appendix C - CIS Controls Internet of Things Companion Guide Applicable Controls	117
Appendix D - ENISA Baseline Security Recommendations for IoT in the context of Critical Information Infrastructures Applicable Controls	121
Appendix E - Recommended Hardening Procedure	125
Appendix F - Script Utilised in Case Scenarios	134
Appendix G - Vulnerability Scan Report for Case Scenario 1	135
Appendix H - Vulnerability Scan Report for Case Scenario 2	139
Appendix I - Vulnerability Scan Report for Case Scenario 3	142
Appendix J - Vulnerability Scan Report for Case Scenario 4	144
Appendix K - Vulnerability Scan Report for Case Scenario 5	147

List of Figures

Figure	Page
Figure 2.1: Four-layer architecture	14
Figure 2.2: Three and Five Layered Architectures	15
Figure 5.1: Outline of Chapter 5: Hardening Procedures	34
Figure 5.2: ISVS Security Model stack	46
Figure 6.1: Modified Risk Evaluation Process	59
Figure 6.2: iotpass.txt contents	62
Figure 6.3: Experiment Setup Architecture	64
Figure 6.4: Evidence of Pi1 data on Initialstate	65
Figure 6.5: User default added within sudoers	65
Figure 6.6 Nmap port scan	66
Figure 6.7 Hydra attack on pi1 identifying the login credentials	67
Figure 6.8 ICMP flood attack on Pi1	68

List of Tables

Table	Page
Table 1.1: List of vulnerabilities that IoT is susceptible to	4
Table 2.1: Smart Home Devices according to Otelco	10
Table 3.1: OWASP IoT Top 10	18
Table 3.2: IoT Threats and vulnerabilities on each layer	22
Table 5.1: Summary of Hardening Procedures identified through Keyword 1('IoT Hardening Procedures')	37
Table 5.2: Summary of Hardening Procedures identified through Keyword 2 ('IoT Hardening')	38
Table 5.3: Summary of Hardening Procedures identified through Keyword 3 ('Smart Home hardening procedures')	39
Table 5.4: Summary of Hardening Procedures identified through Keyword 4 ('Smart Home hardening')	40
Table 5.5: OWASP IoT Top 10 Vulnerability with the score rating for each control	44
Table 5.6 ENISA Recommendations and Intentions	49
Table 5.7 Hardening Procedures Applicable Controls according OWASP IoT Top 10 Vulnerability	50
Table 5.8 Breakdown of workings per hardening procedure	52
Table 5.9 Hardening Procedures Applicable Controls according OWASP IoT Top 10 Vulnerability including recommended hardening procedure	55
Table 5.10 Breakdown of workings including recommended hardening procedure	56

Table 6.1 CVSSv3 Rating	60
Table 6.2 Configuration Settings per SD card	62
Table 6.3 System Information for the machines (physical or virtual) used within the Case Scenarios	63
Table 6.4: Nmap Scan Open Ports and related service in Case Scenario 1	66
Table 6.5: Vulnerabilities identified through manual testing for Case Scenario 1	69
Table 6.6: Vulnerabilities identified through both manual and automated testing for Case Scenario 1	69
Table 6.7: Vulnerabilities identified through manual testing for Case Scenario 2	71
Table 6.8: Vulnerabilities identified through both manual and automated testing for Case Scenario 2	72
Table 6.9: Vulnerabilities identified through manual testing for Case Scenario 3	75
Table 6.10: Vulnerabilities identified through both manual and automated testing for Case Scenario 3	75
Table 6.11: Vulnerabilities identified through manual testing for Case Scenario 4	77
Table 6.12: Vulnerabilities identified through both manual and automated testing for Case Scenario 4	78
Table 6.13: Actions performed within Case Scenario 5	80
Table 6.14: Vulnerabilities identified through manual testing for Case Scenario 5	81
Table 6.15: Vulnerabilities identified through both manual and automated testing for Case Scenario 5	81

List of Abbreviations

AMQP	Advanced Message Queuing Protocol (AMQP)
API	Application Programming Interface
ARM	Advanced RISC Machines
ARP	Address Resolution Protocol
ASVS	Application Security Verification Standard
BLE	Bluetooth Low Energy (BLE)
COAP	Constrained Application protocol
CVSS	Common Vulnerability Scoring System
CVSSv3	Common Vulnerability Scoring System Version 3
CIS	Centre for Internet Security
DDoS	Distributed Denial of Service
DNS	Domain Name System
DoS	Denial of Service
ENISA	European Union Agency for Cybersecurity
GRC	Governance, risk management, and compliance
GUI	Graphical User Interface
H2C	Human to Computer
H2H	Human to Human
H2M	Human to Machine
HTTP	Hypertext Transfer Protocol
IEEE	Institute of Electrical and Electronics Engineers
ICMP	Internet Control Message Protocol
IoT	Internet of Things
IP	Internet Protocol
ISO	International Organisation for Standardisation
ISVS	Internet of Things Security Verification
IT	Information Technology
ITS	Intelligent Transportation System
LAN	Local Area Network
MAC	Media Access Control
M2M	Machine to Machine

MQTT	Message Queueing Telemetry Transport
MIPS	Microprocessor without Interlocked Pipelined Stages
MITM	Man-in-the-middle
NFC	Near Field Communication
NIST	National Institute of Standards and Technology
NMAP	Network Mapper
NVT	Network Vulnerability Tests
OpenVAS	Open Vulnerability Assessment Scanner
OWASP	Open Web Application Security Project
PCI DSS	Payment Card Industry Data Security Standard
SCTP	Stream Control Transmission Protocol
SSL	Secure Sockets Layer
SQL	Structured Query Language
TCP	Transmission Control Protocol
UDP	User Datagram Protocol

1.Introduction

1.1 Background

The National Institute of Standards and Technology (NIST) [1], a non-regulatory agency within the United States Department of Commerce stated that the Internet of Things (IoT) is quickly advancing and the useability for such devices is increasing significantly. NIST [1] further attested that the IoT can be considered as a group of different types of technologies that leverage the physical world. To this extent, it might be the case that organisations do not comprehend how many IoT devices they are already utilising. Users might not realise how these devices differ from other Information Technology (IT) devices in terms of both privacy risks and cybersecurity.

The Centre for Internet Security (CIS) [2] noted that given that the IoT garnered interest from several areas such as different industries, academia, and governments there is no standard agreed upon definition for IoT. This does not mean that one definition is better than the other, but rather that each definition has its own pros and cons. Each definition can work within its own scenario as seen fit. The European Union Agency for Network and Information Security (ENISA) [3] defined the IoT as an ecosystem of several physical sensors and actuators connected through the internet working together to allow decision making. Information is fundamental for IoT to perform decision making and actions. The NIST [1] considers IoT as a result of mixing both IT and Operational Technology. The Institute of Electrical and Electronics Engineer (IEEE) [4] defines IoT as a network of items which all have sensors and are connected to the internet. If a device such as door locks, window sensors, headsets, thermostats, watches, security cameras and smart speakers can be connected within an IT environment then it can be considered as an IoT device [2].

In 2020, Pelaez [5] stated that there were about 11.7 billion IoT connections around the world. In reality, given that in total there are 21.7 billion active internet connections, IoT devices have managed to surpass non-IoT internet connections for the first time [5]. Furthermore, Lueth [6] predicts that in 2025, the global amount of IoT devices will nearly triple when compared to 2020 with numbers reaching more than 30 billion devices. Thus, averaging almost four devices per person. The ENISA [3] attested that IoT will effectively affect how both consumers and businesses behave. Hence, forcing manufacturers to develop more intelligent solutions. Bickley [7] attested that it is essential to secure both IoT data and the physical devices, but this is very difficult to attain. Furthermore, Intellectsoft [8], stated that with the increase of IoT

devices' popularity, it is assumed that this will see an increase in security challenges and issues. The Open Web Application Security Project (OWASP) is a community sourced project providing free to use hardening procedures, methodologies, documentation, and tools generally related to web applications. In 2018, OWASP [9] published the IoT Top 10 which identified the top ten cybersecurity threats that users or system owners should avoid when working with IoT devices. Bickley [7] attested that frameworks and best practices must be documented by security experts. This is necessary so that both device designers and network planners, implement the necessary security measures to mitigate IoT threats. The ENISA [10] noted that the risk and threat landscape for IoT is significantly large and evolving at a fast pace. This has its repercussions on the safety, security, and privacy of citizens. To protect the IoT from these risks, understanding which devices to secure and building the appropriate security controls is imperative [10]. It is also important that manufacturers, companies and end users implement these hardening procedures within their remit and capacity.

IoT devices prove to be a challenge to secure even for professionals within the cybersecurity field [2]. The Australian Cybersecurity Centre [11] defines system hardening as the process through which IT systems are secured by reducing the attack surface through a multitude of techniques and tools. Several well-renowned entities such as the IEEE, ENISA, CIS and NIST have developed hardening procedures with their main aim being securing IoT devices. It is noted that these organisations provide guidelines to both IoT manufacturers and also to end users. The ENISA [10] published guidelines for manufacturers named 'Good practises for Security of Internet of Things in the context of Smart Manufacturing'. Furthermore, ENISA [10] outlines components of a Critical Information Infrastructures which provide a scope for its baseline recommendations. Networks, services, both physical and information technology equipment and facilities, are all within the scope of the ENISA publication. Should the above infrastructure be tampered with, the safety, economical posture or welfare of the people making use of IoT might be at risk. Hence, this system is considered critical by ENISA. Aiding organisations or agencies to gain further knowledge on how to mitigate the risks that individual IoT devices have throughout their lifecycle is the main reason why NIST [1] published its 'Considerations for Managing Internet of Things'. While the purpose stated by each organisation appears very similar, the implementation of the hardening procedure might differ from one another.

1.2 Motivation to conduct the Study

Having been intrigued by the evidence on smart homes and IoT, three years ago, the author had already submitted a dissertation relating to smart home security as part of his Bachelor's degree. The motivation to conduct this study is partially based on the author's personal and work experience on hardening methodologies designed to secure both IoT and non-IoT devices. At work, the author uses hardening methodologies for non-IoT devices published by organisations such as the International Organisation for Standardisation (ISO), NIST and CIS. When deploying his own IoT devices within his home setup the author found it exceedingly difficult to identify one hardening procedure or standard which included all the necessary steps to harden these IoT devices. While the majority of these standards or procedures agree on the main criteria required, it was noted that in some points these differ, and a mixture of standards or procedures are necessary to significantly mitigate the risks that IoT is prone to. Furthermore, it can be attested that while the risks identified in Table 1.1 cannot be fully mitigated, it is imperative to implement rigorous practices and standards such as hardening procedures to aid the end users in adapting Smart Home technology. The author was inclined to analyse what variations exist within the literature hardening procedures and what impact do these variations have on the IoT devices. The author identified a gap within the current IoT security topology and set out to draft this project. The author endeavours that through this project, further insight relating to IoT and IoT security is gained. The aim is that through these research findings, third-parties, especially smart home users can reduce the time spent researching and analysing which hardening procedure or procedures to use.

1.3 Statement of the Problem

Through literature review [5], [6], it is evident that the amount of IoT devices is significantly increasing year in, year out. It is also evident that cyber threats and risks are on the increase especially in the IoT. According to Hamilton [12], it is noticeable why cybersecurity professionals are worried about the risks of IoT as much as they are enthusiastic about the IoT benefits. Daws [13] noted that the attacks on the Kaspersky IoT honeypots doubled in the first six months of 2021 when compared to the last six months of 2020. The main reasons for 58% of these attacks were primarily linked to attempting to perform distributed denial-of-service (DDoS) attacks, cryptocurrency mining, or extrapolating private information [14].

Various authors [9],[15],[16] identified several threats and vulnerabilities within the IoT space. Table 1.1 outlines the threats and vulnerabilities identified by OWASP [9], Hewitt [15], and Thales [16]. Further

literature and analysis on the vulnerabilities affecting IoT devices will be discussed within the literature review found in Chapter 3.

Table 1.1: List of vulnerabilities that IoT is susceptible to [9], [15], [16] .

Vulnerabilities		
Lack of physical hardening [9], [15]	Insecure data storage, protection and transfer [9], [15]	Lack of visibility and device management [9], [15], [16]
Botnets [9], [15]	Weak password protection [9], [15], [16]	Insecure ecosystem interfaces [9], [15], [16]
AI-based attacks [9], [15]	Lack of regular patches and updates along with weak update mechanism [9], [15], [16]	Insecure network services [9]
Insecure default settings [9]	Use of insecure or outdated components [9]	Insufficient privacy protection [9], [15], [16]
IoT Skills Gap [16]		

The aforementioned threats and vulnerabilities identified within the IoT space as illustrated in Table 1.1, outline the need that such hardening procedures are revised to be efficient and implementable on IoT devices. Raja [17] stated that to decrease IoT risks, there are several steps to be followed and performed. This identifies the need for hardening procedures within the IoT field. Ryerse [18] noted that Operating System hardening is the process of securely setting up the Operating System whilst performing system updates. Moreover, Ryerse [18] affirmed that hardening also extends to creating a Governance, Risk, and Compliance (GRC) ecosystem, whilst also removing any application or service which is not needed. Such hardening procedures aid in reducing the System's exposure to vulnerabilities and threats.

Literature review [12], [17], [19] identified that several websites provide individuals hardening procedures to harden their IoT devices. However, the author set out to identify which reputable institutes and

organisations within the Information Security field provide hardening procedures. Miloslavskaya et al. [20] identified several IoT security standards which can be used to converge on a single universal form for all IoT security criteria. Furthermore, Miloslavskaya et al. [20] noted that through standards, people and organisations are provided with a foundation to fully understand the IoT. Each hardening procedure provides different perspectives on security issues although issues surrounding Information Security are not being given the necessary attention. These standards or hardening procedures are discussed further in Chapter 5 of this project. While Miloslavskaya et al. [20] acknowledged the fact that IoT requires a different standardisation approach when compared to other network devices, the authors also noted that the identified gaps need to be mitigated as soon as possible given the speed of development within the IoT field.

1.4 Statement of Purpose

It is evident that as the IoT sector is developing so is the IoT threat landscape [14]. The continuous increase of risks and vulnerabilities does not bode well for the future of IoT given the physical risk these systems present. A cybercriminal hacking into an IoT camera installed in a household would gain the ability to detect social patterns such as when the house tenants leave for work would be an example of such IoT risks. Other risks such as weak authentication methods including but not limited to systems configured with default passwords, insecure interfaces along insecure network services are all well-known risks identified by Cybersecurity experts. Through evidence-based practice, the author noted that given the ever-evolving range of IoT products, the IoT space is attractive to cyber criminals and the number of vulnerabilities is expected to increase in parallel to the amount of IoT devices. As noted in the literature [2], [10], [21], it can be attested that this challenge was also identified by industry recognised organisations who through their guidelines attempt to prompt IoT users to secure their systems. It was also noted that these organisations focus on standards aimed at manufacturers to reduce risks at development stage given that not all users are knowledgeable or technical enough to secure the IoT systems. Several organisations are continuously publishing information aimed at end users to raise both the overall insight about IoT risks and the technical knowledge.

1.5 Research Question

‘Consolidating IoT Hardening through a Qualitative and Experimental Approach’.

The author acknowledges that within organisations the decision on which hardening procedure is implemented might be influenced by either GRC measures or alignment to chosen standard bodies such as ISO. It must also be taken into consideration that not all IoT users have the technical insight or expertise and thus they might be more exposed to IoT threats and vulnerabilities. Furthermore, organisations might research the gaps of the standard they align to or identify further controls which need to be implemented on their IoT devices. The research question aims at solving this identified gap through both a practical and qualitative approach as discussed further in Chapters 5 and 6 respectively.

Research questions were set to investigate the problem area and list this study’s aims and objectives. Through this research, the author predicts that the following questions will be answered:

- **RQ 1:** Which options do smart home owners have when they are concerned with hardening their IoT devices?
- **RQ 2a:** Which hardening procedures identified as a reply to Research Question 1 originate from established organisations within the cybersecurity field?
- **RQ 2b:** Which three hardening procedures established in Research Question 2a can be considered as safe to be utilised for the experimental and theoretical part of this study?
- **RQ 3a:** Which controls from the three established hardening procedures can be implemented within a smart home and within this study?
- **RQ 3b:** Which OWASP IoT Top 10 vulnerabilities are mitigated through the controls established in Research Question 3a?
- **RQ 4:** Can the controls identified within Research Question 3a be merged into one hardening procedure?
- **RQ 5a:** If the result of Research Question 4 is in the affirmative, how does the proposed hardening procedure compare to the other hardening procedures within Research Question 2?
- **RQ 5b:** What additional value do end users gain from the hardening procedure identified in Research Question 4?

1.6 Aims and Objectives of Study

From the literature available and from the gaps identified within sections 1.1 and 1.3 the aims and objectives of this study were identified and structured. Thus, the main goal of this study is to provide IoT system owners with a comparison of the chosen hardening procedures published by Industry recognised organisations.

Through the analysis of each individual hardening procedure, the author aims at recommending a hardening procedure which would further secure the IoT devices and infrastructure within a smart home environment. Hence, the author aims at improving the overall security of IoT devices within smart homes by eliminating threats or vulnerabilities. To achieve the aforementioned aims, the author set out a series of short-term objectives which helped in devising the structure of the project from Chapter 2 onwards. The objectives of this study are as follows:

- **OBJ1:** To answer RQ1, RQ2a, and RQ2b a literature review was pursued with specific keywords identified. Through this literature review an overview of IoT devices, IoT Security attacks, penetration testing tools which can be utilised within the IoT field, and IoT hardening procedures could be provided.
- **OBJ2:** To answer RQ3a, the author will identify which controls can be implemented with a smart home environment.
- **OBJ3:** An analytic review is performed whereby the controls within 3a are scored according to which OWASP IoT Top 10 vulnerability these controls mitigate to answer RQ3b.
- **OBJ4:** Through the previous objectives RQ4 can be answered and a combined hardening procedure is suggested.
- **OBJ5:** To answer RQ5a, an analytic review is also performed on the hardening procedure suggested in OBJ4. Furthermore, five Case Scenarios will be drafted and tested within an experimental setup. The Case Scenarios are as follows:
 - A Case Scenario whereby an IoT device is not hardened and tested;
 - A Case Scenario whereby an IoT device is hardened using applicable controls from hardening procedure A as identified in RQ3a;
 - A Case Scenario whereby an IoT device is hardened using applicable controls from hardening procedure B as identified in RQ3a;
 - A Case Scenario whereby an IoT device is hardened using applicable controls from hardening procedure C as identified in RQ3a;

- A Case Scenario whereby an IoT device is hardened using the controls identified in OBJ4.
- **OBJ6:** To answer RQ5b a security analysis and review will be performed.

1.7 Description of Study Organisation

The project is divided into eight Chapters as follows:

1. *Chapter 1: Introduction and Orientation to the study:* This chapter lays the foundation for this research by providing a synopsis of the research area and a background on both the IoT risks and IoT hardening procedures. This chapter also outlines the aims and objectives, the significance of this project along with the research question.
2. *Chapter 2: The Internet of Things:* This chapter discusses the insight gained by the author from the literature available with regards to IoT.
3. *Chapter 3: Security Risks in IoT:* This chapter discusses the insight gained by the author from the literature available with regards to IoT risks.
4. *Chapter 4: IoT Pen Testing Tools:* This chapter discusses the insight gained by the author from the literature available with regards to IoT Penetration Testing Tools.
5. *Chapter 5: IoT Hardening Procedures:* This chapter discusses the insight gained by the author on the hardening procedures available for IoT devices. Furthermore, the author provides insight about the theoretical experiment conducted following the evidence found in literature. The experiment consists of identifying three hardening procedures that have a substantial amount of controls which can be applied within a smart home. From these hardening procedures, controls which are applicable for IoT home owners are identified and scored according to a scoring system by the author. Analysis of each hardening procedure is provided within this Chapter along with a proposed hardening procedure.
6. *Chapter 6: Case Scenarios:* Through this chapter, the hardening procedures outlined in Chapter 5 are implemented within a test environment to identify whether the proposed hardening procedure rendered any further benefits when exposed to the same threats as the other hardening methodologies.
7. *Chapter 7: Security Analysis:* Discussion and interpretations of the research findings will be presented in this chapter.
8. *Chapter 8: Conclusions and Recommendations:* Any research conclusions and recommendations will be presented in this chapter.

2. The Internet of Things

IoT is continuously evolving, and there are many ‘things’ within the IoT. These can vary in form, size, industry, and relative importance. It is expected that by 2025, the amount of IoT will grow to 22 billion devices ranging from smart generic household devices to more complicated tools used by businesses, organisations or whole industries [22]. IoT is shaping societies and economies at a fast pace. Gillis [23] attested that a person making use of a heart monitor implant can be considered a ‘thing’ along with any object which can transfer data across a network through the assignment of an IP address. This Chapter discusses the literature available on IoT, IoT devices, IoT Capabilities, IoT Architectures and the supporting IoT protocols.

2.1 Introduction to the IoT and IoT Devices

Oracle [22] defined the IoT as a physical object or ‘things’ which incorporate additional hardware or software to share data with other ‘things’ or other devices on the internet. Over the course of recent years, IoT has become one of the main technological advances of the 21st century providing the possibility of flawless interactions between the ‘things’, people, and underlying processes [22]. When everyday devices are connected to the Internet, smart devices are created and through these smart devices the IoT is constructed [24]. IoT devices are autonomous in the sense that they perform an action without the need for any human-to-human (h2h) interaction or human-to-computer (h2c) interaction. According to Pfleeger et al. [24], there are a multitude of opportunities whereby IoT can be utilised, such as smart appliances, smart home, smart transportation, and smart health.

The initial utility of smart homes was to automate lighting and climate systems. Nowadays, it has evolved to incorporate any electrical component which can be used within the household [25]. Further development to the smart home allows devices to monitor the events being undertaken within the home along with monitoring the home environment (such as light and heat). This enables these devices to identify patterns within the inhabitants’ lifestyle, thus developing tasks to match the detected pattern [25]. For example, the IoT device detects that the inhabitant turns on the water heater at seven in the morning. Through several repetitions of this task the IoT management system detects this pattern and starts turning the water heater on automatically at seven in the morning [26]. Table 2.1 illustrates a number of smart home devices and appliances as identified by Otelco [27].

Table 2.1: Smart Home Devices according to Otelco [27].

Self-Cleaning Litter Box	Smart Mobiles	Smart TVs	Smart Slow Cookers	Smart Feeders	Intruder Sensors	Smart Beds	Video Monitors
Smart Thermostats	Smart Showers	Smart Dishwasher	Smart Mats	Medical Alert Button	Smart Shades	Smart Alert Systems	Activity Tracker
Light Automation	Smart Laundry	Smart Toilets	Smart Fans	Smart Ovens	Smart Locks	Smart Health Monitors	Smart Floors
Enhanced Vanity	Smart Pools	Smart Rocking	Smart Plugs	Smart Pet Doors	Automated Sprinklers	Smart Dishes	Smart Changing Tables
Smart Refrigerators	Privacy Windows	Smart Sensors	Robot Vacuums	Smart Speakers	Smart Coffee Pots	Smart Trash Cans	

Smart appliances can be considered as appliances connected to the internet making use of innovative technologies in order for appliances to communicate with the electricity grid (and vice-versa) while providing consumers real-time analytics about their electricity consumption. These smart appliances aid in saving both money and electrical power [26]. Goossens [26] attested that such smart appliances might be easier to fix and maintain than normal appliances, given that these appliances provide diagnostics remotely along with advice relating to maintenance which aids in prolonging the lifetime of an appliance. Goossens [26] further noted that the benefits of smart appliances go beyond energy consumption since such appliances save time (roughly 100 hours per year) and provide additional convenience compared to normal appliances. Some examples of how these smart appliances provide additional comfort according to Goossens [26] are provided below:

- A smart dishwasher identifying the type of dishes and how dirty these dishes are and suggesting the best cleaning cycle to the user. Furthermore, the smart dishwasher can automatically order detergent refills once these run low.
- Through Machine-to-Machine (M2M) communication the amount of electricity consumed is reduced. Two machines can communicate with each other so that one process starts once the other process on another machine ends.
- Smart refrigerators and freezers can automatically create and order groceries. Another feature of such smart appliances is tracking expiration dates of the food stored in them.
- Through a mobile application smart oven can show the inside while food is being cooked and also track the progress of the meal.

According to S3C [28], there are two principles through which smart appliances work and these are the “modification of the starting time of an appliance cycle” and “interruption of regular appliance operation”. In the first principle, the tenant chooses the completion time, and the machine chooses the activity shift within this limitation. In the subsequent principle, a typical activity is hindered for a restricted time frame which preserves the tenant’s comfort [28].

Varun and Karthikyan [29] attested that the integration of the Intelligent Transportation System (ITS) with IoT will be key in the transformation of transportation systems. Various problems such as carbon emissions, fuel consumption increase, along with traffic congestion are attributed to the ever-increasing rate of vehicles. ITS attempts to solve this issue through integrating traffic management and various other modes of transports to enable commuters to travel safer and more efficiently whilst also making better use of transportation networks. Through information gathered from the computer processing systems, auto control networks and sensors, the ITS can be achieved [29].

Yang et al. [30] attested that through the integration of IoT with healthcare, the level of chronic disease management, life-saving interventions and the quality of life of people will improve. Support of patients’ health, service improvement and monitoring of patients’ health, are all potential applications of IoT within healthcare. The European Commission adopted several policy actions and allocated funds to increase the use and to unleash the full potential of the IoT [31].

2.2 IoT Capabilities

To achieve their goals and purposes, one or more IoT devices provide different capabilities and work together with non-IoT devices [1]. NIST [1] attested that transducer capabilities, interface capabilities and supporting capabilities are all examples of IoT capabilities. Transducer capabilities can be considered as a bridge between the digital and physical world. The two main transducer capabilities are sensing and actuating. IoT devices can perform either one of the two or else both capabilities. Sensing is when the device notes real world observations and converts these into data. Actuating is when an IoT device can actually change the physical environment. Interactions such as h2m or m2m are enabled through interface capabilities and these are further segmented into three interfaces. The application interface provides non-IoT devices the ability to liaise with an IoT device through software intermediaries such as an Application Program Interface (API). The human user interface exists to cater for h2m communication whereas the network interface includes both hardware and software to facilitate data communication to and from IoT devices. The last type of capabilities are the supporting capabilities and these are auxiliary services which provide support to other IoT capabilities such as cybersecurity, privacy capabilities and device management [1].

2.3 IoT Architecture

Various authors [32], [33], [34] have provided their interpretation of what the IoT architecture consists of. Goddard [32] stated that in the concept of the IoT architecture there are four main components. These are 1) applications and analytics, 2) integration, 3) security, and 4) infrastructure. Data analytics tools, machine learning, visualisation capabilities, and artificial intelligence make up the application and analytics component. The integration component ensures that every one of the applications, instruments, security, and framework of the IoT project incorporate flawlessly with the IoT management system. Physical and network security for the IoT infrastructure is governed through security controls and firmware in the security component. The infrastructure component incorporates in it the physical devices within IoT such as the sensors and actuators along with a transportation subcomponent which is used to transport data between IoT devices. Goddard [32] further stated that while there is no agreement in literature about the IoT architecture, the three, four and five layered architectures are the most recognised and used. Discussions on the different layered architectures provided by different authors will be provided below.

Lombardi et al. [33] attested that the three-layer architecture can be considered as a “generic high-level architecture”. As the name implies the three-layer architecture consists of three layers which are the 1)

perception, 2) network and 3) application layers. The application layer ensures that the services are available to the final user and is generally hosted either on the cloud or on servers [32]. The network layer enables the IoT device to communicate with other IoT or non-IoT devices. The perception layer can be considered as the physical layer. As explained earlier, IoT devices have sensors which collect information and actuators that perform actions within the external environment. These actions are performed by the IoT device hosted within the perception layer. Dos Santos et al. [34] attested that the fact that the 3-Layer architecture does not cater for the Business Layer must be taken into consideration. Furthermore, Alshohoumi et al. [35] attested that while this architecture is acceptable it is also considered as a trivial IoT architecture.

The four-layer architecture includes a transport layer (sometimes referred to as the data processing layer) between the application layer and the application layer from within the IoT three layered architecture. While the purpose of each layer is similar, they differ slightly to cater for the additional layer. Gandhi [36] defined the application layer as the topmost layer managing the delivery of applications to the users. The application layer also caters for session generation and the graphic user interface (GUI). Device and information management can be considered as the functions of the edge or transport layer. By leveraging the other layers, the transport layer can also provide access control and data exfiltration [36]. The definition for the Network layer (layer 2) and the Physical or perception layer (layer 3) within the four-layer architecture is identical to that of the three-layer architecture as defined by Goddard [32] and Lombardi [33] respectively. Figure 2.1 provides a visual layout of the four-layer IoT architecture. As defined by Gandhi et al. in Figure 2.1, the Network and Data Acquisition Layer and the Physical layer are considered as physical or hardware layers whereas the Edge or Transport Layer and the Application layer are considered as software layers [36].

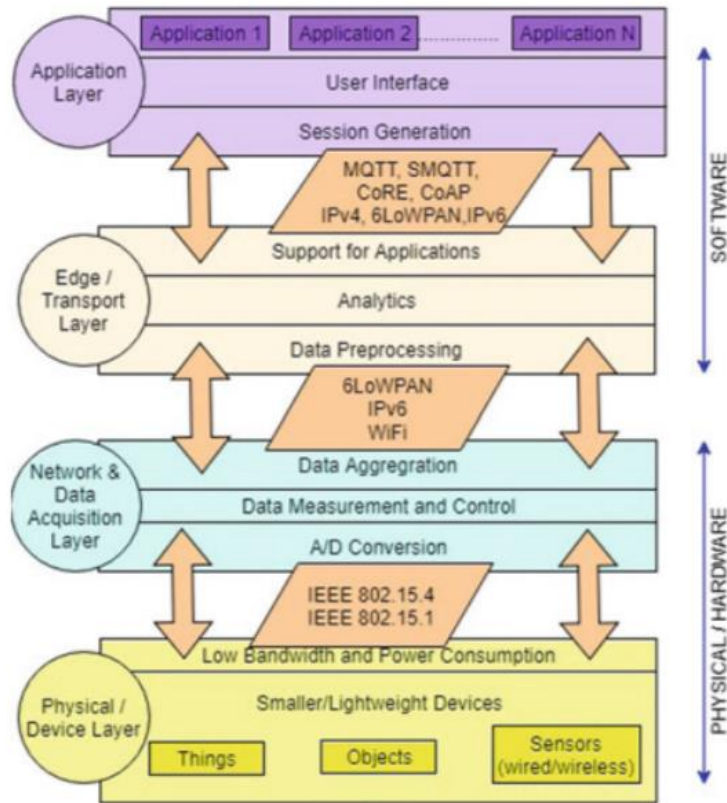


Figure 2.1: Four-layer architecture [36].

Given that the four-layer model does not cater for a business layer, researchers developed a new architecture known as the five-layer architecture [34]. Wu et al. [37] attested that the five-layer IoT architecture consists of an additional two layers, the Processing and Business layers along with the pre-existing three-layer architecture. Both the Perception and Application Layer serve the same functions as within the three-layer architecture whereas the transport layer is responsible for transporting data from the Perception layer to the Processing layer and vice versa (similar to the function of the Network layer within the three-layer architecture). The Processing Layer can be considered as the middleman in the five-layer Architecture with its key role being data extrapolation and filtering from the data received through the transport layer. Databases, intelligent processing, ubiquitous computing, or cloud computing are the main technologies being utilised within this layer [34], [37]. Wu et al. [37] stated that the development of IoT can be dependent on the research and development performed on this layer. The Business layer is imperative to communicate with other non-IoT management systems. Wu et al. [37] further attested that the business layer also serves to provide organisations with additional information to facilitate the decision-making process. Dos Santos et al. [34] noted that while the Business layer caters for data storage and processing, it does not cater for

privacy and security. Sethi and Sarangi [38] provided a graphical illustration (Figure 2.2) of both the Three-Layer (depicted as A) and Five-Layer IoT (depicted as B) architecture.

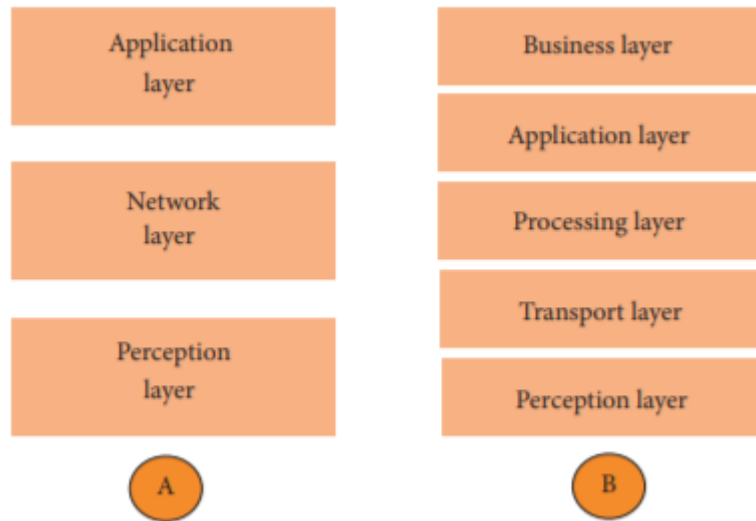


Figure 2.2: Three and Five Layered Architectures [38].

2.4 IoT Protocols

In order for data to traverse from layer to layer or device to device a network protocol is necessary. The Computing Technology Industry Association [39] noted that to transfer data within the same network, a set of predetermined rules need to be established. Network protocols can be considered as small sub-tasks within every network layer which ensure that the bigger task of transporting data is complete. When thinking about IoT, protocols do not tend to come to mind, so much so that the IoT industry focuses mostly on communication [40], [41].

Ansari et al. [42] noted that since IoT devices collect a huge amount of data there is a dependency whereby the protocols chosen can affect the application. Prior to sending data to the internet, IoT devices tend to communicate between each other and aggregate data. Specialised protocols to route communications between sensors had to be designed and developed [43]. Gregersen [40] further noted that it is imperative to choose the appropriate IoT protocols as without the appropriate protocols the IoT stack would fail. Salman and Jain [43] provided a list of IoT protocols used and classified these according to the layer.

Appendix A illustrates the IoT protocols as presented by Salman and Jain [43] along with the full form name and a definition of each protocol as provided by various authors.

3. Security Risks in IoT

Within the last ten years IoT has grown to the point that there are now billions of IoT devices residing within houses and organisations. The IoT environment is rapidly evolving, and companies want to develop and deploy devices to the market in the shortest time possible. Due to this a lot of the IoT devices are laden with security flaws and risks. Should an attack occur on an IoT device it will have a bigger effect when compared to a non-IoT device due to the extensive connectivity that IoT offers [44].

3.1 OWASP IoT Top 10

The OWASP was launched in 2001, as a non-profit organisation to aid companies and individuals improve software security. OWASP provides training and software projects that are open source in an attempt to help organisations create software programs which can be trusted by users [45]. In 2014, OWASP introduced the ‘Internet of Things Project’ to help the IoT stakeholders create and maintain safer IoT systems. In 2018, the team at OWASP identified that several organisations were releasing guidelines on how to secure IOT which were targeted for the different stakeholders in IoT such as developers, manufacturers, and users. To simplify the process, OWASP released the IoT Top 10 vulnerabilities to avoid for all IoT stakeholders [9]. This list was released through consensus amongst global security experts with the aim to converge the highest priorities into one list rather than having several lists aimed at the different participants or different risks, threats, and vulnerabilities within the IoT lifespan [9]. Table 3.1 illustrates the IoT Top 10 vulnerabilities as established by OWASP.

Table 3.1: OWASP IoT Top 10 [9].

Risk Number	Risk Title
1	Weak, Guessable, or Hard Coded Passwords
2	Insecure Network Services
3	Insecure Ecosystem Interfaces
4	Lack of Secure Update Mechanisms
5	Use of Insecure or Outdated Components
6	Insufficient Privacy Protection
7	Insecure Data Transfer and Storage
8	Lack of Device Management
9	Insecure Default Settings
10	Lack of Physical Hardening

Mukherjee [46], a cybersecurity consultant at Sectigo provided a breakdown of what each risk identified in the IoT Top 10 list entails and why these are considered as a risk.

1. Weak, Guessable, or Hard Coded Passwords

Default or fixed passwords are very common within IoT. Thus, leaving these devices prone to various attacks, such as brute force or dictionary attacks [46]. An example of unsecure credentials can be using ‘admin’ as both username and password. Alternatively, if a password is hard coded or left as default, a hacker can easily find the credentials of an IoT device through the user manual which can be easily found online [47]. Furthermore, Mukherjee [46] noted that while it is convenient for remote engineers to embed fixed passwords, it does provide a straightforward way for a hacker to login into the device. Backdoors for debugging reasons through an insecure firmware are also very common within IoT devices.

2. Insecure Network Services

Open ports are another means of how a hacker can gain access to the IoT device [46]. To reduce the possibilities of a hacker accessing IoT devices through insecure network services, Mukherjee [46] recommended that ideally a user should:

- Disable all unnecessary ports and any services which are either unnecessary or vulnerable;
- Segregate IoT devices from non-IoT devices using different networks;
- Perform regular patch management;
- Do not connect remotely to the devices using untrusted networks such as public Wi-Fi; and
- Disable any remote access services.

3. Insecure Ecosystem Interfaces

It is very easy for a perpetrator to access a device's web interface if this is not properly secured. Although APIs are very useful, they can be leveraged by a hacker, thus providing another entry point to the IoT devices [48]. Mukherjee [46] identified several flaws which can lead to an unauthorised user gaining access and the device being compromised. These flaws are 1) vulnerable authentication/authorisation methods, 2) no authentication/authorisation methods, 3) encryption cipher being used is weak and 4) weaknesses in data filtering.

4. Lack of Secure Update Mechanisms

Harper [49] noted that some of these IoT devices never get security updates after being launched in the market. The majority of IoT devices get updates until the manufacturers start designing and manufacturing new devices. Other devices run on legacy and probably unsupported Linux kernels. Thus, a device that is initially perceived by the user as a secure device, is not only insecure, but also vulnerable to hacking attempts. To ensure an update lifecycle, the United Kingdom proposed a law whereby a minimum period of time is established through which a manufacturer must provide security updates for a device [46]. Anti-rollback mechanisms, firmware validation and the security delivery of updates should also be implemented to further reduce risks and vulnerabilities [46].

5. Use of Insecure or Outdated Components

Third-party software or hardware which originate from a compromised supply chain along with the operating system customisation, that might be insecure can all be considered as IoT vulnerabilities [50]. Furthermore, since a supply chain vulnerability occurs early in the process these can remain unnoticed, impacting the security of the device greatly [46].

6. Insufficient Privacy Protection

Personal and sensitive information about the tenants or IoT owners is collected and collated by IoT devices [51]. This data can be compromised from either hacking the device through any of the means mentioned in the proceeding paragraphs or through leveraging the API interfaces that the IoT device interacts with [51]. In California, legislative actions are being discussed to secure the users' personal data [46].

7. Insecure Data Transfer and Storage

Data within the IoT ecosystem is stored, processed or transmitted. While the user can secure his means when transferring the data, the role of securing the passwords and databases rests on the manufacturer [47]. Access to the database and any sensitive data should be restricted, while data should be encrypted within any stage of its lifecycle [46]. Data is left vulnerable and a major IoT concern arises if data is not encrypted [46].

8. Lack of Device Management

There are various elements which need to be considered when assessing device management and these include security support and update management, system management (inventory of devices), monitoring and securely disposing and decommissioning of the devices when they reach their end of life [52]. Failing to perform any of the above can lead to the network or IoT device being compromised [46].

9. Insecure Default Settings

Hardcoded passwords or services using root or 'sudo' permissions are not exceedingly rare within IoT [53]. To apply default settings, the manufacturer has to keep the security of both the device and the user in mind [53]. Often it can be difficult or time consuming for a manufacturer to configure security when the time and cost can be spent implementing further smart functions [47]. Mukherjee [46] noted that in California a law has been passed whereby IoT device manufacturers cannot assign a default password but must either

implement a unique password to devices or force users to change these passwords prior to using the device. Even though this law has some practical limitations it does provide reassurance and is well-intended [54]. For any stakeholder making use of IoT devices such a law means that security capabilities now become a matter of always improving best practices and performing due diligence [54].

10. Lack of Physical Hardening

A hacker might gain access to the physical device and tamper with it at a physical layer. This poses risks for home tenants or organisations as a hacker might disable the sensors and consequently these do not function properly in cases of flood and fire [55]. Mukherjee [46] attested, that to physically secure IoT devices, manufacturers must 1) Design IoT devices with security in mind to mitigate all the attacks a hacker might attempt and 2) identify possible means of device modification or damage that a user or hacker might attempt.

Mukherjee [46] attested that OWASP published the IoT Top 10 list every two years and it was anticipated that in 2020 an updated version of the IoT top 10 would be released. However, this was not the case as the latest version of the OWASP IoT Top 10 was published in 2018.

3.2 Security Risks in IoT

Aydos et al. [56] attested that IoT attacks can be mapped according to layers within the IoT architecture which these attacks occur in. Furthermore, Table 3.2 provides a breakdown of the IoT threats and vulnerabilities which occur on the Physical, Network and Data Processing layers as defined by Aydos et al. [56] and the Application layer as defined by Raghuvanshi et al. [57]. A description of each vulnerability will follow in the succeeding paragraphs.

Table 3.2: IoT Threats and vulnerabilities on each layer [56], [57] .

	<i>Network Layer</i>			
	<i>Physical Layer</i>	<i>Network Layer</i>	<i>Data Processing layer</i>	<i>Application Layer</i>
IOT Threats And Vulnerabilities	Tampering	Man-in-the-middle (MITM)	Exhaustion	Cross Site Scripting
	Jamming	Spoofing	Malware	Malicious Code Attack
	Eavesdropping	Desynchronisation	Collision	Buffer Overflow
	Denial of Service (DoS)	Selective Forwarding		Phishing Attack
		Unfairness		Sensitive Data Permission or Manipulation
		Wormhole		Web Browser Attack
		Sybil		SQL Injection Attack
		Flooding		

Physical Layer

Marnewack [58] attested that a physical vulnerability can be either an invasive or non-invasive attack. In a non-invasive attack a hacker needs to be in close proximity to exploit the device and in consequence tweak the devices’ behaviour or attain information from the device. An invasive attack requires the chip surface to be physically manipulated. Aydos [56] stated that since the sensors directly collect the data, they are the

most vulnerable. Sensors are generally targeted by tampering and jamming; such attacks often lead to a DoS attack [56].

Tampering

An attacker can physically or electronically meddle with the program to either gather information or gain control of the device for ulterior purposes [59]. Tampering is considered as an invasive attack and a hacker might even attempt to alter the behaviour of the device [58].

Jamming

Jamming occurs when a hacker broadcasts malicious noise into the network to take control of the device [60]. A hacker might attempt to drain the device's battery faster than intended through retransmitting data continuously [61]. Such techniques can also be used to create a DoS attack [61].

Eavesdropping

When data is being communicated within the network, other people can intercept and listen to this communication [24]. The data communication occurs either on the wire or wireless which are both susceptible to attacks. Krishna et al. [62] attested that in IoT eavesdropping, traffic generated by IoT devices are sniffed and the data of the users is stolen through the creation of similar IoT devices. Another instance of eavesdropping in IoT would be leveraging devices which have microphones incorporated to eavesdrop on the tenants [62].

Network layer

Table 3.2 illustrated that vulnerabilities also exist at the network layer.

Man-in-the-middle attack

Čekerevac et al. [63] attested that through IoT, man-in-the-middle (MITM) attacks will be more common since IoT devices are normally hosted on unmonitored networks and never turned off. There are various techniques that can be used in an effort to cause a MITM attack such as Domain Name System (DNS) spoofing, evil twin, Secure Sockets Layer (SSL) hijacking, Address Resolution Protocol (ARP) cache poisoning and session hijacking. The main aim of a MITM attack is to create a proxy node which intercepts communication occurring between two other nodes. Čekerevac et al. [63] noted that through unvalidated SSL certificates and IoT refrigerators displaying Google calendars, attackers were able to steal the tenant's Google account.

Spoofing

Spoofing occurs when an attacker obtains access to a network by using the Media Access Control (MAC) or Internet Protocol (IP) address of a genuine device to mimic this device [64]. After gaining access to the network, the attacker can launch other types of attacks accordingly.

Desynchronisation

Aydos et al. [56] noted that in order for a desynchronisation attack to occur within IoT, an attacker needs to meddle and modify the parameters through which IoT devices communicate. The aim is to desynchronise device communication which was previously synchronous to cause network traffic to malfunction. The attack is achievable within the IoT space since these devices rely on wireless communication to communicate.

Selective forwarding

To maintain optimal communication within IoT multiple routing channels need to be established. A selective forwarding attack occurs when a perpetrator takes control of a node and modifies network traffic which the node receives to either redirect traffic or omit certain data packets from being sent. The other nodes as such receive either corrupted or missing data [56].

Unfairness

Unfairness can also be considered as an exhaustion-based attack or a repeated collision attack [56]. This type of attack aims at disabling load balancing mechanisms within the wireless networks. The verbosity of this attack increases according to the number of devices on the network and this might lead to disrupting or disabling a service.

Wormhole

Goyal and Dutta [65] noted that a wormhole is when a node on the network is compromised and listens into all the ongoing communication within the network. The wormhole attack is not easily detectable as it does not interrupt any network communication. Furthermore, the authors attested that apart from disrupting the routing optimisation, this attack violates both network confidentiality and integrity.

Sybil

A Sybil attack occurs when an attacker creates fake nodes on the network. To the other devices on the network these devices appear as genuine and as such communicate with the fake devices [66]. Additional network traffic and resources are consumed due to the fake nodes interacting with other nodes [56]. On a network operating with limited resources, this might lead to the network shutting down.

Flooding

Manda and Nalini [67] attested that a data flooding attack occurs when through route establishment with all network nodes, a device or node floods the network by sending useless data packets to all network nodes. The authors further noted that detecting which node is sending this data is very difficult.

Data processing layer

Aydos et al. [56] noted that within the data processing layer, all the data which is collected by the IoT sensors and devices gets processed within cloud systems. Malware within the data from edge nodes or sensors constitute the attacks within this layer.

Exhaustion

Exhaustion occurs when a jamming attack continues until the device's battery is depleted [68]. Aydos et al. [56] attested that these types of attacks attempt to disrupt the processing of data within the IoT infrastructure. Furthermore, with the distribution of cloud-based systems such attacks can be mitigated as protective measures can be implemented.

Malware

Malware is a generic name for viruses or trojan horses which can be injected within IoT data with the aim of gaining access into cloud or distributed systems. Such malware is difficult to detect and mitigate and protective measures prior to the data processing layer is necessary to mitigate such attacks [56].

Collision

A collision attack occurs when a compromised device transmits packets at the same time that a legitimate node transmits packets within the same channels [68]. As a consequence, the receiver receives two packets nearly at the same time rendering the message meaningless. Such attacks may cause the network to become unusable [56]. The chances of detection are very low with this type of attack and the energy consumed by devices to transmit messages is also low; making this type of attack attractive for hackers as it has less risks of being detected than jamming [68].

Application layer

Aydos et al. [56] attested that securing the application layer is complicated due to the amount of data being generated and processed in IoT environments. Traditional ways of securing IoT platforms such as authentication and data restriction are two means of securing IoT platforms.

Cross Site Scripting

When a hacker includes malicious code within a genuine link this is considered as cross-site scripting [69]. Once the user clicks on the link a genuine request is sent to the website along with the malicious script which is executed within the local browser as it is assumed as a trusted source. This leads to the user being compromised without his knowledge and the attacker gaining access to browser sessions and possibly hijacking the user's web account.

Malicious Code Attack

Malicious code can take form in many ways such as scripting languages, pushed content, ActiveX controls, Java Applets and plug-ins [70]. It tricks the user into installing a genuine looking program thus allowing an application backdoor and attackers remote access. Once the attacker has remote access, the attacker can perform further attacks such as stealing the user's data or deleting his files.

Buffer Overflow

When an attacker sends arbitrary code to a program, it will try to store such input in a buffer. Should the buffer be smaller than the input data, it will proceed to overwrite data onto parallel memory slots. The original data within the buffer would generally include a return pointer which can be considered as the next address for the process. If an attacker changes the values of the return pointer to a location where a malicious file is located, the process will eventually redirect control to the malicious file [71].

Phishing Attack

Abbas et al. [72] attested that phishing attacks are becoming more sophisticated and are always on the rise. Such attacks prove to be one of the biggest risks causing data breaches. This type of attack aims at using different means (mainly emails) to deceive the user into inputting his credentials through genuine looking messages. As previously noted, various means can be used to construct phishing attacks such as social

media, emails, forms, websites and mobile applications. Attacks sometimes use spearfishing which is a well-crafted phishing attempt which includes sensitive information and is more personalised to users. This gives rise to faster reaction times from targeted users. Abbas et al. [72] also noted that even IoT devices can be used to spam users by sending a huge number of emails.

Sensitive Data Permission or Manipulation

In these types of attacks, the privacy of tenants or users is breached due to the fact that the data being held on the IoT infrastructure is accessed by unauthorised or unauthenticated users. Raghuvanshi et al. [57] attested that in such attack scenarios, attackers identify and leverage faults within the device's authorisation model. Physical theft or device shutdown is also attainable if such attacks occur on IoT devices.

Web Browser Attack

To achieve unauthorised or unidentified access attackers leverage the fact that browsers do not generate XML tokens by themselves. Furthermore web-based services generate a lot of metadata which pose a security risk should it fall in the hands of hackers or attackers [57].

SQL Injection Attack

Bitdefender [73] attested that although a Structured Query Language (SQL) attack seems like it is a complicated attack, this is not the case. These types of attacks attempt to gain unauthorised access to the database through inputting a genuine request along with malicious instructions that are executed along with the request. The reason why a hacker would want to access the database is that a database holds sensitive information such as login credentials along with permissions which can be leveraged to gain further access.

4. IoT Pen Testing Tools

As discussed in the previous Chapters, the availability of diverse IoT products and their respective functions provide a multitude of ways to improve environments, organisations, industries, cities and human lives alike. However, through research findings it was noted that these devices are predisposed to vulnerabilities, threats and security risks. Guaranteeing that these smart devices are secure must be the common obligation and responsibility of all stakeholders involved.

Regalado [74], principal security engineer at ZingBox, attested that when conducting IoT penetration testing the challenges vary and are more complicated when compared to traditional penetration testing. This is due to the fact that IoT is much more diverse than traditional systems, since IoT have different operating systems, communication protocols and architectures. Communication protocols such as Zigbee, Near Field Communication (NFC) and Bluetooth Low Energy (BLE) require different tools and insight to perform pen testing on IoT. Furthermore, there are various architectures in IoT such as Advanced RISC Machines (ARM) and Microprocessor without Interlocked Pipelined Stages (MIPS) whereby traditional computers make use of x86 and x64-bits systems. Regalado [74] attested that traditional penetration testers might get confused by the vulnerabilities and protocols of such embedded devices. Penetration testing is becoming a fundamental process within an IoT lifecycle. There are several penetration testing tools to choose from. Some of them have similar functions. As discussed in the paragraphs below, it can be noted that there are several tools that can assess the security state of IoT devices. This is a significant and continuous process carried out through the effective and appropriate use of security tests. From a research point of view, the significance of the penetration testing (Pentest) is clear. This is mainly due to the fact that during the past years there was a significant increase in the number of flaws and vulnerabilities within the IoT field.

From the conclusions and recommendations within the literature, the author established which tools or methodologies will potentially be used within this research study. Due to the word limit of this project the author limited the discussion on a number of penetration testing tools with the aim of utilising these tools within the Case Scenarios presented within Chapter 6. The established tools will be discussed below.

4.1 PENIOT

Accelerating the security testing process to enable security testers to detect and mitigate security flaws through automation is the main goal of PENIOT. PENIOT is considered as one of the first penetration

testing tools specific to IoT. It offers the functionality to perform both passive and active security attacks in either an automated or semi-automated environment. This tool performs mostly protocol-based security attacks [44]. Cankar [44], the author who published PENIOT on GitHub, attested that since IoT devices are on the rise, the team opted to integrate those protocols which are mostly used within IoT. PENIOT caters and provides several exploits such as DoS, Sniffing, replay attacks and fuzzing for devices using the Advanced Message Queuing Protocol (AMQP), BLE, Constrained Application protocol (CoAP) and Message Queueing Telemetry Transport (MQTT) protocols. Furthermore, the tool offers the possibility to export or import attacks and exploits from a user-friendly GUI thus making it extensible [75].

4.2 Tenable Nessus

Nessus is a network vulnerability scanning tool by Tenable which leverages common threats and vulnerabilities. Some of the features that Nessus offers are vulnerability scanning and assessment on the network, web, server or services along with Policy Management, Vulnerability mitigation prioritisation and asset discovery. Furthermore, Nessus allows for the security vulnerability detection through simulated attacks or the detection of missing security updates and patches. Nessus also caters for compliance testing such as Payment Card Industry Data Security Standard (PCI DSS) compliance tests. Scans can also be scheduled to run at any time of the day. Through the use of plugins and by rigorously testing each open port on a server or computer, Nessus is able to identify the services that are running and detect any vulnerability which these services might have. Nessus is capable of detecting whether authorisation vulnerabilities reside on the device or if the device has been configured using either common or default passwords. It is also able to detect if the device is susceptible to DoS attacks, whether the running services on the device have missing patches or any other software flaws that might exist on the machine [76].

4.3 OpenVAS

Open Vulnerability Assessment Scanner (OpenVAS) is a tool distributed by Greenbone Networks and used for vulnerability scanning. Greenbone bases its business on three pillars. These pillars are to provide intuitive vulnerability scanning solutions, to provide an out of the box solutions for enterprises and to ensure that the project remains an open-source concept that ensures transparent security [77]. This tool is freely distributed and easy to use whilst also updated on a daily basis through Network Vulnerability Tests (NVTs) to ensure that it remains relevant [78]. A paid for version exists offering further options to perform vulnerability scanning which are aimed mostly at enterprises or organisations level. Meanwhile, the free version is generally a good choice for small businesses and independent testers who wish to quickly identify

potential threats and vulnerabilities residing on their devices. Various tests come built-in and like Nessus, provides scheduled and credentialed scans. Out of the box, OpenVAS provides eight security scan configurations, however it allows for the creation of new scan configurations to be as customisable as possible [77].

4.4 ZAP

ZAP is a tool designed to be both a flexible and extensible web application security testing tool being maintained by OWASP. It is an easy-to-use tool intended for a variety of IT professionals such as developers and security testing specialists [79]. The tool can be considered as a proxy between the web application and user's browser which intercepts analyses and, in some occasions, modifies the messages prior to forwarding them to the recipient. ZAP can also be used to perform web application security automated testing whereby it scans for any OWASP Top 10 vulnerabilities present on the website. Several add-ons can be added onto ZAP whilst the source code can be analysed as it is an open-source solution.

4.5 BURP Suite

Burp Suite, a tool developed by Portswigger, allows for automated or manual web application penetration testing. The aim of Burp Suite is to be a multi-tool with the possibility to further extend the number of tools through BApps which are add-ons [80]. The number of tools which Burp Suite offers along with the amount of extension tools makes it a very popular web application penetration testing tool for both researchers and bug bounty hunters. Burp Suite has eight main features which consists of 1) Spider, 2) Proxy, 3) Repeater, 4) Sequencer, 5) Decoder, 6) Extender, 7), Intruder, and 8) Scanner [80]. A brief description of each function is provided below.

1. Spider: This reconnaissance functionality provides a list of all the endpoints within the web application. Through this list, each endpoint can be investigated for potential vulnerabilities.

2. Proxy: This feature is similar to the features defined for ZAP whereas Burp can serve as a man in the middle proxy between the browser and web application.

3. Repeater: The repeater function within Burp Suite allows for users to modify the web requests manually and forward these to the browser. This feature is one of the most versatile and can be used for various Hypertext Transfer Protocol (HTTP) requests modifications.

4. Sequencer: The sequencer functions check for any common patterns which can be identified in tokens used for authentication of cookies or anti-CSRF tokens.

5. Decoder: As the name suggests this function enables Uniform Resource Locator (URL), BASE64, HEX and other encoding methods to be decoded. This tool is very commonly used to extract value from parameters or headers.

6. Intruder: This function is also known as a fuzzer whereby batches of values are pre-loaded into an input point and the output is analysed to check whether a success or a failure has been received. Brute-force and dictionary attacks can be crafted using this function.

7. Extender: Burp Suite also caters for external extensions which can be downloaded and integrated with the tool to further its capabilities. These are called BApps.

8. Scanner: The scanner tool within Burp Suite allows for the tool to automatically scan the website to detect vulnerabilities whilst also providing Burp's confidence level and the difficulty for a hacker to exploit these vulnerabilities. The scanner functionality is updated on a regular basis to cater for new or less frequently exploited vulnerabilities [80].

4.6 Nmap

Network Mapper (Nmap) is a free port-scanning tool which is open-source. It can be used for network discovery and vulnerability scanning. The main aim of Nmap is for users to determine which services are running on a device through open ports on the machine. This is achieved by Nmap sending raw packets to specific ports and listening to responses to define whether a port is open, closed or filtered. Since different protocols or services use one specific port such as HTTP using port 80, a user can quickly enumerate services through Nmap [81].

4.7 Kali Linux

Day [82] considers Kali to be a Swiss army knife operating system when it comes to penetration testing. Built with digital forensics and penetration testing in mind, Kali is a free Debian-based Linux distribution. Since it comes with over 600 tools pre-installed, Kali is very popular with penetration testers and

cybersecurity professionals as these professionals do not have to install tools individually. The Kali operating system is regularly updated and improved upon by Offensive Security.

4.8 Hydra

Singh [83] attested that the Hydra tool is a password detection or cracking tool which can be utilised in various scenarios. To detect or crack a password this tool uses a dictionary attack or a brute force attack attempting various credentials on the login form or authentication in protocols such as Telnet, Hypertext Transfer Protocol Secure (HTTPS), and other protocols [83]. This tool comes preinstalled on the Kali operation system mentioned above.

5. IoT Hardening Procedures

5.1 Introduction

The IoT is an emerging concept and the same can be attested to the security challenges, risks, threats, and vulnerabilities brought by the implementation and usage of these devices. As previously discussed in some sections of the preceding Chapters, the mitigation of the latter mentioned challenges is important as in most cases, attacks on IoT devices can severely threaten individuals' safety, privacy, and security [3].

In quest to address the Questions presented below, the author followed the flow illustrated in Figure 5.1.

- **RQ 1:** *Which options do smart homeowners have when they are concerned with hardening their IoT devices?*
- **RQ 2a:** *Which hardening procedures identified as a reply to Question 1 originate from established organisations within the cybersecurity field?*
- **RQ 2b:** *Which three hardening procedures established in Question 2a can be considered as safe to be utilised for the experimental and theoretical part of this study?*
- **RQ 3a:** *Which controls from the three established hardening procedures can be implemented within a smart home and within this study?*
- **RQ 3b:** *Which OWASP IoT Top 10 vulnerabilities are mitigated through the controls established in Question 3a?*
- **RQ 4:** *Can the controls identified within Question 3a be merged into one hardening procedure?*

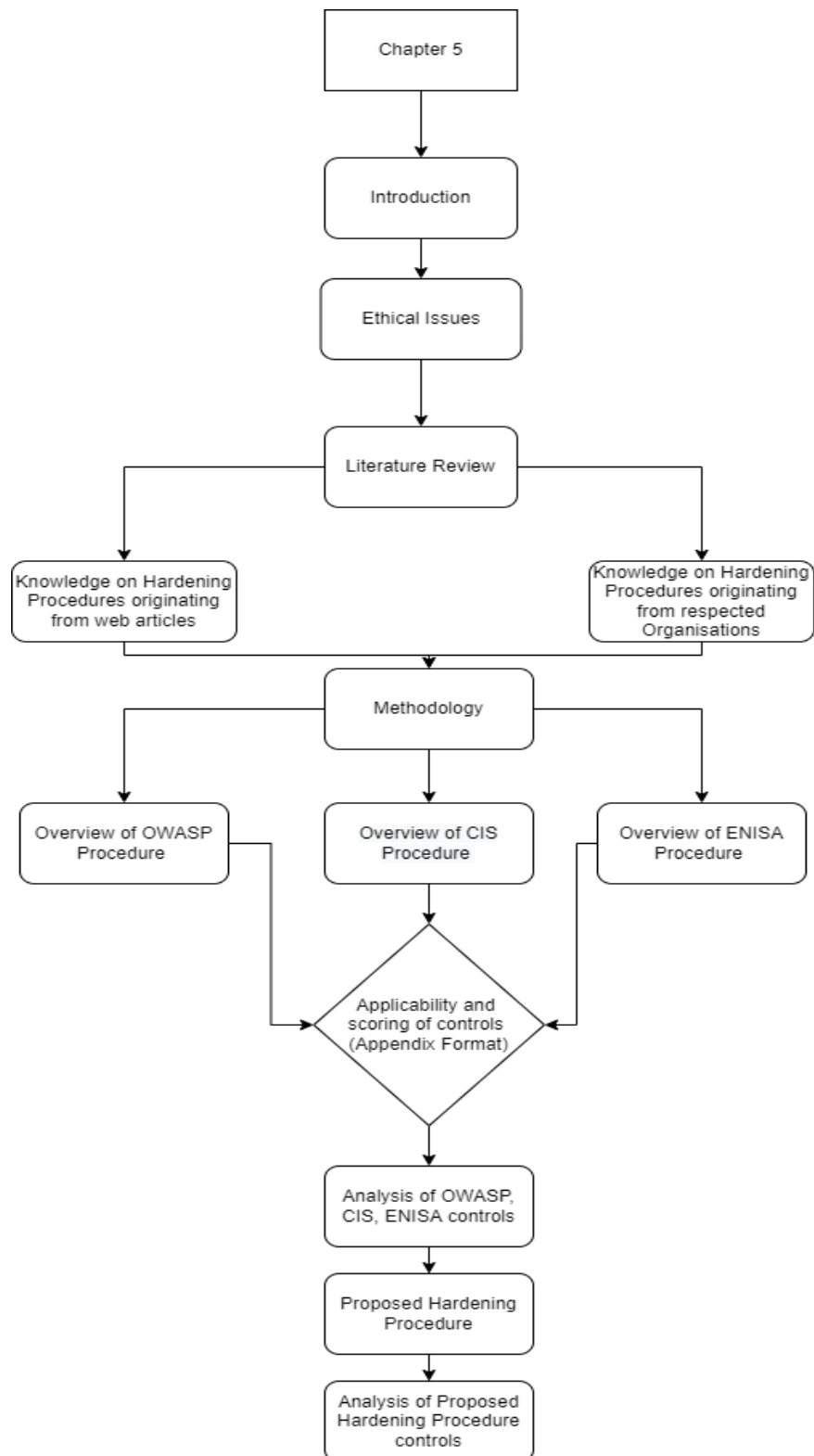


Figure 5.1: Outline of Chapter 5: Hardening Procedures.

Throughout this project the author had to make a decision on which methodological approach is to be utilised. The structure of this research was defined through the research questions being answered and through the author's knowledge and insight. The methodological approaches chosen by the author for this study were an analytical review on IoT hardening procedures and Case Scenarios to test the effectiveness of the hardening procedures within an experimental environment. The findings of the two methodological approaches used in this research study will be discussed within the security analysis provided in Chapter 7.

5.2 Ethical Issues

As a researcher of this study, the author provided the Project Description Form (PDF) and sought permission to conduct the research from the Royal Holloway, University of London. The author acknowledges that researchers have responsibilities to carry out and produce research findings in an ethical manner to uphold research integrity [84]. Moreover, when designing and while conducting this study's research, the author was cognisant that there were legal and ethical principles that had to be followed [84]. Fundamental to these ethical principles, there are (a) beneficence (to do good) and (b) non-maleficence (to do no harm). To ensure professional integrity throughout the research process, the author as a researcher practised within established ethical boundaries that guide every research [84]. The author strived to be honest in respect to his actions throughout the whole research process.

Furthermore, since the study did not include organisations and participants as a sample, the researcher did not need to obtain informed consent from prospective research participants or organisations. Additionally, since the study did not involve participants and since the tests performed were carried in a safe testing environment (no data could have been compromised) some ethical principles such as (a) minimise the risk of harm to participants; (b) protection of the participants' anonymity and confidentiality; (c) participants given the right to withdraw from this research, where not applicable and were not adopted to guide this research. However, although the latter mentioned ethical principles were not applicable to this research study, the author is aware of their respective implications when research is being carried out.

5.3 Literature Review on Hardening Procedures

In quest to answer Research Questions *RQ1*, *RQ2a*, and *RQ2b*, the author conducted a literature review aiming to establish what IoT hardening procedures exist. The search focused on IoT; specifically, those hardening procedures targeted at smart home users securing their devices. While conducting the literature search, the author identified specific keywords to be used. In particular, the search was done by using the keywords listed below:

1. *IoT Hardening Procedures;*
2. *IoT Hardening;*
3. *Smart Home Hardening Procedures;*
4. *Smart Home Hardening.*

To gather the full picture of what literature research is available through websites and web-articles the author performed a Google Search with each keyword outlined above. It was noted that from this search the results were extensive, and thus, the author had to limit the review to two research-based articles deemed relevant as providing the most secure controls for a smart home user. The author does understand that ideally more articles could have been reviewed and analysed for this section. However, due to time constraints the author established and selected articles on each keyword that enabled him to identify, in a structured way, the appropriate, relevant, and related information. Furthermore, the author used the articles that in his opinion and professional judgement were considered as the most likely to generate knowledge to IoT devices homeowners and that will be valuable in informing the general public. These articles are summarised in Tables format as indicated from Table 5.1 till Table 5.4. Each Table will present the synthesis of the two articles reviewed for each keyword. Hence, the keyword used, the websites, article titles, and synthesis of the articles are presented.

Table 5.1: Summary of Hardening Procedures identified through Keyword 1 ('IoT Hardening Procedures')

Keyword Used	Website Name	Article Title	Summary
IoT Hardening Procedures	Techopedia	10 Steps to Strengthen Your IoT Security	Raja [17] provided 10 hardening steps as per below: <ol style="list-style-type: none"> 1. Harden the network; 2. Apply all updates to devices as they are released; 3. Dig into documentation; 4. Ban auto-connection; 5. Use protections against DDoS attacks; 6. Embrace the IoT with strong protections; 7. Partition off the IoT from the rest of the network; 8. Perform secure endpoint hardening; 9. Use security as a part of the buying process; 10. Update the passwords.
	Hewlett Packard Enterprise	9 ways to improve IoT device security	Strom [85] provided 9 hardening steps as per below: <ol style="list-style-type: none"> 1. Implement restrictive network communications policies, and set up virtual Local Area Network (LANs); 2. Use encrypted protocols to secure communications; 3. Improve failover design; 4. Create an industry-backed program for "Secure IoT Device" labelling; 5. Create bug bounty programs and vulnerability reporting systems; 6. Design explicitly for privacy and security; 7. Understand that device firmware can be insecure; 8. Create more effective and secure password policies; 9. Secure and centralise the access logs of IoT devices.

Table 5.2: Summary of Hardening Procedures identified through Keyword 2 ('IoT Hardening')

Keywords Used	Website Name	Article Title	Summary
IoT Hardening	SumoLogic	What is IoT Security?	<p>SumoLogic [86] identified six challenges of IoT Security and provided solutions on how these challenges can be solved. The challenges identified and the respective solutions provided by SumoLogic [86] are identified below:</p> <ol style="list-style-type: none"> 1. Challenge: Device monitoring and management. Recommendation: Ensure that IoT devices are onboarded in asset tracking systems. Unique identifiers per device should be provided by manufacturers. 2. Challenge: Physical hardening. Recommendation: Manufacturers should design devices to be tamper-proof. If a device goes offline unexpectedly users should investigate whether the device was physically tampered with. 3. Challenge: Lack of device authentication. Recommendation: Users should require the devices to authenticate prior to joining a network. 4. Challenge: Outdated components. Recommendation: Devices which are vulnerable should either be updated or replaced. 5. Challenge: Patching and upgrading. Recommendation: Upgrades should be designed to either be automatic or a one-step process. 6. Challenge: Embedded passwords. Recommendation: Create a strong password during device setup.
	GitHub	IoT Hardening	<p>Gandhi [87] recommended five hardening steps and the controls necessary so that these steps can be performed. The steps are:</p> <ol style="list-style-type: none"> 1. Setup Anomaly monitoring; 2. Change install-time defaults; 3. Configure and enable a firewall; 4. Software updates; and 5. Review running services.

Table 5.3: Summary of Hardening Procedures identified through Keyword 3 ('Smart Home Hardening Procedures')

Keyword Used	Website Name	Article Title	Summary
Smart Home hardening procedures	Norton	12 Tips to Help Secure Your Smart home and IoT devices	<p>Symanovich [88] provided 12 hardening steps as per below:</p> <ol style="list-style-type: none"> 1. Watch out for outages; 2. Do the two-step; 3. Keep your software up to date; 4. Check the setting for your devices; 5. Change default usernames and passwords; 6. Use a strong encryption method for Wi-Fi; 7. Avoid public Wi-Fi networks; 8. Audit the IoT devices already on your home network; 9. Disable features you may not need; 10. Use strong, unique passwords for Wi-Fi networks and device accounts; 11. Set up a guest network; 12. Give your router a name.
	Cybertakes	How to Improve the Security of Your Smart Home Devices	<p>Allen [89] provided ample information relating how to improve the smart home environment mainly by:</p> <ol style="list-style-type: none"> 1. Securing the home network wireless router; 2. Improving the overall security of the home network; 3. Never using default passwords; 4. Replacing non-supported devices; 5. Turning off unnecessary features; 6. Verifying the device's privacy and security settings; 7. Making use of two-factor authentication where possible; and 8. Checking and installing security and firmware updates.

Table 5.4: Summary of Hardening Procedures identified through Keyword 4 ('Smart Home Hardening')

Keyword Used	Website Name	Article Title	Summary
Smart Home hardening	Tom's Guide	How to Secure Your (Easily Hackable) Smart Home	Rashid [90] provided information relating to the risks of IoT along with ways of protecting both the IoT perimeter and the IoT devices themselves. Such means are: <ol style="list-style-type: none"> 1. Securing the wireless network; 2. Disabling guest network access; 3. Having more than one different Wi-Fi networks; 4. Giving these Wi-Fi networks an obscure name; 5. Making use of a firewall; 6. Adopting a good password management strategy; 7. Installing security software; 8. Paying attention to brands; and 9. Checking for firmware updates.
	PC Mag	How to Protect Your Smart Home from Hackers	Cohen [91] provided several steps and information about how to protect smart homes mainly through the hardening the network through the below: <ol style="list-style-type: none"> 1. Securing the Wi-Fi Network; 2. Managing Account Passwords; 3. Enabling Two-factor authentication; 4. Updating firmware; 5. Replacing Outdated Routers; 6. Splitting up the network; 7. Monitoring the network; 8. Considering what the user needs.

From the evidence established in the literature and as presented in the Tables 5.1 till 5.4 it can be concluded that the authors ([17], [85], [86], [87], [88], [89], [90], [91]) of each respective article provided their own IoT hardening procedures or recommendations. However, the author concluded that although the findings from this literature review provided information about IoT hardening procedures, the number of steps recommended within each article are very limited and some hardening steps might overlap. For this reason, further analysis of the literature was carried out and primarily focused on organisations providing the most consumer oriented hardening procedures using the same set of keywords.

Allen [92] attested that there are few IoT security standards which are published by organisations within the cybersecurity field, and these are not commonly available. Furthermore, from the literature review the author concluded that the vast majority of these IoT Security standards provide guidelines for manufacturers and developers, rather than for smart home users. The ENISA has published several publications related to IoT such as the ‘Good practices for security of IoT ’ [93] targeting IoT software developers, and ‘Security and Resilience of Smart Home Environments’ [94] which targets manufacturers and third-party developers. In 2017, ENISA published another publication named ‘Baseline Security Recommendations for IoT’ [3] that had several controls which could be applicable to smart home users albeit targeted for the same target audience as the two other publications. The European Telecommunications Standards Institute (ETSI) [95] published ‘Cyber Security for Consumer Internet of Things: Baseline Requirements’ in 2020 which provides controls that can be implemented within a smart home scenario, although it relies on the manufacturer to implement these controls beforehand. Furthermore, The Cloud Security Alliance (CSA) [96] has issued the ‘CSA IoT Security Framework version 2’ in 2021. Through the literature review it was noted that the CIS [97] had also launched their own ‘CIS Control Internet of Things’ version 8 along with a companion guide titled ‘CIS Controls Internet of Things Companion Guide’ which provides further insight on how each control within ‘CIS Controls Internet of Things’ can be implemented. It can be noted that although this publication was published to target enterprises, a significant amount of these controls can be implemented by a smart home user. The ‘Internet of Things (IoT) Security Best Practices’ is the security standard developed by the Institute of Electrical and Electronics Engineers (IEEE) [98]. With reference to the standard, ‘Internet of Things (IoT) Security Best Practices’ Corser et al. [98] stated that:

“This paper is intended for an educated lay audience. The recommendations offered in this paper are generally intended for implementation by manufacturers of IoT products, however they are also designed to be readable by nontechnical but well-educated

lawmakers, corporate and governmental policy makers, and participants in standard setting bodies” [98].

The OWASP [21] established a standard which provides applicable controls that can be used within a smart home Case Scenario. This publication is called the ‘IoT Security Verification Standard’ or the OWASP ISVS [21]. Furthermore, in 2018, the International Organisation for Standards (ISO) started developing their own standard titled ‘ISO/IEC FDIS 27400’ which would provide guidelines on IoT security and privacy. At the time of writing of this study, the latest status update on ‘ISO/IEC FDIS 27400’ was in the final stages of approval.

5.4 Methodology

The literature review on IoT, Smart homes, and IoT hardening procedures enabled the author to collect and synthesise previous research findings. The literature review was also a methodological tool that provided answers; while the knowledge gained enabled the author to address the research questions [99]. Furthermore, the knowledge gained through the literature review and insight on research paradigms guided the author to articulate theoretically informed approaches to produce data [100], [101], [102], [103]. The literature review enabled the author to answer Research Question *RQ2b*. Following an analysis of the literature, the author identified the three IoT hardening procedures which originate from established cybersecurity organisations that were used in this research study. These three hardening procedures chosen are 1) ENISA’s ‘Baseline Security Recommendations for IoT’ [3], 2) the CIS ‘Controls Internet of Things Companion Guide’ [2] and 3) OWASP IoT ‘Security Verification Standard’ [21]. Following this step, the author identified the methodological approach to be used in this part of this study and thus provided answers for the research questions *RQ3a* and *RQ3b*. An answer for Research Question *RQ4* was then established through the findings of the Research Questions *RQ3a* and *RQ3b*. The research design supported detailed and deeper investigation to answer the research questions that guided the research project. The research design linked the data to be collected with what was being foreseen to be achieved from the initial questions of the study [104].

5.4.1 Identifying which controls are applicable and implementable within a smart home environment

From the literature review it was established that the three hardening procedures to be used in this study were developed to target manufacturers or organisations. Thus, it was noted that not all controls were applicable and/or implementable within a smart home environment. For this reason, a set of criteria were developed by the author to verify the applicability of each control to this study. Each control within each respective hardening procedure was assessed on the criteria listed below, taking into consideration the area being studied, i.e., smart homes:

- 1) Is the control only implementable by a manufacturer or organisation?
- 2) Can the control be implemented within a smart home?

Depending on the answers to the questions identified within the criteria, a control was verified to be either applicable or not. If the answer to criteria 1 was ‘no’ and the answer to criteria 2 was ‘yes’ then the control was considered as applicable. As an example, from CIS controls, control 15.7 stating ‘Securely Decommission Service Providers’ would only be implemented by organisations or manufacturers and thus was not applicable to this study. However, CIS control 12.4 stating ‘Establish and maintain Architecture Diagram(s)’ can be implemented by a manufacturer or organisation but it can also be implemented by a smart home user.

5.4.2 Score each applicable control according to which OWASP IoT Top 10 vulnerability it aims to mitigate

Additionally, to evaluate applicable controls and hardening procedures, the author sought to identify a way to score each control. Given that these hardening controls are designed to limit or mitigate vulnerabilities or threats within IoT, the author identified the OWASP IoT Top 10 list as a baseline score to be assigned to each control. A scoring system was devised depending on the criticality of the vulnerability. Each IoT Top 10 vulnerability was assigned a number from one (1) to ten (10) depending on the criticality of the vulnerability. Thus, vulnerability number 1 was assigned 10 points on the basis that this vulnerability is the most critical and needs to be mitigated immediately. Table 5.5 illustrates how the score rating of each control was assigned.

Table 5.5: OWASP IoT Top 10 Vulnerability with the score rating for each control

OWASP IoT Vulnerability Number [9]	OWASP IoT Vulnerability Name [9]	Score Assigned in Controls
1	Weak, guessable or hard coded passwords	10
2	Insecure network services	9
3	Insecure ecosystem interfaces	8
4	Lack of secure update mechanisms	7
5	Use of insecure or outdated components	6
6	Insufficient privacy protection	5
7	Insecure data transfer and storage	4
8	Lack of device management	3
9	Insecure default settings	2
10	Lack of physical hardening	1

Each control deemed applicable for this research; as established according to the criteria in section 5.4.1 was assigned a score depending on the IoT Top 10 vulnerability mitigated. For example, CIS Control 1.1 was deemed an applicable control and was considered by the author to mitigate OWASP IoT Top 10 vulnerability number 8 ('Lack of device management') and thus this control was assigned 3 points. In cases where a control was considered to mitigate more than one vulnerability, the control was assigned points according to which mitigated vulnerability has the highest number of points. As an example, CIS control 4.7 stating 'Manage Default Account on Enterprise Assets and Software' was deemed an applicable control and mitigates OWASP IoT Top 10 vulnerability number 9 ('Insecure default settings') and OWASP IoT Top 10 vulnerability number 1 ('Weak, guessable or hard coded passwords'). Since OWASP IoT Top 10 vulnerability number 1 has a higher number of points (10 points) assigned to it when compared to IoT Top 10 vulnerability number 9 (2 points) this respective control was assigned 10 points.

The applicable controls were identified by performing the methodological steps mentioned above on each hardening procedure. Furthermore, analysis on the applicable controls identified through the aforementioned process was performed and thus results answered Research Question *RQ4*. Through this process a merged hardening procedure was proposed. A comparison of the proposed hardening procedure with the three hardening procedures outlined above was carried out.

The forthcoming sections of this Chapter are divided into five as indicated below:

1. An overview of the OWASP ‘IoT Security Verification Standard’ along with the applicable controls and the score of each control;
2. An overview of the CIS ‘Controls Internet of Things Companion Guide’ along with the applicable controls and the score of each control;
3. An overview of the ENISA ‘Baseline Security Recommendations for IoT in the context of Critical Information Infrastructures’ along with the applicable controls and the score of each control;
4. An analysis of the three hardening procedures mentioned above;
5. A proposed hardening procedure which entails all the controls identified above, will be provided whilst omitting any similar controls will be provided. Analysis/Comparison of the proposed hardening procedure with the aforementioned hardening procedures will be provided.

5.5 OWASP IoT Security Verification Standard (ISVS)

The OWASP ‘Internet of Things Security Verification Standard’ (ISVS) was designed around the ‘Application Security Verification Standard’ (ASVS), which is nowadays being used within various industries to verify web-applications and web services security controls. The main aim of the ISVS is to provide baseline security requirements for IoT applications [105]. OWASP [21] attested that the ISVS was designed to provide best practices and requirements for IoT devices, which would in return increase the confidence level of IoT application’s security posture. Since IoT applications are made up of a multitude of applications and devices connected together, securing the ecosystems in which these applications reside will also secure the applications. Whilst also making use of already developed industry used standards, the ISVS provides security requirements for IoT applications, devices and the infrastructure they reside in [21].

ISVS security controls are divided into five segments which can be represented as a five-layer architecture as illustrated in Figure 5.2. The first segment of requirements relates to the IoT ecosystem and are aimed to secure the ecosystem where the IoT device is connected. To secure the application, the user space

application requirement layer was developed, and this is found just below the IoT ecosystem requirements. The next layer is divided into two segments which are the Software Platform and the Communication Requirements. Lastly, the Hardware Platform Requirements are found at the bottom of the architecture which provide requirements to secure the various hardware components found within an IoT device [21].

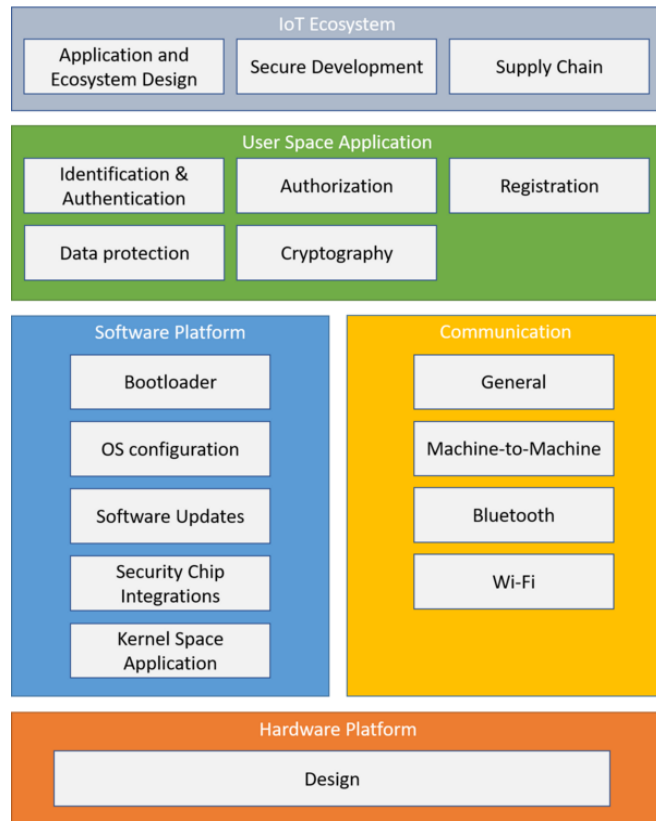


Figure 5.2 ISVS Security Model stack [21]

To cater for the various security-sensitive IoT devices' features and capabilities, OWASP [21] established three security verification levels within the ISVS, with the security depth further increasing per level. Level one requirements attempt to protect against attacks which are non-physical and attacks which do not require physical access to the machine. The aforementioned attacks generally consist of software attacks and devices should be protected to this level when no sensitive data is being hosted on the device or if no lateral movement onto the IoT network is possible should the hardened device be compromised [21]. The aim of the ISVS level two requirements is to harden devices from hardware attacks rather than software attacks. Devices should be hardened at least up to ISVS level two when hosting personal or sensitive information. The third level provides further controls through defence-in-depth mechanisms, which build upon the requirements set in level one and level two to reduce the chances of physical tampering and reverse

engineering. This level is applicable to devices where compromise of the IoT device would lead to fraud or when these devices store very confidential information [21]. OWASP [21] noted that the controls are written in a way that can be measured and achieved in practice. Furthermore, each requirement can be used within a different stage of the IoT's device lifecycle.

The OWASP ISVS controls that were deemed applicable within a smart home environment along with the score assigned to each control are included in Appendix B. Furthermore, insight regarding the OWASP ISVS scoring and applicable controls is provided within sub-heading 5.8 of this Chapter.

5.6 CIS Controls Internet of Things Companion Guide

Each industry or author provides a different definition for IoT. This bares its own issues and is considered by CIS as one of the many challenges that the IoT environment has. CIS is a non-profit community-driven organisation which launched the worldwide and industry recognised CIS Benchmarks and CIS Controls. These Benchmarks and Controls are documents utilised by system owners providing best practices to secure IT data and systems whilst also safeguarding against newly discovered threats [106]. CIS attested that IoT has three issues which are ubiquity, uniqueness and ecosystem, meaning that the IoT environment includes a lot of devices which consist of software, hardware and firmware that are all developed by several manufacturers [97].

The IoT provides security professionals elaborate and distinctive challenges especially when used within organisations or enterprises. Standard hardening does not apply for these devices as often they are connected within the same networks used by employees or otherwise connected directly to the internet [97]. In 2021, CIS launched the 'Internet of Things Companion Guide version 8'. This version provides a list of tasks that a user can utilise to secure IoT devices whilst also mitigating common threats and vulnerabilities [97]. The CIS 'Internet of Things Companion guide' is broken down into 20 controls, each having several sub-controls. CIS [97] stated that the 'Internet of Things companion guide' is not aimed at specific sectors or organisations but it is aimed to be a generic guide for anyone (be it an organisation or an IoT system owner) attempting to secure their system or systems. Furthermore, while the CIS controls are aimed to provide common baseline controls to secure a system, these controls do not cater for environments requiring unique or additional controls [97]. By working together with subject matter expert volunteers, the CIS developed the 'Internet of Things Companion Guide version 8' to help organisations implement best

practises across a range of connected devices [97].The CIS attested that the approach utilised through the IoT guide was by assessing:

- The applicability of the CIS Controls and Sub-Controls in relation to IoT; as an example, controls relating to the network might not be directly applicable to IoT.
- The challenges that surround the control or sub-control implementation; for example, an IoT device may be ‘smarter’ and have more functionalities than others.
- Further discussions needed to make a device safer.

The CIS ‘Internet of Things Companion Guide version 8’ controls that were deemed applicable within a smart home environment along with the score assigned to each control are included in Appendix C. Furthermore, insight regarding the CIS ‘Internet of Things Companion Guide version 8’ scoring and applicable controls is provided within sub-heading 5.8 of this Chapter.

5.7 ENISA Baseline Security Recommendations for IoT in the context of Critical Information Infrastructures

ENISA [3] described IoT as a technological, social and economic new concept which is fast evolving consisting of various devices and services that are all interconnected. These devices range from sensors or everyday objects to more complex industrial or health components. IoT devices and services transform both the business world or the user’s lifestyle in general due to the fact that through the collection, exchange and the processing of data, a specific context is dynamically adapted. The development of IoT raises several challenges such as legal and regulatory challenges which are being amplified as the development of IoT expands. The legal and regulatory measures being drafted and implemented are outpaced by the fast pace of IoT advancements thus leaving the IoT security framework in a legal limbo. This legal limbo and no standardisation cause organisations to implement their own IoT designs which brings about other issues such as interoperability between different devices and other legacy systems. The ENISA [3] developed the ‘Baseline Security Recommendations for IoT in the context of Critical Information Infrastructure’ to 1) provide a mapping between IoT assets and the threats which they might be susceptible to; 2) provide further knowledge as to the security requirements for IoT; 3) identify and quantify IoT threats, risk and attacks and 4) provide recommendations and good practices which need to be implemented within IoT systems to protect the IoT ecosystem [3]. These recommendations were developed in the context of Critical

Information Infrastructures to provide a series of recommendations aimed at various stakeholders within the IoT field as illustrated in Table 5.6.

Table 5.6 ENISA Recommendations and Intentions [3]

Recommendation	Intended for
Identify which IoT stakeholder is liable	IoT industry and regulators
Develop a complete (software, hardware, etc) lifecycle management for IoT devices or services which are as secure as possible	IoT industry, developers, manufacturers and platform operators
Within the IoT ecosystem, accomplish agreement for interoperability	IoT industry, manufacturers, providers, regulators and associations
Provides the necessary incentives be it economic or administrative to achieve IoT security	IoT industry, academics, manufacturers, providers, regulators and associations
Raise the necessary awareness that is needed to increase IoT cybersecurity	IoT industry, academics, manufacturers, providers, regulators and associations
Improve and harmonise regulations and security initiatives within the IoT field	IoT industry, manufacturers, providers and associations

The ENISA [3] ‘Baseline Security Recommendations for IoT in the context of Critical Information Infrastructure’ controls that were deemed applicable within a smart home environment along with the score assigned to each control are included in Appendix D. Furthermore, insight regarding the ENISA [3] ‘Baseline Security Recommendations for IoT in the context of Critical Information Infrastructure’ scoring and applicable controls is provided within sub-heading 5.8 below.

5.8 Hardening Procedures Analysis

To answer Research Questions *RQ3b* and *RQ4*, the author sought to extrapolate statistical evidence from the applicable controls identified within the OWASP, CIS and ENISA hardening procedures mentioned in the preceding paragraphs. As already attested the workings are included in Appendices B, C, and D respectively. Table 5.7 illustrates the total amount of controls in each hardening procedure, established as applicable for this research study and classified according to the OWASP IoT Top 10 vulnerability they mitigate.

Table 5.7 Hardening Procedure Applicable Controls according OWASP IoT Top 10 Vulnerability

OWASP IoT Top 10 vulnerability		ISVS (controls)	ENISA (controls)	CIS (controls)
1	Weak, guessable or Hardcoded Password	11	9	8
2	Insecure Network Services	13	12	13
3	Insecure Ecosystem Interfaces	10	2	8
4	Lack of Secure Update Mechanism	4	3	6
5	Use of Insecure or Outdated Components	4	1	4
6	Insufficient Privacy Protection	1	0	0
7	Insecure Data Transfer or Storage	2	0	0
8	Lack of Device Management	3	4	14
9	Insecure Default Settings	0	0	0
10	Lack of Physical Hardening	0	1	0
Not Applicable		101	39	100

As outlined in Table 5.7, it is evident that the majority of the applicable controls for all the three hardening procedures, focus mostly on the top three vulnerabilities identified within the OWASP IoT Top 10 list, with a total of 86 controls out of 133 applicable controls (65%).

Furthermore, to compare the hardening procedures, the author sought to identify the percentage of applicable controls that each hardening procedure had. This was calculated using the Formula 1 below.

Formula:

$$1) \text{ Percentage(\%) of applicable Controls} = (100 \times \text{Not Applicable Controls}) \div \text{Applicable Controls}$$

Moreover, to gather additional data the author summed up all the scores of each hardening procedure and then calculated the average points per control using Formula 2. The findings of the workings of both formulae for each hardening procedure are illustrated in Table 5.8 below.

Formula:

$$2) \text{ Average Points per question} = \text{Total Number of Points} \div \text{Total Controls}$$

Table 5.8 Breakdown of workings per hardening procedure

		Hardening Procedures		
		ISVS	CIS	ENISA
C O N T R O L S	Total Controls	149	153	71
	Applicable Controls	48	53	32
	Not Applicable Controls	101	100	39
	Percentage (%) of Applicable Controls	32.21%	34.64%	45.07%
P O I N T S	Total Points	381	369	254
	Average Points per Control	7.94	6.96	7.94

From the data identified in Table 5.8, it can be noted that a discrepancy in the number of controls between ENISA and the other two hardening procedures exists. This does not necessarily devalue the merit of this hardening procedure; as a matter of fact, it has the highest percentage of applicable controls with 32 applicable controls out of a total of 71 controls (45.07%). Furthermore, the ENISA hardening procedure also had the most average points per control (7.94), the same as OWASP ISVS. With this being said it is to be noted that it also had the least number of applicable controls with 32 applicable controls, 15 less than OWASP ISVS and 21 less than CIS. The hardening procedure with the most applicable controls is the CIS hardening procedure with 53 controls, 6 more applicable controls than OWASP ISVS. However, it was noted that the CIS hardening procedure had the least number of average points per control. This can be attributed to the fact that the CIS hardening procedure has 14 controls mitigating ‘Lack of device management’ which provides 3 points per control according to the scoring system provided by the author.

The hardening procedures with the highest average points per control are the OWASP ISVS and ENISA. For OWASP ISVS this can be attributed to the fact that the procedure has the most controls in the top three vulnerabilities within the OWASP IoT Top 10 list. The top three controls within OWASP IoT Top 10 provide 10, 9 and 8 points per control respectively. This is also reflected in the total amount of points scored by each hardening procedure whereby OWASP ISVS has the most points scored with 381 points when compared to CIS (369 points) and ENISA (254 points).

5.9 Recommended Hardening Procedure

It was evident from the methodological approaches used on hardening procedures that they converge and can be merged into one hardening procedure. Thus, encapsulating all the various applicable controls. Through this evidence-based approach, Research Question RQ4 was answered. From the statistical evidence it can be attested that the applicable controls within the hardening procedures can be merged. Since there is an overlap when it comes to smart home owner controls within the three hardening procedures an exercise was conducted to omit any control which can be considered similar or duplicate. An example of the aforementioned overlap between hardening controls is OWASP ISVS Control 2.1.10 stating ‘Verify that provisioned credentials for device authentication are unique per device’ and CIS control 5.2 which states ‘Use Unique Passwords’. Given that ISVS Control 2.1.10 is more descriptive, CIS control 5.2 was omitted. For any similar entries which were identified, one control was chosen on the merit of whichever control was most descriptive and easier to understand. The controls which are similar along which controls were omitted are established below:

1. CIS Control 5.2 similar to OWASP ISVS Control 2.1.10:
 - Omitted CIS Control 5.2;
2. CIS Control 4.7 similar to OWASP ISVS Control 2.1.8:
 - Omitted CIS Control 4.7;
3. OWASP ISVS Control 2.2.2 similar to ENISA GP-TM-27:
 - Omitted OWASP ISVS Control 2.2.2;
4. ENISA GP-TM-09 similar to OWASP ISVS Control 2.1.10:
 - Omitted ENISA GP-TM-09;
5. ENISA GP-TM-24 similar to OWASP ISVS Control 2.3.1:
 - Omitted ENISA GP-TM-24;

6. OWASP ISVS Control 1.1.2 similar to ENISA GP-TM-45:
 - Omitted OWASP ISVS Control 1.1.2;
7. ENISA GP-TM-45 and GP-TM-50 similar to OWASP ISVS Control 3.2.2:
 - Omitted GP-TM-45 and GP-TM-50.

Once all the duplicates were omitted, the remaining controls were sorted according to the OWASP IoT Top 10 score which was assigned to each control. The merged list is included as Appendix E. To compare the results of this merged procedure the author included the statistics of the proposed hardening procedure in Tables 5.9 and 5.10.

Table 5.9 Hardening Procedures Applicable Controls according OWASP IoT Top 10 Vulnerability including recommended hardening procedure

	OWASP IoT Top 10 List Vulnerability	ISVS Controls	ENISA Controls	CIS Controls	Recommended Procedure Controls	Omitted Controls for Recommended Procedure
1	Weak, guessable or Hardcoded Password	11	9	8	23	5
2	Insecure Network Services	13	12	13	35	3
3	Insecure Ecosystem Interfaces	10	2	8	18	0
4	Lack of Secure Update Mechanism	4	3	6	13	0
5	Use of Insecure or Outdated Components	4	1	4	9	0
6	Insufficient Privacy Protection	1	0	0	1	0
7	Insecure Data Transfer or Storage	2	0	0	2	0
8	Lack of Device Management	3	4	14	21	0
9	Insecure Default Settings	0	0	0	0	0
10	Lack of Physical Hardening	0	1	0	1	0
Not Applicable		102	39	100	0	0

Table 5.10 Breakdown of workings including recommended hardening procedure

		Hardening Procedures			
		ISVS	CIS	ENISA	Recommended Procedure
C O N T R O L S	Total controls	149	153	71	126
	Applicable Controls	48	53	32	118
	Not Applicable Controls	101	100	39	Null
	Omitted Controls	Not Applicable			8
	Percentage (%) of Applicable Controls	32.21%	34.64%	45.07%	93.65%
P O I N T S	Total Points	381	369	254	911
	Average Points per Control	7.94	6.96	7.94	7.72

As can be noted from Table 5.9, it is evident that the recommended hardening procedure provided more controls than any of the other three hardening procedures especially in the Top 3 OWASP IoT Top 10 vulnerabilities. This is due to the fact that the recommended hardening procedure was a merger of all the three hardening procedures established to be used in this study. Given that the recommended hardening procedure was based around only the applicable controls from the other three hardening procedures it obviously did not have any non-applicable commands thus, this is null as can be illustrated from Table 5.10. Findings indicate that the recommended hardening procedure had scored a lower average points per control than both OWASP ISVS and the ENISA hardening procedure by 0.22 points but a higher average score than CIS. This can be attributed to the fact that the recommended procedure has 21 controls relating to

‘Lack of Device Management’ which rendered 3 points per control. Excluding the fact that the recommended hardening procedure has more controls than the other hardening procedures, from the established data it could be noted that no further benefits were identified. In the following Chapter the recommended hardened procedure along with the three other hardening procedures will be implemented within a test environment to verify whether it renders any other benefits when it is tested in a simulated Case Scenario.

6. Case Scenarios

In pursuit to gain further insights into aspects of the proposed hardening procedure, for the second part of the research study, a Case Scenario approach was used. The author used the case study research design to test if the proposed hardening procedure suggested in Chapter 5 rendered any benefits when applied within a simulated Case Scenario. Case studies are extensively used as they provide insights that are not possibly achieved by other approaches [104]. This research design is widely used by professionals within different disciplines to investigate scientific situations. Crow et al. [107] attested that this methodological approach:

“Is particularly useful to employ when there is a need to obtain an in-depth appreciation of an issue, event or phenomenon of interest, in its natural real-life context” [107]

The design, planning and execution of the Case Scenarios was a thorough process. Reporting of findings were systematically and meticulously carried out, with the aim being to address the research question and to ensure that all relevant data is collected.

6.1 Setup

The author used a modified risk evaluation process identified by Echeverria et al. [108] to identify a Common Vulnerability Scoring System Version 3 (CVSSv3) severity rating for each hardening procedure identified in Chapter 5. Furthermore, the author used the modified risk evaluation process to verify whether the proposed hardening procedure has a lower severity rating than the other three procedures. Figure 6.1 illustrates the modified risk evaluation process proposed by the author.

As can be noted from Figure 6.1 The risk evaluation process is divided into three main processes whereby the step consists of port and service enumeration which can be performed using NMAP. OpenVAS can be used to perform the vulnerability assessment. The attack surface is then analysed using various tools which can be found with the Kali Operating System.

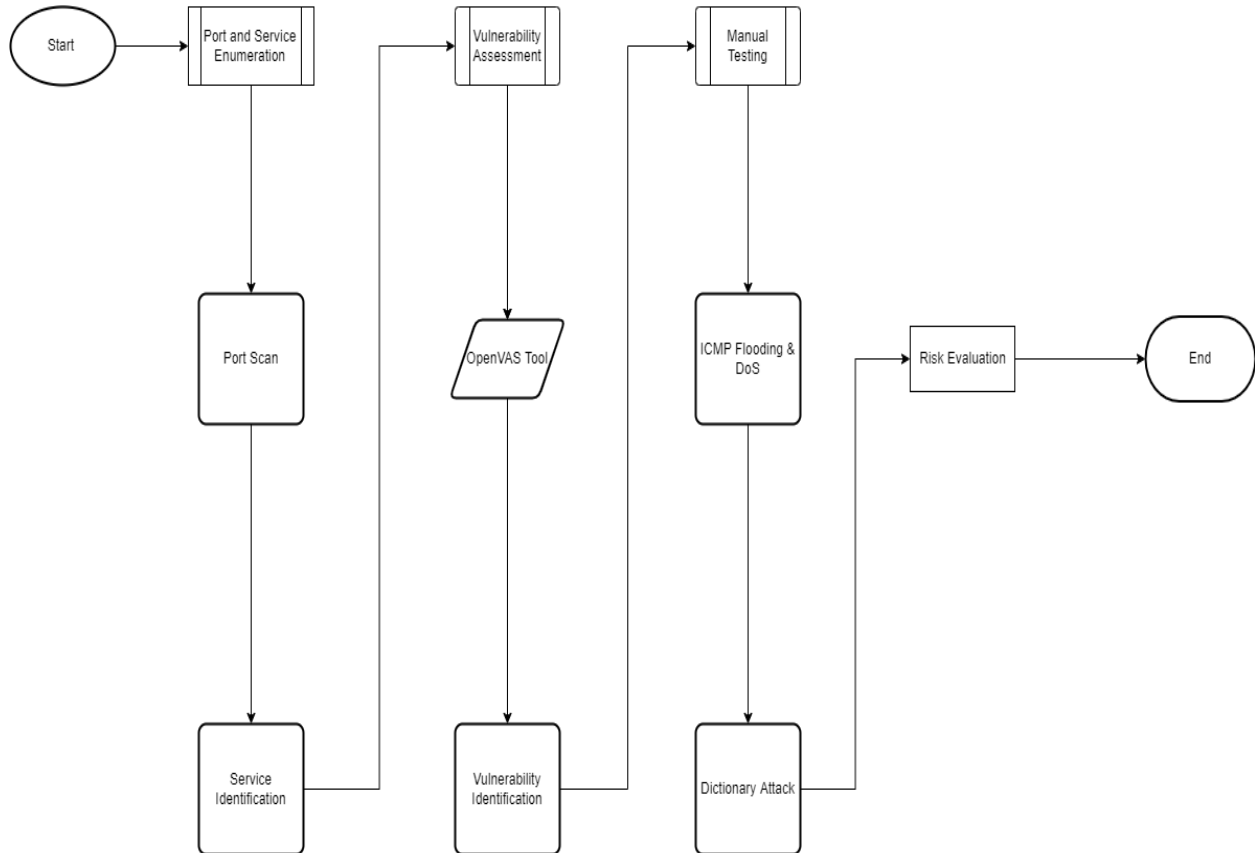


Figure 6.1 Modified Risk Evaluation Process

Bacon [109] noted that in the past, various vendors implemented their own scoring methods for vulnerabilities without providing any insight as to how a score is calculated. Furthermore, system administrators would be unsure which vulnerability they should fix first. The United States National Infrastructure Assurance Council (NIAC) attempted to solve this conundrum through the establishment of the Common Vulnerability Scoring System (CVSS). The aim of the CVSS was to have a baseline scoring methodology that can accurately reflect the severity and impact of vulnerabilities in an IT environment [109]. Furthermore, Bacon [109] noted that since this widely accepted and used public framework is vendor and application neutral it enables organisations to establish one scoring framework throughout a range of software such as different operating systems, databases and web applications. To calculate the risk rating, all risks identified in both the attack surface evaluation and the vulnerability assessment were rated using the Common Vulnerability Scoring System Version 3 (CVSSv3). Furthermore, the CVSSv3 maps five different CVSSv3 score ranges with a severity rating as per Table 6.1. The average CVSSv3 rating was

calculated by adding all the identified vulnerability ratings and then divided it by the total number of vulnerabilities as per Formula (3).

$$3. \text{ CVSSv3 vulnerability average} = \frac{(\sum_{i=1}^n \text{CVSSv3}_{\text{Step } n})}{n}$$

Table 6.1: CVSSv3 Rating

CVSSv3 Score	Severity
0	Null
0.1 - 3.9	Low
4.0 - 6.9	Medium
7.0 - 8.9	High
9.0 - 10.0	Critical

6.2 Research Setup

To verify whether the recommended procedure provides any further benefits when compared to the three other hardening procedures, testing was carried out under the five Case Scenarios indicated below:

1. A Case Scenario whereby an IoT device was not hardened and tested with the Modified Risk Evaluation Process defined in Figure 6.1;
2. A Case Scenario whereby an IoT device was hardened using OWASP ISVS and tested with the Modified Risk Evaluation Process defined in Figure 6.1;
3. A Case Scenario whereby an IoT device was hardened using CIS Controls and tested with the Modified Risk Evaluation Process defined in Figure 6.1;
4. A Case Scenario whereby an IoT device was hardened using ENISA and tested with the Modified Risk Evaluation Process defined in Figure 6.1;
5. A Case Scenario whereby an IoT device was hardened using the recommended procedure as per Chapter 5 and tested with the Modified Risk Evaluation Process defined in Figure 6.1.

For each Case Scenario a CVSSv3 Severity Rating was assigned. The severity rating for Case Scenario 1 was a ‘High’ severity rating outlining several Operating System vulnerabilities. A legacy version of Raspbian (Raspbian Buster) was utilised to mimic a device being launched into the market without vulnerabilities but possibly being laden with vulnerabilities once the end-user purchases and starts using the device. The start point of the other four Case Scenarios were the non-hardened device outlined with the vulnerabilities established in Case Scenario 1. The target CVSSv3 Severity Rating for Case Scenarios 2 to 5 were then considered ‘Null’.

To carry out the Case Scenarios mentioned above, the author set up a Raspberry Pi Model 3B with a Quad-core 64-bit CPU and 1GB of RAM. The Raspberry Pi was connected to a DHT22 Temperature and Humidity sensor following the instructions provided by Initial State [110]. Raspbian Buster was installed on five SanDisk Extreme 32GB microSD Cards so that each SD card was mapped out to a Case Scenario. Table 6.2 illustrates the configurations that were assigned to each Case Scenario (or SD card).

Table 6.2 Configuration Settings per SD card

	SD Card 1	SD Card 2	SD Card 3	SD Card 4	SD Card 5
Host Name	pi1	pi2	pi3	pi4	pi5
User Name	pi	pi	pi	pi	pi
Password	raspberrry	raspberrry	raspberrry	raspberrry	raspberrry
IP Address	192.168.1.160	192.168.1.170	192.168.1.180	192.168.1.190	192.168.1.200

Apart from the default ‘pi’ user, on each SD Card a default user was added to mimic the default accounts set on IoT devices. Stahie [111] identified various generic credentials that are generally used as default credentials on IoT devices. From the list provided, the author chose the username ‘default’ and the password ‘12345’. The author followed the guide provided by Garcia [112] to create a default user on each PiOS and also assign administrative rights to this user. A list of all the credentials identified by Stahie [111] was also extracted into a text file (iotpass.txt file) so that it was utilised as a dictionary when performing dictionary attacks later on in the Case Scenarios. Figure 6.2 identifies the contents of ‘iotpass.txt’ which are the default credentials as presented by Stahie [111] after omitting any duplicate entries.

```
(root@kali)~/usr/share/wordlists
# cat iotpass.txt
admin
CenturyLink
root
xc3511
root
Zte521
Pon521
default
vizxv
support
123456
guest
1234
12345
S2fGqNFs
0xhlwSG8
```

Figure 6.2 iotpass.txt contents

To present the data gathered by the temperature and humidity sensor the author opted to use an online portal named InitialState. Initialstate is an IoT Platform for Data Visualisation which accesses real-time streaming of data whilst also displaying the data received through real-time dashboards [113]. The system relies on a

shared key and a script to share data between the IoT device and the online platform. A 14-day trial was initiated, and the script was installed on all five SD cards with each script being mapped to a different dashboard to confirm that each respective script and the devices were working. A copy of the script used for SD Card 1 can be found in Appendix F. For data presentation means and to verify that the script was working on each SD card the Sensor_Location_Name, Bucket_Name, Bucket_Key entries in the script were changed for each script to the host name of each respective SD card.

The author's own personal computer as the infrastructure used to host different virtual machines and tools mentioned in Chapter 4. The virtualisation software used to host the virtual machines was the Oracle VM VirtualBox version 6.1.28 r147628. Being considered as the Swiss knife of cybersecurity tools [82], Kali version 2022.1 was also deployed on VirtualBox and updated with the latest tools. With regards to Greenbone OpenVAS, an Open Virtualisation Format file (.ova) was downloaded and this was imported as an appliance and configured accordingly. The version of Greenbone OpenVAS installed was the 21.04.13 version. The network configuration for all virtual machines was set to Bridged Adapter. Thus, the Dynamic Host Configuration Protocol (DHCP) server on the author's router provided an individual IP address for each machine rather than using the computer's IP address. It is to be noted that all virtual machines along with the Raspberry Pi were all connected on the same VLAN (192.168.1.0/24). This made it easier to connect via Secure Shell (ssh) to each machine from the author's computer and to address each appliance individually. Table 6.3 illustrates the system information for the author's personal computer and the system information for each virtual machine.

Table 6.3 System Information for the machines (physical or virtual) used within the Case Scenarios

	Computer	OpenVAS	Kali Linux
Physical or Virtual?	Physical	Virtual	Virtual
CPU	8 Logical Processors	2 Processors	2 Processors
RAM	32GB	5GB	4GB
Hard disk Space	1.5TB	15GB	80GB
IP Address	192.168.1.137	192.168.1.185	192.168.1.124

At a network layer, all the devices were connected to the Author's modem through a wired connection using a Cat6 cable. All devices were on the same subnet (192.168.1.0/24) whilst the router was a Technicolor TG789VAC2 provided by the author's local Internet Service Provider. The architecture described above is outlined in Figure 6.3 below.

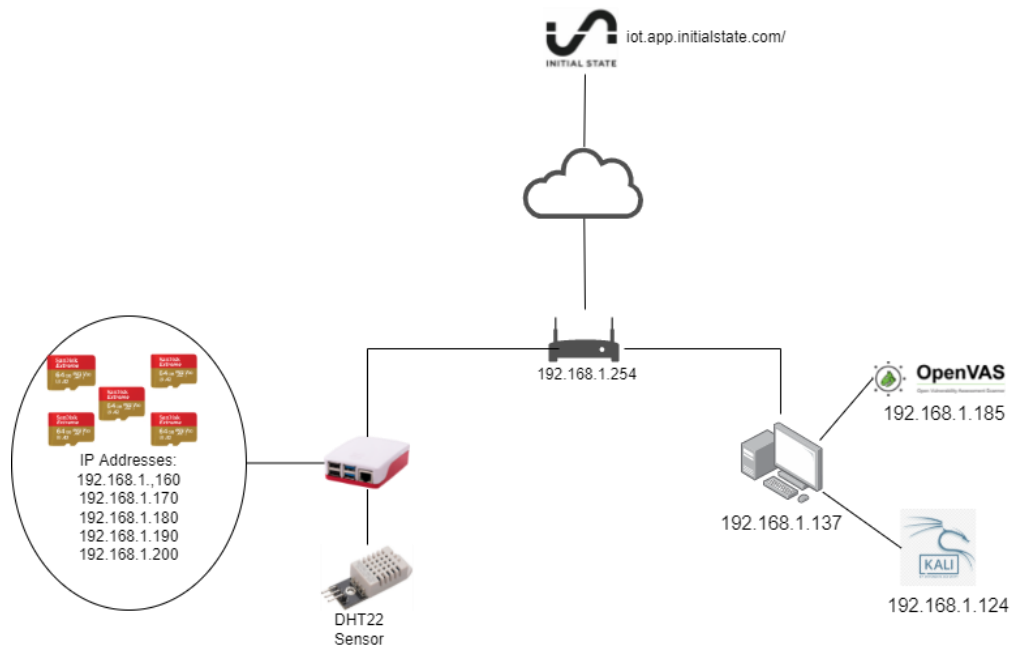


Figure 6.3: Experiment Setup Architecture

6.3 Case Scenarios Initial Configuration

Given that the initial starting point of each Case Scenario was the same, the following configurations were implemented on each SD card. An SD card was inserted into the Raspberry PI and once the system booted the username and password indicated within Table 6.2 were inserted. The communication between the DHT22 Temperature and Humidity sensor and Initialstate was established using the below commands:

- `sudo apt-get install python3-pip`
- `pip3 install ISSstreamer`
- `\curl -sSL https://get.initialstate.com/python -o - | sudo bash`
- `pip install adafruit-circuitpython-dht`

- sudo apt-get install libgpod2
- cat > script.py
- Pasted the script found in Appendix F
- python3 script.py

The author confirmed that the script was working, and readings were being sent to the Initialstate dashboard as per Figure 6.4. The script was set to run in the background using the “CTRL+Z” and “bg” commands.

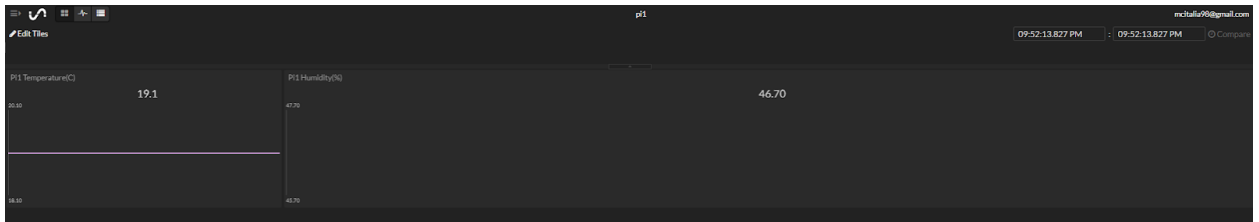


Figure 6.4: Evidence of Pi1 data on Initialstate

The user ‘default’ was added on the machine and to the sudoers file within the Operating System using the steps indicated below:

1. Sudo adduser default
2. Enter Password: ‘12345’
3. Visudo
4. Added ‘default’ user within the ‘User privilege specification’ section as illustrated within Figure 6.5

```
# User privilege specification
root    ALL=(ALL:ALL) ALL
default ALL=(ALL:ALL) ALL
# Allow members of group sudo to execute any command
%sudo  ALL=(ALL:ALL) ALL
```

Figure 6.5: User default added within sudoers

To perform the ICMP flood attack as illustrated in Figure 6.1 the author opened TCP port 80 using the “sudo python3 -m http.server 80” command. This initiated a simple web server listening on port 80. Once these settings were set out the author proceeded to perform the steps established in Figure 6.1.

6.4 Case Scenario 1

6.4.1 Case Scenario 1 - Port and Service Enumeration

As illustrated in Figure 6.1 the author initiated the port scan using Nmap which was installed on the Kali Machine. The command used to perform the port scan was ‘nmap -sTU -O 192.168.1.160’. The command can be segmented as ‘<tool> <command> <command> <Host>’. Command ‘-sTU’ performs a TCP Connect and a UDP scan whereas ‘-O’ is used to perform an OS Detection scan. The results of the Nmap scan are illustrated in Figure 6.6 whilst Table 6.4 identifies the services which each port was utilised for.

```

└─# nmap -sTU -O 192.168.1.160
Starting Nmap 7.92 ( https://nmap.org ) at 2022-03-31 03:35 EDT
Stats: 0:01:04 elapsed; 0 hosts completed (1 up), 1 undergoing UDP Scan
UDP Scan Timing: About 13.33% done; ETC: 03:43 (0:06:56 remaining)
Stats: 0:01:09 elapsed; 0 hosts completed (1 up), 1 undergoing UDP Scan
UDP Scan Timing: About 13.75% done; ETC: 03:44 (0:07:13 remaining)
Stats: 0:07:15 elapsed; 0 hosts completed (1 up), 1 undergoing UDP Scan
UDP Scan Timing: About 48.97% done; ETC: 03:50 (0:07:33 remaining)
Stats: 0:13:24 elapsed; 0 hosts completed (1 up), 1 undergoing UDP Scan
UDP Scan Timing: About 84.20% done; ETC: 03:51 (0:02:31 remaining)
Stats: 0:16:22 elapsed; 0 hosts completed (1 up), 1 undergoing UDP Scan
UDP Scan Timing: About 99.99% done; ETC: 03:52 (0:00:00 remaining)
Nmap scan report for pi1.lan (192.168.1.160)
Host is up (0.0012s latency).
Not shown: 998 closed tcp ports (conn-refused), 955 closed udp ports (port-unreach), 45 open/filtered udp ports (no-response)
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
MAC Address: B8:27:EB:2E:E7:C3 (Raspberry Pi Foundation)
Device type: general purpose
Running: Linux 4.X|5.X
OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5
OS details: Linux 4.15 - 5.6
Network Distance: 1 hop

OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 1033.36 seconds

```

Figure 6.6 Nmap port scan

Table 6.4: Nmap Scan Open Ports and related service in Case Scenario 1

Port	State	Service
22 / tcp	open	ssh
80/tcp	Open	http

6.4.2 Case Scenario 1 - Vulnerability Assessment

The next step in the Risk evaluation was the Vulnerability Assessment procedure. An OpenVAS credentialed scan was run on the Raspberry Pi device. The scan identified 42 vulnerabilities broken down as 11 Critical, 21 High, 8 Medium and 2 Low vulnerabilities according to CVSS Version 3. The total score of the vulnerabilities amounted to 318. The scan results are illustrated in Appendix G.

6.4.3 Case Scenario 1 - Manual Attacks

- **Dictionary Attack using Hydra**

A dictionary attack is a variation of a brute-force technique whereby a list of words or phrases (referred to as a dictionary) are used to identify the credentials of an account. Such attacks generally utilise less resources than a common brute-force attack and require less time although this type of attack might not always be successful [114].

The author used the brute forcing tool Hydra and the dictionary listed in Figure 6.2 to gain the credentials for the account 'default'. Through a terminal on the Kali Linux machine using the root user the author ran the command 'hydra -l default -P /usr/share/wordlists/iotpass.txt 192.168.1.160 -t 4 ssh'. The '-l default' defines the login account which Hydra will use to attempt the login, '-P /usr/share/wordlists/iotpass.txt' defines the dictionary of password which will be used to attempt to login, '192.16.1.160' is the IP of the pi1 machine, '-t 4' limits hydra to run four login attempts simultaneously whilst 'ssh' defines the service on which the attack will be performed.

As can be seen in Figure 6.7 the Hydra attack on pi1 was successful, given that the correct password was identified.

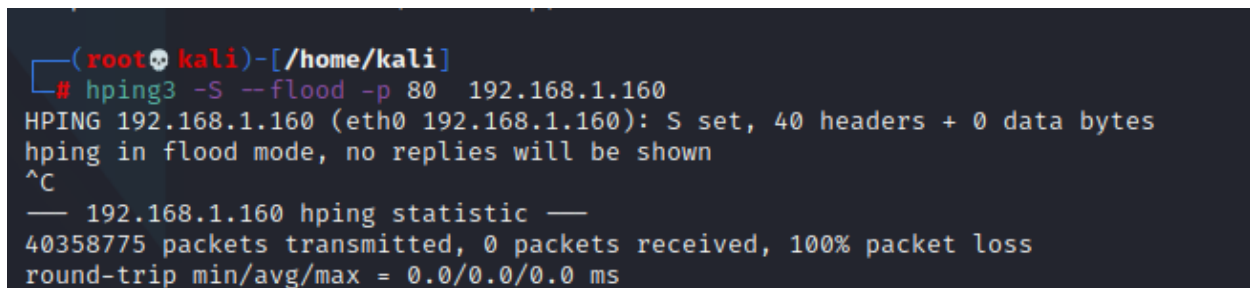
```
└─$ hydra -l default -P /usr/share/wordlists/iotpass.txt 192.168.1.160 -t 4 ssh
Hydra v9.2 (c) 2021 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-03-06 15:34:48
[DATA] max 4 tasks per 1 server, overall 4 tasks, 17 login tries (l:l/p:17), ~5 tries per task
[DATA] attacking ssh://192.168.1.160:22/
[22][ssh] host: 192.168.1.160 login: default password: 12345
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2022-03-06 15:35:02
```

Figure 6.7 Hydra attack on pi1 identifying the login credentials

- **ICMP Flood Attack**

Ohri [115] attested that like any DoS attack, an ICMP flood prevents the users from accessing an endpoint. This is done by flooding the endpoint with multiple ICMP echo requests or pings. The endpoint has to reply to these requests by sending the same amount of reply packets [115]. Such request-response scenarios would ensue throughout the ICMP attack. These types of attacks would consume significant bandwidth and resources resulting in a denial of service for the smart home user [115]. Hping3 was used to perform this type of attack on the Case Scenarios (1 to 5). Hping3 is a network tool which comes pre-installed on Kali Linux. As a tool it is able to send custom TCP, ICMP, or UDP requests to a target machine on the network and to display the replies the target machine sends back [116]. The command used in this scenario is ‘hping3 -S --flood -p 80 192.168.1.160’ which sends ICMP requests on port 80 until the command is stopped. This rendered the machine in Case Scenario 1 unreachable from SSH and also very slow when attempting to type controls in the terminal of this machine.



```
(root@kali)-[~/home/kali]
└─# hping3 -S --flood -p 80 192.168.1.160
HPING 192.168.1.160 (eth0 192.168.1.160): S set, 40 headers + 0 data bytes
hping in flood mode, no replies will be shown
^C
— 192.168.1.160 hping statistic —
40358775 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

Figure 6.8 ICMP attack on pi1

6.4.4 Case Scenario 1 - Risk Evaluation

From the findings of the OpenVAS scan it was noted that 11 Critical, 21 High, 8 Medium and 2 Low vulnerabilities with a total CVSSv3 score of 318 were identified. Furthermore, as per Table 6.5 below, it can be noted that the vulnerabilities identified within the manual testing added an additional 1 Critical and 2 High vulnerabilities. This added another 24.8 points to the total amount of CVSSv3 score and 3 vulnerabilities to the total number of vulnerabilities. Table 6.6 identifies the total amount of CVSS v3 score according to Criticality, the total CVSS v3 score, total number of vulnerabilities and the CVSSv3 Average Score as per Formula 3. Through Table 6.6 it can be concluded that Case Scenario 1 has a severity rating classified as ‘High’.

Table 6.5: Vulnerabilities identified through manual testing for Case Scenario 1

Attack	CVSSv3	Severity
DoS using Hping3	7.5	High
ICMP Flood using Hping3	7.5	High
Dictionary Attack Using Hydra	9.8	Critical

Table 6.6: Vulnerabilities identified through both manual and automated testing for Case Scenario 1

Severity	Score
Critical	116.9
High	175.9
Medium	44.6
Low	5.4
Total Number of CVSS v3 score	342.8
Total Number of Vulnerabilities	45
CVSSv3 Average Score	7.62

6.5 Case Scenario 2 - ISVS

Once the configuration steps identified in Case Scenario 1 were performed on the second SD card the author immediately started hardening the Operating System.

So that controls 1.1.2, 3.2.2, and 3.2.3 of the OWASP ISVS were satisfied the author turned off the web server running on port 80 through the command `'ps aux | grep python'` and then identified the process identification number (PID) of the command. Once the PID was identified, the author ran the command `'kill <PID>'`, where `<PID>` is the PID of the task `'sudo python3 -m http.server 80'` which immediately closed port 80. Neither Telnet, FTP, or any other legacy protocol were used within this Case Scenario.

During this hardening procedure, one risk was identified when verifying control 1.2.2 since an access code was hard coded within the script provided by InitialState. The risk here would be minimal as should a hacker gain access he could potentially send the data to another Initialstate account rather than the author's account. Moreover, a hacker would need access to the Author's home network as the machine is only accessible through SSH locally. Possible mitigations would be to 1) change the access code on a regular basis, 2) introduce an Intrusion Prevention System in front of the network, 3) change the passwords on a regular basis and use secure passwords, or 4) disable SSH.

To cater for the requirements within controls 2.1.1, from controls 2.1.3 till 2.2.2, and 2.3.1 the author changed the password for both the 'default' and the 'pi' with the randomly generated passwords `'Hs@uC5(u){R>!Q5&'` and `'S%@3+b^MMSuq'~E,'` respectively. As can be seen, these passwords are 16 characters long and a mix of alphanumeric characters. This mitigated the Hydra dictionary attack as can be seen in Appendix H.

Authentication logs on the machine were verified that they satisfy the criteria established within the applicable controls starting from 1.4.1 to 1.4.7. Furthermore, through Figure 6.3 each connected device was uniquely identified as such verifying control 2.1.2. Communication with other components was verified to occur over secure channels and no unencrypted channels were used. This established that both controls 4.1.1 and 4.2.1 were satisfied.

To cater for the updates (applicable controls 1.3.4, 3.2.1, and from 3.2.4 till 3.4.5) the author first updated the device using `'apt-get upgrade'`. While this took some time to implement, this ensured that all the libraries within the Operating System were updated to their latest version. This step improved the security of the machine significantly as no vulnerability was identified within the OpenVAS vulnerability assessment. The OpenVAS

Vulnerability assessment report is illustrated in Appendix H.

It is worth noting that the applicable controls 4.3.1 till 4.6.3 were not enforced within this scenario given that the Raspberry Pi didn't make use of the protocols mentioned within these controls. In fact, for this particular Case Scenario both Bluetooth and Wi-Fi were disabled since Bluetooth was not necessary whereas Cat6 was used for internet connectivity rather than Wi-Fi.

From the observations mentioned above the following conclusions were drawn. Table 6.7 denotes whether the manual testing performed on this machine was successful or not. Furthermore, Table 6.8 provides a breakdown of the CVSSv3 points within each severity classification along with the CVSS v3 average score using Formula 3. The data illustrated in Table 6.8, indicated that the Severity Rating of Case Scenario 2 is Null. Evidence of the above testing performed is illustrated in Appendix H.

Table 6.7: Vulnerabilities identified through manual testing for Case Scenario 2

Attack	Successful? (Yes/ No)
DoS using Hping3	No
ICMP Flood using Hping3	No
Dictionary Attack Using Hydra	No

Table 6.8: Vulnerabilities identified through both manual and automated testing for Case Scenario 2

Severity	Score
Critical	0
High	0
Medium	0
Low	0
Total Number of CVSS v3 score	0
Total Number of Vulnerabilities	0
CVSSv3 Average Score	0
CVSSv3 Severity Rating	Null

6.6 Case Scenario 3 - CIS

The author noted that part of the hardening procedure identified by CIS consists of establishing and maintaining several inventories. The relevant controls and required inventories are listed below along with the inventories or descriptions which satisfied the requirements within these controls.

1. Control 1.1: Asset
 - a. Tables 6.2 and 6.3 can be considered as the Asset Inventories for this Case Scenario.
2. Control 2.1: Software Inventory
 - a. Since one software was used (Initialstate) no inventory was created; To note that this software was supported and as such satisfied the criteria listed within Control 2.2. This software is considered up-to-date and trusted. Thus control 16.5 is also satisfied.
3. Controls 5.1 and 5.4: Accounts (both user and Service accounts)
 - a. Two user accounts were utilised and these are pi and the default account; No service accounts were used during this Case Scenario.
4. Control 6.6: Authentication and Authorisation Systems
 - a. Apart from the Linux login, two authentication and authorisation systems were identified and these were the 1) Initialstate authentication through the access code within the script and 2) the SSH authentication when logging into the system remotely. Access and account management can be considered to be centralised given the limited number of accounts and authentication means. As such the criteria of controls 5.6 and 6.7 are both satisfied.
5. Control 12.2: Secure Network Architecture
 - a. Figure 6.3 depicts the Network Architecture.
6. Control 12.4: Architecture diagrams
 - a. Figure 6.3 depicts the Network Architecture.

The criteria established within controls 18.1, 18.2, 18.3, 18.4 and 18.5 was considered satisfied through the penetration testing established within the manual testing as identified within Figure 6.1. Furthermore, through scans performed using OpenVAS the vulnerabilities were identified and immediately remediated through the command 'apt-get upgrade'. Furthermore, the author configured automated Operating System and Application Patch Management through the guide provided by Ondara [117]. In order to verify that the vulnerabilities were remediated and to close off the established Vulnerability Management Process, the author re-ran the OpenVAS scan and verified that all vulnerabilities were remediated as can be seen in Appendix I. Through the operations performed above the author managed to satisfy the criteria listed in

controls 7.1, 7.2, 7.3, 7.4, 7.5 and 7.7. The severity Rating System for both Application vulnerabilities and Operating System vulnerabilities was the CVSSv3 as highlighted previously within this Chapter. Thus, Control 16.6 was considered as satisfied.

The author notes that Anti-malware software was not installed on the Raspberry Pi. However, the author's personal computer had Microsoft Defender which is considered as Anti-Malware Software which automatically updated its own Anti-Malware Signature. Through this software removable media is immediately scanned and doesn't automatically run on the device. Anti-Exploitation Features were also enabled on Microsoft Defender. As such, controls 10.1 till 10.7 of the CIS controls was considered as implemented through the above. Furthermore, Microsoft Defender is also enabled to act as a firewall and protect the network as such satisfying the control 4.5. Through Microsoft Defender any file types which are deemed as unnecessary or unsecure were not accepted. Whenever the author needed to access the Internet and/or his email address he only used Fully Support Browsers (Google Chrome) and Email Clients (Gmail). As such it can be confirmed that controls 9.1 and control 9.6 were also implemented within this Case Scenario. All devices (virtual or physical) used within this scenario were set to automatically lock after 5 minutes in order to satisfy the criteria listed in control 4.3.

From a networking perspective the networking infrastructure was configured with security in mind and all the network components were updated prior to starting the Case Scenarios. The router was configured to automatically address unauthorised assets and is considered as an Active Discovery tool. Furthermore, each approved device was immediately assigned an IP through DHCP. The router also caters for DNS requests. Through these commands a number of controls were addressed such as controls 1.2, 1.3, 1.4, 4.9, 12.1, 12.3 and 12.5. Uncomplicated Firewall (UFW) was installed and configured to accept SSH connections only from the author's personal computer using IP address 192.168.1.137, ensuring that the criteria identified in control 4.4 was met. Furthermore, in order to ensure that only necessary services were being used within this Case Scenario, the author killed the process running 'sudo python3 -m http.server 80'. Control 4.8 was also satisfied through this process.

Sub-chapter 6.3 can be considered as defining a secure configuration process for the device and the network and was deemed applicable to this Scenario. Furthermore, the author changed the passwords of both default accounts ('default' and 'pi') to unique passwords. The above steps ensured that controls 4.1, 4.2, 4.7 and 5.2 were satisfied.

From the observations mentioned above the following conclusions were drawn. Table 6.9 denotes whether the manual testing performed on this machine was successful or not. Furthermore, Table 6.10 provides a breakdown of the CVSSv3 points within each severity classification along with the CVSS v3 average score using Formula 3. The data illustrated in Table 6.10, indicated that the Severity Rating of Case Scenario 3 is 'Null'. Evidence of the above testing performed is illustrated within Appendix I.

Table 6.9: Vulnerabilities identified through manual testing for Case Scenario 3

Attack	Successful? (Yes/ No)
DoS using Hping3	No
ICMP Flood using Hping3	No
Dictionary Attack Using Hydra	No

Table 6.10: Vulnerabilities identified through both manual and automated testing for Case Scenario 3

Severity	Score
Critical	0
High	0
Medium	0
Low	0
Total Number of CVSS v3 score	0
Total Number of Vulnerabilities	0
CVSSv3 Average Score	0
CVSSv3 Severity Rating	Null

6.7 Case Scenario 4 - ENISA

Following the configuration of the machine as per sub-chapter 6.3 and confirmation that the device (Pi4) has the same risks rating as Case Scenario 1; the author began to harden the device as per the applicable controls illustrated in Appendix J.

To satisfy the controls relating to authentication means (GP-TM-09, GP-TM-TM21, GP-TM-TM22, GP-TM-TM23, GP-TM-TM24, GP-TM-TM25, GP-TM-TM26, GP-TM-TM27 and GP-TM-TM29) the author initially changed the passwords of both accounts ('pi' and 'default') to '|z3,+!<KnD-74mdzS' and 'cfwPVrq?#33!{4R' ' respectively using the control 'sudo passwd pi' and 'sudo passwd default'. Through this command the passwords were changed using a robust password reset mechanism. However, the Linux based operating system does indicate whether the account exists. After this was performed both accounts were removed using the command 'userdel <user>' where <user> was either 'pi' or 'default'. A new administrator account ('piuser4') was set up to ensure that the criteria laid in control GP-TM-22 was satisfied. The new administrator account was assigned the password ' 6M8(S%yru"H9Jn5R '. As evidenced in Appendix J the aforementioned controls mitigated the risk of the dictionary attack. It can be noted that the given Case Scenario does not cater for two-factor authentication (2FA) or multi-factor authentication (MFA). Hence, this part of the control was considered as not applicable. Furthermore, it was confirmed that within this Case Scenario no credentials were exposed in the clear either in internal or external network traffic.

Firmware and software updates were configured to be updated automatically using the guide provided by Ondara [117]. This ensured that the controls outlined in GP-TM-18, GP-TM-19 and GP-TM-20 were satisfied and that security updates were implemented immediately once they were launched through automatic updates.

From a networking perspective, various tasks were performed in order to satisfy the controls within the ENISA 'Baseline Security Recommendations for IoT in the context of Critical Information Infrastructures'. As performed within Case Scenario 2, port 80 was closed using the same controls ('ps aux | grep python' and 'kill <PID>') outlined in the aforementioned Case Scenario. This procedure satisfied the controls GP-TM-44, GP-TM-45 and GP-TM-50 is illustrated within Appendix J. Uncomplicated Firewall (UFW) was installed and configured to accept SSH connections only from the author's personal computer using IP

address 192.168.1.137, ensuring that the criteria identified in GP-TM-41, GP-TM-42, GP-TM-43 and GP-TM-46 were satisfied, since the traffic received and transmitted was limited to only necessary connections.

Since the Case Scenario takes into consideration just one IoT device, GP-TM-48 and GP-TM-49 were not applicable to this Case Scenario. Hence, from the above observations the following conclusions could be drawn. Table 6.11 denotes whether the manual testing performed on this machine was successful or not whilst Table 6.12 provides a breakdown of the CVSSv3 points within each severity classification along with the CVSS v3 average score using Formula 3. As evidenced in Table 6.12 the Severity Rating of Case Scenario 4 is Null. Evidence of the above testing performed is illustrated within Appendix J.

Table 6.11: Vulnerabilities identified through manual testing for Case Scenario 4

Attack	Successful? (Yes/ No)
DoS using Hping3	No
ICMP Flood using Hping3	No
Dictionary Attack Using Hydra	No

Table 6.12: Vulnerabilities identified through both manual and automated testing for Case Scenario 4

Severity	Score
Critical	0
High	0
Medium	0
Low	0
Total Number of CVSS v3 score	0
Total Number of Vulnerabilities	0
CVSS v3 Average Score	0
CVSS v3 Severity Rating	Null

6.8 Case Scenario 5 - Recommended Hardening Procedure

Following the configuration of the machine as per Sub-Chapter 6.3 and confirmation that the device (Pi5) has the same risk rating as Case Scenario 1; the author began to harden the device as per the applicable controls as illustrated in Appendix K.

Since the recommended hardening procedure is a mix of controls from the three hardening procedures; the controls performed in each hardening procedure were implemented on the Raspberry Pi in this Case Scenario. Table 6.13 illustrates the steps taken during Case Scenario 5 along with which vulnerability from the OWASP IoT Top 10 list was mitigated.

Table 6.13: Actions performed within Case Scenario 5

Steps	Action Taken	OWASP Score
1	Established an inventory of accounts that included the accounts: 'default' and 'pi' .	OWASP Score 10
2	Changed passwords of both 'default' and 'pi' to passwords that are 16 characters long and include a mix of alphanumeric characters and symbols.	OWASP Score 10
3	A new account 'pi5' was created and inserted within the sudoers file.	OWASP Score 10
4	Both 'default' and 'pi' accounts were deleted.	OWASP Score 10
5	Ensure that administrative tasks can only be run by elevating privilege to root.	OWASP Score 10
6	Verified that only secure protocols were being used within the Case Scenario.	OWASP Score 9
7	Closed off any ports which were not necessary. Notably port 80.	OWASP Score 9
8	Downloaded and configured UFW to allow accept connections from trusted sources	OWASP Score 9
9	Performed Manual Penetration Testing and remediated any vulnerabilities identified	OWASP score 8
10	Performed all the Operating System and applications updates	OWASP Score 7
11	Configured auto-update on operating system and applications updates	OWASP Score 7
12	Performed automated vulnerability assessment	OWASP Score 7
13	Mitigated vulnerabilities identified within step 12	OWASP Score 7
14	Confirmed that the software used within this scenario was secure, trusted and up-to-date	OWASP Score 6
15	Confirmed and verified that logging was being collected and that confidentiality, integrity and authenticity of these logs were ensured	OWASP Score 5
16	Ensured that antimalware and anti-exploitation tools were installed and updated on the author's personal computer.	OWASP Score 3

Hence, from the above observations the following conclusions were drawn. Table 6.14 denotes whether the manual testing performed on this machine was successful or not. Table 6.15 provides a breakdown of the CVSSv3 points within each severity classification along with the CVSSv3 average score using Formula 3. As is evidenced by Table 6.15 the Severity Rating of Case Scenario 5 is 'Null'. Evidence of the below testing being performed is illustrated in Appendix K.

Table 6.14: Vulnerabilities identified through manual testing for Case Scenario 5

Attack	Successful? (Yes/ No)
DoS using Hping3	No
ICMP Flood using Hping3	No
Dictionary Attack Using Hydra	No

Table 6.15: Vulnerabilities identified through both manual and automated testing for Case Scenario 5

Severity	Score
Critical	0
High	0
Medium	0
Low	0
Total Number of CVSS v3 score	0
Total Number of Vulnerabilities	0
CVSS v3 Average Score	0
CVSS v3 Severity Rating	Null

Data from the Case Scenarios will also be discussed in the following Chapter.

7. Security Analysis

7.1 Introduction

The data collected from Chapter 5 and Chapter 6 of this study rendered the findings to answer the Research Questions of this study. This chapter will present and discuss the overall findings of the data analysis in relation to the statement of the problem, the research aims and questions.

7.2 Discussion of results

The study was appropriately informed by theory and can be deemed to substantiate the already available literature. It can be concluded that IoT devices need to be secured using well-established hardening procedures. The different sources used to generate data yielded different kinds of knowledge and insight. As the evidence was being collected through the analysis and testing of hardening procedures, the author had an active role in the process. This was achieved as the author being the main researcher while understanding, collating, managing and constructing the findings, asked valid research questions, to interpret the findings. To remain focused, while analysing the data, the author was constantly referring to the Research Questions. Each process was perceived as a learning experience. Furthermore, appropriate documentation and report keeping were maintained to ensure that through the evidence on the operations of the study, data collection produced can be repeated with the same results. In this section the author will discuss the research findings around the following research questions:

7.2.1 RQ 1: Which options do smart home owners have when they are concerned with hardening their IoT devices?

It was established through the literature that IoT devices are always increasing and are predicted to reach up to 30 billion devices in 2025 [6]. It is also evident that IoT security challenges and issues will increase as IoT devices' increase in popularity [8]. Hence, it is imperative that stakeholders within the IoT field secure their devices through standards or procedures established by reputable entities. However, it is noted that the pace by which IoT is growing and developing is outpacing the developments in IoT hardening procedures originating from well-established organisations. This can be evidenced by the fact that a standard such as 'ISO/IEC FDIS 27004' has been proposed in 2018 and up till the time of writing of this

report (March 2022) the standard was not yet published. This view further supported the findings by Allen [92], who attested that very few security standards were published by organisations within the cybersecurity field. From the literature analysed within this research, it was noted that from these published standards or hardening procedures, none targeted smart home users or end users. The published hardening procedures target manufacturers, enterprises or organisations implementing IoT within their network. However, it was also noted that some controls within these hardening procedures can be adapted to serve as a hardening procedure for users.

Moreover, it can be argued that possibly smart home users who are not security oriented might not know about these hardening procedures or about organisations such as ENISA, OWASP and ISO/IEC. As evidenced in Chapter 5, through the literature review, it was noted that through a Google search, smart home owners can access several web articles on hardening procedures. However, from the analysed literature review, the author concluded that in most cases these web articles lack the number of hardening steps required. Furthermore, these hardening procedures do not offer granular and concise controls but are generic. It was also noted that the maximum number of controls within these hardening procedures was twelve controls and some of these controls can be considered as general security best practices such as securing the home network, changing passwords and using MFA. These hardening procedures originating from web articles tend to overlap in the steps they provide to the end user. Additionally, the controls identified within the web articles are also listed within the hardening procedures published by cybersecurity organisations. Thus, users would have to gather evidence from more than one article in order to collect the amount of information necessary to strengthen their Smart Home security posture. Thereby, through the findings of the literature review, the author concluded that smart home end users need to be conversant with well-respected organisations and hardening procedures as these can offer a holistic solution to reduce the IoT threat landscape.

7.2.2a RO 2a: Which hardening procedures identified as a reply to Question 1 originate from established organisations within the cybersecurity field?

7.2.2b RO 2b: Which three hardening procedures established in Question 2a can be considered as safe to be utilised for the experimental and theoretical part of this study?

As identified through the findings in Chapter 5 various hardening procedures originate from established organisations. Organisations such as ENISA, CSA, IEEE, OWASP and CIS have developed their own hardening procedures whilst other organisations such as ISO/IEC are in the process of developing and establishing their own. However, it was also noted that none of the hardening procedures were focused on providing controls directly related to smart home owners. The majority of these hardening procedures were targeted at manufacturers or organisations, whilst others were generic hardening procedures. The author acknowledges that devices should be launched into the market with security safeguards in place in order to guarantee that the device owner is protected. In some areas of the world, notably California, this is no longer a nice-to-have but is now a legal must have [46]. However, security shouldn't be a one-sided perspective smart home owners have the responsibility to keep abreast with the latest information and update their devices to reduce vulnerabilities. Additionally, IoT owners, be it organisations or the general public should also perform steps when configuring their devices and continuously strive to protect themselves and their data. While organisations might be legally bound or bound through the organisation's GRC approach to protect their devices, smart home owners do not have similar safeguards. The need for a hardening procedure which provides smart home owners with a concise IoT hardening procedure targeted at them cannot be understated taking into consideration the widespread vulnerability and threats they are exposed to [20], [46].

Therefore, given that no hardening procedures targeting specifically smart home owners were found in the literature, the author identified three hardening procedures, originating from established organisations, that have a substantial number of controls implementable within a smart home. As already established in Chapter 5, these three hardening procedures are 1) ENISA's 'Baseline Security Recommendations for IoT' [3], 2) the CIS 'Controls Internet of Things Companion Guide' [2] and 3) OWASP IoT 'Security Verification Standard'[21]. The author acknowledges that while these hardening procedures are targeted at a bigger audience rather than smart home owners, several controls can be applied within a smart home scenario to secure an IoT device and the network it is deployed in. Thus, these three hardening procedures

were used for both the theoretical and the experimental aspects of this study. Results of the theoretical and experimental aspects will be discussed in the answers to the Research Questions below.

7.2.3a RO 3a: Which controls from the three established hardening procedures can be implemented within a smart home and within this study?

7.2.3b RO 3b: Which OWASP IoT Top 10 vulnerabilities are mitigated through the controls established in Question 3a?

Through the data captured from the comparison of the three established hardening procedures (used throughout the research), it was noted that not all controls can be implemented within a smart home environment. Results indicate that for the ENISA hardening procedure from 71 controls, only 32 controls were identified as applicable. Furthermore, it was noted that the OWASP ISVS has 48 applicable controls out of a total 149 controls. The last analysed hardening procedure was the CIS and it had the most applicable controls with 53 applicable controls out of a total of 153. Based on these findings it would be arguable that from a smart home owner's perspective, the CIS hardening procedure might be considered as the suggested hardening procedure to implement. However, from a security perspective, it can be argued that while it offers the most controls when it comes to mitigating 'Insecure Network Service', 'Lack of Secure Update Mechanisms' and 'Lack of Device Management'; the CIS hardening procedure does not offer any controls relating to 'Insufficient Privacy Protection', 'Insecure Data Transfer or Storage', 'Insecure Default Settings' and 'Lack of Physical hardening'. Thus, making the device vulnerable to tampering or jamming attacks within the physical layer [56] and man-in-the middle attacks within the network layer [63].

Moreover, it was noted that only OWASP ISVS provided controls mitigating 'Insufficient Privacy Protection' and 'Insecure Data Transfer or Storage' and the number of controls provided was very minimal. This can be attributed to the fact that the majority of controls provided within this area rely on the manufacturers to implement such controls at the development stage [47]. Similarly with regards to 'Lack of Physical Hardening', only ENISA provided one control and this can also be attributed to the fact that the physical hardening provided within the hardening procedures was targeted at manufacturer's level. Thus, as attested in the literature, it is imperative that manufacturers must identify any possible means of device tampering and design IoT devices with possible risks and physical attacks in mind [46]. With regards to 'Insecure Default Setting' no hardening procedure identified any control as applicable. However, this is due to the fact that default settings usually relate to passwords and these were assigned a higher score (10

points). Thus, it can be concluded that controls explicitly aimed for smart home owners and mitigating 1) 'Insufficient Privacy Protection', 2) 'Insecure Data Transfer or Storage', 3) 'Insecure Default Settings' and 4) 'Lack of Physical Hardening' need to be developed by renowned organisations.

Through the findings of the three established hardening procedures, it was noted that the majority of controls focus on mitigating the top three vulnerabilities [1) 'Weak, guessable or hard coded passwords' , 2) 'Insecure network services' and 3) 'Insecure ecosystem interfaces'] from the OWASP IoT Top 10 vulnerability list. This is due to the fact that the home users might have more control over which passwords they use, which network services they utilise, and to some extent how well they manage their IoT Ecosystem. When comparing the three hardening procedures, OWASP ISVS was found to have the most controls in each of the top three vulnerabilities within the OWASP IoT Top 10 list. From these findings, it can be attested that smart home owners should change the default passwords during the configuration stage to passwords which can be considered as secure and unique. It is imperative that manufacturers proactively implement controls which force the users to change default passwords upon first logon. Alternatively, as noted in the hardening procedures, manufacturers can assign a random and unique password to the default account of each individual device. Furthermore, it can be recommended that where possible manufacturers should implement MFA authentication methods to ensure that should the user's password be compromised a hacker would still need access to the second authentication password. Thus, securing devices from brute force or dictionary attacks [46].

7.2.4 RQ 4: Can the controls identified within Question 3a be merged into one hardening procedure?

From the research findings established in Chapter 5 and through the answers of Research Questions *RQ3a* and *RQ3b* it was found that the applicable controls originating from the three hardening procedures could be merged and adapted to serve as a tool in the Case Scenarios. Thus, this process provided an answer to Research Questions *RQ5a* and *RQ5b*.

7.2.5a RQ 5a: If question 4 is in the affirmative, how does the proposed hardening procedure compare to the other hardening procedures within Question 2?

7.2.5b RQ 5b: What additional value do end users gain from the hardening procedure identified in Question 4?

Furthermore, it was noted that the proposed hardening procedure provided more applicable controls when compared to the three standalone hardening procedures. This is because the applicable controls identified from the three hardening procedures were merged to form the recommended hardening procedure. However, further findings indicated that if the average points per control are taken into consideration, albeit by a small margin, the findings established that the recommended hardening procedure ranks second behind both the OWASP and ENISA hardening procedures. It can be argued that if a smart home owner implements the controls identified by either OWASP or ENISA, this would be considered as sufficient in protecting the smart home environment. However, the recommended hardening procedure offers more controls which can translate into additional protection gained. For example, the OWASP ISVS provides additional controls relating to specific protocols such as Bluetooth, LoRaWAN and Wi-Fi, whereas the ENISA hardening procedure does not offer such controls. Similarly, ENISA offers additional controls relating to authentication means and mitigations to weak, guessable or hardcoded passwords. Furthermore, ISVS does not provide controls relating to vulnerability management or penetration testing, contrary to both ENISA and CIS. The latter mentioned two hardening procedures offer several controls on how to perform vulnerability management and penetration testing. The controls provided by the CIS [97] hardening procedures recommended that designs, inventories and architectures concerning different elements within the smart home environment are established. This potentially can help smart home users gain further insight about their IoT setup or architecture and potentially identify approaches to protecting their environments.

Furthermore, CIS provides controls which relate to protecting the endpoints used to access the IoT infrastructure such as deploying Anti-Malware or Anti-Exploitation tools. Findings indicated that both ENISA and OWASP hardening procedures do not offer controls relating to either the deployment of Anti-Malware and Anti-Exploitation tools nor the establishing of designs, inventories or architecture within the IoT environment. Thus, from the findings the author concluded, that through a mix of controls, the recommended hardening procedure ensures that the advantages that each hardening procedure offers outlined above are converged into one hardening procedure.

From the research findings established through the Case Scenarios, the author noted that by implementing minor changes on the device (Raspberry Pi), the security of IoT devices can be increased drastically. These minor changes can be implemented on IoT devices by 1) upgrading libraries and software or enabling auto-updates, 2) changing the passwords to passwords which are longer than 12 characters, unique and contain a mix of alphanumeric characters and 3) disabling unnecessary services. Such findings provide evidence to smart home owners that continuous security updates need to be implemented on IoT devices, thus reducing the threats and vulnerabilities that a device is prone to [49] [46]. Furthermore, through the implementation of the hardening steps outlined above, it was noted that the manual attacks (dictionary attack, ICMP flood and DDoS attack) were mitigated. Moreover, it was noted that any service dependent attacks can be mitigated through updating these services from a vulnerable version to a non-vulnerable version.

Moreover, it was noted that an operating system which is over two years old comes with several operating system vulnerabilities. This can mean that if a user purchases a device launched in the market two years ago and no safeguards (such as auto-update during configuration) are implemented at manufacturer level; the user is making use of a vulnerable device with a lot of vulnerabilities in the operating system. This is reaffirming the findings of Harper [49] and Mukherjee [46] who attested that devices either never get security updates after being launched in the market or only receive updates until the manufacturers launch a new device rendering them susceptible to vulnerabilities. Further findings from the Case Scenario 1, established that a device which was not hardened is vulnerable to attacks such as dictionary attack and ICMP flood. Additionally, this Case Scenario had a CVSSv3 severity score of High. From these results, it was noted that when applied within a Case Scenario with one device and exposed to the aforementioned attacks, each of the three hardening procedures reduced the CVSSv3 severity score from High to Null. Furthermore, results indicated that by implementing the recommended hardening procedure within the same Case Scenario the same CVSSv3 severity score (Null) was achieved. Thus, it can be attested that when exposed to the same attacks on the same device, the recommended hardening procedure rendered the same

level of confidence as the three other hardening procedures. Although these findings render the same level of confidence, from a security perspective, the proposed hardening procedure recommends more controls to be implemented. Hence, it can provide further assurance that the threat landscape is reduced. The author noted that the implementation of other hardening procedures or different IoT devices could have affected the findings of the Case Scenario/s differently.

From the collective discussion mentioned above, it can be noted that both the literature and the research findings established that IoT security needs to ameliorate from three fronts. Firstly, at manufacturing and designing level the IoT device must be designed with security as a priority and that updates are released regularly until the device's end-of-life. Furthermore, prior to deploying IoT devices into the market it is imperative that manufacturers perform hardening on the device in order to reduce the risks shipped to the IoT owner. Secondly, at user level, hardening of the IoT devices must be performed utilising hardening procedures originating from established organisations. Finally, organisations, academics, experts in the field and researchers must provide regularly updated recommendations that will aid both the manufacturers and end users with standards and hardening procedures.

7.3 Limitations of this study

Price et al. [118] attested that a study's limitations can be considered as those design or methodological characteristics that had an effect on the research findings. Study limitations are considered as the boundaries placed on the generalisation of the results. Limitations can be a basis on which recommendations are established and further research is built upon. In order to avoid pitfalls and possible limitations, resources were appropriately used and the research was well organised and planned throughout. However, like any other research, through reflection and analysis the author established that this study does have its own flaws and limitations; some of which were beyond his control. The author identified and listed some of the limitations within this research. Discussion on the limitations will be done to help other researchers or individuals practising through evidence-based methods to establish and possibly not repeat these limitations again in other similar studies [118]. The limitations identified by the author will be discussed below.

From the literature review no hardening procedures aimed at smart home owners were identified. Thus, three hardening procedures which could be adopted to target smart home owners were chosen. This rendered two study limitations which are 1) not all controls from the hardening procedures chosen where

applicable to this study and 2) possibly there were other hardening procedures available that could have been chosen and analysed.

The author understands that qualitative research is time-consuming and expensive [119]. The volume of the gathered data along with time restrictions in place limited the depth of the analysis performed and can be considered as limitations in themselves. Furthermore, due to the aforementioned limitations, penetration testing and vulnerability assessments had to be performed on one device, i.e., a Raspberry Pi. Being a study conducted on one device from the plethora of smart home devices available on the market, the author acknowledges that this study might not be representative of the wider body of smart home devices. As such conclusions drawn from this study can be considered as transferable only in the context of IoT devices with the same or better computing power than the Raspberry Pi 3 Model B v1.2. With a longer timeline the author believes that more testing scenarios with more smart home devices could be performed and more reliable and comparable data could be gathered. Additionally, more data would have been yielded and further validation of the research findings could be drafted. The author noted that further research can be established using different IoT devices from other brands or manufacturers. The author acknowledges that through triangulation of data by using different IoT devices as a means of testing and thus data collection, triangulation could be achieved by comparing different hardening procedures on various smart home devices.

Another limitation established by the author is the work experience he has been exposed to. The author acknowledges that his work experience was mainly focused on implementing controls to defend the infrastructure (blue teaming) and not attacking devices. Furthermore, the author acknowledges that his knowledge and insight on penetration testing and red teaming is limited. Thus, due to both time limitation and the author's knowledge, the penetration testing was limited to use only a few tools. As an example, no source code analysis or reverse engineering was performed during this study. The author acknowledges that with more time and insight on penetration testing further vulnerabilities could be identified.

This study relied on automated tests and on manual testing which were easy to perform. This study assumes that the smart home owner has little to no experience with vulnerability assessment and penetration testing. As such this study does not include any manual testing which includes a high level of knowledge in penetration testing such as reading and/or reverse engineering the source code.

The author notes that this research was focused on comparing IoT procedures within a smart home

environment. From the literature review, it was noted that this form of comparison was never performed on the three hardening procedures nor on other hardening procedures. Similar comparison on such hardening procedures within an organisation or a manufacturer environment might render different results considering that all controls would be taken into consideration.

8. Conclusions and Recommendations

The literature review established that IoT devices are gradually increasing. In fact, research [5][6] indicates that IoT connections have already exceeded non-IoT connections and by 2025 the total number of IoT devices is predicted to reach more than 30 billion devices. This means that the amount of IoT devices will be increasing, while security within IoT remains a major concern. The author notes that while these statistics can be seen as a positive and promising, however security issues need to be tackled to ensure that mitigations are in place to protect the smart home owner. Various reputable organisations such as ENISA, IEEE, CIS, CSA and OWASP have already published several recommendations and publications in order to streamline security through a device's lifecycle from conception stage till the decommissioning stage. Furthermore, these reputable organisations have also established standards which can be used by organisations or enterprises when deploying and maintaining IoT devices within the organisation's network. However, through this research findings it was established that the options available for end users such as smart home users are limited and provide few steps in order to secure an IoT device. As evidenced by the research findings of this study there are various tools that can be used in order to perform penetration testing on IoT devices. The same vulnerabilities that can be attributed to devices in other areas of technology, can be attributed to IoT devices as well. Thus, IoT devices should be held to the same security standards as other non-IoT devices. Penetration testing should also be performed to mitigate vulnerabilities.

Based on the research of this study there are some recommendations that can be brought forward to lessen the vulnerabilities and risks present within IoT devices. Considering the above findings, the researcher proposes the following recommendations that are important to researchers, academics, manufacturers and smart home owners:

- It is imperative that smart home users perform automated vulnerability assessments and penetration testing. Furthermore, common best practices such as updating their devices, changing default passwords and disabling unnecessary services should be performed on the IoT devices.
- IoT manufacturers, organisations making use of IoT devices and also smart home users adopt a security-focused approach to reduce the attack surface and unlock the full potential of IoT.
- It is imperative that established organisations such as ENISA, IEEE, ISO and CIS develop and maintain standards and hardening procedures which are specific and targeted at smart home users or IoT end users. All this while ensuring that the published hardening procedures targeting manufacturers and organisations continue to be maintained and developed.
- Organisations, researchers and authors of web articles should base their controls and

recommendations to mitigate well-established vulnerabilities within the IoT (such as OWASP IoT Top 10 vulnerabilities) when developing hardening procedures or recommendations for smart home users.

- The three hardening procedures used throughout this study (or other hardening procedures) should be analysed and tested in an organisation or manufacturer Case Scenario in order to verify whether the controls provided do secure the IoT environment within an organisation or manufacturer.
- Tests can be performed on different IoT devices to verify whether different results are obtained, refuted or affirmed.
- Tests can be performed using different attack methodologies or tools to verify whether different results are obtained, refuted or affirmed.
- Other testing scenarios can be established whereby other hardening procedures are identified and compared in order to identify whether they provide additional controls to smart home users.

To conclude, despite the limitations of the project, the author attests that the results of this study provide relevant information to smart home owners. Furthermore, it can be attested that both the aims and objectives of this project have been achieved through the literature review, Case Scenarios and analysis of the hardening procedures. It was noted that smart home owners were provided with a consolidated hardening procedure that affirmed the literature findings stating that the implementation of hardening procedures in a smart home environment is necessary. The author notes that further research can be developed based on the outcomes and recommendations set forth within the project.

Bibliography

- [1] K. Boeckl, M. Fagan, W. Fisher, N. Lefkovitz, K.N. Megas, E. Nadeau, D. Gabel O'Rourke, B. Piccarreta and K. Scarfone, 'Considerations for managing Internet of Things (IoT) cybersecurity and privacy risks', National Institute of Standards and Technology, Gaithersburg, MD, Jun. 2019. doi: 10.6028/nist.ir.8228.
- [2] Centre for Internet Security (CIS), 'New release: CIS controls® Internet of Things companion guide', *Centre for Internet Security (CIS)*, Jun. 27, 2019. <https://www.cisecurity.org/blog/new-release-cis-controls-internet-of-things-companion-guide/> (accessed Oct. 02, 2021).
- [3] European Union Agency for Network and Information Security, 'Baseline Security Recommendations for IoT in the context of Critical Information Infrastructures', *European Union Agency for Network and Information Security*, Nov. 17, 2017. <https://www.enisa.europa.eu/publications/baseline-security-recommendations-for-iot> (accessed Oct. 17, 2021).
- [4] R. Minerva, A. Biru, and D. Rotondi, 'Towards a definition of the Internet of Things (IoT)', *IEEE Internet Initiative*, vol. 1, no. 1, pp. 1–86, 2015, Accessed: Oct. 17, 2021. [Online]. Available: https://iot.ieee.org/images/files/pdf/IEEE_IoT_Towards_Definition_Internet_of_Things_Revision1_27MAY15.pdf
- [5] A. Pelaez and C. Botero, '5 reasons why 2020 became the year of IoT', *Ubidots*, Dec. 30, 2020. <https://ubidots.com/blog/2020-the-year-of-iot/> (accessed Oct. 02, 2021).
- [6] K. L. Lueth, 'State of the IoT 2020: 12 billion IoT connections, surpassing non-IoT for the first time', *IoT Analytics*, Nov. 19, 2020. <https://iot-analytics.com/state-of-the-iot-2020-12-billion-iot-connections-surpassing-non-iot-for-the-first-time/> (accessed Oct. 02, 2021).
- [7] A. Bickley, 'The Challenges of IoT Security and IoT Device Hardening', *Arrow Electronics*, Aug. 23, 2018. <https://www.arrow.com/en/research-and-events/articles/the-challenges-of-iot-security-and-how-to-harden-the-edge> (accessed Oct. 02, 2021).
- [8] Intellectsoft, 'Top 10 IoT security issues: Ransom, botnet attacks, spying', *Intellectsoft*, Jul. 30, 2020. <https://www.intellectsoft.net/blog/biggest-iot-security-issues/> (accessed Oct. 02, 2021).
- [9] OWASP, 'OWASP IoT Top 10 2018', *OWASP*, 2018. <https://owasp.org/www-pdf-archive/OWASP-IoT-Top-10-2018-final.pdf> (accessed Oct. 20, 2021).
- [10] European Union Agency for Network and Information Security, *Good practices for security of Internet of things in the context of smart manufacturing*. Publications Office of the European Union, 2018. doi: 10.2824/851384.
- [11] Australian CyberSecurity Centre, 'Guidelines for System Hardening', *Australian CyberSecurity Centre*, Sep. 2021. <https://www.cyber.gov.au/acsc/view-all-content/advice/guidelines-system-hardening> (accessed Oct. 19, 2021).
- [12] D. Hamilton, 'Best practices for IoT security', *Networkworld*, Mar. 27, 2018. <https://www.networkworld.com/article/3266375/best-practices-for-iot-security.html> (accessed Oct. 04, 2021).

- [13] R. Daws, 'Kaspersky: Attacks on IoT devices double in a year', *IoTNews*, Sep. 07, 2021. <https://iottechnews.com/news/2021/sep/07/kaspersky-attacks-on-iot-devices-double-in-a-year/> (accessed Oct. 06, 2021).
- [14] C. Cyrus, 'IoT cyberattacks escalate in 2021, according to Kaspersky', *IoT World Today*. <https://www.iotworldtoday.com/2021/09/17/iot-cyberattacks-escalate-in-2021-according-to-kaspersky/> (accessed Mar. 14, 2022).
- [15] K. Hewitt, '7 internet of things threats and risks to be aware of', *SecurityScorecard*, Aug. 04, 2021. <https://securityscorecard.com/blog/internet-of-things-threats-and-risks> (accessed Oct. 06, 2021).
- [16] Thales Group, 'IoT security issues in 2021: A business perspective', *Thales Group*, Sep. 2021. <https://www.thalesgroup.com/en/markets/digital-identity-and-security/iot/magazine/internet-threats> (accessed Oct. 06, 2021).
- [17] M. A. Raja, '10 steps to strengthen your IoT security', *Techopedia*, Jul. 23, 2018. <https://www.techopedia.com/10-steps-to-strengthen-your-iot-security/2/33447> (accessed Oct. 10, 2021).
- [18] W. S. Jay Ryerse, '8 important OS hardening tips to protect your clients', *ConnectWise*, Nov. 18, 2021. <https://www.connectwise.com/blog/cybersecurity/8-important-os-hardening-tips-to-protect-your-clients> (accessed Oct. 06, 2021).
- [19] Microsoft, 'Internet of Things (IoT) security best practices'. <https://docs.microsoft.com/en-us/azure/iot-fundamentals/iot-security-best-practices> (accessed Mar. 14, 2022).
- [20] N. G. Miloslavskaya, A. Nikiforov, K. Plaksiy, and A. I. Tolstoy, 'Standardization Issues for the Internet of Things', in *WorldCIST (2)*, 2019, pp. 328–338. Accessed: Oct. 06, 2021. [Online]. Available: https://www.researchgate.net/profile/Natalia-Miloslavskaya/publication/332875684_WorldCIST2019-Miloslavskaya-IoT-Standardspdf/data/5ccff7d3a6fdccc9dd9022f2/WorldCIST2019-Miloslavskaya-IoT-Standards.pdf
- [21] OWASP, 'Using the ISVS', *Github*, Feb. 02, 2021. <https://github.com/OWASP/IoT-Security-Verification-Standard-ISVS> (accessed Nov. 25, 2021).
- [22] Oracle, 'What is the Internet of Things (IoT)?', *Oracle*, 2021. <https://www.oracle.com/internet-of-things/what-is-iot/> (accessed Oct. 12, 2021).
- [23] A. S. Gillis, 'What is IoT (Internet of Things) and How Does it Work?', *TechTarget*, Aug. 13, 2021. <https://internetofthingsagenda.techtarget.com/definition/Internet-of-Things-IoT> (accessed Oct. 08, 2021).
- [24] C. P. Pfleeger, S. L. Pfleeger, and J. Margulies, *Security in Computing*. Prentice Hall, 2015. [Online]. Available: <https://play.google.com/store/books/details?id=VjMqBgAAQBAJ>
- [25] V. Ricquebourg, D. Menga, D. Durand, B. Marhic, L. Delahoche, and C. Loge, 'The Smart Home Concept : our immediate future', in *2006 1ST IEEE International Conference on E-Learning in Industrial Electronics*, Dec. 2006, pp. 23–28. doi: 10.1109/ICELIE.2006.347206.

- [26] S. Goossens, 'Smart appliances for a circular society: Changing the world from home', *H & PC Today - Household and Personal Care Today*, no. Vol 13(1), Feb. 2018.
- [27] Otelco, 'Everything You Need to Know About Smart Home Technology', *Otelco*, May 09, 2018. <https://www.otelco.com/resources/smart-home-guide/> (accessed Nov. 15, 2021).
- [28] S3C, 'GUIDELINE: INTRODUCING SMART APPLIANCES', S3C, 2021. https://www.smartgrid-engagement-toolkit.eu/fileadmin/s3ctoolkit/user/guidelines/GUIDELINE_INTRODUCING_SMART_APPLIANCES.pdf (accessed Nov. 12, 2021).
- [29] C. H. Varun and J. Karthikeyan, 'Survey On The Role Of IoT In Intelligent Transportation System', *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 11, no. 3, pp. 936–941, Sep. 2018, doi: 10.11591/ijeecs.v11.i3.pp936-941.
- [30] X. Yang, X. Wang, X. Li, D. Gu, C. Liang, K. Li, G. Zhang and J. Zhong, 'Exploring emerging IoT technologies in smart health research: a knowledge graph analysis', *BMC Med. Inform. Decis. Mak.*, vol. 20, no. 1, p. 260, Oct. 2020, doi: 10.1186/s12911-020-01278-9.
- [31] S. S. Costigan and G. Lindstrom, 'Policy and the Internet of Things', *Connect.*, vol. 15, no. 2, pp. 9–18, 2016, [Online]. Available: <http://www.jstor.org/stable/26326436>
- [32] W. Goddard, 'IoT network & architecture', *IT Chronicles*, Oct. 30, 2020. <https://itchronicles.com/iot/iot-network-architecture/> (accessed Oct. 12, 2021).
- [33] M. Lombardi, F. Pascale, and D. Santaniello, 'Internet of Things: A General Overview between Architectures, Protocols and Applications', *Information*, vol. 12, no. 2, p. 87, Feb. 2021, doi: 10.3390/info12020087.
- [34] M. G. dos Santos, D. Ameyed, F. Petrillo, F. Jaafar, and M. Cheriet, 'Internet of Things architectures: A comparative study', *arXiv [cs.SE]*, Apr. 27, 2020. Accessed: Oct. 08, 2021. [Online]. Available: <http://arxiv.org/abs/2004.12936>
- [35] F. Alshohoumi, M. Sarrab, A. AlHamadani, and D. Al-Abri, 'Systematic review of existing IoT architectures security and privacy issues and concerns', *Int. J. Adv. Comput. Sci. Appl.*, vol. 10, no. 7, pp. 232–251, 2019, Accessed: Oct. 08, 2021. [Online]. Available: https://www.researchgate.net/profile/Abdullah-Al-Hamdani/publication/334944162_Systematic_Review_of_Existing_IoT_Architectures_Security_and_Privacy_Issues_and_Concerns/links/5f97b76592851c14bceabdb9/Systematic-Review-of-Existing-IoT-Architectures-Security-and-Privacy-Issues-and-Concerns.pdf
- [36] V. Gandhi and J. Singh, 'IoT: Architecture, Technology, Applications, and Quality of Services', in *Microcalorimetry of Biological Molecules*, unknown, 2019, pp. 79–92. doi: 10.1007/978-981-13-5934-7_8.
- [37] M. Wu, T.-J. Lu, F.-Y. Ling, J. Sun, and H.-Y. Du, 'Research on the architecture of Internet of Things', in *2010 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE)*, Aug. 2010, vol. 5, pp. V5–484–V5–487. doi: 10.1109/ICACTE.2010.5579493.
- [38] P. Sethi and S. R. Sarangi, 'Internet of Things: Architectures, Protocols, and Applications', *J. Electr.*

Comput. Eng., vol. 2017, Jan. 2017, doi: 10.1155/2017/9324035.

- [39] CompTIA, 'Network Protocol Definition', *CompTIA*, 2021. <https://www.comptia.org/content/guides/what-is-a-network-protocol> (accessed Nov 10 2021).
- [40] C. Gregersen, 'A complete guide to IoT protocols & standards in 2021', Dec. 18, 2020. <https://www.nabto.com/guide-iot-protocols-standards/> (accessed Nov 10 2021).
- [41] AVSystem, 'IoT Standards and protocols guide — protocols of the Internet of Things', Mar. 04, 2020. <https://www.avsystem.com/blog/iot-protocols-and-standards/> (accessed Nov 10 2021).
- [42] D.B Ansari, A.U. Rehman, and R. Ali, 'Internet of things (IoT) protocols: A brief exploration of MQTT and CoAP', *Int. J. Comput. Appl.*, vol. 179, no. 27, pp. 9–14, Mar. 2018, doi: 10.5120/ijca2018916438.
- [43] T. Salman and R. Jain, 'Networking protocols and standards for internet of things', *Internet of Things and Data Analytics Handbook*, vol. 7, pp. 14–18, 2015, Accessed: Nov 11 2021. [Online]. Available: https://www.cse.wustl.edu/~jain/cse570-15/ftp/iot_prot.pdf
- [44] B. Cankar, *PENIOT: Penetration Testing Tool for IoT*. Github, 2020. Accessed: Nov. 28, 2021. [Online]. Available: <https://github.com/yakuza8/peniot>
- [45] OWASP, 'About Us', *OWASP*, 2021. <https://owasp.org/about/> (accessed Dec. 08, 2021).
- [46] L. Mukherjee, 'The OWASP IoT Top 10 List of Vulnerabilities - InfoSec Insights', *Sectigo*, Apr. 30, 2020. <https://sectigostore.com/blog/owasp-iot-top-10-iot-vulnerabilities/> (accessed Nov. 18, 2021).
- [47] C. Jariwala, 'OWASP top 10 overview and vulnerabilities', *SecureLayer7*, Jul. 18, 2020. <https://blog.securelayer7.net/owasp-top-10-overview-and-vulnerabilities/> (accessed Nov. 18, 2021).
- [48] E. Boehm, 'Top 10 IoT vulnerabilities in your devices', *Keyfactor*, Oct. 28, 2020. <https://www.keyfactor.com/blog/top-10-iot-vulnerabilities-in-your-devices/> (accessed Nov. 18, 2021).
- [49] A. Harper, '17 Biggest security challenges for IoT', *Peerbits*, Jul. 25, 2021. <https://www.peerbits.com/blog/biggest-iot-security-challenges.html> (accessed Nov. 17, 2021).
- [50] N. Johnston, 'OWASP IoT Top 10 - A gentle Introduction and an exploration of root causes', Nov. 12, 2019. [Online]. Available: <https://owasp.org/www-chapter-toronto/assets/slides/2019-12-11-OWASP-IoT-Top-10---Introduction-and-Root-Causes.pdf>
- [51] M. Burkhalter, 'Top IoT security vulnerabilities: 2020 and beyond', *Perle*, Jun. 16, 2020. <https://www.perle.com/articles/top-iot-security-vulnerabilities-2020-and-beyond-40189357.shtml> (accessed Nov. 17, 2021).
- [52] A. R. Patel, 'Cyber Security | OWASP IoT Top-10 2020', *Medium*, Dec. 07, 2020. <https://akashranjanpatel.medium.com/cyber-security-owasp-iot-top-10-2020-5c6bb23f58c> (accessed Nov. 17, 2021).
- [53] P. Panda, 'OWASP's Top 10 IoT vulnerabilities and what you can do', *Intertrust*, Nov. 12, 2020.

- <https://www.intertrust.com/blog/owasps-top-10-iot-vulnerabilities-and-what-you-can-do/> (accessed Nov. 16, 2021).
- [54] ‘California’s security of connected devices law to take effect January 1st’, *Hopkins & Carley*. <https://www.hopkinscarley.com/blog/client-alerts-blogs-updates/data-privacy-law-client-alerts/californias-security-of-connected-devices-law-to-take-effect-january-1st> (accessed Mar. 15, 2022).
- [55] A. Arampatzis, ‘Top 10 vulnerabilities that make IoT devices insecure’, *Venafi*, Mar. 15, 2021. <https://www.venafi.com/blog/top-10-vulnerabilities-make-iot-devices-insecure> (accessed Nov. 16, 2021).
- [56] M. Aydos, Y. Vural, and A. Tekerek, ‘Assessing risks and threats with layered approach to Internet of Things security’, *Meas. Control*, vol. 52, no. 5–6, pp. 338–353, Jun. 2019, doi: 10.1177/0020294019837991.
- [57] A. Raghuvanshi, R. K. Veluri, U. K. Singh, P. Panse, and M. Saxena, ‘Internet of things: Taxonomy of various attacks’, *European Journal of Molecular & Clinical Medicine*, vol. 7, no. 10, pp. 3853–3864, Feb. 2021, Accessed: Nov. 22, 2021. [Online]. Available: https://ejmcm.com/article_7690.html
- [58] K. Marneweck, ‘The role of physical security in IoT’, *Arm Community*, Mar. 14, 2019. <https://community.arm.com/arm-community-blogs/b/internet-of-things-blog/posts/the-role-of-physical-security-in-iot> (accessed Nov. 20, 2021).
- [59] I. Andrea, C. Chrysostomou, and G. Hadjichristofi, ‘Internet of Things: Security vulnerabilities and challenges’, in *2015 IEEE Symposium on Computers and Communication (ISCC)*, Jul. 2015, pp. 180–187. doi: 10.1109/ISCC.2015.7405513.
- [60] C. Wheelus and X. Zhu, ‘IoT Network Security: Threats, Risks, and a Data-Driven Defense Framework’, *IoT*, vol. 1, no. 2, pp. 259–285, Oct. 2020, doi: 10.3390/iot1020016.
- [61] N. Namvar, W. Saad, N. Bahadori, and B. Kelley, ‘Jamming in the Internet of Things: A Game-Theoretic Perspective’, *arXiv [cs.IT]*, Jul. 21, 2016. Accessed: Nov. 21, 2021. [Online]. Available: <http://arxiv.org/abs/1607.06255>
- [62] R. R. Krishna, A. Priyadarshini, A. V. Jha, B. Appasani, A. Srinivasulu, and N. Bizon, ‘State-of-the-Art Review on IoT Threats and Attacks: Taxonomy, Challenges and Solutions’, *Sustain. Sci. Pract. Policy*, vol. 13, no. 16, p. 9463, Aug. 2021, doi: 10.3390/su13169463.
- [63] Z. Čekerevac, Z. Dvorak, L. Prigoda, and P. Čekerevac, ‘INTERNET OF THINGS AND THE MAN-IN-THE-MIDDLE ATTACKS – SECURITY AND ECONOMIC RISKS’, *MEST Journal*, vol. 5, no. 2, pp. 15–15, Jul. 2017, doi: 10.12709/mest.05.05.02.03.
- [64] N. Wang, L. Jiao, P. Wang, M. Dabaghchian, and K. Zeng, ‘Efficient Identity Spoofing Attack Detection for IoT in mm-Wave and Massive MIMO 5G Communication’, in *2018 IEEE Global Communications Conference (GLOBECOM)*, Dec. 2018, pp. 1–6. doi: 10.1109/GLOCOM.2018.8647707.
- [65] M. Goyal and M. Dutta, ‘Intrusion Detection of Wormhole Attack in IoT: A Review’, in *2018 International Conference on Circuits and Systems in Digital Enterprise Technology (ICCSDET)*, Dec.

2018, pp. 1–5. doi: 10.1109/ICCSDET.2018.8821160.

- [66] S. Aggarwal and N. Kumar, ‘Chapter Twenty - Attacks on blockchain☆☆Working model’, in *Advances in Computers*, vol. 121, S. Aggarwal, N. Kumar, and P. Raj, Eds. Elsevier, 2021, pp. 399–410. doi: 10.1016/bs.adcom.2020.08.020.
- [67] S. Manda and N. Nalini, ‘Denial-of-service or flooding attack in IoT routing’, *Int. J. Pure Appl. Math.*, vol. 118, pp. 29–42, 2018, [Online]. Available: <https://www.acadpubl.eu/jsi/2018-118-19/articles/19a/3.pdf>
- [68] I. Butun, P. Österberg, and H. Song, ‘Security of the Internet of Things: Vulnerabilities, Attacks and Countermeasures’, *arXiv [cs.CR]*, Oct. 29, 2019. Accessed: Nov. 21, 2021. [Online]. Available: <http://arxiv.org/abs/1910.13312>
- [69] BitDefender, ‘Understanding IoT Vulnerabilities: Code Injection Attacks Can Steal Your Web Life’, *BitDefender*, Oct. 18, 2017. <https://www.bitdefender.com/blog/hotforsecurity/code-injection-attack-can-steal-web-life> (accessed Nov. 21, 2021).
- [70] A. G. Petcu, ‘7 examples of malicious code to keep in mind’, *Heimdall Security*, Jan. 20, 2021. <https://heimdalsecurity.com/blog/examples-of-malicious-code/> (accessed Nov. 21, 2021).
- [71] M. Cobb, ‘What is a buffer overflow? How do these types of attacks work?’, *TechTarget*, Jul. 28, 2021. <https://www.techtarget.com/searchsecurity/definition/buffer-overflow> (accessed Nov. 22, 2021).
- [72] S.G Abbas, I. Vaccari, F. Hussain, S. Zahid, U.U Fayyaz, G.A. Shah, T. Bakhshi, E.Cambiaso, ‘Identifying and Mitigating Phishing Attack Threats in IoT Use Cases Using a Threat Modelling Approach’, *Sensors*, vol. 21, no. 14, Jul. 2021, doi: 10.3390/s21144816.
- [73] BitDefender, ‘Understanding IoT Vulnerabilities: SQL injection or Hackers Can Hit Connected Things with Tricky Requests’, *BitDefender*, Jun. 27, 2017. <https://www.bitdefender.com/blog/hotforsecurity/hackers-can-hit-connected-things-tricky-requests> (accessed Nov. 21, 2021).
- [74] A. Harper, D. Regalado, R. Linn, S. Sims, B. Spasojevic, L. Martinez, M. Baucom, C. Eagle and S. Harris, *Gray Hat Hacking: The Ethical Hacker’s Handbook, Fifth Edition*. McGraw-Hill Education, 2018. [Online]. Available: <https://play.google.com/store/books/details?id=xsZaDwAAQBAJ>
- [75] A. Vasileiadis, ‘PENIOT: Pentesting tool for IoT devices’, *iGuRu.gr*, Jul. 21, 2020. <https://en.iguru.gr/peniot-ergaleio-pentesting-gia-iot-syskeves/> (accessed Nov. 26, 2021).
- [76] I. T. Perfection, ‘What is NESSUS and How Does it Work?’, *IT Perfection*, Jan. 18, 2021. <https://www.itperfection.com/network-security/network-monitoring/what-is-nessus-and-how-does-it-work-network-munitoring-vulnerabilit-scanning-security-data-windows-unix-linux/> (accessed Nov. 30, 2021).
- [77] H. Poston, ‘A brief introduction to the OpenVAS vulnerability scanner’, *InfoSec Institute*, Oct. 30, 2018. <https://resources.infosecinstitute.com/topic/a-brief-introduction-to-the-openvas-vulnerability-scanner/> (accessed Nov. 30, 2021).

- [78] Shaheer, 'OpenVAS Free Download (2021 Latest) - #1 Vulnerability Scanner Tool - SecuredYou', *SecuredYou*, Jan. 06, 2021. <https://www.securedyou.com/download-opensvas-linux-windows-free/> (accessed Nov. 30, 2021).
- [79] ZAPProxy, 'OWASP ZAP – getting started', *ZAPProxy*, 2021. <https://www.zaproxy.org/getting-started/> (accessed Nov. 27, 2021).
- [80] Geeks for Geeks, 'What is burp suite?', *Geeks for Geeks*, Aug. 22, 2019. <https://www.geeksforgeeks.org/what-is-burp-suite/> (accessed Nov. 29, 2021).
- [81] M. Ferranti, 'What is Nmap? Why you need this network mapper', *Network World*, Aug. 17, 2018. <https://www.networkworld.com/article/3296740/what-is-nmap-why-you-need-this-network-mapper.html> (accessed Feb. 27, 2022).
- [82] M. Day, '7 Things You Need to Know about Kali Linux – StartaCyberCareer.com', *Start a Cyber Career*. <https://startacybercareer.com/7-things-you-need-to-know-about-kali-linux/> (accessed Feb. 26, 2022).
- [83] A. Singh 'What Is Hydra Tool In Kali Linux And How Does It Work?', *OFFICIAL HACKER*, Jun. 15, 2017. <https://www.officialhacker.com/hydra-tool/> (accessed Mar. 15, 2022).
- [84] Royal Holloway University of London, 'Code of Good Practice for Research'. Dec. 11, 2020. [Online]. Available: <https://intranet.royalholloway.ac.uk/staff/assets/docs/pdf/policies-hub/research-and-enterprise/codeofgoodresearchpractice.pdf>
- [85] D. Strom, 'IoT device security: 9 ways to secure IoT devices', *HPE*, Jan. 26, 2017. <https://www.hpe.com/us/en/insights/articles/9-ways-to-make-iot-devices-more-secure-1701.html> (accessed Mar. 19, 2022).
- [86] S. Logic, 'What is IoT Security?', *Sumo Logic*, May 23, 2019. <https://www.sumologic.com/blog/iot-security/> (accessed Mar. 19, 2022).
- [87] R. Gandhi, 'IoT hardening · nebraska-gencyber-modules'. https://mlhale.github.io/nebraska-gencyber-modules/2021/iot_hardening/README/ (accessed Mar. 19, 2022).
- [88] S. Symanovich, '12 tips to secure your smart home and IoT devices'. <https://us.norton.com/internetsecurity-iot-smart-home-security-core.html> (accessed Mar. 19, 2022).
- [89] K. Allen, 'How to Improve the Security of Your Smart Home Devices [2022]', *CyberTakes*, Jan. 14, 2021. <https://www.cybertakes.com/improve-the-security-of-your-smart-home-devices/> (accessed Mar. 19, 2022).
- [90] F. Y. Rashid, 'How to secure your (easily hackable) smart home', *Tom's Guide*, Oct. 16, 2017. <https://www.tomsguide.com/us/secure-smart-home-how-to,news-19380.html> (accessed Mar. 19, 2022).
- [91] J. Cohen, 'How to protect your smart home from hackers', *PCMag*. <https://www.pcmag.com/how-to/how-to-protect-your-smart-home-from-hackers> (accessed Mar. 19, 2022).

- [92] Allen, 'Security Standards in IoT', *Security Boulevard*, Dec. 16, 2021. <https://securityboulevard.com/2021/12/security-standards-in-iot/> (accessed Mar. 16, 2022).
- [93] ENISA, *Good Practices for Security of IoT: Secure Software Development Lifecycle*. ENISA, 2019. [Online]. Available: <https://play.google.com/store/books/details?id=TatjzQEACAAJ>
- [94] C. Lévy-Bencheton, E. Darra, G. Tétu, G. Dufay, and M. Alattar, *Security and Resilience of Smart Home Environments: Good Practices and Recommendations*. Publications Office of the European Union, 2015. [Online]. Available: <https://play.google.com/store/books/details?id=ofSGnQAACAAJ>
- [95] European Telecommunications Standards Institute, 'Cyber Security for Consumer Internet of Things: Baseline Requirements', European Telecommunications Standards Institute, 650 Route des Lucioles F-06921 Sophia Antipolis Cedex - FRANCE, EN 303 645 V2.1.1, Jun. 2020. [Online]. Available: https://www.etsi.org/deliver/etsi_en/303600_303699/303645/02.01.01_60/en_303645v020101p.pdf
- [96] Cloud Security Alliance, 'CSA IoT Security Controls Framework v2'. Jan. 28, 2021. [Online]. Available: <https://cloudsecurityalliance.org/artifacts/csa-iot-security-controls-framework-v2/>
- [97] Center for Internet Security, 'CIS Controls v8 Internet of Things Companion Guide'. Sep. 02, 2021. [Online]. Available: <https://learn.cisecurity.org/CIS-Controls-v8-IoT-Companion-Guide>
- [98] G. Corser, G. Fink, and J. Bielby, 'Internet of Things (IoT) Security Best Practices; IEEE Internet Technology Policy Community; White Paper', *IEEE: Piscataway, NJ, USA*, 2017.
- [99] H. Snyder, 'Literature review as a research methodology: An overview and guidelines', *J. Bus. Res.*, vol. 104, pp. 333–339, Nov. 2019, doi: 10.1016/j.jbusres.2019.07.039.
- [100] D. J. Jones, 'Ethnographic Research: A Guide to General Conduct. R.F. ELLEN, ed', *American Ethnologist*, vol. 15, no. 2, pp. 403–404, 1988. doi: 10.1525/ae.1988.15.2.02a00290.
- [101] M. Crotty, *The foundations of social research: Meaning and perspective in the research process*. Routledge, 2020. [Online]. Available: <https://www.taylorfrancis.com/books/mono/10.4324/9781003115700/foundations-social-research-michael-crotty>
- [102] J. Grix, *The Foundations of Research*. Macmillan International Higher Education, 2018. [Online]. Available: <https://play.google.com/store/books/details?id=eqZyDwAAQBAJ>
- [103] A. A. Rehman and K. Alharthi, 'An introduction to research paradigms', *International Journal of Educational Investigations*, vol. 3, no. 8, pp. 51–59, 2016, [Online]. Available: <http://www.ijeionline.com/attachments/article/57/IJEI.Vol.3.No.8.05.pdf>
- [104] J. Rowley, 'Using case studies in research', *Management Research News*, vol. 25, no. 1, pp. 16–27, Jan. 2002, doi: 10.1108/01409170210782990.
- [105] S. Langkemper, 'Comparison of IoT Security Frameworks', *Eurofins Cybersecurity*, Sep. 23, 2020. <https://www.eurofins-cybersecurity.com/news/comparison-iot-security-frameworks/> (accessed Dec. 11, 2021).
- [106] Center for Internet Security (CIS), 'About us', *Center for Internet Security (CIS)*, Jan. 27, 2017.

<https://www.cisecurity.org/about-us/> (accessed Nov. 25, 2021).

- [107] S. Crowe, K. Cresswell, A. Robertson, G. Huby, A. Avery, and A. Sheikh, 'The case study approach', *BMC Med. Res. Methodol.*, vol. 11, p. 100, Jun. 2011, doi: 10.1186/1471-2288-11-100.
- [108] A. Echeverría, C. Cevallos, I. Ortiz-Garces, and R. O. Andrade, 'Cybersecurity Model Based on Hardening for Secure Internet of Things Implementation', *NATO Adv. Sci. Inst. Ser. E Appl. Sci.*, vol. 11, no. 7, p. 3260, Apr. 2021, doi: 10.3390/app11073260.
- [109] M. Bacon, 'CVSS (Common Vulnerability Scoring System)', *SearchSecurity*, Dec. 15, 2020. <https://www.techtarget.com/searchsecurity/definition/CVSS-Common-Vulnerability-Scoring-System> (accessed Mar. 26, 2022).
- [110] Initial State, 'How to build a raspberry pi temperature monitor', *Initial State*, Oct. 08, 2019. <https://medium.com/initial-state/how-to-build-a-raspberry-pi-temperature-monitor-8c2f70acaea9> (accessed Mar. 16, 2022).
- [111] S. Stahie, 'Common Credentials Criminals Use in IoT Dictionary Attacks Revealed', *Hot for Security*. <https://www.bitdefender.com/blog/hotforsecurity/common-credentials-criminals-use-in-iot-dictionary-attacks-revealed/> (accessed Mar. 16, 2022).
- [112] J. Garcia, 'Create a sudo user in Ubuntu -', May 20, 2020. <https://docs.rackspace.com/support/how-to/create-a-sudo-user-in-ubuntu/> (accessed Mar. 16, 2022).
- [113] Initial State, 'About Initial State'. <https://www.initialstate.com/about/> (accessed Mar. 26, 2022).
- [114] D. Swinhoe, 'What is a dictionary attack? And how you can easily stop them', *CSO Online*, Aug. 05, 2020. <https://www.csoonline.com/article/3568794/what-is-a-dictionary-attack-and-how-you-can-easily-stop-them.html> (accessed Mar. 19, 2022).
- [115] A. Ohri, 'Ping Flood or ICMP Flood Attack - A Simple Guide In 3 Points', *Jigsaw Academy*, Feb. 13, 2021. <https://www.jigsawacademy.com/blogs/cyber-security/ping-flood/> (accessed Mar. 26, 2022).
- [116] E. Fleming, 'What is hping3 command?' <https://www.sidmartinbio.org/what-is-hping3-command/> (accessed Mar. 26, 2022).
- [117] W. Ondara, 'How to enable automatic updates on Ubuntu 20.04', *LinOxide*, Jan. 26, 2021. <https://linoxide.com/enable-automatic-updates-on-ubuntu-20-04/> (accessed Mar. 23, 2022).
- [118] J. H. M. Price, 'Research Limitations and the Necessity of Reporting Them', *American Journal of Health Education; Reston volume*, vol. 35, no. 2. pp. 66–67, Apr. 2004. [Online]. Available: <https://search.proquest.com/openview/b6991f124333fca111dfbc6ef96d080c/1?pq-origsite=gscholar&cbl=44607>
- [119] S. McLeod, 'Case study Method', 2007, Accessed: Mar. 28, 2022. [Online]. Available: <https://www.simplypsychology.org/case-study.html>

- [120] D. De Guglielmo, S. Brienza, and G. Anastasi, 'IEEE 802.15.4e: A survey', *Comput. Commun.*, vol. 88, pp. 1–24, Aug. 2016, doi: 10.1016/j.comcom.2016.05.004.
- [121] J. DeLisle, 'What's the Difference Between IEEE 802.11af and 802.11ah?', Apr. 24, 2015. <https://www.mwrf.com/technologies/active-components/article/21846205/whats-the-difference-between-ieee-80211af-and-80211ah> (accessed Nov 11 2021).
- [122] M. Nixon and T. R. Rock, 'A Comparison of WirelessHART and ISA100. 11a', *Whitepaper, Emerson Process Management*, pp. 1–36, 2012, Accessed: Nov 11 2021. [Online]. Available: <https://www.controlglobal.com/assets/12WPpdf/120904-emerson-wirelesshart-isa.pdf>
- [123] S. Shea, 'Z-Wave', *TechTarget*, Aug. 29, 2018. <https://internetofthingsagenda.techtarget.com/definition/Z-Wave> (accessed Dec. 08, 2021).
- [124] TechTerms, 'BLE (Bluetooth Low Energy) Definition', *TechTerms*, 2019. <https://techterms.com/definition/ble> (accessed Nov 12 2021).
- [125] B. Garcia, 'Zigbee Smart Energy', *Teldat*, Nov. 06, 2018. <https://www.teldat.com/blog/zigbee-smart-energy-smart-metering-home-automation/> (accessed Nov. 12, 2021).
- [126] W. Ayoub, A. E. Samhat, F. Nouvel, M. Mroue, and J.-C. Prévotet, 'Internet of Mobile Things: Overview of LoRaWAN, DASH7, and NB-IoT in LPWANs Standards and Supported Mobility', *IEEE Communications Surveys Tutorials*, vol. 21, no. 2, pp. 1561–1581, 2019, doi: 10.1109/COMST.2018.2877382.
- [127] L. Yonge, J. Abad, K. Afkhamie, L. Guerrieri, S. Katar, H. Lioe, P. Pagani, R. Riva, D.M. Schneider, and A. Schwager, 'An Overview of the HomePlug AV2 Technology', *J. Electr. Comput. Eng.*, vol. 2013, Mar. 2013, doi: 10.1155/2013/892628.
- [128] E. Dashevsky, 'FAQ: How is LTE-Advanced different from regular LTE?', *PCWorld*, Jan. 21, 2014. <https://www.pcworld.com/article/443178/faq-how-is-lte-advanced-different-from-regular-lte.html> (accessed Nov. 09, 2021).
- [129] B. Foubert and N. Mitton, 'Long-Range Wireless Radio Technologies: A Survey', *Future Internet*, vol. 12, no. 1, p. 13, Jan. 2020, doi: 10.3390/fi12010013.
- [130] B. Buckiewicz, 'A Technical Overview of DECT ULE', *Laird Conenctivity*, Feb. 04, 2019. <https://www.lairdconnect.com/resources/white-papers/technical-overview-of-dect-ule> (accessed Nov. 08, 2021).
- [131] T. Clausen, U. Herberg, and M. Philipp, 'A critical evaluation of the IPv6 Routing Protocol for Low Power and Lossy Networks (RPL)', *2011 IEEE 7th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*. 2011. doi: 10.1109/wimob.2011.6085374.
- [132] A. Triantafyllou, P. Sarigiannidis, and T. D. Lagkas, 'Network Protocols, Schemes, and Mechanisms for Internet of Things (IoT): Features, Open Challenges, and Trends', *Proc. Int. Wirel. Commun. Mob. Comput. Conf.*, vol. 2018, Sep. 2018, doi: 10.1155/2018/5349894.

- [133] J. Olsson, '6LoWPAN demystified', *Texas Instruments*, vol. 13, 2014, Accessed: Nov. 11, 2021. [Online]. Available: <http://caxapa.ru/thumbs/784686/swry013.pdf>
- [134] X. Vilajosana, T. Watteyne, T. Chang, M. Vučinić, S. Duquennoy, and P. Thubert, 'IETF 6TiSCH: A Tutorial', *IEEE Communications Surveys Tutorials*, vol. 22, no. 1, pp. 595–615, 2020, doi: 10.1109/COMST.2019.2939407.
- [135] S. Chakrabarti, 'Working Group Update: 6lo', *IETF Journal*, Nov. 01, 2016. <https://www.ietfjournal.org/working-group-update-6lo/> (accessed Nov. 11, 2021).
- [136] C. Bernstein, K. Brush, and A. S. Gillis, 'What is MQTT and How Does it Work?', *TechTarget*, Jan. 27, 2021. <https://internetofthingsagenda.techtarget.com/definition/MQTT-MQ-Telemetry-Transport> (accessed Nov. 13, 2021).
- [137] Priya, 'Application layer protocols for IOT : IOT part 11', *Engineers Garage*, Mar. 06, 2021. <https://www.engineersgarage.com/application-layer-protocols-for-iot-iot-part-11/> (accessed Nov. 13, 2021).
- [138] L. Johansson, 'What is AMQP and why is it used in RabbitMQ?', *Cloud AMQP*, Nov. 21, 2019. <https://www.cloudamqp.com/blog/what-is-amqp-and-why-is-it-used-in-rabbitmq.html> (accessed Nov. 13, 2021).
- [139] Z. Shelby, K. Hartke, and C. Bormann, 'The constrained application protocol (CoAP)', Jun. 2014, Accessed: Nov. 08, 2021. [Online]. Available: https://iottestware.readthedocs.io/en/master/coap_rfc.html
- [140] Mike, 'XMPP refresher: The open instant messaging protocol then & now', *GetStream*, Jan. 2021. <https://getstream.io/blog/xmpp-extensible-messaging-presence-protocol/> (accessed Nov. 13, 2021).

Appendices

Appendix A - IoT Protocols

Table A1: IoT Data Link Protocols [43]

Network protocols for IoT	Full form	Definition
IEEE 802.15.4e		<ul style="list-style-type: none"> • Defines both the physical and the MAC layers; • Referenced standard for commercial Wireless Sensor Networks (WSNs); • The protocol improves upon its predecessor (802.15.4e) as it introduces mechanisms such as multichannel communication, channel hopping and time slotted access; • Five MAC behaviour modes or protocols are defined to cater for specific or application domains [120].
IEEE 802.11 ah		<ul style="list-style-type: none"> • Low-power and low-throughput; • Allows data transmission with low on-time for sensors; • Enables sensor traffic priority, beaconless paging modes and smaller-frame formats for low-power applications [121].
WirelessHART	Wireless Highway Addressable Remote Transducer	<ul style="list-style-type: none"> • Based on the Highway Addressable Remote Transducer (HART) communication protocol; • HART progressed from the late 1980s to a protocol which now caters for both wired and wireless technologies and for more rigorous security; • WirelessHART was designed to fully support several devices such as sensors and actuators [122].
Z-Wave		<ul style="list-style-type: none"> • Used mostly in smart homes since it helps reduce power consumption in devices; • Allows for sensors to connect smart home devices and also enables data and control commands exchange between devices; • Is considered better than Wi-Fi since it consumes less power and also better than Bluetooth since the range is effectively longer [123].
BLE	Bluetooth Low Energy	<ul style="list-style-type: none"> • As the name denotes it is based on Bluetooth but aimed for devices with low power capabilities; • Introduced in December 2009; • Used in smart homes and wearable devices since these devices often do not require a steady connection and have limited electrical power [124].

Zigbee Smart Energy	Not Applicable	<ul style="list-style-type: none"> ● Uses radio frequencies; ● Used mainly to manage, automate and monitor the generation and consumption of water, energy and gas [125].
D7AP		<ul style="list-style-type: none"> ● Specified by the DASH7; ● Optimised high level communication stack from physical to application layer for Radio Frequency Identification (RFID) and Wireless Sensors and Actuators Networks (WSAN); ● Interoperability between different devices and operators is achieved through this protocol [126].
HomePlug		<ul style="list-style-type: none"> ● Enables full interoperability with other technologies through existing power line wires whilst also achieving Gigabit-class connection speeds; ● Homeplug v2 defined further features at Physical and MAC layers [127].
G.9959		<ul style="list-style-type: none"> ● Designed by ITU to cater for half-duplex reliable wireless communication within low cost and low bandwidth environments; ● Operates within the MAC layer; ● Designed for applications operating in real time where low power consumption and reliability are required [43].
LTE-A	Long Term Evolution Advanced	<ul style="list-style-type: none"> ● Stands for “long term evolution”; ● Globally recognised and used standard (LTE); ● More advanced than previous LTE standard providing faster and larger wireless-data; ● Promised to deliver 4G speeds [128].
LoRaWAN	Low Power Wide Area Network	<ul style="list-style-type: none"> ● Protocol stack operating on unlicensed bands over the Long Range (LoRa) physical layer designed by the LoRa Alliance. ● Features include several operating classes for various applications, low complexity and low data rate [126].
Weightless		<ul style="list-style-type: none"> ● Developed by the Weightless Special Interest Group; ● Three variants: W, P and N; ● Weightless-W: <ul style="list-style-type: none"> ○ Uses TV white space frequencies; ○ Data rate between 1 kbps and 10 Mbps using a narrow-band signal; ● Weightless-N: <ul style="list-style-type: none"> ○ Based on UNB; ○ Similar to Sigfox but with a higher data rate (30 to 100 kbps); ● Weightless-P:

		<ul style="list-style-type: none">○ Bidirectional connectivity;○ Based on ISM sub-GHz bands;○ Data rate between 0.2kbps and 100 kbps [129].
DECT/ULE	Digital enhanced cordless telecommunications /Ultra Low Energy	<ul style="list-style-type: none">● DECT launched in 1987 to cater for cordless phone communications;● Continuously evolved with the latest version being DECT Ultra Low Energy (ULE);● ULE features low energy consumption and cost, long range without interface and very stable bit-rates making it perfect for home automation and security;● Without deploying a mesh topology ULE can provide up to over 600 metres outdoors and over 70 metres range indoors [130].

Table A2: Network Layer Routing Protocols [43]

Network protocols for IoT	Full form	Definition
RPL	Routing Protocol for Low-Power and Lossy Networks	<ul style="list-style-type: none"> ● Low-power Lossy Networks (LLNs) IPv6 routing protocol; ● Proposed by ROLL Working Group within IETF; ● Intended for all LLNs; ● Aimed to provide an IP-based routing standard through preventing fragmentation; ● Optimisation of multipoint-to-point traffic, support of point-to-multipoint traffic and provisioning of basic features in point-to-point traffic are goals of RPL [131].
CORPL	Cognitive Routing Protocol for Low-Power and Lossy Networks	<ul style="list-style-type: none"> ● Extension of RPL developed for cognitive networks; ● Uses DODAG topology generation; ● Through opportunistic data transmission, packets are forwarded to a forwarder set; ● Data is sent to the best hop whilst each node informs other nodes to update the priorities within the forwarder set [132].
CARP	Channel-Aware Routing Protocol	<ul style="list-style-type: none"> ● Routing protocol for Underwater Wireless Sensor Networks (UWSNs); ● Aimed to deliver packets within acceptable time frames within low energy devices; ● Through previous successful data transfer collected from neighbouring sensors, link quality information is supported [132].

Table A3: Network Layer Encapsulation Protocols [43]

Network protocols for IoT	Full form	Definition
6LoWPAN		<ul style="list-style-type: none"> ● Defined in RFC 6282 by IETF; ● Supports IEEE 802.15.4 low-power networks in the 2.4 GHz band; ● Adapted and utilised by other devices even within the sub-1 GHz band [133].
6TiSCH	IETF IPv6 over the TSCH mode	<ul style="list-style-type: none"> ● Enables low power-industrial-grade IPv6 networks; ● Multi-hop topologies through IPv6 RPL is supported; ● Established the control protocols to link the application communication needs and routing topology to the link-layer resources; ● Established the means to address the difficulties when attempting to build IPv6 networks with low capacities [134].
6Lo		<ul style="list-style-type: none"> ● 6Lo provides the specifications for IPv6 through constrained nodes networks; ● Optimises network bandwidth and device energy usage in scenarios where memory, CPU resources and power are limited [135].
IPv6 over G.9959 (RFC 7428)		<ul style="list-style-type: none"> ● Provides the definitions on how the frames must be formatted when a IPv6 packets are transmitted on ITU-T G.9959 networks; ● Same header compression technique used that of 6lowPAN; ● Through a shared network key used for encryption provides a basic level of security [43].
IPv6 over Bluetooth Low Energy		<ul style="list-style-type: none"> ● Also known as Bluetooth Smart; ● Initially introduced in Bluetooth 4.0 and further improved in Bluetooth 4.1; ● Uses techniques for compression as noted within 6LowPAN; ● Fragmentation from 6LowPAN is not utilised; ● Uses a central node acting as a router in the middle of other nodes rather than supporting multi-hop networks [43].

Table A4: Session Layer Protocols [43]

Network protocols for IoT	Full form	Definition
MQTT	Message Queue Telemetry Transport	<ul style="list-style-type: none"> • Lightweight open messaging protocol designed to cater for limitations in CPU and bandwidth; • Designed to provide communication in a reliable and effective manner; • Used for m2m communication [136].
SMQTT	Secure Message Queue Telemetry Transport	<ul style="list-style-type: none"> • Based on MQTT with encryption; • Includes four steps which are setup, encryption, publish and decryption; • Similar to MQTT but with included secret master key registration between subscriber and publisher; • Publisher encrypts data whereas subscriber decrypts the data [137].
AMQP	Advanced Message Queuing Protocol	<ul style="list-style-type: none"> • Application layer protocol allowing messaging between systems on whichever broker or platform used; • Allows interactions between client applications and the server; • Provides a high-level architecture for message brokers [138].
CoAP	Constrained Application Protocol	<ul style="list-style-type: none"> • Designed to provide web transfer protocol for machine-to-machine applications within low-power networks; • Aims to interface with HTTP for integration with the Web; • Meets requirements such as low overhead, simplicity and multicast support [139].
XMPP	Extensible Messaging and Presence Protocol	<ul style="list-style-type: none"> • Supports nearly real time messaging through the interchange of XML data through the network; • Through TCP, XML snippets (also known as stanzas) are passed reliably through a server in the middle of the communication [140].
DDS	Data Distribution Service	<ul style="list-style-type: none"> • Designed for M2M communications by the Object Management Group (OMG); • Relies on a broker-less architecture thus being very useful for IoT and M2M communication whilst also providing reliability and quality of service; • Offers other services such as security, priority, reliability and durability [43].

Appendix B - OWASP IoT Security Verification Standard (ISVS) Applicable Controls

Table B1: OWASP ISVS [21] Applicable Controls and Control score according to OWASP IoT Top 10

Control Number	Control Title	Control Score
1.1.1	Verify that all applications in the IoT ecosystem are developed with a level of security that is in line with the security criticality of the application.	8
1.1.2	Verify that all components and communication channels in the IoT application's ecosystem have been identified and are known to be needed. Remove or disable any that aren't necessary.	8
1.1.4	Verify that security controls are enforced server-side and that data and instructions are not blindly trusted by server-side components.	8
1.2.2	Verify that potential areas of risk that come with the use of third-party and open-source software have been identified and that actions to mitigate such risks have been taken.	6
1.3.4	Verify packages are downloaded and built from trusted sources.	8
1.4.1	Verify that devices can collect logs about events with security implications, such as successful and failed authentication attempts, access to debugging functionality etc.	3
1.4.2	Verify that collected logs have the adequate granularity to enable actionable insights and alerts. Logs should include, at a minimum, the type of event, timestamp, source, outcome, and identification of involved actors.	3
1.4.3	Verify that devices contain or are synchronised with a reliable time source, to ensure the validity of log timestamps.	3

1.4.4	Verify that collected logs do not include sensitive information, such as PII, credentials and cryptographic keys.	5
1.4.5	Verify that collected logs can be securely retrieved from the devices over an online collection, either periodically or on-demand.	4
1.4.7	Verify that the confidentiality, integrity and authenticity of collected logs is appropriately protected, both on the devices that created them and on other systems that store or process them.	4
2.1.1	Verify that all forms of users and accounts in the IoT ecosystem can be uniquely identified.	8
2.1.2	Verify that all connected devices within the IoT ecosystem can be uniquely identified including connected to the cloud, hubs, as well as to other devices (sensors).	8
2.1.3	Verify strong user and device authentication is enforced across the IoT ecosystem.	10
2.1.4	Verify that user, service, and device authentication schemes share a common framework centrally managed in the IoT ecosystem.	10
2.1.5	Verify that passwords used for user authentication are at least 12 characters long.	10
2.1.6	Verify that passwords used for user authentication can be changed by the user and that the password change functionality requires the user's current and new password.	10
2.1.7	Verify that passwords used for device authentication are sufficiently long and complex.	10
2.1.8	Verify that default user or device credentials can be changed by authorised administrators or end-users.	10

2.1.9	Verify that authentication credentials for users, devices, or services are not hardcoded in firmware or ecosystem applications.	10
2.1.10	Verify that provisioned credentials for device authentication are unique per device.	10
2.1.11	Verify that authentication schemes are designed to revoke credentials of compromised or decommissioned devices.	10
2.2.1	Verify that IoT system accounts across users, services and devices share a common authorization framework.	8
2.2.2	Verify that devices enforce the concept of least privilege by limiting applications and services that run as root or administrator.	8
2.3.1	Verify that sensitive information such as personal identifiable information (PII) and credentials are stored securely using strong encryption to protect from data leakage and integrity checking to protect against unauthorised modification.	10
3.2.1	Verify that the embedded operating system is configured according to the latest industry best practices, CIS or SCAP benchmarks (if applicable), and uses secure defaults.	6
3.2.2	Verify that all network services exposed by the device on every network interface are necessary services and unnecessary services are removed or disabled.	9
3.2.3	Verify that the device does not make use of legacy or insecure protocols such as Telnet and FTP.	9
3.2.4	Verify that the OS kernel and software components are up to date and do not contain known vulnerabilities.	6
3.2.5	Verify that persistent filesystem storage volumes are encrypted.	6

3.4.1	Verify that packages and user space applications use over the air updates decoupled from firmware updates.	7
3.4.2	Verify that devices can be updated automatically upon a predefined schedule.	7
3.4.3	Verify that updates are cryptographically signed by a trusted source and their authenticity is verified before execution.	7
3.4.5	Verify that updates do not modify user-configured preferences, security, and/or privacy settings without notifying the user.	7
4.1.1	Verify that communication with other components in the IoT ecosystem (including sensors, gateway and supporting cloud) occurs over a secure channel in which the confidentiality and integrity of data is guaranteed and in which protection against replay attacks is built into the communication protocol.	8
4.2.1	Verify that unencrypted communication is limited to data and instructions that are not of a sensitive nature.	8
4.3.1	Verify that pairing and discovery is blocked in Bluetooth devices except when necessary.	9
4.3.2	Verify that PIN or Passkey codes are not easily guessable (e.g., don't use 0000 or 1234).	10
4.3.3	Verify that devices using old versions of Bluetooth with simple modes of authentication enabled require a PIN for pairing.	9
4.3.4	Verify that for modern versions of Bluetooth, at least 6 digits are required for Secure Simple Pairing (SSP) authentication under all versions except "Just Works".	9
4.4.1	Verify Wi-Fi connectivity is disabled unless required as part of device functionality. Devices with no need for network connectivity or which support other types of network connectivity, such as Ethernet, should have the Wi-Fi interface disabled.	9

4.4.2	Verify that WPA2 or higher is used to protect Wi-Fi communications.	9
4.5.1	Verify that Zigbee version 3.0 is used for new applications.	9
4.5.2	Verify that a suitable Zigbee security architecture (Centralised or Distributed) is selected, depending on the application's security level requirements and threat model. The Centralised architecture generally offers higher security at the cost of flexibility.	9
4.5.3	Verify that the most secure way of joining the Zigbee network is used, depending on the selected security architecture. For example, for the Centralised architecture, use out-of-band install codes. For the Distributed one, use pre-configured link keys.	9
4.6.1	Verify that LoRaWAN version 1.1 is used by new applications.	9
4.6.2	Verify that the network, join and application servers of the LoRaWAN ecosystem are appropriately hardened according to industry best practices and benchmarks.	9
4.6.3	Verify that all communication between the LoRaWAN gateway and the network, join and application servers occur over a secure channel (for example TLS or IPsec), guaranteeing at least the integrity and authenticity of the messages.	9

Appendix C - CIS Controls Internet of Things Companion Guide

Applicable Controls

Table C1 : CIS [97] Applicable Controls and Control score according to OWASP IoT Top 10

Control Number	Control Title	Control Score
1.1	Establish and Maintain Detailed Enterprise Asset Inventory	3
1.2	Address Unauthorised Assets	3
1.3	Utilise an Active Discovery Tool	3
1.4	Use Dynamic Host Configuration Protocol (DHCP) Logging to Update Enterprise Asset Inventory	3
2.1	Establish and Maintain a Software Inventory	6
2.2	Ensure Authorised Software Is Currently Supported	6
2.7	Allowlist Authorised Scripts	6
4.1	Establish and Maintain a Secure Configuration Process	9
4.2	Establish and Maintain a Secure Configuration Process for Network Infrastructure	9
4.3	Configure Automatic Session Locking on Enterprise Assets	8
4.4	Implement and Manage a Firewall on Servers	9
4.5	Implement and Manage a Firewall on End-User Devices	9
4.6	Securely Manage Enterprise Assets and Software	3

4.7	Manage Default Accounts on Enterprise Assets and Software	10
4.8	Uninstall or Disable Unnecessary Services on Enterprise Assets and Applications	9
4.9	Configure Trusted DNS Servers on Enterprise Assets	9
5.1	Establish and Maintain an Inventory of Accounts	10
5.2	Use Unique Passwords	10
5.4	Restrict Administrator Privileges to Dedicated Administrator Accounts	10
5.5	Establish and Maintain an Inventory of Service Accounts	10
5.6	Centralize Account Management	10
6.6	Establish and Maintain an Inventory of Authentication and Authorization Systems	10
6.7	Centralize Access Control	10
7.1	Establish and Maintain a Vulnerability Management Process	7
7.2	Establish and Maintain a Remediation Process	7
7.3	Perform Automated Operating System Patch Management	7
7.4	Perform Automated Application Patch Management	7
7.5	Perform Automated Vulnerability Scans of Internal Enterprise Assets	7
7.7	Remediate Detected Vulnerabilities	7
9.1	Ensure Use of Only Fully Supported Browsers and Email Clients	8
9.6	Block Unnecessary File Types	9
10.1	Deploy and Maintain Anti-Malware Software	3

10.2	Configure Automatic Anti-Malware Signature Updates	3
10.3	Disable Autorun and Autoplay for Removable Media	3
10.4	Configure Automatic Anti-Malware Scanning of Removable Media	3
10.5	Enable Anti-Exploitation Features	3
10.6	Centrally Manage Anti-Malware Software	3
10.7	Use Behaviour-Based Anti-Malware Software	3
12.1	Ensure Network Infrastructure is Up-to-Date	9
12.2	Establish and Maintain a Secure Network Architecture	9
12.3	Securely Manage Network Infrastructure	9
12.4	Establish and Maintain Architecture Diagram(s)	9
12.5	Centralise Network Authentication, Authorization, and Auditing (AAA)	9
12.6	Use of Secure Network Management and Communication Protocols	9
16.4	Establish and Manage an Inventory of Third-Party Software Components	3
16.5	Use Up-to-Date and Trusted Third-Party Software Components	6
16.6	Establish and Maintain a Severity Rating System and Process for Application Vulnerabilities	3
16.7	Use Standard Hardening Configuration Templates for Application Infrastructure	8
18.1	Establish and Maintain a Penetration Testing Program	8
18.2	Perform Periodic External Penetration Tests	8
18.3	Remediate Penetration Test Findings	8

18.4	Validate Security Measures	8
18.5	Perform Periodic Internal Penetration Tests	8

Appendix D - ENISA Baseline Security Recommendations for IoT in the context of Critical Information Infrastructures Applicable Controls

Table D1: ENISA [3] Applicable Controls and Control score according to OWASP IoT Top 10

Control Number	Control Title	Control Score
GP-TM-07	Use protocols and mechanisms able to represent and manage trust and trust relationships.	8
GP-TM-08	Any applicable security features should be enabled by default, and any unused or insecure functionalities should be disabled by default.	6
GP-TM-09	Establish hard to crack, device-individual default passwords.	10
GP-TM-18	Ensure that the device software/firmware, its configuration and its applications have the ability to update Over-The-Air (OTA), that the update server is secure, that the update file is transmitted via a secure connection, that it does not contain sensitive data (e.g. hardcoded credentials), that it is signed by an authorised trust entity and encrypted using accepted encryption methods, and that the update package has its digital signature, signing certificate and signing certificate chain, verified by the device before the update process begins.	7
GP-TM-19	Offer an automatic firmware update mechanism.	7
GP-TM-20	Backward compatibility of firmware updates. Automatic firmware updates should not modify user-configured preferences, security, and/or privacy settings without user notification.	7
GP-TM-21	Design the authentication and authorisation schemes (unique per device) based on the system-level threat models.	10

GP-TM-22	Ensure that default passwords and even default usernames are changed during the initial setup, and that weak, null or blank passwords are not allowed.	10
GP-TM-23	Authentication mechanisms must use strong passwords or personal identification numbers (PINs), and should consider using two-factor authentication (2FA) or multi-factor authentication (MFA) like Smartphones, Biometrics, etc., on top of certificates.	10
GP-TM-24	Authentication credentials shall be salted, hashed and/or encrypted.	10
GP-TM-25	Protect against 'brute force' and/or other abusive login attempts. This protection should also consider keys stored in devices.	10
GP-TM-26	Ensure password recovery or reset mechanism is robust and does not supply an attacker with information indicating a valid account. The same applies to key update and recovery mechanisms.	10
GP-TM-27	Limit the actions allowed for a given system by Implementing fine-grained authorisation mechanisms and using the Principle of least privilege (POLP): applications must operate at the lowest privilege level possible.	10
GP-TM-28	Device firmware should be designed to isolate privileged code, processes and data from portions of the firmware that do not need access to them. Device hardware should provide isolation concepts to prevent unprivileged from accessing security sensitive code.	8
GP-TM-29	Data integrity and confidentiality must be enforced by access controls. When the subject requesting access has been authorised to access particular processes, it is necessary to enforce the defined security policy.	10
GP-TM-30	Ensure a context-based security and privacy that reflects different levels of importance.	3
GP-TM-33	Ensure that devices only feature the essential physical external ports (such as	1

	USB) necessary for them to function and that the test/debug modes are secure, so they cannot be used to maliciously access the devices. In general, lock down physical ports to only trusted connections.	
GP-TM-40	Ensure credentials are not exposed in internal or external network traffic.	9
GP-TM-41	Guarantee data authenticity to enable reliable exchanges from data emission to data reception. Data should always be signed whenever and wherever it is captured and stored.	9
GP-TM-42	Do not trust data received and always verify any interconnections. Discover, identify and verify/authenticate the devices connected to the network before trust can be established, and preserve their integrity for reliable solutions and services.	9
GP-TM-43	IoT devices should be restrictive rather than permissive in communicating.	9
GP-TM-44	Make intentional connections. Prevent unauthorised connections to it or other devices the product is connected to, at all levels of the protocols.	9
GP-TM-45	Disable specific ports and/or network connections for selective connectivity.	9
GP-TM-46	Rate limiting. Controlling the traffic sent or received by a network to reduce the risk of automated attacks.	9
GP-TM-47	Risk Segmentation. Splitting network elements into separate components to help isolate security breaches and minimise the overall risk.	9
GP-TM-48	Protocols should be designed to ensure that, if a single device is compromised, it does not affect the whole set.	9
GP-TM-49	Avoid provisioning the same secret key in an entire product family, since compromising a single device would be enough to expose the rest of the	9

	product family.	
GP-TM-50	Ensure only necessary ports are exposed and available.	9
GP-TM-51	Implement a DDoS-resistant and Load-Balancing infrastructure.	9
GP-TM-55	Implement a logging system that records events relating to user authentication, management of accounts and access rights, modifications to security rules, and the functioning of the system. Logs must be preserved in durable storage and retrievable via authenticated connections.	3
GP-TM-56	Implement regular monitoring to verify the device behaviour, to detect malware and to discover integrity errors.	3
GP-TM-57	Conduct periodic audits and reviews of security controls to ensure that the controls are effective. Perform penetration tests at least biannually.	3

Appendix E - Recommended Hardening Procedure

Table E1: Recommended Hardening Procedure Controls and Control score according to OWASP IoT Top 10

Control Number	Control Title	Control Score
2.1.3	Verify strong user and device authentication is enforced across the IoT ecosystem.	10
2.1.4	Verify that user, service, and device authentication schemes share a common framework centrally managed in the IoT ecosystem.	10
2.1.5	Verify that passwords used for user authentication are at least 12 characters long.	10
2.1.6	Verify that passwords used for user authentication can be changed by the user and that the password change functionality requires the user's current and new password.	10
2.1.7	Verify that passwords used for device authentication are sufficiently long and complex.	10
2.1.8	Verify that default user or device credentials can be changed by authorized administrators or end-users.	10
2.1.9	Verify that authentication credentials for users, devices, or services are not hardcoded in firmware or ecosystem applications.	10
2.1.10	Verify that provisioned credentials for device authentication are unique per device.	10
2.1.11	Verify that authentication schemes are designed to revoke credentials of compromised or decommissioned devices.	10
2.3.1	Verify that sensitive information such as personal identifiable information (PII) and credentials are stored securely using strong encryption to protect from data leakage and integrity checking to protect against unauthorized modification.	10
4.3.2	Verify that PIN or PassKey codes are not easily guessable (e.g. don't use 0000 or 1234).	10

5.1	Establish and Maintain an Inventory of Accounts	10
5.4	Restrict Administrator Privileges to Dedicated Administrator Accounts	10
5.5	Establish and Maintain an Inventory of Service Accounts	10
5.6	Centralize Account Management	10
6.6	Establish and Maintain an Inventory of Authentication and Authorization Systems	10
6.7	Centralize Access Control	10
GP-TM-21	Design the authentication and authorisation schemes (unique per device) based on the system-level threat models.	10
GP-TM-22	Ensure that default passwords and even default usernames are changed during the initial setup, and that weak, null or blank passwords are not allowed.	10
GP-TM-23	Authentication mechanisms must use strong passwords or personal identification numbers (PINs), and should consider using two-factor authentication (2FA) or multi-factor authentication (MFA) like Smartphones, Biometrics, etc., on top of certificates.	10
GP-TM-25	Protect against 'brute force' and/or other abusive login attempts. This protection should also consider keys stored in devices.	10
GP-TM-26	Ensure password recovery or reset mechanism is robust and does not supply an attacker with information indicating a valid account. The same applies to key update and recovery mechanisms.	10
GP-TM-27	Limit the actions allowed for a given system by Implementing fine-grained authorisation mechanisms and using the Principle of least privilege (POLP): applications must operate at the lowest privilege level possible.	10
GP-TM-29	Data integrity and confidentiality must be enforced by access controls. When the subject requesting access has been authorised to access particular processes, it is necessary to enforce the defined security policy.	10
3.2.2	Verify that all network services exposed by the device on every network interface are necessary services and unnecessary services are removed or disabled.	9

3.2.3	Verify that the device does not make use of legacy or insecure protocols such as Telnet and FTP.	9
4.3.1	Verify that pairing and discovery is blocked in Bluetooth devices except when necessary.	9
4.3.3	Verify that devices using old versions of Bluetooth with simple modes of authentication enabled require a PIN for pairing.	9
4.3.4	Verify that for modern versions of Bluetooth, at least 6 digits are required for Secure Simple Pairing (SSP) authentication under all versions except “Just Works”.	9
4.4.1	Verify Wi-Fi connectivity is disabled unless required as part of device functionality. Devices with no need for network connectivity or which support other types of network connectivity, such as Ethernet, should have the Wi-Fi interface disabled.	9
4.4.2	Verify that WPA2 or higher is used to protect Wi-Fi communications.	9
4.5.1	Verify that Zigbee version 3.0 is used for new applications.	9
4.5.2	Verify that a suitable Zigbee security architecture (Centralized or Distributed) is selected, depending on the application's security level requirements and threat model. The Centralized architecture generally offers higher security at the cost of flexibility.	9
4.5.3	Verify that the most secure way of joining the Zigbee network is used, depending on the selected security architecture. For example, for the Centralized architecture, use out-of-band install codes. For the Distributed one, use pre-configured link keys.	9
4.6.1	Verify that LoRaWAN version 1.1 is used by new applications.	9
4.6.2	Verify that the network, join and application servers of the LoRaWAN ecosystem are appropriately hardened according to industry best practices and benchmarks.	9
4.6.3	Verify that all communication between the LoRaWAN gateway and the network, join and application servers occurs over a secure channel (for example TLS or IPsec), guaranteeing at least the integrity and authenticity of the messages.	9

4.1	Establish and Maintain a Secure Configuration Process	9
4.2	Establish and Maintain a Secure Configuration Process for Network Infrastructure	9
4.4	Implement and Manage a Firewall on Servers	9
4.5	Implement and Manage a Firewall on End-User Devices	9
4.8	Uninstall or Disable Unnecessary Services on Enterprise Assets and Applications	9
4.9	Configure Trusted DNS Servers on Enterprise Assets	9
9.6	Block Unnecessary File Types	9
12.1	Ensure Network Infrastructure is Up-to-Date	9
12.2	Establish and Maintain a Secure Network Architecture	9
12.3	Securely Manage Network Infrastructure	9
12.4	Establish and Maintain Architecture Diagram(s)	9
12.5	Centralize Network Authentication, Authorization, and Auditing (AAA)	9
12.6	Use of Secure Network Management and Communication Protocols	9
GP-TM-40	Ensure credentials are not exposed in internal or external network traffic.	9
GP-TM-41	Guarantee data authenticity to enable reliable exchanges from data emission to data reception. Data should always be signed whenever and wherever it is captured and stored.	9
GP-TM-42	Do not trust data received and always verify any interconnections. Discover, identify and verify/authenticate the devices connected to the network before trust can be established, and preserve their integrity for reliable solutions and services.	9
GP-TM-43	IoT devices should be restrictive rather than permissive in communicating.	9
GP-TM-44	Make intentional connections. Prevent unauthorised connections to it or other devices the product is connected to, at all levels of the protocols.	9

GP-TM-46	Rate limiting. Controlling the traffic sent or received by a network to reduce the risk of automated attacks.	9
GP-TM-47	Risk Segmentation. Splitting network elements into separate components to help isolate security breaches and minimise the overall risk.	9
GP-TM-48	Protocols should be designed to ensure that, if a single device is compromised, it does not affect the whole set.	9
GP-TM-49	Avoid provisioning the same secret key in an entire product family, since compromising a single device would be enough to expose the rest of the product family.	9
GP-TM-51	Implement a DDoS-resistant and Load-Balancing infrastructure.	9
1.1.1	Verify that all applications in the IoT ecosystem are developed with a level of security that is in line with the security criticality of the application.	8
1.1.4	Verify that security controls are enforced server-side and that data and instructions are not blindly trusted by server-side components.	8
1.3.4	Verify packages are downloaded and built from trusted sources	8
2.1.1	Verify that all forms of users and accounts in the IoT ecosystem can be uniquely identified.	8
2.1.2	Verify that all connected devices within the IoT ecosystem can be uniquely identified including connected to the cloud, hubs, as well as to other devices (sensors).	8
2.2.1	Verify that IoT system accounts across users, services and devices share a common authorization framework.	8
4.1.1	Verify that communication with other components in the IoT ecosystem (including sensors, gateway and supporting cloud) occurs over a secure channel in which the confidentiality and integrity of data is guaranteed and in which protection against replay attacks is built into the communication protocol.	8
4.2.1	Verify that unencrypted communication is limited to data and instructions that are not of a sensitive nature.	8

4.3	Configure Automatic Session Locking on Enterprise Assets	8
9.1	Ensure Use of Only Fully Supported Browsers and Email Clients	8
16.7	Use Standard Hardening Configuration Templates for Application Infrastructure	8
18.1	Establish and Maintain a Penetration Testing Program	8
18.2	Perform Periodic External Penetration Tests	8
18.3	Remediate Penetration Test Findings	8
18.4	Validate Security Measures	8
18.5	Perform Periodic Internal Penetration Tests	8
GP-TM-07	Use protocols and mechanisms able to represent and manage trust and trust relationships.	8
GP-TM-28	Device firmware should be designed to isolate privileged code, processes and data from portions of the firmware that do not need access to them. Device hardware should provide isolation concepts to prevent unprivileged from accessing security sensitive code.	8
3.4.1	Verify that packages and user space applications use over the air updates decoupled from firmware updates.	7
3.4.2	Verify that devices can be updated automatically upon a predefined schedule.	7
3.4.3	Verify that updates are cryptographically signed by a trusted source and their authenticity is verified before execution.	7
3.4.5	Verify that updates do not modify user-configured preferences, security, and/or privacy settings without notifying the user.	7
7.1	Establish and Maintain a Vulnerability Management Process	7
7.2	Establish and Maintain a Remediation Process	7
7.3	Perform Automated Operating System Patch Management	7
7.4	Perform Automated Application Patch Management	7

7.5	Perform Automated Vulnerability Scans of Internal Enterprise Assets	7
7.7	Remediate Detected Vulnerabilities	7
GP-TM-18	Ensure that the device software/firmware, its configuration and its applications have the ability to update Over-The-Air (OTA), that the update server is secure, that the update file is transmitted via a secure connection, that it does not contain sensitive data (e.g. hardcoded credentials), that it is signed by an authorised trust entity and encrypted using accepted encryption methods, and that the update package has its digital signature, signing certificate and signing certificate chain, verified by the device before the update process begins.	7
GP-TM-19	Offer an automatic firmware update mechanism.	7
GP-TM-20	Backward compatibility of firmware updates. Automatic firmware updates should not modify user-configured preferences, security, and/or privacy settings without user notification.	7
1.2.2	Verify that potential areas of risk that come with the use of third-party and open-source software have been identified and that actions to mitigate such risks have been taken.	6
3.2.1	Verify that the embedded operating system is configured according to the latest industry best practices, CIS or SCAP benchmarks (if applicable), and uses secure defaults.	6
3.2.4	Verify that the OS kernel and software components are up to date and do not contain known vulnerabilities.	6
3.2.5	Verify that persistent filesystem storage volumes are encrypted.	6
2.1	Establish and Maintain a Software Inventory	6
2.2	Ensure Authorized Software Is Currently Supported	6
2.7	Allowlist Authorized Scripts	6
16.5	Use Up-to-Date and Trusted Third-Party Software Components	6
GP-TM-08	Any applicable security features should be enabled by default, and any unused or insecure functionalities should be disabled by default.	6
1.4.4	Verify that collected logs do not include sensitive information, such as PII, credentials and cryptographic keys.	5

1.4.5	Verify that collected logs can be securely retrieved from the devices over an online collection, either periodically or on-demand.	4
1.4.7	Verify that the confidentiality, integrity and authenticity of collected logs is appropriately protected, both on the devices that created them and on other systems that store or process them.	4
1.4.1	Verify that devices can collect logs about events with security implications, such as successful and failed authentication attempts, access to debugging functionality etc.	3
1.4.2	Verify that collected logs have the adequate granularity to enable actionable insights and alerts. Logs should include, at a minimum, the type of event, timestamp, source, outcome, and identification of involved actors.	3
1.4.3	Verify that devices contain or are synchronized with a reliable time source, to ensure the validity of log timestamps.	3
1.1	Establish and Maintain Detailed Enterprise Asset Inventory	3
1.2	Address Unauthorised Assets	3
1.3	Utilize an Active Discovery Tool	3
1.4	Use Dynamic Host Configuration Protocol (DHCP) Logging to Update Enterprise Asset Inventory	3
4.6	Securely Manage Enterprise Assets and Software	3
10.1	Deploy and Maintain Anti-Malware Software	3
10.2	Configure Automatic Anti-Malware Signature Updates	3
10.3	Disable Autorun and Autoplay for Removable Media	3
10.4	Configure Automatic Anti-Malware Scanning of Removable Media	3
10.5	Enable Anti-Exploitation Features	3
10.6	Centrally Manage Anti-Malware Software	3
10.7	Use Behavior-Based Anti-Malware Software	3
16.4	Establish and Manage an Inventory of Third-Party Software Components	3

16.6	Establish and Maintain a Severity Rating System and Process for Application Vulnerabilities	3
GP-TM-30	Ensure a context-based security and privacy that reflects different levels of importance.	3
GP-TM-55	Implement a logging system that records events relating to user authentication, management of accounts and access rights, modifications to security rules, and the functioning of the system. Logs must be preserved on durable storage and retrievable via authenticated connections.	3
GP-TM-56	Implement regular monitoring to verify the device behaviour, to detect malware and to discover integrity errors.	3
GP-TM-57	Conduct periodic audits and reviews of security controls to ensure that the controls are effective. Perform penetration tests at least biannually.	3
GP-TM-33	Ensure that devices only feature the essential physical external ports (such as USB) necessary for them to function and that the test/debug modes are secure, so they cannot be used to maliciously access the devices. In general, lock down physical ports to only trusted connections.	1

Appendix F - Script Utilised in Case Scenarios

```
import adafruit_dht
from ISSreamer.Streamer import Streamer
import time
import board

# ----- User Settings -----
SENSOR_LOCATION_NAME = "Pi1"
BUCKET_NAME = "Pi1"
BUCKET_KEY = "Pi1"
ACCESS_KEY = "ist_MATRdteMEvrob5TmjpyZD5OWi5JyJpJL"
MINUTES_BETWEEN_READS = 2
METRIC_UNITS = True
# -----

dhtSensor = adafruit_dht.DHT22(board.D4)
streamer = Streamer(bucket_name=BUCKET_NAME, bucket_key=BUCKET_KEY,
access_key=ACCESS_KEY)

while True:
    try:
        humidity = dhtSensor.humidity
        temp_c = dhtSensor.temperature
    except RuntimeError:
        print("RuntimeError, trying again...")
        continue

    if METRIC_UNITS:
        streamer.log(SENSOR_LOCATION_NAME + " Temperature(C)", temp_c)
    else:
        temp_f = format(temp_c * 9.0 / 5.0 + 32.0, ".2f")
        streamer.log(SENSOR_LOCATION_NAME + " Temperature(F)", temp_f)
    humidity = format(humidity, ".2f")
    streamer.log(SENSOR_LOCATION_NAME + " Humidity(%)", humidity)
    streamer.flush()
    time.sleep(60*MINUTES_BETWEEN_READS)
```

Appendix G - Vulnerability Scan Report for Case Scenario 1

Table G1: Vulnerability Scan Report for Case Scenario 1

Vulnerability Name	CVSS V3 Base Score	CVSS V3 Severity Rating
Debian: Security Advisory for bind9 (DSA-4909-1)	9.8	Critical
Debian: Security Advisory for ffmpeg (DSA-4990-1)	9.8	Critical
Debian: Security Advisory for ffmpeg (DSA-4722-1)	9.8	Critical
Debian: Security Advisory for curl (DSA-4633-1)	9.8	Critical
Debian: Security Advisory for vlc (DSA-4671-1)	9.8	Critical
Debian: Security Advisory for openssl (DSA-4963-1)	9.8	Critical
Debian: Security Advisory for libx11 (DSA-4920-1)	9.8	Critical
Raspberry Pi OS / Raspbian Default Credentials (SSH)	9.8	Critical
Debian Security Advisory DSA 4502-1 (ffmpeg - security update)	9.8	Critical
Debian: Security Advisory for libproxy (DSA-4800-1)	9.8	Critical

Debian: Security Advisory for aspell (DSA-4948-1)	9.1	Critical
Debian: Security Advisory for bluez (DSA-4951-1)	8.6	High
Debian: Security Advisory for bind9 (DSA-4857-1)	8.1	High
Debian: Security Advisory for ntfs-3g (DSA-4971-1)	7.8	High
Sudo Heap-Based Buffer Overflow Vulnerability (Baron Samedit) - Active Check	7.8	High
Debian: Security Advisory for djvulibre (DSA-5032-1)	7.8	High
Debian: Security Advisory for ghostscript (DSA-4748-1)	7.8	High
Debian: Security Advisory for policykit-1 (DSA-5059-1)	7.8	High
Debian: Security Advisory for sudo (DSA-4839-1)	7.8	High
Debian: Security Advisory for vlc (DSA-4834-1)	7.8	High
Debian: Security Advisory for vlc (DSA-4704-1)	7.8	High
Debian: Security Advisory for pygments (DSA-4870-1)	7.5	High
Debian: Security Advisory for git (DSA-	7.5	High

4657-1)		
Debian: Security Advisory for git (DSA-4659-1)	7.5	High
Debian: Security Advisory for openssl (DSA-4661-1)	7.5	High
Debian: Security Advisory for curl (DSA-4881-1)	7.5	High
Debian: Security Advisory for pygments (DSA-4878-1)	7.5	High
Debian: Security Advisory for wpa (DSA-4898-1)	7.5	High
Debian: Security Advisory for bind9 (DSA-4752-1)	7.5	High
Debian: Security Advisory for bind9 (DSA-4689-1)	7.5	High
Debian: Security Advisory for underscore (DSA-4883-1)	7.2	High
Debian: Security Advisory for bluez (DSA-4647-1)	7.1	High
Debian: Security Advisory for openssl (DSA-4875-1)	5.9	Medium
Debian: Security Advisory for openssl (DSA-4807-1)	5.9	Medium
Debian: Security Advisory for apt (DSA-4808-1)	5.7	Medium

Debian: Security Advisory for ghostscript (DSA-5038-1)	5.5	Medium
Debian: Security Advisory for systemd (DSA-4942-1)	5.5	Medium
Debian: Security Advisory for apt (DSA-4685-1)	5.5	Medium
Debian: Security Advisory for openssl (DSA-4855-1)	5.3	Medium
Debian: Security Advisory for bind9 (DSA-4994-1)	5.3	Medium
Debian: Security Advisory for python-apt (DSA-4809-1)	2.8	Low
TCP timestamps	2.6	Low

Appendix H - Vulnerability Scan Report for Case Scenario 2

Evidence of connection with InitialState



Hydra Results prior to hardening

```
(root@kali) ~/home/kali
# hydra -l default -P /usr/share/wordlists/iotpass.txt 192.168.1.170 -t 4 ssh
Hydra v9.2 (c) 2021 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for ill
egal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-03-08 16:49:34
[DATA] max 4 tasks per 1 server, overall 4 tasks, 17 login tries (l:1/p:17), ~5 tries per task
[DATA] attacking ssh://192.168.1.170:22/
[22][ssh] host: 192.168.1.170 login: default password: 12345
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2022-03-08 16:49:46
```

Nmap Scan prior to hardening

```
(root@kali) ~/home/kali
# nmap -sTU -O 192.168.1.170
Starting Nmap 7.92 ( https://nmap.org ) at 2022-03-08 16:49 EST
Stats: 0:01:56 elapsed; 0 hosts completed (1 up), 1 undergoing UDP Scan
UDP Scan Timing: About 13.40% done; ETC: 17:03 (0:12:23 remaining)
Stats: 0:04:53 elapsed; 0 hosts completed (1 up), 1 undergoing UDP Scan
UDP Scan Timing: About 31.00% done; ETC: 17:04 (0:10:52 remaining)
Nmap scan report for pi2.lan (192.168.1.170)
Host is up (0.0010s latency).
Not shown: 998 closed udp ports (port-unreach), 998 closed tcp ports (conn-refused)
PORT      STATE      SERVICE
22/tcp    open      ssh
80/tcp    open      http
68/udp    open|filtered dhcp
5353/udp  open|filtered zeroconf
MAC Address: B8:27:EB:2E:E7:C3 (Raspberry Pi Foundation)
Device type: general purpose
Running: Linux 4.X|5.X
OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5
OS details: Linux 4.15 - 5.6
Network Distance: 1 hop

OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 1030.71 seconds
```

Dictionary Attack post Hardening

```

root@kali:~/home/kali# hydra -l default -P /usr/share/wordlists/iotpass.txt 192.168.1.170 -t 4 ssh
Hydra v9.2 (c) 2021 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these ** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-03-08 17:19:36
[DATA] max 4 tasks per 1 server, overall 4 tasks, 17 login tries (l1:p:17), -5 tries per task
[DATA] attacking ssh://192.168.1.170:22/
1 of 1 target completed, 0 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2022-03-08 17:19:48
    
```

ICMP Flood attack Post Hardening

```

len=46 ip=192.168.1.170 ttl=64 DF id=0 tos=0 iplen=40
sport=80 flags=RA seq=0 win=0 rtt=0.0 ms^C
— 192.168.1.170 hping statistic —
165727299 packets transmitted, 2263642 packets received, 99% packet loss
round-trip min/avg/max = 1.5/17.6/36.1 ms
sport=80 flags=RA seq=0 win=0 rtt=0.0 ms
    
```

Evidence of Vulnerability Assessment Perform by OpenVAS post hardening

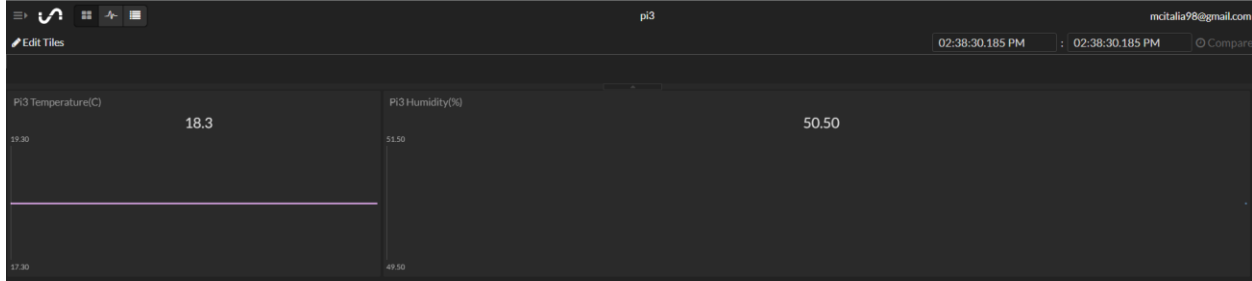
Vulnerability	Severity	QoD	Host		Location	Created
			IP	Name		
Authenticated Scan / LSC Info Consolidation (Linux/Unix SSH Login)	0.0 (Log)	80 %	192.168.1.170	pi2.lan	general/tcp	Sun, Mar 6, 2022 10:32 PM UTC
Avahi Version Detection (Linux)	0.0 (Log)	80 %	192.168.1.170	pi2.lan	general/tcp	Sun, Mar 6, 2022 10:19 PM UTC
CPE Inventory	0.0 (Log)	80 %	192.168.1.170	pi2.lan	general/CPE-T	Sun, Mar 6, 2022 10:32 PM UTC
Detection of Linux Kernel mitigation status for hardware vulnerabilities	0.0 (Log)	80 %	192.168.1.170	pi2.lan	general/tcp	Sun, Mar 6, 2022 10:20 PM UTC
Determine OS and list of installed packages via SSH login	0.0 (Log)	97 %	192.168.1.170	pi2.lan	22/tcp	Sun, Mar 6, 2022 10:18 PM UTC
dmi decode Detection (Linux/Unix SSH Login)	0.0 (Log)	80 %	192.168.1.170	pi2.lan	general/tcp	Sun, Mar 6, 2022 10:21 PM UTC
FFmpeg Version Detection (Linux)	0.0 (Log)	80 %	192.168.1.170	pi2.lan	general/tcp	Sun, Mar 6, 2022 10:18 PM UTC
GNU Bash Version Detection (Linux)	0.0 (Log)	80 %	192.168.1.170	pi2.lan	general/tcp	Sun, Mar 6, 2022 10:21 PM UTC
GZip Version Detection (Linux)	0.0 (Log)	80 %	192.168.1.170	pi2.lan	general/tcp	Sun, Mar 6, 2022 10:19 PM UTC
Hostname Determination Reporting	0.0 (Log)	80 %	192.168.1.170	pi2.lan	general/tcp	Sun, Mar 6, 2022 10:32 PM UTC

Vulnerability	Severity	QoD	Host		Location	Created
			IP	Name		
ICMP Timestamp Detection	0.0 (Log)	80 %	192.168.1.170	pi2.lan	general/icmp	Sun, Mar 6, 2022 10:21 PM UTC
ISC DHCP Client Version Detection	0.0 (Log)	80 %	192.168.1.170	pi2.lan	general/tcp	Sun, Mar 6, 2022 10:18 PM UTC
jQuery Detection Consolidation	0.0 (Log)	80 %	192.168.1.170	pi2.lan	general/tcp	Sun, Mar 6, 2022 10:21 PM UTC
OpenSSH Detection Consolidation	0.0 (Log)	80 %	192.168.1.170	pi2.lan	general/tcp	Sun, Mar 6, 2022 10:18 PM UTC
OpenSSL Detection Consolidation	0.0 (Log)	80 %	192.168.1.170	pi2.lan	general/tcp	Sun, Mar 6, 2022 10:18 PM UTC
OS Detection Consolidation and Reporting	0.0 (Log)	80 %	192.168.1.170	pi2.lan	general/tcp	Sun, Mar 6, 2022 10:18 PM UTC
Perl Detection Consolidation	0.0 (Log)	80 %	192.168.1.170	pi2.lan	general/tcp	Sun, Mar 6, 2022 10:20 PM UTC
Perl Modules Detection (Linux/Unix SSH Login)	0.0 (Log)	80 %	192.168.1.170	pi2.lan	general/tcp	Sun, Mar 6, 2022 10:20 PM UTC
Perl Modules Detection (Linux/Unix SSH Login)	0.0 (Log)	80 %	192.168.1.170	pi2.lan	general/tcp	Sun, Mar 6, 2022 10:20 PM UTC
Report running Linux Kernel	0.0 (Log)	97 %	192.168.1.170	pi2.lan	general/tcp	Sun, Mar 6, 2022 10:21 PM UTC

Vulnerability	Severity	QoD	Host	Name	Location	Created
			IP			
rpcbind Detection (Linux/Unix SSH Login)	0.0 (Log)	80 %	192.168.1.170	pi2.lan	general/tcp	Sun, Mar 6, 2022 10:20 PM UTC
Secure File Transfer Protocol (SFTP) Detection (SSH)	0.0 (Log)	80 %	192.168.1.170	pi2.lan	22/tcp	Sun, Mar 6, 2022 10:17 PM UTC
Services	0.0 (Log)	80 %	192.168.1.170	pi2.lan	22/tcp	Sun, Mar 6, 2022 10:17 PM UTC
SSH Authorization Check	0.0 (Log)	80 %	192.168.1.170	pi2.lan	22/tcp	Sun, Mar 6, 2022 10:17 PM UTC
SSH Login Successful For Authenticated Checks	0.0 (Log)	80 %	192.168.1.170	pi2.lan	22/tcp	Sun, Mar 6, 2022 10:17 PM UTC
SSH Protocol Algorithms Supported	0.0 (Log)	80 %	192.168.1.170	pi2.lan	22/tcp	Sun, Mar 6, 2022 10:17 PM UTC
SSH Protocol Versions Supported	0.0 (Log)	95 %	192.168.1.170	pi2.lan	22/tcp	Sun, Mar 6, 2022 10:17 PM UTC
SSH Server type and version	0.0 (Log)	80 %	192.168.1.170	pi2.lan	22/tcp	Sun, Mar 6, 2022 10:17 PM UTC
sudo / sudoers Detection (Linux/Unix SSH Login)	0.0 (Log)	80 %	192.168.1.170	pi2.lan	general/tcp	Sun, Mar 6, 2022 10:19 PM UTC
Traceroute	0.0 (Log)	80 %	192.168.1.170	pi2.lan	general/tcp	Sun, Mar 6, 2022 10:17 PM UTC

Appendix I - Vulnerability Scan Report for Case Scenario 3

Evidence of Connection with Initial State



Dictionary Attack prior to hardening

```
(root@kali)~# hydra -l default -P /usr/share/wordlists/iotpass.txt 192.168.1.180 -t 4 ssh
Hydra v9.2 (c) 2021 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-03-10 16:34:54
[DATA] max 4 tasks per 1 server, overall 4 tasks, 17 login tries (l:/p:17), ~5 tries per task
[DATA] attacking ssh://192.168.1.180:22/
[22][ssh] host: 192.168.1.180 login: default password: 12345
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2022-03-10 16:35:11
```

Dictionary Attack post hardening

```
(root@kali)~# hydra -l default -P /usr/share/wordlists/iotpass.txt 192.168.1.180 -t 4 ssh
Hydra v9.2 (c) 2021 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-03-10 16:36:06
[DATA] max 4 tasks per 1 server, overall 4 tasks, 17 login tries (l:/p:17), ~5 tries per task
[DATA] attacking ssh://192.168.1.180:22/
1 of 1 target completed, 0 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2022-03-10 16:36:17
```

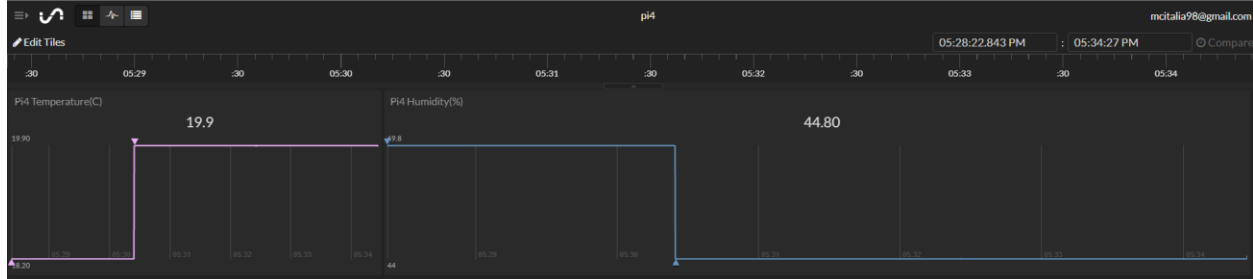
Evidence of Vulnerability Assessment Perform by OpenVAS post hardening

Vulnerability	Severity	QoD	Host IP	Name	Location	Created
Authenticated Scan / LSC Info Consolidation (Linux/Unix SSH Login)	0.0 (Log)	80 %	192.168.1.180	pi3.lan	general/tcp	Mon, Mar 7, 2022 8:55 AM UTC
Avahi Version Detection (Linux)	0.0 (Log)	80 %	192.168.1.180	pi3.lan	general/tcp	Mon, Mar 7, 2022 8:43 AM UTC
CPE Inventory	0.0 (Log)	80 %	192.168.1.180	pi3.lan	general/CPE-T	Mon, Mar 7, 2022 8:55 AM UTC
Detection of Linux Kernel mitigation status for hardware vulnerabilities	0.0 (Log)	80 %	192.168.1.180	pi3.lan	general/tcp	Mon, Mar 7, 2022 8:43 AM UTC
Determine OS and list of installed packages via SSH login	0.0 (Log)	97 %	192.168.1.180	pi3.lan	22/tcp	Mon, Mar 7, 2022 8:41 AM UTC
dmidecode Detection (Linux/Unix SSH Login)	0.0 (Log)	80 %	192.168.1.180	pi3.lan	general/tcp	Mon, Mar 7, 2022 8:44 AM UTC
FFmpeg Version Detection (Linux)	0.0 (Log)	80 %	192.168.1.180	pi3.lan	general/tcp	Mon, Mar 7, 2022 8:42 AM UTC
GNU Bash Version Detection (Linux)	0.0 (Log)	80 %	192.168.1.180	pi3.lan	general/tcp	Mon, Mar 7, 2022 8:44 AM UTC
GZip Version Detection (Linux)	0.0 (Log)	80 %	192.168.1.180	pi3.lan	general/tcp	Mon, Mar 7, 2022 8:43 AM UTC
Hostname Determination Reporting	0.0 (Log)	80 %	192.168.1.180	pi3.lan	general/tcp	Mon, Mar 7, 2022 8:55 AM UTC

Vulnerability	Severity	QoD	Host IP	Name	Location	Created
ICMP Timestamp Detection	0.0 (Log)	80 %	192.168.1.180	pi3.lan	general/icmp	Mon, Mar 7, 2022 8:44 AM UTC
ISC DHCP Client Version Detection	0.0 (Log)	80 %	192.168.1.180	pi3.lan	general/tcp	Mon, Mar 7, 2022 8:42 AM UTC
jQuery Detection Consolidation	0.0 (Log)	80 %	192.168.1.180	pi3.lan	general/tcp	Mon, Mar 7, 2022 8:44 AM UTC
OpenSSH Detection Consolidation	0.0 (Log)	80 %	192.168.1.180	pi3.lan	general/tcp	Mon, Mar 7, 2022 8:42 AM UTC
OpenSSL Detection Consolidation	0.0 (Log)	80 %	192.168.1.180	pi3.lan	general/tcp	Mon, Mar 7, 2022 8:42 AM UTC
OS Detection Consolidation and Reporting	0.0 (Log)	80 %	192.168.1.180	pi3.lan	general/tcp	Mon, Mar 7, 2022 8:42 AM UTC
Perl Detection Consolidation	0.0 (Log)	80 %	192.168.1.180	pi3.lan	general/tcp	Mon, Mar 7, 2022 8:44 AM UTC
Perl Modules Detection (Linux/Unix SSH Login)	0.0 (Log)	80 %	192.168.1.180	pi3.lan	general/tcp	Mon, Mar 7, 2022 8:44 AM UTC
Perl Modules Detection (Linux/Unix SSH Login)	0.0 (Log)	80 %	192.168.1.180	pi3.lan	general/tcp	Mon, Mar 7, 2022 8:44 AM UTC
Report running Linux Kernel	0.0 (Log)	97 %	192.168.1.180	pi3.lan	general/tcp	Mon, Mar 7, 2022 8:44 AM UTC
Vulnerability	Severity	QoD	Host IP	Name	Location	Created
rpcbind Detection (Linux/Unix SSH Login)	0.0 (Log)	80 %	192.168.1.180	pi3.lan	general/tcp	Mon, Mar 7, 2022 8:44 AM UTC
Secure File Transfer Protocol (SFTP) Detection (SSH)	0.0 (Log)	80 %	192.168.1.180	pi3.lan	22/tcp	Mon, Mar 7, 2022 8:41 AM UTC
Services	0.0 (Log)	80 %	192.168.1.180	pi3.lan	22/tcp	Mon, Mar 7, 2022 8:40 AM UTC
SSH Authorization Check	0.0 (Log)	80 %	192.168.1.180	pi3.lan	22/tcp	Mon, Mar 7, 2022 8:40 AM UTC
SSH Login Successful For Authenticated Checks	0.0 (Log)	80 %	192.168.1.180	pi3.lan	22/tcp	Mon, Mar 7, 2022 8:41 AM UTC
SSH Protocol Algorithms Supported	0.0 (Log)	80 %	192.168.1.180	pi3.lan	22/tcp	Mon, Mar 7, 2022 8:40 AM UTC
SSH Protocol Versions Supported	0.0 (Log)	95 %	192.168.1.180	pi3.lan	22/tcp	Mon, Mar 7, 2022 8:40 AM UTC
SSH Server type and version	0.0 (Log)	80 %	192.168.1.180	pi3.lan	22/tcp	Mon, Mar 7, 2022 8:40 AM UTC
sudo / sudoers Detection (Linux/Unix SSH Login)	0.0 (Log)	80 %	192.168.1.180	pi3.lan	general/tcp	Mon, Mar 7, 2022 8:42 AM UTC
Traceroute	0.0 (Log)	80 %	192.168.1.180	pi3.lan	general/tcp	Mon, Mar 7, 2022 8:41 AM UTC

Appendix J - Vulnerability Scan Report for Case Scenario 4

Evidence of Connection with Initial State



Dictionary attack prior to hardening

```
(root@kali)~# hydra -l default -P /usr/share/wordlists/iotpass.txt 192.168.1.190 -t 4 ssh
Hydra v9.2 (c) 2021 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-03-23 12:32:57
[DATA] max 4 tasks per 1 server, overall 4 tasks, 17 login tries (l:1/p:17), ~5 tries per task
[DATA] attacking ssh://192.168.1.190:22/
[22][ssh] host: 192.168.1.190 login: default password: 12345
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2022-03-23 12:33:10
```

Nmap scan prior to hardening

```
(root@kali)~# nmap -sTU -O 192.168.1.190
Starting Nmap 7.92 ( https://nmap.org ) at 2022-03-23 12:35 EDT
Stats: 0:02:27 elapsed; 0 hosts completed (1 up), 1 undergoing UDP Scan
UDP Scan Timing: About 13.78% done; ETC: 12:53 (0:15:20 remaining)
Stats: 0:04:03 elapsed; 0 hosts completed (1 up), 1 undergoing UDP Scan
UDP Scan Timing: About 23.38% done; ETC: 12:52 (0:13:16 remaining)
Stats: 0:06:40 elapsed; 0 hosts completed (1 up), 1 undergoing UDP Scan
UDP Scan Timing: About 39.08% done; ETC: 12:52 (0:10:25 remaining)
Stats: 0:06:42 elapsed; 0 hosts completed (1 up), 1 undergoing UDP Scan
UDP Scan Timing: About 39.28% done; ETC: 12:52 (0:10:23 remaining)
Nmap scan report for pi4.lan (192.168.1.190)
Host is up (0.0013s latency).
Not shown: 998 closed udp ports (port-unreach), 998 closed tcp ports (conn-refused)
PORT      STATE      SERVICE
22/tcp    open      ssh
80/tcp    open      http
68/udp    open|filtered dhcpc
5353/udp  open|filtered zeroconf
MAC Address: B8:27:EB:2E:E7:C3 (Raspberry Pi Foundation)
Device type: general purpose
Running: Linux 4.X|5.X
OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5
OS details: Linux 4.15 - 5.6
Network Distance: 1 hop

OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 1020.85 seconds
```

Evidence of the 'sudo python3 http.server 80' PID

```

root@raspberrypi:/home/pi# ps aux | grep python
root    1084  0.0  1.4 30440 13868 pts/0    S   16:28   0:01 python3 script.py
root    1085  0.2  0.0 10324   420 pts/0    Sl  16:28   0:20 /usr/local/lib/python3.7/dist-packages/adafruit_blin
ka/microcontroller/bcm283x/pulseio/libgpiod_pulsein --pulses 81 --queue 20917 -i gpiochip0 4
root    1116  0.0  0.3  9948  3316 pts/0    S   16:29   0:00 sudo python3 -m http.server 80
root    1121  2.5  1.4 59012 13524 pts/0    S   16:29   3:02 python3 -m http.server 80
root    22557  0.0  1.6 38724 15912 ?        Ssl 18:17   0:00 /usr/bin/python3 /usr/share/unattended-upgrades/unat
tended-upgrade-shutdown --wait-for-signal
root    22947  0.0  0.0  7348   540 pts/0    S+  18:27   0:00 grep python
root@raspberrypi:/home/pi#

```

Evidence that upgrades have been performed

```

root@raspberrypi:/home/pi# apt-get upgrade
Reading package lists... Done
Building dependency tree
Reading state information... Done
Calculating upgrade... Done
The following packages were automatically installed and are no longer required:
  libmicrodns0 python-colorzero rpi-eeeprom-images
Use 'sudo apt autoremove' to remove them.
The following packages have been kept back:
  binutils binutils-arm-linux-gnueabi-hf binutils-common libbinutils python-rpi.gpio python3-rpi.gpio
  raspberrypi-sys-mods raspberrypi-ui-mods
0 upgraded, 0 newly installed, 0 to remove and 8 not upgraded.
root@raspberrypi:/home/pi#

```

Evidence of new user (piuser4) creation

```

root@raspberrypi:/home/pi# useradd piuser4
root@raspberrypi:/home/pi# passwd piuser4
New password:
Retype new password:
passwd: password updated successfully

```

Evidence that both accounts (Pi and Default) were removed from administrators and only piuser4 was set to administrator.

```

# User privilege specification
root    ALL=(ALL:ALL) ALL
piuser4 ALL=(ALL:ALL) ALL

```

ICMP Flood attack evidence

```
└─# hping3 -S --flood -p 22 192.168.1.190 (hydra) starting at 2022-03-23 12:12:12
HPING 192.168.1.190 (eth0 192.168.1.190): S set, 40 headers + 0 data bytes
hping in flood mode, no replies will be shown
^C
— 192.168.1.190 hping statistic —
9796495 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

Appendix K - Vulnerability Scan Report for Case Scenario 5

Evidence of start

```
root@raspberrypi:/home/pi# python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
^Z
[1]+  Stopped                  python3 -m http.server 80
root@raspberrypi:/home/pi# bg
[1]+ python3 -m http.server 80 &
```

Evidence of default user within privileged account

```
# User privilege specification
root    ALL=(ALL:ALL) ALL
default ALL=(ALL:ALL) ALL
# Allow members of group sudo to execute any command
```

Evidence of dictionary attack prior to hardening

```
L# hydra -l default -P /usr/share/wordlists/iotpass.txt 192.168.1.200 -t 4 ssh
Hydra v9.2 (c) 2021 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for ill
egal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2022-03-26 12:02:25
[DATA] max 4 tasks per 1 server, overall 4 tasks, 17 login tries (l:1/p:17), ~5 tries per task
[DATA] attacking ssh://192.168.1.200:22/
[22][ssh] host: 192.168.1.200  login: default  password: 12345
```

Nmap scan prior to hardening

```
(root@kali)-[~/kali]
└─# nmap -sTU -O 192.168.1.200
Starting Nmap 7.92 ( https://nmap.org ) at 2022-03-26 15:57 EDT
Stats: 0:02:25 elapsed; 0 hosts completed (1 up), 1 undergoing UDP Scan
UDP Scan Timing: About 15.68% done; ETC: 16:12 (0:13:00 remaining)
Stats: 0:02:30 elapsed; 0 hosts completed (1 up), 1 undergoing UDP Scan
UDP Scan Timing: About 16.18% done; ETC: 16:12 (0:12:57 remaining)
Stats: 0:03:12 elapsed; 0 hosts completed (1 up), 1 undergoing UDP Scan
UDP Scan Timing: About 20.38% done; ETC: 16:13 (0:12:30 remaining)
Stats: 0:04:06 elapsed; 0 hosts completed (1 up), 1 undergoing UDP Scan
UDP Scan Timing: About 25.78% done; ETC: 16:13 (0:11:48 remaining)
Stats: 0:04:11 elapsed; 0 hosts completed (1 up), 1 undergoing UDP Scan
UDP Scan Timing: About 26.28% done; ETC: 16:13 (0:11:44 remaining)
Stats: 0:05:37 elapsed; 0 hosts completed (1 up), 1 undergoing UDP Scan
UDP Scan Timing: About 34.88% done; ETC: 16:13 (0:10:29 remaining)
Nmap scan report for pi5.lan (192.168.1.200)
Host is up (0.0017s latency).
Not shown: 998 closed udp ports (port-unreach), 998 closed tcp ports (conn-refused)
PORT      STATE      SERVICE
22/tcp    open      ssh
80/tcp    open      http
68/udp    open|filtered dhcpc
5353/udp  open|filtered zeroconf
MAC Address: B8:27:EB:2E:E7:C3 (Raspberry Pi Foundation)
Device type: general purpose
Running: Linux 4.X|5.X
OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5
OS details: Linux 4.15 - 5.6
Network Distance: 1 hop

OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 1020.91 seconds
```

Evidence of the 'sudo python3 http.server 80' PID being disabled

```
root@raspberrypi:/home/pi# ps aux | grep python
root    1033  0.0  1.0 30072 10120 ?        S    19:49   0:01 python3 -m http.server 80
root    22391 3.5  1.4 30444 13728 pts/1    S    20:48   0:00 python3 script.py
root    22392 2.3  0.0 10324   460 pts/1    Sl   20:48   0:00 /usr/local/lib/python3.7/dist-packages/adafruit_blinka/microcontroller/bcm283x/pulseio/libgpiod_pulsein --pulses 81 --queue 27042 -i gpiochip0 4
root    22402 0.0  0.0  7348   532 pts/1    S+   20:48   0:00 grep python
root@raspberrypi:/home/pi# kill 1033
root@raspberrypi:/home/pi# ps aux | grep python
root    22391 0.8  1.4 30444 13728 pts/1    S    20:48   0:00 python3 script.py
root    22392 0.5  0.0 10324   460 pts/1    Sl   20:48   0:00 /usr/local/lib/python3.7/dist-packages/adafruit_blinka/microcontroller/bcm283x/pulseio/libgpiod_pulsein --pulses 81 --queue 27042 -i gpiochip0 4
root    22404 0.0  0.0  7348   532 pts/1    S+   20:49   0:00 grep python
```


Nmap scan post hardening

```

Nmap scan report for pi5.lan (192.168.1.200)
Host is up (0.00071s latency).
Not shown: 999 closed tcp ports (conn-refused), 956 closed udp ports (port-unreach), 44 open|filtered udp ports (no-response)
PORT      STATE SERVICE
22/tcp    open  ssh
MAC Address: B8:27:EB:2E:E7:C3 (Raspberry Pi Foundation)
No exact OS matches for host (If you know what OS is running on it, see https://nmap.org/submit/ ).
TCP/IP fingerprint:
OS:SCAN(V=7.92%E=4%D=3/26%OT=22%CT=1%CU=2%PV=Y%DS=1%DC=D%G=Y%M=B827EB%TM=62
OS:3F8E20%P=x86_64-pc-linux-gnu)SEQ(SP=101%GCD=1%ISR=10A%TI=Z%CI=Z%II=I%TS=
OS:U)OPS(O1=M5B4NNSNW7%O2=M5B4NNSNW7%O3=M5B4NW7%O4=M5B4NNSNW7%O5=M5B4NNSNW7
OS:%O6=M5B4NNS)WIN(W1=FAF0%W2=FAF0%W3=FAF0%W4=FAF0%W5=FAF0%W6=FAF0)ECN(R=Y%
OS:DF=Y%T=40%W=FAF0%M5B4NNSNW7%CC=Y%Q=)T1(R=Y%DF=Y%T=40%S=0%A=S+%F=AS%RD=
OS:0%Q=)T2(R=N)T3(R=N)T4(R=Y%DF=Y%T=40%W=0%S=A%A=Z%F=R%O=%RD=0%Q=)T5(R=Y%DF
OS:=Y%T=40%W=0%S=A+S+%F=AR%O=%RD=0%Q=)T6(R=Y%DF=Y%T=40%W=0%S=A+A=Z%F=R%O=
OS:%RD=0%Q=)T7(R=Y%DF=Y%T=40%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)U1(R=Y%DF=N%T=40%
OS:IPL=164%UN=0%RIPL=G%RID=G%RIPCK=G%RUCK=G%RUD=G)IE(R=Y%DFI=N%T=40%CD=S)

Network Distance: 1 hop

OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 1043.66 seconds
    
```

Evidence of Vulnerability Assessment Perform by OpenVAS post hardening

Vulnerability	Severity	QoD	Host IP	Name	Location	Created
Authenticated Scan / LSC Info Consolidation (Linux/Unix SSH Login)	0.0 (Log)	80 %	192.168.1.200	pi5.lan	general/tcp	Tue, Mar 8, 2022 12:59 AM UTC
Avahi Version Detection (Linux)	0.0 (Log)	80 %	192.168.1.200	pi5.lan	general/tcp	Tue, Mar 8, 2022 12:46 AM UTC
CPE Inventory	0.0 (Log)	80 %	192.168.1.200	pi5.lan	general/CPE-T	Tue, Mar 8, 2022 12:59 AM UTC
Detection of Linux Kernel mitigation status for hardware vulnerabilities	0.0 (Log)	80 %	192.168.1.200	pi5.lan	general/tcp	Tue, Mar 8, 2022 12:47 AM UTC
Determine OS and list of installed packages via SSH login	0.0 (Log)	97 %	192.168.1.200	pi5.lan	22/tcp	Tue, Mar 8, 2022 12:45 AM UTC
dmidecode Detection (Linux/Unix SSH Login)	0.0 (Log)	80 %	192.168.1.200	pi5.lan	general/tcp	Tue, Mar 8, 2022 12:47 AM UTC
Ffmpeg Version Detection (Linux)	0.0 (Log)	80 %	192.168.1.200	pi5.lan	general/tcp	Tue, Mar 8, 2022 12:45 AM UTC
GNU Bash Version Detection (Linux)	0.0 (Log)	80 %	192.168.1.200	pi5.lan	general/tcp	Tue, Mar 8, 2022 12:47 AM UTC
GZip Version Detection (Linux)	0.0 (Log)	80 %	192.168.1.200	pi5.lan	general/tcp	Tue, Mar 8, 2022 12:46 AM UTC
Hostname Determination Reporting	0.0 (Log)	80 %	192.168.1.200	pi5.lan	general/tcp	Tue, Mar 8, 2022 12:59 AM UTC
Vulnerability	Severity	QoD	Host IP	Name	Location	Created
ICMP Timestamp Detection	0.0 (Log)	80 %	192.168.1.200	pi5.lan	general/icmp	Tue, Mar 8, 2022 12:48 AM UTC
ISC DHCP Client Version Detection	0.0 (Log)	80 %	192.168.1.200	pi5.lan	general/tcp	Tue, Mar 8, 2022 12:45 AM UTC
JQuery Detection Consolidation	0.0 (Log)	80 %	192.168.1.200	pi5.lan	general/tcp	Tue, Mar 8, 2022 12:48 AM UTC
OpenSSH Detection Consolidation	0.0 (Log)	80 %	192.168.1.200	pi5.lan	general/tcp	Tue, Mar 8, 2022 12:46 AM UTC
OpenSSL Detection Consolidation	0.0 (Log)	80 %	192.168.1.200	pi5.lan	general/tcp	Tue, Mar 8, 2022 12:46 AM UTC
OS Detection Consolidation and Reporting	0.0 (Log)	80 %	192.168.1.200	pi5.lan	general/tcp	Tue, Mar 8, 2022 12:46 AM UTC
Perl Detection Consolidation	0.0 (Log)	80 %	192.168.1.200	pi5.lan	general/tcp	Tue, Mar 8, 2022 12:47 AM UTC
Perl Modules Detection (Linux/Unix SSH Login)	0.0 (Log)	80 %	192.168.1.200	pi5.lan	general/tcp	Tue, Mar 8, 2022 12:47 AM UTC
Perl Modules Detection (Linux/Unix SSH Login)	0.0 (Log)	80 %	192.168.1.200	pi5.lan	general/tcp	Tue, Mar 8, 2022 12:47 AM UTC
Report running Linux Kernel	0.0 (Log)	97 %	192.168.1.200	pi5.lan	general/tcp	Tue, Mar 8, 2022 12:47 AM UTC

Vulnerability	Severity	QoD	Host IP	Name	Location	Created
rpcbind Detection (Linux/Unix SSH Login)	0.0 (Log)	80 %	192.168.1.200	pi5.lan	general/tcp	Tue, Mar 8, 2022 12:47 AM UTC
Secure File Transfer Protocol (SFTP) Detection (SSH)	0.0 (Log)	80 %	192.168.1.200	pi5.lan	22/tcp	Tue, Mar 8, 2022 12:45 AM UTC
Services	0.0 (Log)	80 %	192.168.1.200	pi5.lan	22/tcp	Tue, Mar 8, 2022 12:44 AM UTC
SSH Authorization Check	0.0 (Log)	80 %	192.168.1.200	pi5.lan	22/tcp	Tue, Mar 8, 2022 12:44 AM UTC
SSH Login Successful For Authenticated Checks	0.0 (Log)	80 %	192.168.1.200	pi5.lan	22/tcp	Tue, Mar 8, 2022 12:44 AM UTC
SSH Protocol Algorithms Supported	0.0 (Log)	80 %	192.168.1.200	pi5.lan	22/tcp	Tue, Mar 8, 2022 12:44 AM UTC
SSH Protocol Versions Supported	0.0 (Log)	95 %	192.168.1.200	pi5.lan	22/tcp	Tue, Mar 8, 2022 12:44 AM UTC
SSH Server type and version	0.0 (Log)	80 %	192.168.1.200	pi5.lan	22/tcp	Tue, Mar 8, 2022 12:44 AM UTC
sudo / sudoers Detection (Linux/Unix SSH Login)	0.0 (Log)	80 %	192.168.1.200	pi5.lan	general/tcp	Tue, Mar 8, 2022 12:46 AM UTC
Traceroute	0.0 (Log)	80 %	192.168.1.200	pi5.lan	general/tcp	Tue, Mar 8, 2022 12:44 AM UTC