



# Master Thesis

A Repository for Testing Compliance to the Internet  
of Things (IoT) Security Standards

Kes Olga Greuter

*March 2024*

Supervisors:  
Dr. M. Daneva  
Dr. D. K. Sarmah  
Dr. F. Bukhsh

## Executive Summary

The rising popularity of IoT devices creates new attack vectors, exposing user data, privacy, and safety to potential threats. To mitigate these risks, manufacturers must prioritize secure development and adhere to established security standards like IEC 62443 [1] and ETSI EN 303 645 [2]. The EN 303 645 and IEC 62443 standards, developed for consumers' and professional IoT devices respectively, provide requirements for the development and maintenance of secure products. The intention of these standards is to protect the confidentiality, integrity, and availability of the device. In addition to securing the IoT ecosystem, standards like IEC 62443 and ETSI EN 303 645 play a crucial role in helping manufacturers navigate the landscape of emerging EU cybersecurity regulations. Manufacturers can mitigate the risks and liabilities from security issues and non-compliance through certification against the established standards. To achieve this, organizations must adequately implement the requirements of IEC 62443 and EN 303 645.

However, achieving compliance with these standards presents challenges. While they outline security objectives, they lack concrete implementation guidance, leading to uncertainties during verification processes. For example, it requires organizations to evaluate and choose an approach for each requirement, without knowing whether the approach is adequately covering the requirement. This is especially problematic when organizations go for certifications, as they do not know whether certification bodies will view their implemented approaches as sufficient for compliance with the requirements. This can lead to uncertainty for organizations seeking certification and potential security risks for end users.

The objective of this thesis is to propose a solution to the challenge of achieving compliance with security standards such as IEC 62443 and EN 303 645. The solution is twofold:

1. **An architecture for a centralized test repository** that provides IoT manufacturers with clear and actionable guidance for compliance testing. This architecture addresses the current limitations experienced by IoT manufacturing companies through several functionalities. It offers (i) broader security standard coverage (e.g., IEC 62443, EN 303 645), (ii) ability to filter candidate testing strategies for interested companies, based on security levels, and (iii) ability to improve IoT testing efficiency. Requirements imposed by the two selected standards are grouped by standard type and clearly linked to test cases, enhancing traceability and visibility. The architecture streamlines non-compliance identification through failed test case analysis and generates multifaceted compliance reports for better insights. Finally, it leverages existing access control for user security.
2. **A test repository** that maps comprehensive test cases to the requirements of EN 303 645, the two modules of IEC 62443, namely IEC 62443-3-3 and IEC 62443-4-2. This test repository empowers manufacturers with clear and actionable guidance for compliance testing. The test cases encompass both generic security principles and application-specific requirements, ensuring a well-rounded assessment of an IoT device's security posture. To guarantee thorough system evaluation, test cases integrate both positive and negative testing paradigms. Positive testing validates expected functionality, while negative testing probes for vulnerabilities through unexpected inputs. Additionally, the repository acknowledges dependencies between certain test cases, reflecting a logical testing sequence for comprehensive evaluation. The developed repository is publicly available on GitHub under Kes-G/Master-Thesis [1] and presented in Appendix B.

An exploratory methodology following a case study approach with a corporation specializing in IoT device and system manufacture was employed for this research. The four phases of the research process ensured its real-world applicability and impact:

1. **Investigation:** This phase examines the cybersecurity standards landscape and the role of IEC 62443 and EN 303 645. It analyzes the content and nature of these standards. Finally, it includes the execution of a literature review.

2. **Design:** This phase evaluates the current system of the case study organization by using interviews and observations. It also constructs the proposed architecture based on the evaluation.
3. **Development:** This phase analyzes the types and techniques used in the test cases that are candidates for inclusion in a test repository. The phase then creates the test repository by mapping test cases to the requirements of the chosen standards.
4. **Integration:** This phase achieves the integration of the test repository within the proposed architecture.

The main strength of the adopted research approach is that it leverages the strengths of multiple qualitative research methods, including interviews, observations, and document analysis. Additionally, expert interviews are conducted to validate the test repository.

This research makes a valuable contribution to both the scientific and business communities. The centralized test repository architecture directly benefits practitioners by providing broader security standard coverage, making compliance reporting and gap identification easier. In addition, it improves information access and generates insightful reports, all while integrating seamlessly with existing tools for streamlined issue tracking and test automation.

From a scientific standpoint, the architecture serves as a standardized model for future research, highlighting the potential of integrated test repositories for a holistic approach. Moreover, the results of this research also contribute to the on-going discussion in the security compliance research community about the cost-effectiveness of security certification approaches. The case study context described in this thesis serves as an example of demonstrating the benefits of a centralized repository in the process of implementing testing strategies as part of assuring security compliance.

The developed test repository empowers practitioners with standardized test cases, reducing risk and improving testing efficiency. Beyond the context of the case study organization, the repository's modular structure allows for adaptation to various organizations, devices, and security standards, for example the inclusion of ISO 27001. New requirements can be easily integrated, while existing tests can be reused, promoting efficiency and reducing development costs. These standardized tests are also valuable for security education, providing concrete examples for teaching best practices.

Finally, the research in this thesis acknowledges the limitations of the proposed solution and proposes directions for future research in order to improve its impact. These lines for future research include the automation of the tests relevant for each standard, extending the range of the standards covered, and the execution of follow-up replication case studies in order to make the findings more generalizable. By addressing these areas, the research has the potential to significantly improve IoT security by providing manufacturers with a valuable framework.

Overall, this research makes a valuable step towards a more secure IoT landscape by providing a comprehensive framework for testing and promoting secure development practices.

## Acknowledgments

I am deeply grateful to my supervisors, Dipti K. Sarmah, Maya Daneva and Faiza Bukhsh, for their unwavering guidance and support during the research process. Their insightful discussions and constructive feedback have been a constant motivation for me to strive for quality and make a valuable contribution to the field. Dipti and Maya also deserve a special mention for giving me the freedom to explore my own topics and take initiatives that aligned with my interests. They encouraged me to follow the path I wanted to take, which allowed me to fully engage with the research and produce work that I am proud of.

I would also like to express my appreciation to the security experts who generously shared their knowledge and expertise with me. Dimitrios Papadopoulos and Barbara Oosterveld provided invaluable insights that greatly enriched the content of this paper, and I am grateful for everything I learned from them. Their contributions have played a crucial role in enhancing the quality and accuracy of this research.

Finally, I would like to express my sincere gratitude to my family and friends for their support throughout this master's program and interest during this thesis. I am particularly grateful to Michalis for his constant encouragement and belief in me. His willingness to take the time to delve deeper into complex topics, especially those related to mathematics, was invaluable during challenging times and helped me stay motivated. Additionally, I am thankful to my best friend, May, for always being by my side and pulling me through challenging aspects of the student life. Furthermore, I want to thank my family who tried their best to support me in materials related to security and programming and showed sincere interest in my student life. My deepest gratitude goes to my grandfather, Bob. Throughout my entire education, he generously opened his home to me, offering not just a place to live but a warm and welcoming environment. His embrace extended to my friends as well, especially for his so called "Bobtails". This living situation proved invaluable in allowing me to focus fully on my studies, and I am incredibly grateful for his generosity and affection. The time spent getting closer to him during these years has been a true source of joy and something I will always treasure. I am truly fortunate to have such a supportive network of people in my life.

# Contents

Executive Summary.....	2
Acknowledgments.....	4
List of Acronyms.....	7
List of Figures .....	8
List of Tables .....	9
1 Introduction .....	10
1.1 Research Context .....	10
1.2 Problem Statement.....	13
1.3 Research Objectives and Research Questions .....	13
1.4 The Case Study.....	14
1.5 Purpose and Industrial Relevance.....	15
1.6 Research Approach .....	17
2 Investigation: Background and Related Work .....	18
2.1 Cybersecurity standards .....	18
2.1.1 IEC 62443: A Framework for IACS Security .....	19
2.1.2 ETSI EN 303 645: Consumer IoT .....	20
2.1.3 Summary of IoT Standards.....	21
2.2 Related Work on the Standards of Interest to this Thesis .....	22
2.2.1 Research on IEC 62443.....	22
2.2.2 Research on EN 303 645 .....	23
2.2.3 Research on the relation between IEC 62443 and EN 303 645 .....	23
2.2.4 Research Gap .....	23
2.2.5 Summary of Related Work.....	23
2.3 Literature review.....	24
2.3.1 Deliverables.....	24
2.3.2 Summary of the Literature Review .....	28
3 Research Methodology .....	29
3.1 Research Framework .....	29
3.2 Research Design.....	31
3.2.1 Research Process for the Design of the Architecture .....	32
3.2.2 Research Process for the Development of Test Repository .....	35
3.2.3 Research Approach for Integrating the Test Repository .....	42
3.3 Participants in the Expert-based Validation of the Proposed Mappings.....	42
3.3.1 Summary .....	43
4 Design of the Architecture .....	44

4.1	Description of the Current System.....	44
4.2	Observations of the Current System and its Limitations .....	45
4.3	Interview Findings regarding the Limitations of the Current System .....	47
4.4	Proposal for a New Architecture.....	47
4.4.1	Attributes: Information Fields and Presentation .....	48
4.4.2	Attributes: Platform, Configurations, and Access Control .....	49
4.4.3	Summary .....	51
5	Development of the Test Repository.....	52
5.1	Structure of the Test Repository.....	52
5.2	The Foundation for the Test Cases .....	52
5.3	Test Cases Techniques and Dependencies .....	53
5.4	Distribution of the Test Cases .....	54
5.5	Integration of the Test Repository into the Architecture .....	54
5.6	Summary .....	54
6	Discussion.....	55
6.1	Discussion on the Architecture .....	55
6.1.1	Implications for Practice .....	56
6.1.2	Implications for Research.....	56
6.2	Discussion of the Test Repository .....	57
6.2.1	Implications for Practice .....	57
6.2.2	Implications for Research and Education .....	58
6.3	Discussion of the Repository Integration.....	58
6.4	Summary .....	59
7	Conclusion, Limitations, and Future Work.....	60
7.1	Conclusion.....	60
7.2	Limitations.....	60
7.3	Future Work.....	61
8	References .....	63
	Appendix A.....	71
	Appendix B.....	75

## List of Acronyms

<b>Acronym</b>	<b>Meaning</b>
IoT	Internet of Things
NIS	Network and Information Systems
CRA	Cyber Resilience Act
RED	Radio Equipment Directive
OWASP	Open Web Application Security Project
CSCIoT	Cyber Security for Consumer Internet of Things
IACS	Industrial Automation and Control Systems
FDA	Food and Drug Administration
BSI	Bundesamt für Sicherheit in der Informationstechnik
SoGP	Standard of Good Practice
SL	Security Level
IIoT	Industrial Internet of Things
RBAC	Role-Based Access Control

**Table 1:** Acronyms and their meanings.

## List of Figures

<b>Figure 1:</b> Research approach. ....	17
<b>Figure 2:</b> Cyber Security Standards [39, 41]. ....	18
<b>Figure 3:</b> Modules of IEC 62443 [6].....	19
<b>Figure 4:</b> Thesis' research approach adapted from Saunders et al. [38]. ....	29
<b>Figure 5:</b> The flowchart for answering for RQ1. ....	33
<b>Figure 6:</b> Flowchart for RQ2 for IEC 62443-4-2.....	36
<b>Figure 7:</b> Flowchart for RQ2 for IEC 62443-3-3.....	40
<b>Figure 8:</b> Flowchart for RQ2 for EN 303 645.....	41
<b>Figure 9:</b> Construction of test sets, test cases, steps, and the links between test sets and requirements in the current system of IoTCorp. ....	45
<b>Figure 10:</b> Requirements and Test Cases of IoT Security Standards. ....	54



## List of Tables

<b>Table 1:</b> Acronyms and their meanings. ....	7
<b>Table 2:</b> <i>Example of a test set of IoTCorp including test cases and steps.</i> .....	15
<b>Table 3:</b> <i>Expected benefits of this research for various stakeholder groups.</i> .....	16
<b>Table 4:</b> Security Levels of IEC 62443 [6]. ....	20
<b>Table 5:</b> The main thirteen guidelines of EN 303 645.....	21
<b>Table 6:</b> Security Testing Methods per SDLC phase. ....	25
<b>Table 7:</b> Legend for figures used in Chapter 3.2. ....	31
<b>Table 8:</b> Attributes and their description. ....	33
<b>Table 9:</b> The principles of Norman that impact the attribute selection.....	34
<b>Table 10:</b> Roles of the interviewed engineers. ....	34
<b>Table 11:</b> Example of test set with linked requirements and test cases. ....	38
<b>Table 12:</b> An example of a mapping for requirements includes in the test set of Table 10. ....	38
<b>Table 13:</b> Example of a Test Set in the database of IoTCorp. ....	44
<b>Table 14:</b> Example of a Test Case in the database of IoTCorp.....	45
<b>Table 15:</b> Attributes (detailed in table 8) of the current system of IoTCorp. ....	46
<b>Table 16:</b> Attributes for the new repository architecture. ....	48
<b>Table 17:</b> Example of a requirement in the test repository. ....	48
<b>Table 18:</b> Example of a test case in test repository and its corresponding information.....	49
<b>Table 19:</b> An example of a failed test case. ....	50
<b>Table 20:</b> Example of a test coverage of a product. ....	50
<b>Table 21:</b> Example of an overview of test coverage per requirement. ....	50
<b>Table 22:</b> Columns of the EN 303 645 mapping in the repository.....	52
<b>Table 23:</b> Columns of IEC 62443-4-2 mapping in the repository.....	52
<b>Table 24:</b> Security testing methods and the SDLC phase they belong to, along with their description and examples of requirements that they test. ....	71
<b>Table 25:</b> Test cases mapped to requirements of IEC 62443-4-2.....	75
<b>Table 26:</b> Mapping of test cases to requirements of IEC 62443-3-3. ....	118
<b>Table 27:</b> Mapping of test cases to requirements of EN 303 645 [7]. ....	138

# 1 Introduction

The widespread adoption of Internet of Things (IoT) devices brings numerous benefits in terms of convenience and efficiency [3]. However, the rapid growth of IoT applications simultaneously resulted in security vulnerabilities that expose users to potential threats of various degrees of severity. The increasing number of IoT devices introduces new attack vectors that can compromise user data, privacy, and even physical safety [4, 5]. In response to this, manufacturers need to prioritize secure development and compliance with established security standards such as IEC 62443 [6] and ETSI EN 303 645 [7] to mitigate these risks and potential legal repercussions. However, compliance with these IoT security standards presents challenges as they lack implementation guidance.

This thesis explores the challenges faced by IoT manufacturers when implementing the requirements for IEC 62443 and ETSI EN 303 645 and proposes a solution in the form of an architecture of a test repository with comprehensive test cases and the implementation of this architecture in the context of a real-world organization. Using a case study approach [8, 9], this repository is specifically designed to evaluate compliance with the requirements outlined in both standards. The proposed practical tool is meant to empower manufacturers to streamline their testing processes, navigate the evolving compliance landscape, and ultimately deliver secure IoT products.

In the rest of this chapter, we first introduce the research context, which provides definitions of terms concerning IoT devices, the associated risks, security standards, and the certification process in Chapter 1.1. We then present the problem statement in Chapter 1.2, and outline our research objectives and questions in Chapter 1.3. Next, we provide an overview of the case study used in this thesis in Chapter 1.4. In Chapter 1.5, we discuss the purpose and industrial relevance of this thesis, and finally in Chapter 1.6. we provide the structure of the thesis.

## 1.1 Research Context

Internet of Things (IoT) devices are Internet-connected devices, such as, smart light bulbs [10], Bluetooth-connected toothbrushes, and mobile phones. These devices are now widely adopted across diverse sectors, including homes, offices, transportation, healthcare, telecommunication, and agriculture [11]. The popularity of IoT devices is continuously increasing. International Data Corporation's market forecast [3] predicts that there will be 55.7 billion connected devices worldwide by 2025 and 75 percent of those will be connected to an IoT platform. This growth is usually explained with the potential of IoT devices to bring benefits across strategical, tactical, and operational levels. By harnessing data-driven insights, IoT becomes instrumental to enhance efficiency to both consumers and businesses [4].

However, despite their benefits, security remains a critical concern for IoT devices. The technology powering the devices often falls short of ensuring truly secure communication and device protection [11]. In what follows, we explain this context in more detail.

The core function of IoT devices is to collect and transmit sensor data. Cameras, microphones, and other sensors are embedded within these devices [12]. They act as digital eyes and ears, capturing information from the physical world to facilitate remote monitoring and control [4]. Their dependence on sensor data, coupled with the lack of robust security measures, creates vulnerabilities stemming from a combination of factors, including:

- **Inadequately securitized software:** This factor is often traceable to bugs, weak encryption, and lack of secure coding practices [5]. This aligns with the Open Web Application Security Project's (OWASP) identification of insecure software as one of the most important technical challenges [13].
- **Insufficient hardware security elements:** This could involve outdated device firmware, weak processors, and a lack of physical security measures like tamper detection [5, 14].
- **Security creation faults** [5]: This encompasses a wide range of issues, including the use of default passwords, insecure communication protocols, and poorly configured devices.

Vulnerabilities traceable to the types of factors presented above may have severe consequences for IoT users, as illustrated by the following incident involving the Ring indoor camera, a popular smart

home security device [15]. Due to a security flaw, hackers were able to exploit a vulnerability to gain access to the camera's live feed and stored video footage. As a result, the privacy and security of users were compromised as hackers were able to remotely observe and even interact with individuals within their homes. This incident highlights the potential for insecure IoT devices to leave users vulnerable to data breaches, compromised device functionality, and disrupted services [16].

The potential impact of security issues extends well beyond individual privacy concerns and poses a threat to various stakeholders. Consumers face a multitude of risks, including unauthorized data breaches exposing sensitive information such as personal preferences and financial details [4, 5]. Compromised device functionality poses further threats, as hackers could manipulate smart devices as cameras and thermostats, which may result in physical harm [17]. Businesses face reputational damage and financial losses from data breaches and service disruptions [4, 18]. Furthermore, insufficient security measures may result in non-compliance with the applicable legal requirements on IoT device security, which may lead to regulatory fines [19].

Operating within the dynamic environment of IoT device and system development, IoT manufacturers face the challenge of protecting their brand image from the damaging effects of cyber-attacks. Implementing robust security measures shields their brand image from such effects. However, platforms as social media, instant news outlets, and open review sites can amplify security vulnerabilities into widespread reputational damage [18]. In an effort to mitigate the risks posed by insecure devices, manufacturers of IoT devices and systems strive to adhere to the applicable IoT security standards. Such standards provide guidance and good practices for IoT security architecture and design. Based on these standards, cybersecurity certification processes carried out by dedicated certification bodies usually include verifying the device's security features against cybersecurity best practices for storing consumers' information, password and security management standards, and over-the-air mechanisms for software updates.

While in the recent past, demonstrating compliance of IoT products was desirable, the year of 2024 marked a turning point for the IoT marketplace by putting it de-facto in a state of transition towards nearly mandatory certification of IoT devices. This is because the IoT regulatory environment matured to a point that in 2024 many countries introduced — or are in the process of introducing — IoT regulations to govern various aspects of IoT deployment, from data creation to infrastructure and business operations. For example, starting 2024, the European Commission has imposed minimum requirements for the security of IoT products, which in turn means that products failing to meet these standards would be banned from the European Union market. To IoT manufacturers, this change means that in order to keep their access in the IoT market, they have to undergo a lengthy and expensive certification process to assure security compliance of their produced IoT devices to internationally recognized standards.

This thesis focuses on the two key standards for IoT security devices and systems: the **IEC 62443** [6] for industrial systems and the **ETSI EN 303 645** [7] for consumer IoT. While they both share the common goal of promoting secure development throughout the device lifecycle, they cater to distinct markets. Specifically, the IEC 62443 [6] standard safeguards Industrial Automation and Control Systems (IACS). Its purpose is to provide information and requirements to manufacture, install, and operate IoT devices securely. This standard offers a modular and flexible framework, addressing current and future security vulnerabilities in professional systems [1]. The standard is split into modules that each cover a different aspect of security on the process, people, and technology within the lifecycle of an IoT device. For a more in-depth analysis of the IEC 62443 standard and its modular structure, we refer the readers to Chapter 2.1.1. Next, the ETSI EN 303 645 [7] standard, also named the “*International Cyber Security for Consumer Internet of Things (CSCIoT)*”, tackles requirements for the secure development of IoT devices according to data protection rights. This standard is specifically focused on securing the consumer IoT device throughout its lifecycle.

It is worthwhile noting that in addition to securing the IoT ecosystem, the two chosen standards for this master thesis (IEC 62443 and ETSI EN 303 645) play a crucial role in helping manufacturers navigate the landscape of emerging EU cybersecurity regulations. As already mentioned earlier, many

EU countries are in the process of introducing regulations that will address different areas of IoT product security, motivating manufacturers to proactively adhere to established standards. Four key upcoming regulations, namely the NIS2 Directive [20], Cybersecurity Act [21], Cyber Resilience Act (CRA) [22], and Radio Equipment Directive (RED) [23], significantly impact the security posture of IoT devices and their manufacturers. Each have a different impact on the IoT ecosystem and relation to IEC 62443 and EN 303 645. Below we explain these as follows:

- **NIS2 Directive (Oct 2024)** [20]: This directive expands on the previous Network and Information Systems (NIS) Directive [24]. The NIS2 broadens the scope of NIS to reach beyond critical infrastructure sectors and include additional sectors, such as waste management and public administration [20]. Many of these new sectors rely heavily on IoT devices. The directive specifically adds cybersecurity risk-management measures and reporting obligations for security incidents. Notably, the IEC 62443 standard offers guidance in addressing key NIS2 requirements like risk management, policies, security measures, and incident response mechanisms [25]. While EN 303 645 is limited to consumer IoT devices, it can still contribute to specific aspects like data protection, vulnerability management, and software updates [7], which are also covered in NIS2 requirements [20].
- **Cybersecurity Act (2019)** [21]: This act fosters EU-wide cooperation and incident response. It establishes a network of national cybersecurity authorities and mandates incident reporting obligations. While it does not mandate specific standards, aligning with best practices as IEC 62443 and EN 303 645 can contribute to overall security, which better prepares organizations for incident handling.
- **Cyber Resilience Act (proposed)** [22]: This act aims to mandate essential security requirements for the product, vulnerability handling, information to the user, and technical documentation. In case the act mandates specific security requirements, modules of IEC 62443 such as IEC 62443-4-2 and IEC 62443-4-1 could be recognized as compliant standards for meeting those requirements [26]. Depending on the final scope and product categories of the CRA, certain parts of EN 303 645 could be used for compliance, especially for consumer IoT products [27].
- **Radio Equipment Directive (August 2025)** [23]: The Radio Equipment Directive (RED) builds upon its previous version RED 2014/53/EU [28] to incorporate mandatory cybersecurity requirements. This version is taking effect from August 1<sup>st</sup>, 2025. Both the IEC 62443 and EN 303 645 have been mapped to the essential requirements of the RED [29].

These upcoming regulations, coupled with increasing security threats, further emphasize the importance of adhering to established standards like IEC 62443 and EN 303 645. IoT manufacturers can mitigate the risks and liabilities from security issues and non-compliance through certification against the established standards. This certification demonstrates the organization's commitment to rigorous testing of their products and services and that they meet the required safety or performance standards [30]. To achieve this, organizations must adequately implement the requirements of IEC 62443 and EN 303 645 [31]. After completing product development in accordance with a chosen standard, organizations initiate the certification process by assembling and submitting required documentation for assessment. Independent experts from an external company then evaluate the product's adherence to the standard's requirements. Ultimately, if the assessment confirms the product's compliance, a certificate or Statement of Conformity is issued. While passing the initial assessment confirms a product's compliance, it does not automatically grant certification. Only Notified Bodies, designated by the European Union, can issue official certifications [32, 33]. The path of certification includes inherent uncertainties regarding final costs due to factors like device type and testing methods [32]. Adding to the complexity, identifying the specific testing requirements demands deep understanding of the product's architecture and ecosystem. Moreover, the separate certification fees are additional to the final assessment cost. However, organizations can reduce the costs of assessment and certification with proactive compliance efforts. Regularly conducting internal security tests of IoT devices during development, such as the reviewing of logs during Unit and System Testing,

can contribute to this goal [34]. This proactive approach strengthens the device's security posture by uncovering and addressing security issues early and streamlines the organization's testing practices, potentially optimizing the overall cost.

## 1.2 Problem Statement

While the importance of securing IoT devices and systems through established standards such as IEC 62443 and EN 303 645 is acknowledged by manufacturers, achieving compliance remains a challenge due to the lack of implementation guidance with these standards. The standards define the “*what*” of secure development but leave the “*how*” ambiguous [31]. This creates knowledge gaps and uncertainties for organizations in regard to two key areas:

1. **Verification of Compliance:** Determining whether the implemented security measures truly fulfil the standards' requirements remains unclear. This lack of clarity may lead to confusion, delays, and inefficiencies in the compliance process.
2. **Selection of the Optimal Approach:** The requirements allow for multiple approaches for compliance to a requirement. As each approach has distinct strengths and weaknesses, this can add to the complexity of the requirement implementation. For example, virus scanners [35] and intrusion detection systems [36] are two different approaches that can be used for compliance with a requirement on malicious code protection mechanisms; however, each of them provides a different level of security. The virus scanner may be more efficient in detecting known threats, while the intrusion detection system may be more efficient in detecting unknown threats [35, 36]. While both of these approaches provide malicious code protection mechanisms, each one covers different aspects of the requirement. This in itself creates complexity for organizations when choosing the optimal approach out of a number of candidate approaches available to them. More often than not, the complexity in requirement implementation demands organizations to evaluate and choose an approach without knowing for sure whether the certification body will see it as sufficient for the requirement and accept it. This lack of knowledge can make it challenging for organizations to implement the security approach in question in the way that will match the requirements of IEC 62443 [6] and EN 303 645 [7].

These knowledge gaps ultimately hinder the manufacturers' efforts to develop secure and compliant IoT devices, leaving them and their users exposed to security vulnerabilities. Understanding the available implementation options in terms of extents to which they meet the requirements imposed by the applicable standards, will therefore be of help to IoT manufacturers preparing for security certification of their IoT devices. This present thesis is a step towards this.

## 1.3 Research Objectives and Research Questions

As already indicated, the rise of IoT devices increased the need for robust security measures and adherence to industry standards such as EN 303 645 and IEC 62443. However, the lack of concrete guidance on compliance testing hinders organizations seeking to secure and certify their products. This thesis addresses this critical gap by proposing:

1. **An environment for centralized testing:** This centralized approach streamlines the testing process and facilitates collaboration.
2. **A comprehensive test repository:** This repository maps test cases to the requirements of to the requirements of EN 303 645, the two modules of IEC 62443, namely IEC62443-3-3, and IEC 62443-4-2 (detailed in Chapter 2.1.1). The test repository is meant to provide manufacturers with clear and actionable guidance for compliance testing.

The developed repository, which is publicly available on GitHub under Kes-G/Master-Thesis [37] and presented in Appendix B, showcases its readiness and suitability for real-world application in the field of IoT security through a case study involving a corporation specializing in IoT device and system manufacture. This ensures the repository's practical relevance and applicability within the industry. More information about the case study is presented in Chapter 1.4, which outlines the specific context

of the case study organization in implementing security testing for compliance with the standards. Three research questions guide this study:

- **Research Question 1 (RQ1):** What are the key functionalities and architecture of a centralized test repository that effectively supports compliance testing of IoT devices, considering the specific needs identified in the case study?
- **Research Question 2 (RQ2):** Which test cases can be developed or adapted to ensure compliance with each requirement within EN 303 645, IEC 62443-3-3, and IEC 62443-4-2?
- **Research Question 3 (RQ3):** In what way can the developed testcases be integrated with the centralized test repository in RQ1?

This research directly addresses the challenge of inefficient and inconsistent testing practices, which hinder manufacturers' ability to achieve compliance with critical IoT security standards like EN 303 645 and IEC 62443. By developing a centralized test repository, this thesis aims to empower organizations with robust and efficient compliance testing practices.

#### 1.4 The Case Study

The underlying research process for carrying out the work in this master thesis is informed and inspired by a case study research methodology [8]. This research takes place in the context of a multinational corporation, anonymized here as "IoTCorp," which manufactures and develops IoT devices and systems. This organization provided the case study in which we investigate the challenges faced by organizations like IoTCorp when implementing security testing for compliance with standards such as IEC 62443 and EN 303 645. The paragraphs below provide detailed information about the context as follows.

As part of their efforts to improve compliance testing, they developed a system which we analyze and build upon in this thesis. To ensure the security of their products, IoTCorp adheres to industry best practices, including striving for compliance with two relevant standards, namely ETSI EN 303 645 for consumer products and IEC 62443 for professional systems. IoTCorp's diverse organizational structure consists of multiple departments responsible for developing and maintaining both consumer and professional systems. Each department houses various teams specializing in different aspects of the security of the device and system lifecycle, such as development, maintenance, and security testing. However, security testing remains decentralized, requiring each team to manually assess their products for compliance. This current decentralized approach presents several drawbacks:

1. **Redundant Work:** Each team within IoTCorp conducts security assessments for their respective products independently, resulting in redundant work. This inefficiency arises from teams performing similar tasks without coordinated efforts, leading to wasted resources and potentially inconsistent outcomes.
2. **Inconsistencies and Lack of Coherence:** The absence of centralized coordination between teams sometimes leads to diverse testing methodologies being employed across different departments. This lack of coherence can compromise the clarity and effectiveness of the security testing process. Also, the inconsistent use of methodologies may introduce inconsistencies in the evaluation of security vulnerabilities, potentially leading to missed vulnerabilities or false positives.
3. **Increased Risk of Errors and Oversights:** Adopting testing approaches without a centralized overview and coordinated communication, may increase the risk of errors and oversights. This could occur due to incomplete testing coverage, inadequate knowledge sharing, or failure to identify potential vulnerabilities due to inconsistencies in testing procedures. Such oversights would ultimately jeopardize product security and expose vulnerabilities to potential attackers.

IoTCorp's system for security testing includes a database of test sets mapped to specific requirements within the IEC 62443-4-2 standard (detailed in Chapter 2.1.1). A test set consists of multiple test cases designed to evaluate specific security requirements based on compliance standards. Each test case outlines a series of steps to be performed for verifying conformity with a particular aspect of a security requirement or control.

Table 2 presents an example of a test set designed to evaluate the principle of least privilege. This set includes various test cases. For illustration purposes, we have chosen one that focuses on enumeration. This specific test case involves three steps to achieve process and service enumeration, as shown in the rightmost column of Table 2.

Test Set	Test Case(s)	Step(s)
Protection – Least Privilege.	Enumerate all processes or services running in the device or system from root shell.	<ol style="list-style-type: none"> <li>1. Authenticate to the command line.</li> <li>2. Enter an elevated command prompt through <i>sudo su</i>.</li> <li>3. Type in <i>ps -aux</i> from the elevated command prompt.</li> </ol>

**Table 2:** Example of a test set of IoT Corp including test cases and steps.

The database with test sets suffers from several limitations, which is discussed in detail in Chapter 4.3. These limitations hinder its effectiveness in addressing the broader challenges of decentralized testing and. Recognizing the inherent challenges associated with decentralized security testing within the organization, IoT Corp sought to establish a centralized test repository as a solution, which is designed in this thesis and presented in Chapter 4.4. This repository aims to facilitate knowledge sharing, enhance coherence, and promote clarity in the security testing process.

#### 1.5 Purpose and Industrial Relevance

As stated earlier, this thesis aims to address the knowledge gap in testing compliance with the security requirements of EN 303 645, IEC 62443-3-3, and IEC 62443-4-2 for IoT devices and systems. To bridge this gap, the research proposes the design of a centralized test repository and the development of that repository in which test cases are mapped to the requirements of these standards. This repository aims to provide clear and actionable guidance for organizations seeking to effectively assess the security of their IoT devices and achieve compliance with industry standards. The test repository is intended to bring advantages for various stakeholders within the IoT ecosystem:

- **Security Professionals** benefit from standardized test cases, improving accuracy and efficiency while facilitating knowledge sharing.
- **Security Strategists** gain comprehensive coverage of potential vulnerabilities, enabling informed decision-making regarding resource allocation and risk management.
- **Financial Managers** experience improved resource allocation and budgeting through reduced redundancy in testing efforts.
- **Organizations as a whole** achieve enhanced testing consistency, compliance, and security posture.
- **Consumers** benefit from increased security and reduced vulnerability in tested IoT devices and systems, protecting their sensitive information.

Table 3 on the following page further details the specific expected benefits associated with each stakeholder group, highlighting the multifaceted value proposition of the proposed centralized test repository. We can conclude that the expected benefits listed in Table 3 are diverse and cater to the various stakeholder groups. This table demonstrates a comprehensive understanding of the needs and priorities of different individuals involved in security and compliance. The benefits are not limited to immediate gains, but also offer a roadmap for future testing and improvement.

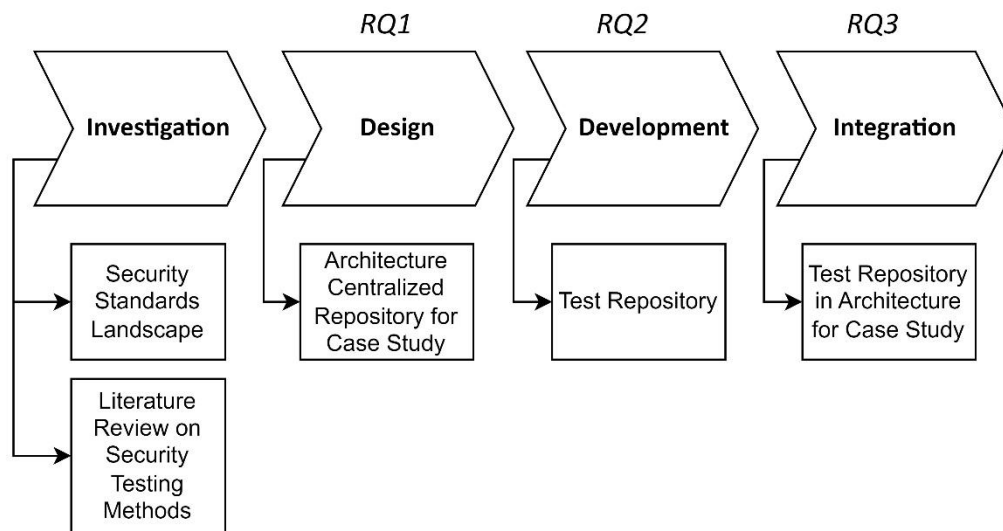
<b>Stakeholder Group</b>	<b>Benefit</b>	<b>Explanation</b>
<i>Security Professionals</i>	Standardized test cases	A consistent set of test cases eliminates the need for individually creating cases, minimizing errors and inconsistencies.
	Improved accuracy	Clear instructions and reporting guidelines enhance the accuracy and reliability of security assessments.
	Knowledge sharing	The repository enables knowledge sharing among professionals, promoting collaboration and sharing of best practices.
<i>Security Strategists</i>	Comprehensive coverage	Standardized test cases ensure thorough assessment of potential vulnerabilities across various organizational assets.
	Informed decision-making	Insights from the repository empower informed decision-making regarding resource allocation, prioritization of vulnerabilities, and implementation of mitigation strategies.
	Compliance assurance	The repository facilitates adherence to industry standards and regulations, reducing compliance risks.
	Effective risk management	The repository provides valuable data for developing and implementing effective risk management strategies to proactively address potential cyber threats.
<i>Financial Managers</i>	Efficient resource allocation	The centralized repository eliminates the need for redundant test case development, optimizing resource allocation and reducing costs associated with security testing.
	Improved budgeting	The repository provides transparency into testing expenses, enabling accurate budgeting for security and compliance initiatives.
	Reduced financial impact	Proactive identification and remediation of vulnerabilities helps minimize the potential financial impact of security incidents.
<i>Organization as a whole</i>	Enhanced consistency and efficiency	Standardized test cases and centralized management promote consistent and efficient testing practices across the organization, saving time and resources.
	Strengthened compliance posture	The repository ensures adherence to relevant standards and regulations, mitigating compliance risks and fostering trust with stakeholders.
	Elevated security posture	Systematic identification and remediation of vulnerabilities contribute to a more robust security posture, protecting sensitive data and critical infrastructure.
	Reduced costs and enhanced competitiveness	Improved efficiency and reduced redundancy in testing efforts lead to cost savings. Additionally, delivering secure and reliable IoT products enhances competitiveness.
	Roadmap for future testing	The centralized repository serves as a valuable resource for continuous improvement in security testing practices, enabling adaptation to evolving threats and industry standards.
<i>Consumers</i>	Increased security and reduced vulnerability	Devices and systems tested using the repository undergo rigorous procedures, reducing the likelihood of exploitable vulnerabilities.
	Protection of confidentiality and integrity	This reduces the risk of data breaches and protects the confidentiality and integrity of sensitive consumer information.

**Table 3:** *Expected benefits of this research for various stakeholder groups.*



## 1.6 Research Approach

To address the research questions, the research approach of this thesis consists of four phases: (1) Investigation, (2) Design, (3) Development, and (4) Implementation. This approach is depicted in Figure 1, which includes the building blocks of the phases and the research questions they address.



**Figure 1:** Research approach.

Chapter 2 covers the Investigation phase of this research, which aims to build the foundational knowledge for this thesis. We begin by exploring the current cybersecurity standards landscape, and specifically focus on the relation of IEC 62443 and EN 303 645 to this ecosystem. We then further analyze the content and nature of these two standards. Furthermore, we introduce the results of a literature review conducted as part of the UT Research Topic Assignment to inform the research presented in this thesis. We consider this review as part of the Investigation phase (see the left side of Figure 1). This chapter also discussed related work to identify the gap in research that this thesis addresses.

Chapter 3 covers the research methodology used in this thesis during the Design, Development, and Integration phases. This thesis is an exploratory research that is structured following the ‘research onion’ model of Saunders et. al. [38]. We employ a multifaceted approach in which interviews, document analysis, and observations inform the research phases. Additionally, Experts interviews are conducted to validate the findings of the Development phase.

Chapter 4 covers the Design phase, in which we address RQ1 and propose a new architecture for the current system of IoT Corp. This system is evaluated using semi-structured interviews and direct observations. The proposed architecture is constructed upon the findings of those interviews and observations.

Chapter 5 covers the Development phase, in which we address RQ2 and RQ3. We first present the created test repository (RQ2) that maps test cases to the requirements of IEC 62443 and EN 303 645. We elaborate on the types and techniques of the test cases. Additionally, in this chapter we cover the Integration phase, in which we elaborate on the integration of the test repository in the architecture proposed in the Design phase (RQ3).

Chapter 6 discusses the answers to the three RQs of this thesis. For each RQ, the chapter presents the key findings and explores their implications for both practice and research, fostering a deeper understanding of the research contributions.

Chapter 7 covers the conclusion, in which we acknowledge the limitations inherent to the chosen approach and explores promising avenues for future work that can extend upon the established foundation.

## 2 Investigation: Background and Related Work

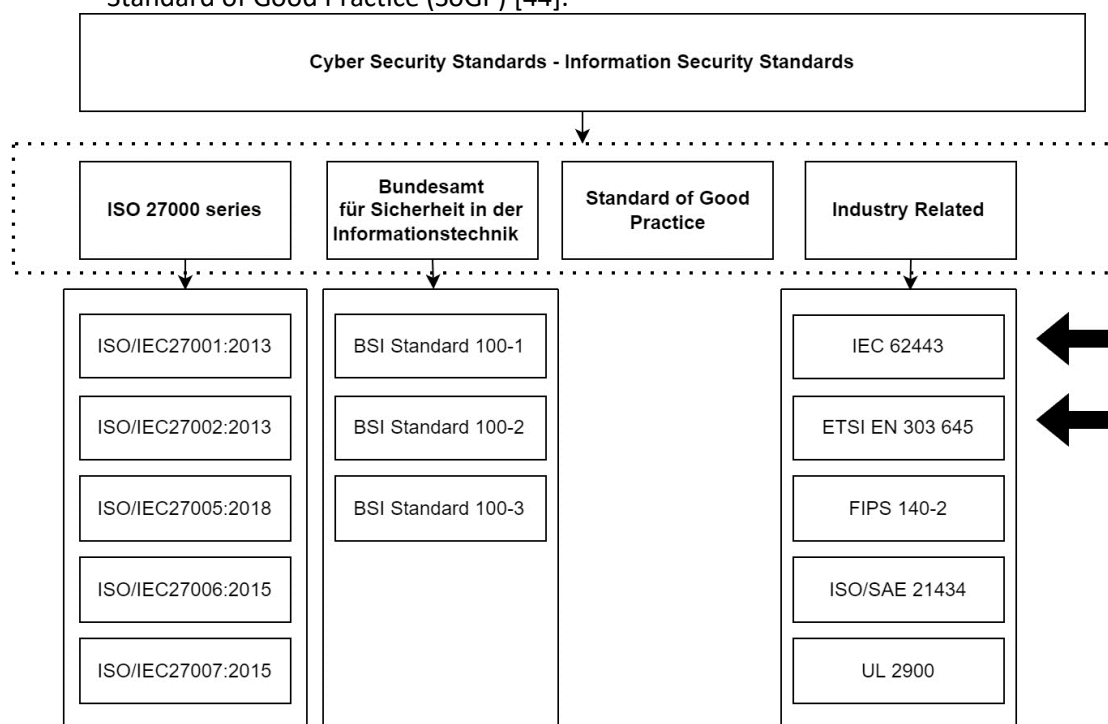
This chapter first examines the landscape of cybersecurity standards in Chapter 2.1. Furthermore, Chapter 2.2 describes related work and previous research on IEC 62443 and EN 303 645. Finally, Chapter 2.3 presents a summarized literature review, which provides insights used in the development of the test repository.

### 2.1 Cybersecurity standards

This chapter summarizes the most commonly used cybersecurity standards (Chapter 2.1) and zooms in on the purpose and content of IEC 62443 [6] (Chapter 2.1.1) and EN 303 645 [7] (Chapter 2.1.2). We provide a summary in Chapter 2.1.3.

This thesis focuses on security standards for IoT devices, specifically IEC 62443 and ETSI EN 303 645. Figure 2 below provides a broader landscape of cybersecurity and information security standards, categorized into:

- **The popular and frequently used standards** as indicated by Taherdoost [39];
- **The emerging standard UL 2900** [40], as this standard has gained popularity since its official recognition by the USA's Food and Drug Administration (FDA) [41];
- **The general standards for cyber security and information security**, which are the ISO 27000 series [42], Bundesamt für Sicherheit in der Informationstechnik (BSI) [43], and Standard of Good Practice (SoGP) [44].



**Figure 2: Cyber Security Standards [39, 41].**

Within Figure 2, IEC 62443 [6] and ETSI EN 303 645 [7] represent industry-related standards specifically targeting IoT security. These are highlighted by arrows on the right of Figure 2. While other valuable standards exist, they fall outside this thesis' scope due to their focused areas:

- **ISO/SAE 21434** [45] is focused on cybersecurity risk management requirements in the engineering of electronic systems of road vehicles.
- **FIPS 140-2** [46] includes hardware and software requirements to protect cryptography modules.
- **UL 2900** is focused on connected components of healthcare systems such as medical devices [40].

Choosing IEC 62443 and EN 303 645 aligns with the thesis objective of exploring security test cases for IoT devices.

### 2.1.1 IEC 62443: A Framework for IACS Security

IEC 62443 [6] is an internationally recognized family of standards that provides a complete framework for assessing various actors as manufacturers, asset owners, and system integrators, in the field of IACS [47]. For manufacturers, compliance with IEC 62443 [6] demonstrates the quality of the security of their systems and components. Furthermore, for asset owners and system integrators, compliance with the procedures described in the standards helps in improving the brand image and minimizing the risk of security breaches [47]. Figure 3 shows the standard decomposed into thirteen modules, 62243-1-1 up to 62443-4-2, with four categories [48]:

- **General (modules 62243-1):** provides a basic understanding of IACS security, including important concepts and terminology.
- **Policies and Procedures (modules 62443-2):** guides the creation and maintenance of a comprehensive cybersecurity management system.
- **System (modules 62443-3):** delves into the technical requirements for secure system design, development, and integration.
- **Component (modules 62443-4):** provides specific technical guidelines for secure development of IACS components [49].

Out of all modules depicted in Figure 3, IEC 62443-2-1, IEC 62443-2-4, 62443-3-3, 62443-4-1, and 62443-4-2 specifically mention requirements, and are marked with ‘X’. Other modules of IEC 62443 do not mention requirements and contain just informative text.

General	Policies & Procedures	System	Component
<p>ISA-62443-1-1</p> <p>Concepts and models</p>	<p><b>X</b> ISA-62443-2-1</p> <p>Requirements for an IACS security management system</p>	<p>ISA-TR62443-3-1</p> <p>Security technologies for IACS</p>	<p><b>X</b> ISA-62443-4-1</p> <p>Product development requirements</p>
<p>ISA-TR62443-1-2</p> <p>Master glossary of terms and abbreviations</p>	<p>ISA-TR62443-2-2</p> <p>Implementation guidance for an IACS security management system</p>	<p>ISA-62443-3-2</p> <p>Security risk assessment and system design</p>	<p><b>X</b> ISA-62443-4-2</p> <p>Technical security requirements for IACS components</p>
<p>ISA-62443-1-3</p> <p>System security conformance metrics</p>	<p>ISA-TR62443-2-3</p> <p>Patch management in the IACS environment</p>	<p><b>X</b> ISA-62443-3-3</p> <p>System security requirements and security levels</p>	
<p>ISA-TR62443-1-4</p> <p>IACS security life-cycle and use-cases</p>	<p><b>X</b> ISA-62443-2-4</p> <p>Requirements for the IACS solution suppliers</p>		

**Figure 3: Modules of IEC 62443 [6].**

The IEC 62443 standard connects Security Levels (SL), shown in Table 4, to the requirements. The levels, ranging from SL1 to SL4, indicate the types of security measures needed to meet the corresponding requirements [50]. This approach ensures that security measures are proportionate to the risks of the device’s function, sensitivity of the data it handles, and the assumed nature of an attack [49].

Level	Description
SL1	Protection against casual or coincidental violation.
SL2	Protection against intentional violation using simple means with low resources, generic skills, and low motivation.
SL3	Protection against intentional violation using sophisticated means with moderate resources, system-specific skills, and moderate motivation.
SL4	Protection against intentional violation using sophisticated means with extended resources, system-specific skills, and high motivation.

**Table 4:** Security Levels of IEC 62443 [6].

For example, a connected device processing confidential governmental information is assigned SL4 and has more strict security requirements compared to a simple sensor in a non-critical environment (SL1). Consequently, higher security levels may need a broader range of security measures. For instance, where all security levels (SL1 to SL4) mandate user authentication, higher levels (SL3 or SL4) might need hardware-based security measures like tamper-resistant modules or certified cryptographic chips [50]. By understanding the designated security level for their product, organizations can filter the applicable requirements and corresponding test cases within the test repository. This targeted filtering process, discussed in detail in Chapter 4.4, streamlines the assessment and certification process for IACS devices, ensuring they meet the necessary security benchmarks.

#### 2.1.2 ETSI EN 303 645: Consumer IoT

ETSI EN 303 645 is the first globally applicable cybersecurity standard for consumer IoT devices [2]. The standard outlines requirements for protection against the most common cybersecurity threats and the prevention of attacks against IoT consumer devices. It encompasses smart devices, sensors, and control components within the consumer IoT landscape [51]. ETSI EN 303 645 is structured into 13 security requirement clusters, which are listed in Table 5 on the following page.

Requirement Cluster	Guideline	Requirement Cluster	Guideline
RC1	No universal default passwords.	RC8	Ensure that personal data is secure.
RC2	Implement a means to manage reports of vulnerabilities.	RC9	Make systems resilient to outages.
RC3	Keep software updated.	RC10	Examine system telemetry data.
RC4	Securely store sensitive security parameters.	RC11	Make it easy for users to delete user data.
RC5	Communicate securely.	RC12	Make installation and maintenance of devices easy.
RC6	Minimize exposed attack surfaces.	RC13	Validate input data.
RC7	Ensure software integrity.		

**Table 5:** The main thirteen guidelines of EN 303 645.

Under these 13 clusters, EN 303 645 includes 33 mandatory provisions and 35 recommendations that consumer IoT devices must adhere to achieve compliance [2]. The provisions and recommendations are not included in Table 5 but are included in the test repository available at GitHub under Kes-G/Master-Thesis [37] and presented in Appendix B.

### 2.1.3 Summary of IoT Standards

This chapter explores the crucial role of ETSI EN 303 645 and IEC 62443 in the landscape of cybersecurity standards for IoT devices. It further analyzes how these standards distinguish themselves from others in terms of their focus and requirements. ETSI EN 303 645 stands as the first globally recognized standard specifically addressing consumer IoT security. It outlines comprehensive requirements to safeguard against common cyber threats and prevent attacks on these devices. IEC

62443, on the other hand, caters to a broader industrial context, encompassing various actors involved in IACS. This comprehensive framework categorizes security requirements based on diverse aspects like system design, development, and components. Notably, the standard also incorporates Security Levels, dynamically tailoring requirements based on potential risks. In comparison to other industry-specific standards focusing on areas like vehicle cybersecurity, cryptography module protection, or healthcare systems, both ETSI EN 303 645 and IEC 62443 offer a dedicated focus on securing the realm of IoT devices.

## 2.2 Related Work on the Standards of Interest to this Thesis

This chapter reviews existing research on the two key cybersecurity standards relevant to this thesis: IEC 62443 (Chapter 2.2.1) and EN 303 645 (Chapter 2.2.2). Additionally, we explore how these standards relate to each other (Chapter 2.2.3) and the gap of research we address (Chapter 2.2.4). We finish this chapter with a summary (Chapter 2.2.5).

### 2.2.1 Research on IEC 62443

A substantial body of published research has investigated various aspects of IEC 62443, encompassing its benefits and challenges, implementation strategies, and testing methodologies:

- **Benefits and Challenges:** A study by Steward [52] emphasizes the advantage of IEC 62443 compliance for organizations, such as enabling them to prevent security vulnerabilities at the source. Moreover, the research efforts of Leander et al. [53] delve deeper, exploring the standard's alignment with the unique challenges of securing Industrial Internet of Things (IIoT) systems. Their work identifies potential roadblocks faced by process owners, such as outdated legacy systems and varying security level requirements and proposes a roadmap to navigate these hurdles. Additionally, Hassani et al. [54] leverage IEC 62443 to establish a risk assessment approach for IIoT objects, facilitating validation and corrective measures before integration into industrial systems.
- **Implementation Strategies:** Research efforts have explored various strategies for effectively implementing IEC 62443. Astorga et al. [55] present an overview of security measures and recommendations for securing IIoT based the IEC 62443's requirements. Shabaan et al. [56] introduce a novel concept in which they leverage the standard's security levels to define secure zones and communication conduits with IIoT systems. A zone is a group of cyber assets with the same cybersecurity requirements and a conduit is a group of cyber assets dedicated exclusively to communications. Components within a conduit share the same cybersecurity requirements as the zone it connects to [57]. Their work also led to the development of a dedicated tool to facilitate the definition and management of these zones and conduits within complex IIoT architectures. Furthermore, Fockel et al. [58] showcase a practical example of integrating a standard-compliant threat analysis process into the development workflow of an industrial control systems manufacturer. Their work demonstrates how to seamlessly integrate such analysis into existing practices and tools, ensuring that security considerations are embedded throughout the development lifecycle. Furthermore, Shabaan et al. [59] contribute a comprehensive threat database specifically designed for IoT application domains. This database is mapped to relevant security requirements within the IEC 62443 framework.
- **Testing Methodologies:** Despite various IEC 62443 implementation strategies, comprehensive testing methodologies remain scarce. Currently, only a limited set of algorithms [60, 61] available on GitHub offer automated testing capabilities for specific system and component requirements. While the first algorithm [60] focuses on verifying properties of a cloud-connected vehicle system, and the second on testing security functionalities within Linux kernel modules [61], they only address only a limited subset of the extensive requirements outlined in IEC 62443, which encompasses a broader range of security aspects beyond specific domains or the kernel level. Additionally, the complexity of IACS environments and the subjective nature of certain requirements often need manual testing and expert judgment for comprehensive compliance verification [1].

### 2.2.2 *Research on EN 303 645*

Research on EN 303 645 has played a critical role in clarifying the fragmented landscape of cybersecurity certification frameworks for IoT devices. As such, Puys et al. [62] includes EN 303 645 in their mapping of common frameworks within the context of the European Cybersecurity Act. Their analysis compared various aspects like target audience, structure, and support offered by different frameworks. Additionally, Jaskolka et al. [63] provide a comparative evaluation of representative examples, including EN 303 645, focusing on factors such as target audience and document organization. Similarly, work by Catal et al. [64] compare frameworks based on their orientation and focus on design/test quality. Furthermore, work by Fischer [65] compared the scope and general content IoT security standards, among which EN 303 645. Finally, Langkemper et al. [66] mapped requirements based on their topic to requirements of several IoT documents stating requirements, including EN 303 645.

### 2.2.3 *Research on the relation between IEC 62443 and EN 303 645*

Several studies have investigated the relationship between the cybersecurity requirements of IEC 62443 and EN 303 645. Greuter et al. [49] conducted a comparative analysis of EN 303 645 against IEC 62443 and other relevant IoT security documents. Their work identified deficiencies in EN 303 645's requirements, particularly regarding essential elements deemed necessary for robust consumer IoT security. This finding suggests that EN 303 645, while offering a baseline, might not be sufficient on its own to ensure comprehensive protection for consumer IoT devices. Similarly, Djebbar et. al. [67] further explored the relationship between these standards by conducting a comparative analysis of IEC 62443, EN 303 645, and ISO 27001. Their study revealed significant overlaps in security requirements, with EN 303 645 largely encompassing those outlined in IEC 62443-3-3. However, they also identified gaps attributable to the specific scope of EN 303 645, which is primarily focused on consumer IoT devices.

### 2.2.4 *Research Gap*

While research on IEC 62443 and EN 303 645 has provided valuable insights into their benefits, challenges, and implementation strategies, a critical gap remains in the area of comprehensive compliance testing methodologies. Existing research primarily focuses on other aspects of the standards, with limited exploration of approaches that enable thorough testing of compliance of the security requirements outlined within these standards. For example, while efforts like [60, 61], demonstrate the development of testing libraries for specific functionalities within the standards, they primarily address limited subsets of requirements and often lack the comprehensiveness necessary for ensuring compliance. Additionally, the complexity of IACS environments and the subjective nature of certain requirements often demand manual testing and expert judgment, highlighting the need for more standardized and automated testing solutions. This thesis directly addresses this critical gap by introducing a novel test repository encompassing test cases for key requirements within EN 303 645, IEC 62443-3-3, and IEC 62443-4-2. By facilitating robust compliance testing, this work empowers organizations to proactively identify and address vulnerabilities, significantly reducing the risk of cyberattacks and data breaches, thereby strengthening the overall security landscape for industrial and consumer IoT ecosystems.

### 2.2.5 *Summary of Related Work*

This chapter positions the two key cybersecurity standards relevant to this thesis, IEC 62443 and EN 303 645, against the landscape of existing standards applicable to the IoT marketplace. The chapter also provides a summary of existing research on these two standards. Previously published work [68, 69, 70] explored the benefits and challenges of IEC 62443 compliance for organizations, particularly in preventing security vulnerabilities and aligning with IIoT security challenges. Additionally, research

investigates implementation strategies such as security measures, risk assessment approaches, and zone/conduit definitions for secure IIoT systems [70, 71, 72]. However, existing testing methodologies remain limited, with only a few algorithms on platforms such as GitHub addressing specific requirements, highlighting the need for more comprehensive solutions [73, 74]. Research on EN 303 645 focused so far on comparisons and evaluations of representative examples of IoT security guidance and standards based on various factors [75, 76, 77, 78, 79].

Furthermore, previously published work analyzed the relation between EN 303 645's security requirements compared to other relevant standards, including comparisons with IEC 62443 [80, 81]. While existing research has explored several aspects of IEC 62443 and EN 303 645, a critical gap remains in the area of compliance testing methodologies for both IEC 62443 and EN 303 645, which is crucial for ensuring the security of IoT devices. As we will see in the next chapters, this thesis addresses the gap by providing a test repository with test cases for conducting compliance testing to these standards, contributing to improving the security of industrial systems and IoT devices.

## 2.3 Literature review

As stated in the beginning of Chapter 2, this thesis was informed by a literature, which was performed as a pre-step to our research. In this chapter we present each of the deliverables in the sub-sections of Chapter 2.3.1. We close this chapter with a summary (Chapter 2.3.2).

The goal of this literature review was to develop a clear understanding of the published security testing methods and to identify the most effective security testing methods for organizations in need to evaluate compliance with requirements of IoT security standards, specifically EN 303 645 and IEC 62443. Eventually, the review was the initial exploration conducted as part of the Final Year Project's Research Topics component. In the following sections, we include a summary of the findings of this literature review which pertain to the development of the repository of test cases that was created in this master thesis.

### 2.3.1 Deliverables

As our literature review focused on the identification of the most effective security testing methods for organizations from the perspective of achieving compliance with the requirements of the IoT standards, specifically EN 303 645 [7] and IEC 62443 [47], it brought six deliverables that are of relevance to the master thesis. These are explained further in this chapter:

- **Unified Set of Security Testing Methods:** A comprehensive list of security testing methods categorized by the Software Development Lifecycle (SDLC) phases (Chapter 2.3.1.1).
- **Mapping of Methods to Requirements:** A detailed mapping of the identified security testing methods against the requirements of EN 303 645 and IEC 62443 (Chapter 2.3.1.2).
- **Analysis of Most Contributing Methods:** An evaluation of the security testing methods with the highest impact on achieving compliance with the targeted standards (Chapter 2.3.1.3).
- **Distribution of Requirements across SDLC:** An analysis of how the requirements of the standards are distributed across the different phases of the SDLC (Chapter 2.3.1.4).
- **Recommendations for Prioritization:** Practical guidance for organizations on prioritizing security testing methods to ensure efficient compliance with the standards (Chapter 2.3.1.5).
- **Least Contributing Methods:** Identification of the security testing methods with the least impact on achieving compliance (Chapter 2.3.1.6).

#### 2.3.1.1 Unified Set of Security Testing Methods

The literature review identified security testing methods drawing insights from resources such as the OWASP testing guide [13] and the survey by Felderer et. al. [82]. These methods are categorized according to the five phases of the SDLC: Analysis, Design, Development, Deployment, and Maintenance [13]. Integrating security testing throughout the SDLC ensures the incorporation of



security testing throughout the product’s lifecycle. Some testing methods span multiple phases, being Analysis/Design, Development/Deployment, and Deployment/Maintenance, which reflects their broader applicability. Table 6 provides a categorized list of the identified security testing methods for each SDLC phase.

SDLC Phase	Security Testing Methods
Analysis	SDLC Process Review and Policy and Standards Review.
Analysis/Design	Model-Based Security Testing Approach for Web Applications.
Design	Security Requirements Review, Design and Architecture Review, UML Models Review, and Threat Models Review.
Development	Code Review, Static Application Security Testing, and Code Walkthrough.
Development/Deployment	Unit and System Testing.
Deployment	Penetration Testing, Configuration Management Review, and Acceptance Test.
Deployment/Maintenance	Vulnerability Scanning, Dynamic Taint Analysis, and Fuzzing.
Maintenance	Change Verification, Health Checks, Operational Management Review, and Regression Testing.

**Table 6:** Security Testing Methods per SDLC phase.

Detailed descriptions of the methods listed in Table 6 are included in Table 24 of Appendix A. This table also provides examples of those requirements which the testing methods may assess. The organization of security testing methods on SDLC phases contributes to the analysis of the distribution of requirements across the SDLC described in Chapter 2.3.1.4.

### 2.3.1.2 Mapping of Methods to Requirements

The literature review establishes a comprehensive mapping between the identified security testing methods (refer to Table 6) and the requirements of the IoT security standards IEC 62443 and EN 303 645. This mapping is available on GitHub under Kes-G/Master-Thesis [83] and presented in Appendix B, and helps organizations to identify relevant testing methods and prioritize testing efforts. The mapping process involved assessing each standard's requirements against every listed security testing method. This evaluation determined whether each method would effectively test for adherence to the specific requirement. For example, consider a requirement that mandates secure development processes for device software. The mapping process would assess whether any listed testing methods, such as *SDLC Process Review*, can effectively evaluate adherence to this requirement. Table 24 of Appendix A provides further examples of how specific requirements align with suitable testing methods.

This mapping serves as the foundation for further analysis in subsequent chapters. Chapter 2.3.1.3 identifies the security testing methods with the most significant contribution to achieving compliance. Furthermore, Chapter 2.3.1.4 analyzes the distribution of requirements across the different phases of the SDLC.

### 2.3.1.3 Analysis of Most Contributing Methods

Building upon the established mapping (Chapter 2.3.1.2), this chapter analyzes the security testing methods with the most significant contribution to achieving compliance for each standard: IEC 62443-2-1, IEC 62443-2-4, IEC 62443-3-3, IEC 62443-4-1, IEC 62443-4-2, and EN 303 645. The contribution percentage for each method represents the proportion of standard requirements it can effectively test for compliance. This value is determined through the comprehensive mapping process. The key findings of this analysis are as follows:

- **EN 303 645:** Design and Architecture Review and code walkthrough methods contribute most significantly, while process-oriented methods like SDLC and operational management reviews have lower impact.
- **IEC 62443-2-1:** Policy and standards review and operational management reviews hold the highest and second-highest contribution percentages, respectively.
- **IEC 62443-2-4:** Policy and standards review takes the top spot, followed by methods focusing on process-oriented requirements and system setup.
- **IEC 62443-3-3:** Methods targeting the system's architecture and code have the highest contribution, while those focusing on process or maintenance aspects have the least.
- **IEC 62443-4-1:** Similar to IEC 62443-2-4, policy and standards review holds the top position, followed by methods centered on architecture and code.

Overall, this analysis emphasizes the importance of employing a diverse range of security testing methods for comprehensive compliance with IoT security standards. Organizations should prioritize methods with the highest contribution percentages based on their specific needs and the standards they aim to comply with.

#### *2.3.1.4 Distribution of Requirements across SDLC*

This chapter analyzes how the requirements of various IoT security standards are distributed across the different phases of the SDLC. This analysis provides insights into which phases each standard emphasizes for security testing. The key findings are as follows:

- **EN 303 645:** This standard primarily focuses on requirements during Development, Deployment, and Design phases, with minimal emphasis on Maintenance.
- **IEC 62443-2-1:** This standard prioritizes requirements for the Analysis phase, followed by Design and Maintenance.
- **IEC 62443-2-4:** This standard exhibits a relatively even distribution across all phases, with slightly less emphasis on Maintenance.
- **IEC 62443-3-3:** This standard heavily emphasizes the Development phase, with minimal focus on Maintenance.
- **IEC 62443-4-1:** Due to its focus on lifecycle requirements, this standard exhibits a uniform distribution across all phases.
- **IEC 62443-4-2:** This standard primarily focuses on Development and Design phases, with minimal emphasis on Maintenance.

This analysis highlights the importance of considering the SDLC phase emphasis of each targeted standard when designing and implementing security testing strategies. By aligning testing efforts with the phases most emphasized by each standard, organizations can ensure comprehensive coverage and effective compliance with IoT security requirements.

#### *2.3.1.5 Recommendations for Prioritization*

The literature review emphasizes the importance of combining multiple security testing methods to achieve comprehensive compliance with IoT security standards. This approach ensures organizations assess both the availability and quality of the required security controls. The recommended methods consider the perspective adopted in each standard, the level of depth, and the contribution to testing compliance, and are as follows:

- **EN 303 645:** Design and Architecture Review, Unit and System Testing, Code Review, and Penetration Testing.
- **IEC 62443-2-1:** Policy and Standards Review, Design and Architecture Review, Operational Management Review, and UML Models Review.
- **IEC 62443-2-4:** Policy and Standards Review, Design and Architecture Review, UML Models Review, and Configuration Management Review.

- **IEC 62443-3-3:** Unit and System Testing, Design and Architecture Review, Code Review, and Penetration Testing.
- **IEC 62443-4-1:** Policy and Standards Review, Operational Management Review, Design and Architecture Review, and UML Models Review are recommended.
- **IEC 62443-4-2:** Unit and System Testing, Design and Architecture Review, Code Review, and Configuration Management Review.

The recommendations can help organizations prioritize their security testing efforts and ensure comprehensive testing for IoT security standards.

### 2.3.1.6 *Least Contributing Methods*

The literature review also identified security testing methods with minimal impact on achieving compliance for each IoT security standard. These findings are based on the comprehensive mapping analysis presented in Chapter 2.3.1.2. The least contributing security testing methods are as follows:

- **EN 303 645:** Regression Testing, Fuzzing, Acceptance Testing, Static Application Security Testing, and Model-Based Security Testing Approach for Web Applications.
- **IEC 62443-2-1:** Regression Testing, Operational Management Review, Health Checks, SDLC Process Review, and Vulnerability Scanning.
- **IEC 62443-2-4:** Regression Test, Operational Management Review, Acceptance Test, Security Requirements Review, and Operational Management Review.
- **IEC 62443-3-3:** Regression Test, Fuzzing, Operational Management Review, and Acceptance Test.
- **IEC 62443-4-1:** Regression Test, Change Verification, Vulnerability Scanning, Model-Based Security Testing Approach for Web Applications, and Operational Management Review.

While these methods may have a lower overall contribution, organizations should carefully evaluate their specific context and needs before completely disregarding them. In certain situations, these methods might still hold value depending on the unique security posture and risk profile of the organization's IoT system.

### 2.3.1.7 *Insights of the Literature Review and their Use in this Thesis*

The comprehensive analysis of security testing methods and their alignment with IoT security standards in the literature review yielded valuable insights that were instrumental in achieving the research objectives (Chapter 1.3) of this thesis. These insights primarily focus on the EN 303 645, IEC 62443-3-3, and IEC 62443-4-2 standards, which are the specific targets for test case mapping in this research (RQ2). Test cases are specific procedures that demonstrate how a security testing method is applied to evaluate a system's compliance with a requirement. For example, a test case for "Fuzzing" might involve designing input data that targets a specific function within the code and verifies the expected behavior. The key findings of the review and their applications are as follows:

- **Focus on Design and Development:** The literature review revealed that the most impactful security testing methods for the targeted standards emphasize the device's fundamental design and development phases (Chapter 2.3.1.4). This finding directly informs the selection of test cases during the mapping process (RQ2). Test cases prioritize methods like Design and Architecture Review, Unit and System Testing, and Code Review, which effectively evaluate these crucial phases for compliance with the respective standards (See Table 6).
- **Policy and Standards Review:** While consistently ranked high across all standards in Chapter 2.3.1.3, Policy and Standards Review alone is insufficient for comprehensive testing. This insight from the literature review highlights the importance of combining this method with others during test case mapping. The selected test cases ensure a balanced approach, incorporating Policy and Standards Review alongside methods targeting specific technical aspects of security, for example fuzzing.
- **Specificity and Usage of Methods:** The literature review revealed that methods with lower contribution scores tend to target specific details and are employed less frequently (Section

2.3.1.6). This finding informs the evaluation of existing test cases during the mapping process (RQ2). While not entirely disregarded, methods such as Regression Testing are carefully assessed for their suitability in addressing specific requirements alongside more impactful methods. However, methods with higher contribution scores can be leveraged more extensively during mapping. For instance, when evaluating the sufficiency of regression test cases for compliance with an IEC 62443-3-3 requirement, the literature review suggests that Regression Testing is among the least recommended methods. Consequently, test cases employing recommended methods (Chapter 2.3.1.5), such as Unit and System Testing, are prioritized to ensure comprehensive evaluation.

The insights from the literature review, which we presented in the above bulleted list, guided the selection and evaluation of test cases during the mapping process (RQ2). By carefully considering the identified trends, trade-offs, and recommended methods, this research went further to establish a comprehensive and effective mapping of test cases to the requirements of EN 303 645, IEC 62443-3-3, and IEC 62443-4-2 standards.

### *2.3.2 Summary of the Literature Review*

As already stated, as part of the Investigation phase of the research in this thesis, a literature review was carried out. It identified various security testing methods for IoT devices, categorized by their applicability throughout the SDLC (Table 6). A mapping (available on GitHub under Kes-G/Master-Thesis [83] and presented in Appendix B) was established between these methods and the requirements of the EN 303 645 and IEC 62443 standards, revealing the most impactful methods for achieving compliance. The review also analyzed the distribution of requirements across the SDLC and provided recommendations for prioritizing testing methods. These insights inform the selection and evaluation of test cases during the mapping process, ensuring the development of a comprehensive approach to evaluating compliance with critical IoT security standards.

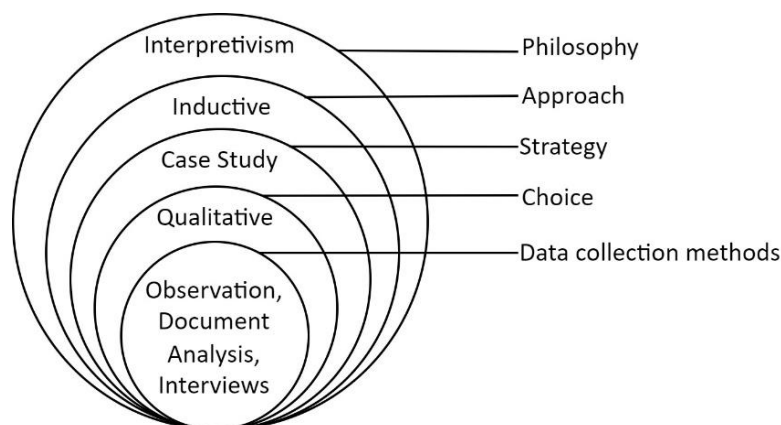
### 3 Research Methodology

This chapter provides a thorough overview of the methodological decisions made to address the research questions. We begin by introducing the research framework (Chapter 3.1) and then elaborate on our research philosophy, approach, strategy, choice, and data collection methods. Next, we outline our research approach (Chapter 3.2) for the Design (Chapter 3.2.1), Development (Chapter 3.2.2), and Integration (Chapter 3.2.3) phases, which outline the steps taken to address RQ1, RQ2, and RQ3 respectively.

#### 3.1 Research Framework

This thesis uses the ‘research onion’ model of the Saunders et al. [38] to guide its planning and execution. The strength of this model lies in its systematic process, offering a step-by-step approach that encourages researchers to consider various aspects of their study, ultimately fostering a comprehensive and well-rounded design. Alternative frameworks, such as Creswell's Framework [84], offer valuable insights into research design. Creswell particularly emphasizes the intricate relationship between research questions, approaches, and designs. In the same vein, the Design Science Research paradigm, for example the design science methodology of R. Wieringa [85] could be a viable alternative for planning and executing our research. Wieringa’s textbook specifically focuses on the relationship between stakeholders’ goals, designs, and evaluation criteria that researchers may choose for the designs. However, for the specific research of this thesis, we preferred the research onion model of Saunders et al. [38] because of its distinct advantages over other frameworks. These are the following:

- **Layered Structure:** The research onion [38] provides a clear and logical path through the research design process. Each layer, encompassing philosophy, approach, strategy, choice, and data collection methods, guides researchers through crucial stages, ensuring all aspects are addressed for a comprehensive design [38]. This structured approach is particularly beneficial for this study as it ensures a systematic and thorough exploration of the research questions (Chapter 1.3).
- **Adaptability:** The framework's adaptability allows researchers to tailor it to their specific research questions and methodologies. Regardless of the chosen question or methodology, the research onion's layers can be effectively applied. This adaptability is crucial in this study as it allows the framework to accommodate the unique research questions and methods employed, such as exploring individual experiences.



**Figure 4:** Thesis' research approach adapted from Saunders et al. [38].

Figure 4 depicts the thesis’ approach adapted from Saunders et. al. [38]. Each layer of the research onion is explained in detail below, highlighting its application within this thesis.








- **Philosophy:** This layer delves into fundamental assumptions about the nature of knowledge and how it can be obtained [38]. RQ1 in Chapter 1.3 explores individual experiences and perspectives with the current system of IoTCorp. To understand these subjective experiences, this thesis adopts an *interpretivism* philosophy [38]. Furthermore, developing the test cases based on the insight of security professionals, as stated in the research objectives outlined in Chapter 1.3, reinforces the interpretivist philosophy by acknowledging the value of subjective understanding and expertise within the security field.
- **Approach:** The second layer outlines the broader strategy for generating knowledge. This thesis follows an *inductive* approach, which uses specific observations and data to progressively form insights and recommendations [38]. This approach aligns with the research objectives: identifying the key features of a suitable centralized test repository for IoTCorp (RQ1) and test cases for the requirements of EN 303 645, IEC 62443-3-3, and IEC 62443-4-1 (RQ2) (Chapter 1.3). Analyzing specific observations, documents, and interviews contributes to proposing an architecture for the test repository and mapping test cases to the requirements. Several factors influence the choice of the inductive approach:
  - **Topic Complexity:** The explored topics are complex, involving numerous factors like available information from the IoT security standards (Chapter 2.1) and the environment at IoTCorp (Chapter 1.4). An inductive approach allows for open-ended exploration, leading to a nuanced understanding of the research questions.
  - **Limited Existing Research:** As described in Chapter 2.2, limited research exists on compliance testing for IEC 62443 and EN 303 645, which calls for an inductive approach. This approach prioritizes observations and data collected from within IoTCorp to inform the development of a methodology for mapping test cases to security requirements.
  - **Understanding Practices and Needs:** Rather than testing pre-established assumptions through hypotheses, the inductive approach facilitates the exploration of current practices and the identification of specific needs within IoTCorp regarding the centralized test repository and test case mapping.
- **Strategy:** This layer defines the specific research design chosen to answer the research questions. This thesis employs a *case study approach* [8, 9], which involves an in-depth investigation of a particular phenomenon within its real-world context [38]. In this case, the case study focuses on IoT security compliance testing within IoTCorp. This approach is chosen to gain a deeper understanding of how security testing for IoT devices and systems is actually conducted in practice. Additionally, the case study provides access to existing organizational material on security testing of IEC 62443 and EN 303 645.
- **Choice:** This layer addresses the specific data collection methods employed within the chosen research design. Since the focus is on understanding individual experiences and perspectives, *qualitative* methods like interviews, observations, and document analyses are selected for this thesis [38]. These methods provide rich descriptive data necessary for exploring complex phenomena like human behavior, opinions, and experiences [86]. In contrast, quantitative research focuses on measuring and analyzing numerical data, aiming to test pre-existing hypotheses and derive generalizable conclusions [86]. While valuable for objective measurement, quantitative approaches are less suitable for this study's core objective of comprehending the subjective aspects of the testing process, where qualitative methods are better equipped to achieve this goal. Although qualitative methods are ideal for capturing rich, descriptive data, it is important to acknowledge their inherent subjectivity. The potential for personal biases in the test repository is addressed further in the limitations (Chapter 7).
- **Data Collection Methods:** This layer details the specific employed methods for data collection. Such methods typically involve developing interview guides, designing questionnaires, or establishing observation protocols to ensure consistency and reliability in data gathering [38]. For this thesis, data collection primarily involves three qualitative methods:

- **Observation:** This method allows for the collection of insights into the technical environment of the case study, including practical aspects of the current system. It facilitates the identification of key features and how they are implemented, enriching the understanding of the system's functionality. The observations are described in Chapter 4.2.
- **Document Analysis:** This method enables the examination of relevant documents such as IoT security standards and related research. It helps to identify essential information for addressing RQ1 (Chapter 1.3) and understand the specific compliance requirements for IoTCorp, forming the foundation for the test repository developed in RQ2.
- **Interviews:** (detailed in Chapter 3.2.1): This method facilitates the exploration of diverse perspectives from key stakeholders involved with the current system and security professionals contributing expertise for the repository. It allows for a deeper understanding of the testing process's practical aspects and potential areas for improvement within the existing system. A purposive sampling technique is employed to select participants with specific knowledge and experience in IoT device compliance testing within IoTCorp.

The data collection methods are included in the figures within Chapter 3.2, to highlight how these methods contribute to addressing the research questions.

### 3.2 Research Design

The research design for this thesis outlines the approach used to answer the three research questions: RQ1 in Chapter 3.2.1, RQ2 in Chapter 3.2.2, and RQ3 in Chapter 3.2.3) and we a summary in Chapter 3.3.1. Figure 5 through Figure 8 represent the flow of information throughout the research process. These figures employ different shapes to denote various elements, as outlined in the legend in Table 7 on the following page.

Shape	Name	Usage
	Source	The specific source from which information is extracted (e.g., "Company").
	Resource	The specific information or data obtained from the data source. It could be either an artefact (e.g., "Architecture", "Requirement") or a human actor (e.g. "Expert to participate in the research").
	Information	Represents processed or contextualized information based on the artefact (e.g., "Employee opinion," "Mapping").
	Contribution	The final outcome or insight derived through data processing and analysis (e.g., "Overview of necessary requirements").
<i>Italic Text</i>	Data Collection Method	Describes the specific approach used to gather information from the source (e.g., "Interview").
	Decision	Used to change the direction of a program. This block indicates a decision to be made. The outgoing arrows are labelled with possible answers to the question posed. (e.g., "Do the necessary requirements address the employee experiences?").
	Direction of flow	Arrows show how to travel through the flowchart.
	Step	The circle denotes the step of the approach, corresponding to the research question the figure addresses.

**Table 7:** Legend for figures used in Chapter 3.2.

The legend in Table 7 details the specific meaning and usage of different shapes employed within the figures. By referring to this legend, readers can effectively interpret the various elements presented within the figures and gain a deeper understanding of the research approach.

### 3.2.1 Research Process for the Design of the Architecture

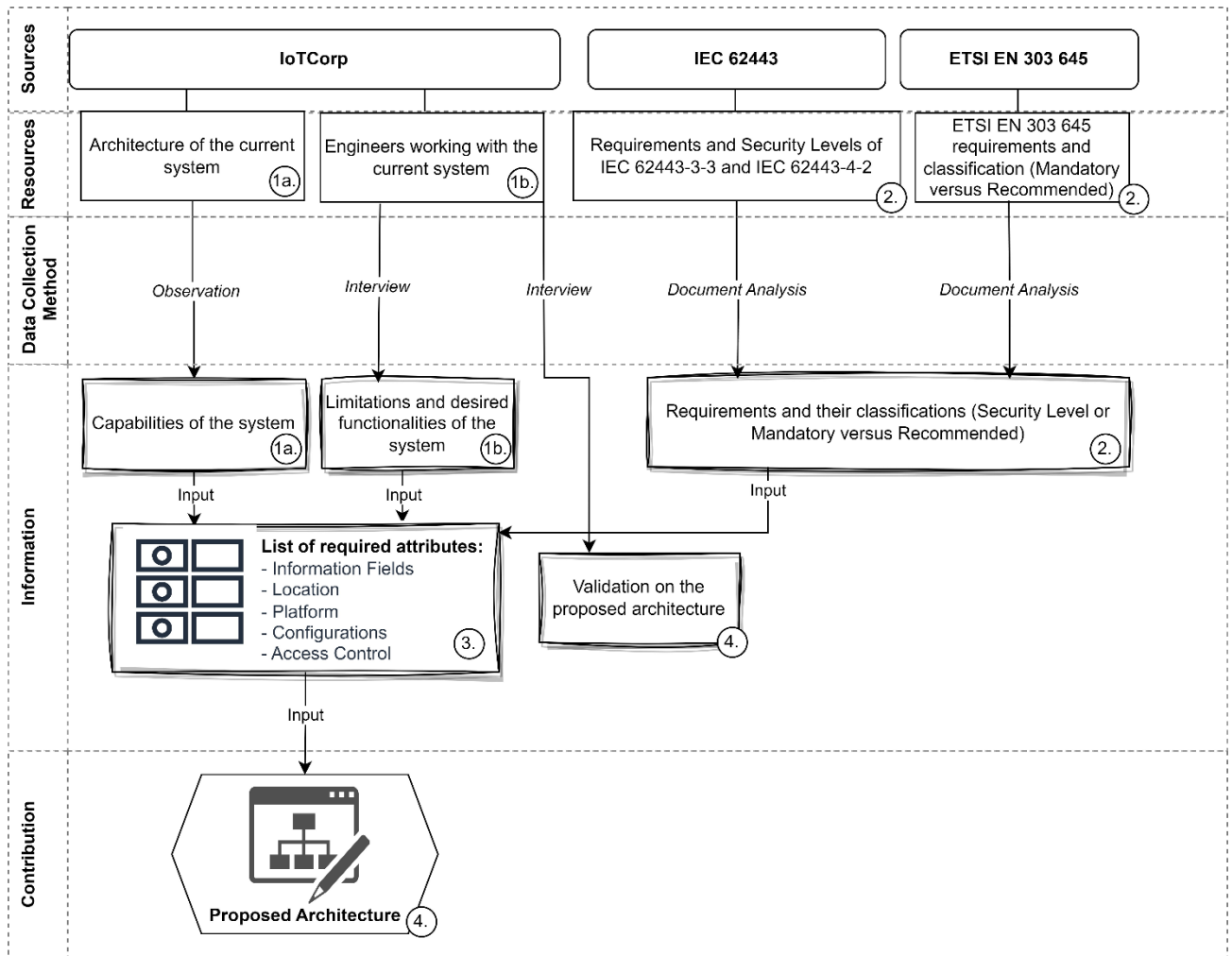
This chapter elaborates on the research process employed to address RQ1 (Chapter 1.3), in which we identify the key features needed for a centralized test repository specifically designed for IoTCorp. We employ a multifaced research approach, involving distinct ordered steps as visualized in Figure 5 on the following page.

The steps for addressing RQ1 are as follows (referencing Figure 5 on the following page):

1. **Analysis of the Current System (Step 1a and 1b):**
  - a. Through **observation** we examine the architecture of the current system to identify current capabilities. Observing the existing system in action allows us to gain insights into its actual usage and workflow.
  - b. Two semi-structured **interviews** are conducted, with a group of eight engineers (detailed in Chapter 3.2.1.1 ) currently working with the existing system at IoTCorp, to gain valuable user perspectives and inform the development process. The first interview focuses on understanding user needs and challenges associated with the current system. This open discussion yields critical insights into:
    - i. **Details of the Current System:** Exploring various aspects of the existing system used for security testing. This includes used functionalities and available information.
    - ii. **Limitations:** Identifying the weaknesses of the existing system from the user's perspective provides valuable context for the design of the new repository.
2. **Analysis of IoT Security Standards (Step 2):** We perform document analyses on the IEC 62443-3-3, IEC 62443-4-2, and EN 303 645 to obtain their requirements and Security Levels or classification of Mandatory or Recommended requirements (detailed in Chapter 2.1).
3. **Defining Key Attributes (Step 3):** We use the current capabilities and limitations of the system, combined with information from the IoT security standards to define key attributes, which are outlined in Table 8.
4. **Proposing and Validating the Architecture (Step 4):** We propose a new architecture and conduct the interview again to obtain validation. The second interview presents the proposed architecture of the test repository, soliciting feedback from the engineers. This validation process aims to ensure that the proposed features effectively address the identified challenges and align with user expectations. The engineers were satisfied with the proposed architecture and therefore no further refinements were made to the architecture.

The analysis of the current system is described in Chapter 4. More specifically, the new architecture is presented in Chapter 4.4.





**Figure 5:** The flowchart for answering for RQ1.

As discussed, Table 8 presents a comprehensive overview of the key attributes that will define the functionality and user experience of the test repository. These attributes serve as the building blocks for a robust and secure system that effectively facilitates compliance testing for IoT devices.

Attribute	Description
Information Field	The specific data points to be captured within the test cases for effective and comprehensive testing.
Presentation	The relationships and organization of information fields within the overall data overview.
Platform	Selecting the most suitable software platform for building the repository, ensuring the platform aligns with both technical requirements and user needs.
Configurations	This attribute refers to the features and capabilities offered by the chosen platform for managing test cases and associated data. It encompasses the various actions users can perform within the system, as well as the ability to configure those functionalities to optimize workflows and user experience.
Access Control	Establishing secure access levels and permissions for various user groups, ensuring data security and integrity.

**Table 8:** Attributes and their description.

The selection of the key attributes for the design of an architecture for a test repository follows Norman's 7 Principles of Design (1988). These principles offer a framework for creating user-centered

interfaces that are intuitive and efficient. Table 9 details how Norman's Usability Principles [87] influence the selection of key attributes for a user-friendly test repository architecture.

<b>Attribute</b>	<b>Principle</b>	<b>Impact of Principle on Attribute Selection</b>
Information Fields	Use both knowledge in the world and knowledge in the head.	Information, e.g., labels, should be inherently understandable and data points should reflect existing knowledge of testing practices, minimizing training needs.
Presentation	Make things visible.	The organization and visual design need to be clear and intuitive, providing a well-structured overview of the data within the information fields.
Platform	Get the mappings right.	The chosen platform should align with the user's mental model of how a test case repository functions. This means leveraging on familiar environments, e.g., similar to the existing system, and ensuring its features map to user expectations.
Configurations	Simplify the structure of tasks.	The system should offer well-defined options and intuitive actions for managing test cases. Clear menus, buttons aligned with testing practices (e.g., "Mark as Pass"), and customizable options (e.g., use of filters in reporting overviews) all contribute to simplifying tasks.
Access Control	Exploit the power of constraints, both natural and artificial.	By establishing secure access levels based on user roles, the system utilizes constraints to promote user safety and data security. This ensures unauthorized actions are restricted and data integrity is maintained.

**Table 9:** The principles of Norman that impact the attribute selection.

Chapter 4.4 delves into the desired attributes envisioned for the new centralized test repository, leveraging the principles in Table 9. Further details regarding the current system's attributed can be found in Chapter 4.2.

### 3.2.1.1 Interview Methodology

To gain valuable insights into the current lotCorp system's limitations and potential improvement areas, we conducted a two-phase interview study with a group of eight engineers from the company. Table 10 provides an overview of their diverse roles, highlighting the expertise brought to the interviews.

<b>Role</b>	<b>Number of Engineers</b>
Security Manager	1
Application Security Manager	1
Product Security Architect	2
Development Engineer	1
System Security Architect	1
Software Operations Security Architect	1
Product Security Manager	1

**Table 10:** Roles of the interviewed engineers.

Selecting engineers from various roles working with the system ensured a well-rounded understanding of its strengths and weaknesses. The Security Manager, for instance, could offer

valuable insights into the system's reporting capabilities. Similarly, the Development Engineer could shed light on completeness of test cases and identify areas for streamlining. By combining these diverse perspectives, the interviews aimed to form a holistic picture of the current system and its opportunities for improvement in terms of process efficiency, collaboration, and knowledge sharing.

The interview study consists of two phases:

- **Phase 1:** The first interview phase aims to identify and gather insights on the limitations of the current system. A semi-structured interview guide was created to focus on areas like test coverage, traceability, and usability for compliance testing.
- **Phase 2:** The second interview phase serves to validate the initial findings from Phase 1 and explore the impact of the discovered limitations on different stakeholder roles. The interview guide for this phase was further refined based on the initial analysis of the collected data in Phase 1.

We employ Charmaz's Constructive Grounded Theory [88] approach to analyze the interview data. This approach is iterative, meaning all stages (initial coding, memoing, and core category development) inform each other throughout the analysis. The stages are as follows:

- **Initial Coding:** Transcripts are reviewed to identify initial concepts and themes related to the limitations of the system.
- **Memoing:** Throughout the analysis, detailed memos are written to document observations, emerging themes, and potential connections between them.
- **Developing Core Categories:** Core categories were identified that represented the central themes of the limitations based on the coded data and memo reflections.

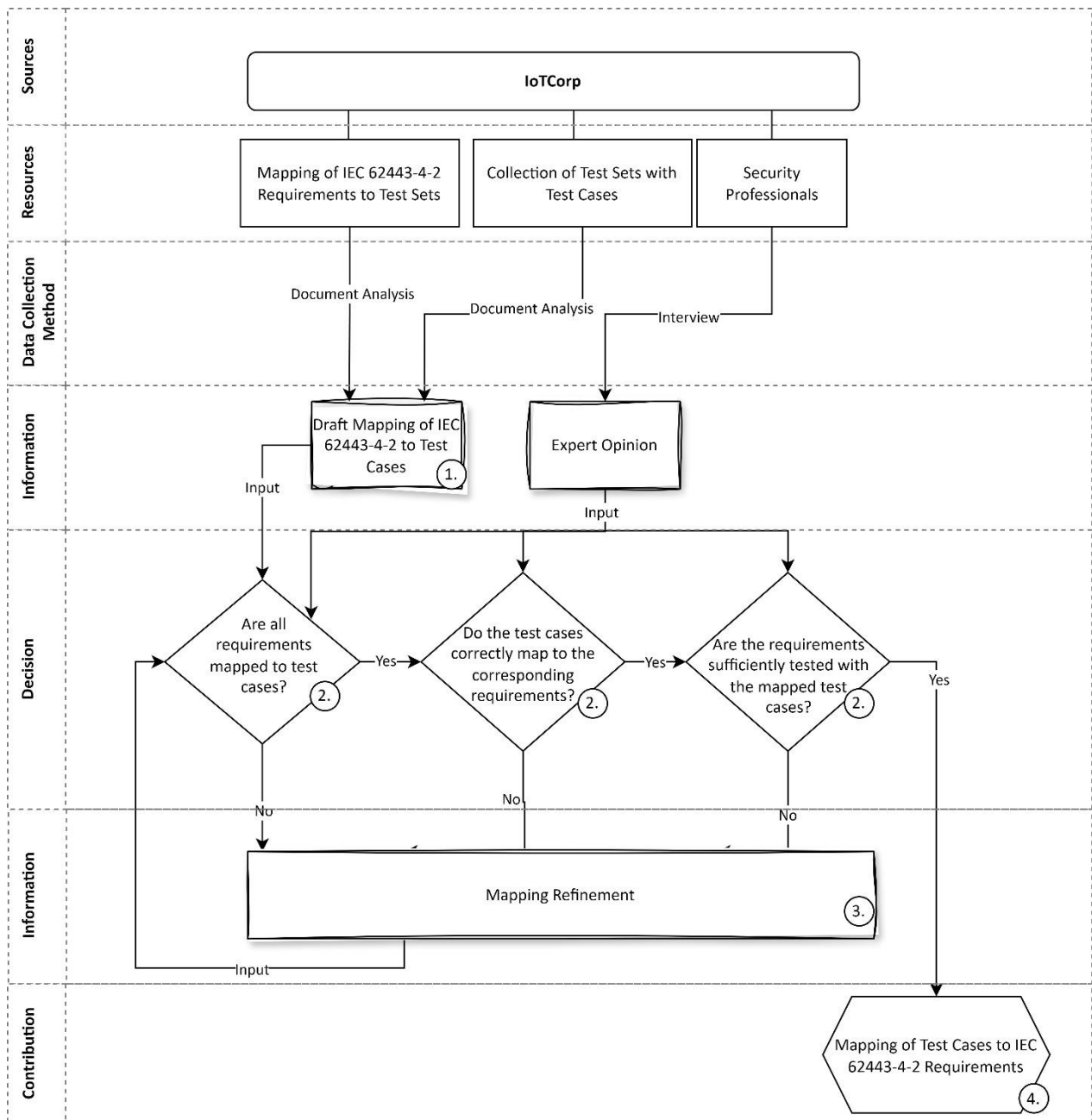
Informed consent was obtained from all participants prior to the interviews. All data was anonymized to ensure participant confidentiality.

### *3.2.2 Research Process for the Development of Test Repository*

This section details the research process employed to map test cases to security requirements and develop a test repository (Chapter 5) to address RQ2 (Chapter 1.3) for each of the three security standards: IEC 62443-4-2, IEC 62443-3-3, and EN 303 645. While a standardized process is maintained, specific procedures differ for each standard, as detailed below. Figure 6 (IEC 62443-4-2), Figure 7 (IEC 62443-3-3), and Figure 8 (EN 303 645) visually depict the flowchart for each standard.

#### *3.2.2.1 Mapping Test Cases to IEC 62443-4-2 Requirements*

This chapter focuses on mapping test cases to the requirements of the IEC 62443-4-2 standard. This process leverages existing resources, including materials provided by IoT Corp, such as existing mappings and test cases. Here, we employ a multifaceted research approach, involving distinct ordered steps as visualized in Figure 5.



**Figure 6: Flowchart for RQ2 for IEC 62443-4-2.**

The steps for addressing RQ2 for IEC 62443-4-2 are as follows (referencing Figure 6):

1. **Draft Mapping (Step 1):** The draft mapping is created using the existing mapping of IEC 62443-4-2 requirements to test sets of IoT Corp, alongside test cases and their details.
  - a. **Requirement Analysis:** By performing document analysis, each test case within a test set is analyzed to identify the specific IEC 62443-4-2 requirement(s) it contributes to testing. For example, Test Set A may contain 20 test cases linked to four requirements. Each test case in this set is examined to determine which of these four requirements it tests compliance with.
  - b. **Mapping Assignment:** If a test case addresses one or more requirements, it is assigned to those specific requirements. This process is repeated for all test cases within the test set.

2. **Validation (Step 2):** The initial mapping is reviewed by two security experts. Their profiles are described in Chapter 3.3. The validation step is to ensure three aspects of the initial mapping:
  - a. **Completeness:** Identify any missing links between test cases and requirements (e.g., untested requirements or redundant test cases).
  - b. **Accuracy:** Verify that all test cases are mapped correctly to the relevant requirements.
  - c. **Sufficiency:** Assess whether all mapped test cases adequately address their corresponding requirements.

These three aspects to be validated, are represented by the three decision nodes in Figure 6, i.e. see the three questions in the three nodes.

3. **Mapping Refinement (Step 3):** Based on the experts evaluations in the validation step, requirements may be:
  - a. **Added:** If the experts identify untested security aspects, new test cases may be added to address these gaps.
  - b. **Altered:** Existing test cases may be refined to provide clearer or more specific testing guidelines.
  - c. **Deleted:** Redundant or unnecessary test cases may be removed to streamline the testing process.

If the experts identify any issues during the refinement process, the mapping is revised accordingly, and step 2 (Requirement Analysis) is repeated to ensure the revised mapping aligns with the refined test cases. This iterative process continues until the experts provide no further feedback on the mapping and requirement definitions.

4. **Finalization (Step 4):** We present the final mapping for the IEC 62443-4-2 requirements and test cases, incorporating any refinements from the validation and requirement refinement step.

To illustrate the above steps, Table 11 on the following page shows a real-life example of a test set. In this example, the Initial Mapping step needs to be performed, for which the test cases will be mapped to corresponding requirements.

Test Set	Linked Requirements	Test Cases
Denial of Service Protection.	IEC 62443-4-2 CR 7.1, IEC 62443-4-2 CR 7.1 RE 1, IEC 62443-4-2 CR 7.2.	Verify a single user cannot overload a system with certain requests, such as the recording and sending of the same packets repetitively.
		Verify that the device has measures in place to prevent or minimize the impact of denial-of-service attacks, such as excessive network traffic, log flooding, and application/protocol traffic.
		Verify that request throttling is in place to prevent automated attacks against common authentication attacks such as brute force attacks or denial of service attacks.
		Verify that there exists some protective mechanism that can prevent DoS attacks, which involve flooding the network with excessive data or using unauthorized applications from remote devices to disrupt the system.

**Table 11:** Example of test set with linked requirements and test cases.

Table 11 illustrates a test set from IoTCorp that includes test cases focusing on the protection of devices and systems against Denial-of-Service attacks. There are several requirements mapped to this test set. During the initial mapping stage (step 1), we evaluated whether the test cases adequately and sufficiently cover the requirement IEC 62443-4-2 CR 7.1, which states; *“Components shall provide the capability to maintain essential functions when operating in a degraded mode as the result of a DoS event.”*. The test case we mapped to this requirement is “Verify a single user cannot overload a system with certain requests, such as the recording and sending of the same packets repetitively.” (see the first row of Table 11 below). This test case verifies if the system can handle repeated requests from a single user, which can mimic a basic DoS attack. If the system can still function under such stress, it suggests a level of resilience against DoS attacks. The other test cases test the other requirements linked to the test set (IEC 62443-4-2 CR 7.1 RE 1 and IEC 62443-4-2 CR 7.2) better and are therefore not linked to IEC 62443-4-2 CR 7.1. The result of this execution of the Initial Mapping step is presented in Table 12.

Requirement	Test Case
IEC 62443-4-2 CR 7.1	Verify a single user cannot overload a system with certain requests, such as the recording and sending of the same packets repetitively.
IEC 62443-4-2 CR 7.1 RE 1	Verify that the device has measures in place to prevent or minimize the impact of denial-of-service attacks, such as excessive network traffic, log flooding, and application/protocol traffic.
	Verify that request throttling is in place to prevent automated attacks against common authentication attacks such as brute force attacks or denial of service attacks.
IEC 62443-4-2 CR 7.2	Verify that the device has measures in place to prevent or minimize the impact of denial-of-service attacks, such as excessive network traffic, log flooding, and application/protocol traffic.
	Verify that there exists some protective mechanism that can prevent DoS attacks, which involve flooding the network with excessive data or using unauthorized applications from remote devices to disrupt the system.

**Table 12:** An example of a mapping for requirements includes in the test set of Table 11.

The test repository (Appendix B) consists of many of such mappings, as illustrated in Table 12. All of these mappings were subjected to validation by the security experts.

### 3.2.2.2 Mapping Test Cases to IEC 62443-3-3 and EN 303 645 Requirements

This chapter builds upon the established mapping for IEC 62443-4-2, leveraging the results and insights gained from the previous steps. The process for IEC 62443-3-3 and EN 303 645 uses the previously defined steps for IEC 62443-4-2, with additional adaptations to address specific nuances the standards. These steps reference Figure 7 and Figure 8, on the following pages.

1. **Leveraging Existing Information (Step 1 in Figure 7 and Figure 8)**
  - a. **IEC 62443-3-3:** The final IEC 62443-4-2 mapping is utilized to identify congruent requirements (identical or highly similar) between the standards. For each identified matching requirement, corresponding test cases from the IEC 62443-4-2 mapping are directly mapped to the matching requirement in IEC 62443-3-3, enabling efficient reuse of existing test cases where applicable.
  - b. **EN 303 645:** Existing research on the alignment between EN 303 645 and IEC 62443-3-3 [89] is utilized to leverage corresponding test cases from the established IEC 62443-3-3 mapping.
2. **Mapping of Unmapped Requirements (Step 2 in Figure 7 and Figure 8):** For requirements not found in the previous alignment (unmatched requirements) for either standard, a new mapping is established using materials from IoT Corp. This process follows an iterative approach, involving:
  - a. **Requirement Analysis:** Each unmapped requirement is carefully analyzed to understand its specific security objective.
  - b. **Mapping Assignment:** Based on the analysis, the requirement is assigned to the most relevant test case(s) from the available pool, ensuring appropriate testing coverage.
3. **Validation (Step 3 in and Refinement (Step 4 in Figure 7 and Figure 8):**
  - a. Similar to the IEC 62443-4-2 process, experts review and refine the mapping for unmapped requirements. This potentially involves adding new requirements, altering existing ones, or adjusting test case assignments. This iterative process continues until a comprehensive and accurate mapping is established for both IEC 62443-3-3 and EN 303 645 requirements and test cases.
4. **Finalization (Step 5 in Figure 7 and Figure 8):** The final mappings for both IEC 62443-3-3 and EN 303 645 are presented, incorporating the results from leveraging existing information and mapping unmapped requirements.

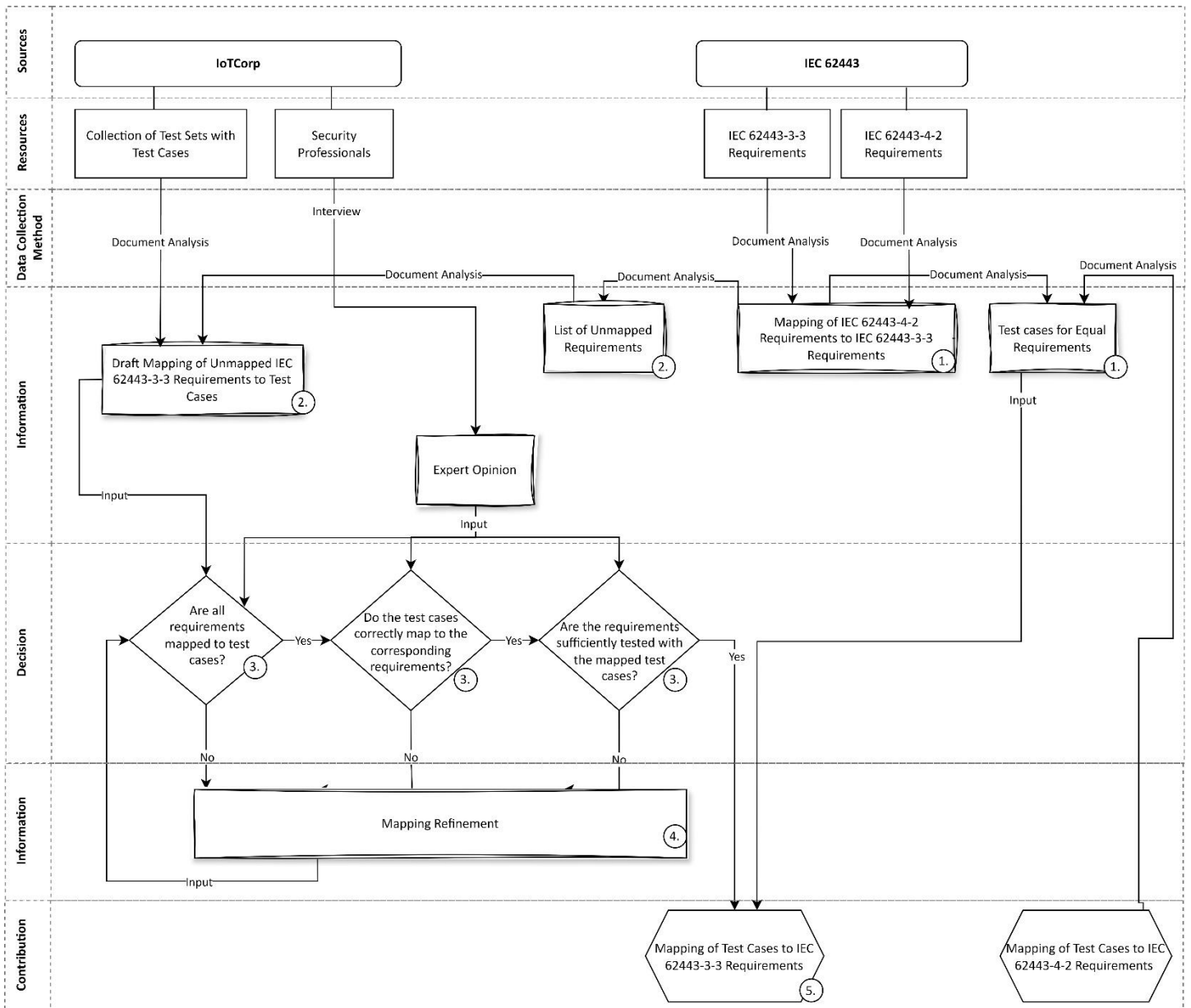


Figure 7: Flowchart for RQ2 for IEC 62443-3-3.



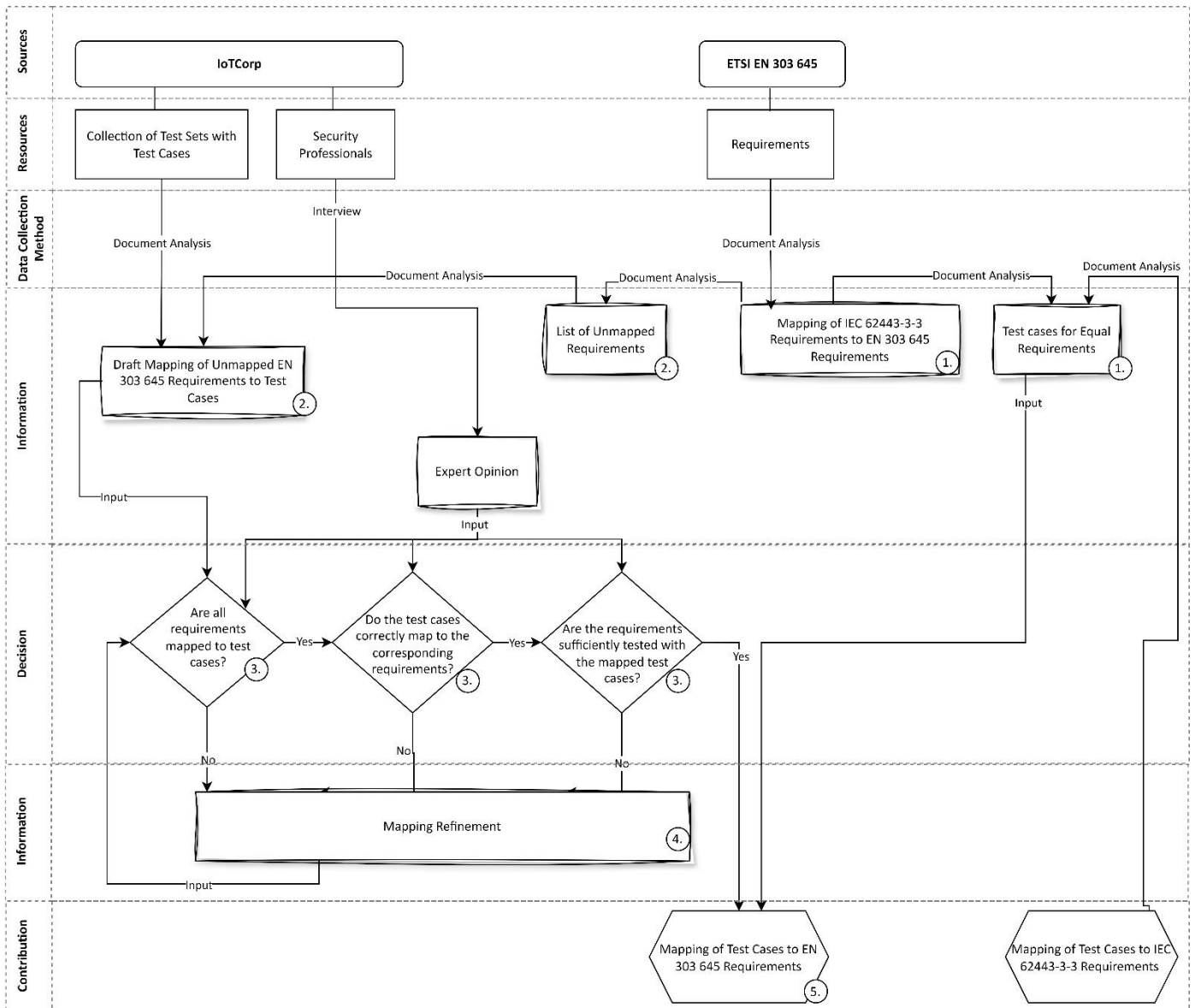


Figure 8: Flowchart for RQ2 for EN 303 645.

### 3.2.3 Research Approach for Integrating the Test Repository

The developed test repository, detailed in Chapter 5 is currently a data compilation within an Excel file [90], which needs to be manually populated into Jira. This manual process is required due to two reasons:

- **Data Transfer to Platform:** The data needs to be transferred from the Excel spreadsheet into the structured format of the Jira platform. This transfer cannot be automated because the Excel data lacks the organization and standardization required by Jira's pre-defined "Issue Type" categories, specifically "Requirement" and "Test Case."
- **Data Interpretation and Classification:** The manual population process also allows for human expertise in interpreting the data and accurately classifying it into the appropriate "Issue Type" categories within Jira. This ensures the data aligns with the platform's specific structure and facilitates efficient retrieval and utilization.

The process of transferring the data is as follows:

1. **Creating Requirements in Jira:** First, we create a new "Issue Type" within the Jira repository called "Requirement." This type has three attributes:
  - a. **Title:** This follows a specific format, including the source (e.g., "IEC 62443") and the requirement title (e.g., "SR 1.3 - Account Management").
  - b. **Description:** This field contains the full text of the requirement. An example is: "The control system shall provide the capability to support the management of all accounts by authorized users, including adding, activating, modifying, disabling, and removing accounts."
  - c. **Labels:** We assign two labels to each requirement. These labels are explained in Chapter 4.4. and help categorize requirements based on the standard they origin from and their security level or classification.
2. **Creating Test Cases in Jira:** Next, we create another "Issue Type" called "Test Case" with two main parts:
  - a. **Title:** This field accommodates the text of the test case itself. An example would be: "Verify that Super User accounts, such as Administrator or Root, are disabled or removed wherever possible."
  - b. **Steps:** If applicable, individual steps pertaining to the test case are entered within this field.

Once all the data is transferred into Jira, it becomes a central location for all testing requirements and test cases for this project. This completes the process of building a comprehensive testing repository for IoT Corp.

### 3.3 Participants in the Expert-based Validation of the Proposed Mappings

The step of validation described in Chapter 3.2.2 includes two cybersecurity experts [91] [92] with extensive experience and relevant certifications (CISSP, CISM, CSSLP), who are recruited to provide expert validation. Here we describe their profiles as follows:

**Expert 1:** [91] Possesses over 25 years of experience in cybersecurity and holds certifications in CISSP, CISM, and CSSLP. Currently holding the role of Application and Product Security Manager, whose expertise focuses on security processes, governance, competence, and adherence to standards and regulations.

**Expert 2:** [92] Holds over 15 years of experience in cybersecurity and certifications in CISSP, CISM, CCSP, and CISA. Their current role as IoT and Cloud Product Security Manager leverages their specialized knowledge of IoT and cloud architectures.

This collaborative effort ensured a thorough evaluation of the mapped and created test cases for IEC 62443-4-2, IEC 62443-3-3, and EN 303 645. Their valuable insights and feedback are incorporated into the test repository (detailed in Chapter 3.2.2), enhancing its overall comprehensiveness and validity.

The two participants were chosen based on four criteria: (1) relevance of their expertise to this research, (2), more than 15 years of experience, (3) availability, and (4) willingness to participate

in the research. Both participants joined the validation steps of the three processes (See Figure 6, Figure 7, and Figure 8) pertaining to RQ2.

### 3.3.1 *Summary*

This chapter describes the research approach for the Design, Development, and Integration phases of the development of a proposal for a new architecture for a centralized test repository for IoT Corp and the test repository mapping test cases to requirements. The research process consists of several steps. For the Design phase, first we analyze the functionalities and limitations of the existing system through document analysis and interviews with engineers at IoT Corp. This analysis helped identify areas for improvement in the new test repository. Second, based on the findings from the first step and an examination of the security standards, we define key attributes for the new test repository. These attributes include information fields, presentation, platform, configurations, and access control.

For the Development phase, we map test cases to the security requirements of each standard. This process involves leveraging existing resources, validating the mapping with security experts, and refining the mapping based on their feedback.

Finally, for the Integration phase the developed test repository, initially compiled in an Excel spreadsheet, is manually transferred into Jira, a project management tool. This manual transfer is necessary due to limitations in data format and to ensure improved data classification within Jira.

## 4 Design of the Architecture

This chapter answers RQ1. It presents the proposed architecture for a centralized test repository, a core contribution of this thesis. First, we describe the current architecture as it is right now at the case study organization (Chapter 4.1). The limitations of the current system are identified through observations (Chapter 4.2) and interviews (Chapter 4.3). This repository aims to address the limitations of the existing testing system at IoTCorp and proposed a new architecture in Chapter 4.4. Finally, we describe how the system from the new architecture is populated with the test repository in Chapter 5.

### 4.1 Description of the Current System

Building upon insights from the initial interview (detailed in Chapter 3.2), this chapter delves into the internal working of IoTCorp's current system. The team uses Jira [93], which is a project management tool used by software development teams to track issues and bugs, manage sprints and releases, and collaborate on projects. Jira offers access control and role-based permissions and is therefore a secure location for information sharing. Within Jira, there are several issue types that can be used to track different types of work, for example, *Test Sets* and *Test Cases*, of which an example is shown in Table 2. The issue type *Test Set* is a collection of *Test Cases* that are grouped based on the topic of the *Test Set*. The issue type *Test Case* is an individual test case designed to test a specific aspect of a software application. Other issue types are discussed in Chapter 4.4 regarding the new architecture. The team uses these issue types for testing compliance with IEC 62443-4-2 requirements. The *Test Sets* created by the team are linked to one or multiple requirements using *Labels* in Jira. *Labels* are keywords or phrases that can group related issues, making it easier to search for and filter issues based on specific criteria.

To illustrate the information linked to each component, Table 13 provides an example. It illustrates the components associated with the issue type Test Set within the system. Each Test Set (e.g. "Testsetname.01") has a unique reference number (e.g., "TS001") for identification within the database. Next, Test Cases (example in Table 14) are linked to each Test Set, referenced by their unique key (e.g., "TC001", see the fourth column in Table 12 below). Furthermore, Labels (e.g., "IEC 62443-4-2 CR 7.5", see the rightmost column of Table 12) are assigned to Test Sets, each representing a specific requirement from the IEC 62443 security standard. This association between Test Sets, Test Cases, and Labels plays a crucial role in organizing and linking test data with relevant security requirements in the current system. Table 13 showcases a concrete example of a Test Set within the system's database. It displays the reference number (TS001), issue type (Test Set), test set name (Testsetname.01), associated Test Cases (TC001 and TC002), and assigned Labels (IEC 62443-4-2 CR 7.5). This information demonstrates how Test Sets are structured and linked to specific test cases and relevant security requirements.

Reference	Issue Type	Test Set	Test case(s)		Labels
			Key	Summary	
TS001	Test Set	Testsetname.01	TC001	Test case text (See Table 14)	IEC 62443-4-2 CR 7.5
			TC002	Test case text (See Table 14)	

**Table 13:** Example of a Test Set in the database of IoTCorp.

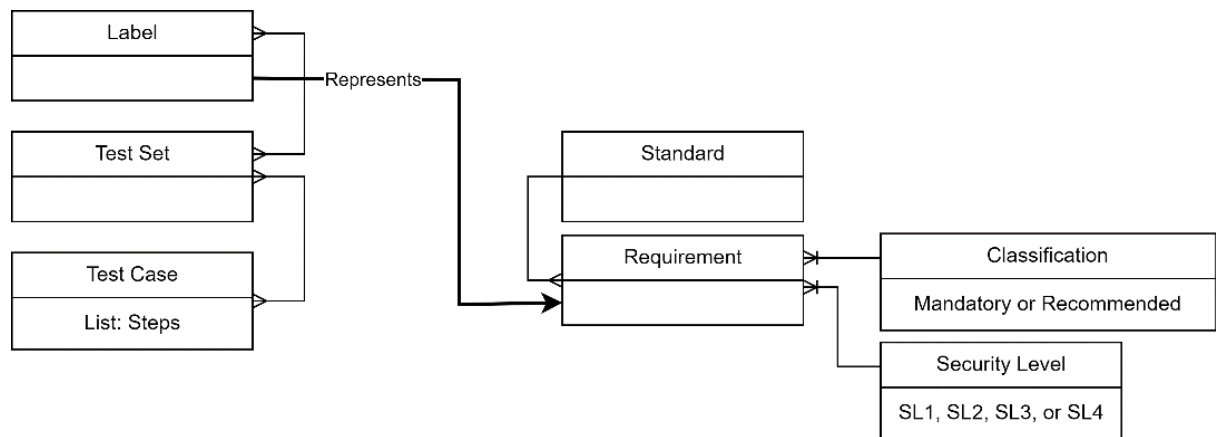
Table 14 provides an example of a Test Case entry within the database. It shows the key (e.g., "TC001") and the summary (e.g., "Perform a power failure simulation test"), which displays the text of the Test Case. Furthermore, it includes the detailed steps (e.g., "Simulate a power failure by

disconnecting the primary power source and switch to an emergency power supply”) outlining the specific procedures for conducting the test.

Key	Summary	Steps
TC001	Perform a power failure simulation test.	<ol style="list-style-type: none"> <li>1. Simulate a power failure by disconnecting the primary power source and switch to an emergency power supply.</li> <li>2. Monitor if functionality remains as expected without degrading security functions.</li> </ol>

**Table 14:** Example of a Test Case in the database of IoTCorp.

The information in Table 14 exemplifies how Test Cases are defined and documented within the current system. The construction of *Test Sets*, *Test Cases*, and *Steps* linked to the requirements of the IoT security standards is represented in Figure 9 below. This diagram depicts that a requirement is a component of a standard and has a Security Level (SL1 – SL4) (these were introduced in Chapter 2.1.1) or a Classification of ‘Mandatory’ or ‘Recommended’. The standards, requirements, Security Levels, and classification are not included in the database of IoTCorp. To link requirements of standards in their current system, IoTCorp assigns *Labels* to *Test Sets*. Figure 9 shows the construction of issue types on the current system of IoTCorp and the relationship to the standards.



**Figure 9:** Construction of test sets, test cases, steps, and the links between test sets and requirements in the current system of IoTCorp.

Chapter 4.2 below reports direct observations of the author regarding the current testing system in the organizations. These observations are instrumental to create an overview of the attributes of the current system as shown in Table 15 (in the next chapter).

#### 4.2 Observations of the Current System and its Limitations

As part of this research, the technique of direct observations [94] was used to collect qualitative data about the functionality of the current system, its workflow, and user interactions at IoTCorp. Our observations revealed valuable insights which informed the design of the proposed new architecture (see Chapter 4.4). These are as follows:

- **Platform:** Jira is used for testing.
- **Testing Method:** Manual testing is fully relied upon, with dedicated testers executing test cases.
- **Test Case Management:** Test cases are categorized and grouped using labels for organized access.
- **Testing Focus:** The current focus is on general security topics, not adherence to specific compliance standards.
- **User Base:** Only one team utilizes the system for testing purposes.

- **Documentation:** No formal documentation exists; the system leverages Jira's built-in functionalities.
- **Test Plan Creation:** Test plans are linked to specific products, outlining relevant test cases.
- **Test Execution & Reporting:** Executed test cases have a status that could be passed, failed, or not applicable. The comprehensive reporting capabilities that Jira offers are not used.

The observed testing practices revealed some limitations in both oversight and potential compliance adherence. The organization's reliance on manual testing and limited utilization of Jira's comprehensive reporting functionalities suggests a lack of data-driven insights for informed decision-making. Additionally, focusing solely on general security topics, without focusing on the requirements of the standards, raises concerns regarding potential compliance gaps. Furthermore, the fact that only one team currently utilizes the system highlights limited collaboration and potentially siloed testing practices. These observations collectively point towards the need for a more strategic approach incorporating standardized testing frameworks to enhance oversight, promote compliance, and facilitate broader adoption across the organization. As automated testing methods are not in scope for this thesis, we included them as a point discussed in Future Work (Chapter 7). Table 15 combines the findings on the current system from observations and interviews (Chapter 4.1) in order to define the current attributes of the system.

Attribute	Current System
Information Fields	While <i>Steps</i> are part of <i>Test Cases</i> , additional information like requirements and standards are not included.
Presentation	The use of <i>Test Sets</i> to group <i>Test Cases</i> based on <i>Labels</i> (representing requirements) provides some structure. However, the lack of information about standards, requirements, and security levels or classification within the database limits the overall data overview.
Platform	While Jira is technically suitable, leveraging underutilized functionalities, like reporting dashboard for non-compliance, could further enhance employee needs for data insights.
Configurations	Basic functionalities for test case management and testing are utilized. However, these functionalities are not fully utilized, limiting the system's potential for efficient test execution and insightful analysis.
Access Control	Jira offers access control and role-based permissions, suggesting secure information sharing.

**Table 15:** Attributes (detailed in table 8) of the current system of IoT Corp.

Table 15 highlights several limitations within the system's current attributes. Collectively, these limitations hinder the comprehensive understanding of the testing process and its effectiveness in addressing security concerns. Test cases lack crucial details such as requirements, standards, and security levels. Furthermore, despite test cases being grouped by "Labels" representing requirements, the system lacks information about the specific standards and security considerations, such as lack of security levels. Next, while the platform that was used (i.e., Jira) did offer functionalities that could improve efficiency and analysis, features such as reporting dashboards remained underutilized. Finally, the observations collected by the author indicated that the system primarily relies on basic functionalities for testing and therefore opportunities for efficient test execution and insightful analysis are missed out upon. Reflecting on the attributes from Table 14 makes us conclude that these limitations collectively hinder the system's ability to provide a clear picture of security coverage and effectively manage the testing process.

### 4.3 Interview Findings regarding the Limitations of the Current System

This chapter highlights the limitations of the current system of lotCorp based on the findings from the conducted interview (detailed in Chapter 3.2.1). We employed Charmaz's Constructive Grounded Theory [88] approach to analyze the interview data and identified the following key limitations:

- **Limited Testing Scope:** One key limitation identified was the limited testing scope. Engineers, particularly Security Specialists and Test Engineers, expressed concerns that the system's focus on IEC 62443-4-2 neglects other relevant standards such as IEC 62443-3-3 and EN 303 645. This hinders their ability to comprehensively assess the security posture of IoT devices. The interview revealed engineers struggle to gain a complete understanding of a device's security because the system focuses on a single standard.
- **Unclear Traceability:** Traceability between requirements and test cases is hampered due to requirements being linked to *Test Sets* using *Labels*, which means that when a *Test Case* fails, it is difficult to pinpoint the root cause of test case failures and ensure compliance with all necessary standards. The interview revealed challenges in identifying the root cause due to this lack of clear traceability. The participating engineers highlighted the difficulty in understanding which specific requirement a failing test case is linked to, hindering their ability to effectively address compliance issues.
- **Limited Accessibility to Requirements:** The current system only includes the requirement reference code (e.g., "*IEC 62443-4-2 CR 7.2*"), and not the full requirement text (e.g., "*Components shall provide the capability to limit the use of resources by security functions to protect against resource exhaustion.*" [1]). This limited accessibility to full requirements hinders understanding and oversight. The engineers highlighted the difficulty in accessing complete details, potentially leading to a lack of understanding and oversight during the testing process. The interview revealed that without the full context of the requirement, engineers struggle to effectively evaluate compliance or identify potential security vulnerabilities.
- **Inability to Group Requirements:** The inability to group requirements by standard or Security Level is identified as a limitation. Standards such as IEC 62443 assign Security Levels, while others like EN 303 645 utilize classifications of "Mandatory" or "Recommended." The lack of grouping based on this information makes it difficult to assess overall compliance for a particular standard or Security Level. The interview revealed that engineers struggle to gain a holistic view of compliance because they cannot easily identify which requirements belong to a specific standard or security level.

The limitations have a differential impact on stakeholders (see Table 10). Management, including the Security Manager and Application Security Manager, lacks essential reporting functions. This hinders their ability to effectively monitor compliance progress. Technical personnel, encompassing Product Security Architects (2), the Development Engineer, System Security Architect, and Software Operations Security Architect, all face challenges due to the lack of a dedicated testing environment. This limitation impedes their ability to efficiently execute test cases.

### 4.4 Proposal for a New Architecture

The proposed architecture for the centralized test repository aims to address the limitations of the current system discussed in Chapters 4.2 and 4.3. Moreover, the proposed architecture is expected to fulfill the key attributes presented in Table 16 below.

Attributes	New Architecture
Information Fields	Comprehensive standards coverage with "Requirement" issue type and labels for Security Levels and classifications.
Presentation	Grouping requirements by standard using specific labels (e.g., "IEC-62443-3-3-REQLIST" and a clear link between test cases and requirements.
Platform	Building upon existing Jira platform.
Configurations	Enables non-compliance identification based on failed test cases and generates multifaceted compliance reports with product and requirement perspectives.
Access Control	Leverage Jira's existing role-based access control (RBAC) to ensure appropriate access levels.

**Table 16:** Attributes for the new repository architecture.

The attributes in Table 16 were first introduced in Table 8 earlier in this thesis. For the purpose of readability, we explain them more in detail in the remaining of this chapter. Chapter 4.4.1 treats the first two attributes of Table 15, namely Information Fields and Presentation. Next, Chapter 4.4.2 treats the attributes Platform, Configurations, and Access Control.

#### 4.4.1 Attributes: Information Fields and Presentation

Table 17 illustrates an example of how the new architecture incorporates the attributes Information Fields and Presentation.

Reference	Issue Type	Requirement	Test case(s)		Labels	Security Level
			Key	Summary		
REQ001	Requirement	IEC 62443-3-3 SR 7.5	TC001	Test case placeholder text (See Table 18)	IEC-62443-3-3-REQLIST	SL2
			TC002	Test case placeholder text (See Table 18)		

**Table 17:** Example of a requirement in the test repository.

We refer to Table 17 during the description in the remaining text for these attributes. The new architecture's attribute **Information Field** addresses the limitations of the current system regarding information availability. The proposed improvements are as follows:

1. **Comprehensive Standards Coverage:** The repository covers various security standards, including IEC 62443 and EN 303 645, ensuring a broader range of testing for IoT devices, systems, and components. This includes creation of the issue type requirements and inclusion of Security Levels and classification:
  - a. **Issue Type "Requirement":** The requirements are incorporated in the system by creation of the issue type "*Requirement*", which is used to track and manage product requirements for software development projects. It can include information such as a description of the *Requirement*, its priority, its status, and any associated attachments or comments. As illustrated in Table 17, the issue type requirement has a unique reference (e.g., "REQ001") and refers to a requirement of IEC 62443 or EN 303 645 (e.g., "IEC 62443-3-3 SR 7.5"). Furthermore, a requirement has test cases associated with it, which are those that test compliance to the requirement. An



example of a test case is shown in Table 18 and is further explained for the attribute Presentation, along with the use of Labels.

- b. Inclusion of Security Levels and Classification:** The four Security Levels (SL) described in Chapter 2.1.1 are represented in the test repository by incorporating them as Labels; “SL-1”, “SL-2”, “SL-3”, and “SL-4”. Furthermore, for requirements of EN 303 645, the classification is integrated by use of the Labels ‘Mandatory’ and ‘Recommended’. Teams are able to filter the requirements based on their Security Level or classification and can therefore obtain a list of test cases specifically intended to test the associated product. For example, in Table 17, “SL-2” is assigned to the requirement, which can be used as a filter to collect all requirements that have security level 2 for devices and services corresponding to that level.

Furthermore, the attribute **Presentation** addresses the limitations of the current system in information display and relations between components:

- 1. Grouping Requirements by Standard:** For the test repository, it is necessary to group all requirements related to a specific standard together, such as those belonging to EN 303 645. This is achieved by linking one of the following labels: ‘IEC-62443-3-3-REQLIST’, ‘IEC-62443-4-2-REQLIST’, and EN-303-645-REQLIST’ to each respective requirement. By doing so, the test repository can display all requirements within a standard. This is illustrated in Table 17, where the label is “IEC-62443-3-3-REQLIST”. Use of this label yields a list of all requirements within IEC 62443-3-3 and their associated test cases.
- 2. Link between Test Cases and Requirements:** The new architecture shows to which requirements Test Cases test compliance to, as shown in Table 18. In this example, the test case tests “REQ001”, which is the unique reference for a requirement.

Key	Summary	Steps	Tests
TC001	Perform a power failure simulation test.	<ol style="list-style-type: none"> <li>1. Simulate a power failure by disconnecting the primary power source and switch to an emergency power supply.</li> <li>2. Monitor if functionality remains as expected without degrading security functions.</li> </ol>	REQ001

**Table 18:** Example of a test case in test repository and its corresponding information.

The Test Case and its associated attributes in Table 18 are, apart from the ‘tests’ element, identical to those of the system of IoT Corp. The information and set-up for the test cases only required a link to the requirement and are therefore not altered.

#### 4.4.2 Attributes: Platform, Configurations, and Access Control

As explained in Chapter 4.3, the current platform is technically suitable. Therefore, for the attribute **Platform** in the new architecture, we propose to continue the use of Jira. This choice is justified for two reasons: (i) Jira is the platform currently used by IoT Corp, and (ii) the limitations of their current system can still be addressed using Jira.

To address the limitations regarding unutilized capabilities of the platform, we propose the following for the attribute **Configurations**:

- **Non-Compliance Identification:** The repository offers functionalities to clearly identify non-compliant requirements based on failed test case results. This enables teams to

prioritize remediation efforts and address specific security vulnerabilities promptly. Table 19 visualizes how the system depicts failed test cases and the requirements they test.

Product	Test Case	Requirement(s)	Status
IoTDevice V1	Verify a single user cannot overload a system with certain requests, such as the recording and sending of the same packets repetitively.	IEC 62443-3-3 SR 7.1, IEC 62443-4-2 CR 7.1.	FAIL

**Table 19:** An example of a failed test case.

The information in Table 19 offers an overview of the testing performance for a specific product (e.g., “IoTDevice V1”). The status of a test case for a product provides the employees the opportunity to monitor compliance per products and investigate security flaws for failed test cases.

- **Compliance Reporting:** Upon completion of product testing, Jira enables the generation of compliance reports. These reports offer a multifaceted view of test results, encompassing both a product-centric perspective (e.g., overall pass/fail percentage) as shown in Table 20, and a requirement-specific perspective (i.e., number of associated tests that passed, failed, or are not applicable) as shown in Table 20.

Product: IoTDevice V1		
PASS (50%)	FAIL (30%)	NOT APPLICABLE (20%)

**Table 20:** Example of a test coverage of a product.

Table 20 depicts the test coverage of a product. This offers opportunities to monitor overall progress per product. For example, the pass rate is an indicator of the overall success of the testing process, while the fail rate indicates areas that require further attention. Furthermore, the reporting of test coverage helps to identify potential issues, such as products with a high number of failed test cases. Next, the testing results may be used to convey the test progress and outcomes to stakeholders, where the overall pass rate may be highlighted. Such information also serves as comparative data across different products and departments, which can provide insights on progress and identify trends. This product-centric view, along with the requirement-specific perspective of Table 21, contributes to comprehensive reporting of compliance with security standards.

Requirement	Total Tests	Tests Passed	Test Failed	Test Not applicable
IEC 62443-3-3 SR 7.1	2	0	2	0

**Table 21:** Example of an overview of test coverage per requirement.

Table 21 offers an organization-centric view of the testing process. This report highlights how the organization's test practices fare in meeting individual requirements. This information assists in ensuring organizational-wide compliance and pinpoints requirements that consistently fail tests. Repeatedly failing requirements might signify security weaknesses requiring prioritization across all products to address these potential vulnerabilities.

Finally, we propose to utilize the existing capabilities of Jira for the attribute **Access control**. Jira uses role-based access control (RBAC) to manage access to projects and information within the platform. The access control in the system of IoTCorp is sufficient and does not need to be altered.

#### 4.4.3 Summary

This chapter presents the current testing system at IoTCorp, outlining its structure and limitations, thereby establishing the need for the proposed architecture. The existing system utilizes Jira, a project management tool, for test case management and execution. Manual testing is employed, with test cases organized under labels corresponding to requirements. However, the focus remains on general security topics rather than adherence to specific compliance standards. Notably, only one team currently leverages the system. For test execution, each case is marked as passed, failed, or not applicable, while the comprehensive reporting capabilities of the platform remain largely underutilized.

Several key limitations hinder the effectiveness of the current system. Firstly, the database solely encompasses test cases for compliance with the IEC 62443-4-2 standard, neglecting other relevant standards like IEC 62443-3-3 and EN 303 645. Secondly, pinpointing specific non-compliant requirements upon test failure proves challenging due to the indirect linking of requirements to test sets through labels. Furthermore, the system only provides reference codes for requirements, hindering accessibility and potentially compromising oversight as the full context remains unavailable. Lastly, the inability to group requirements by standard or security level makes it difficult to assess a product's compliance with specific regulations. These limitations collectively demand the development of a new architecture to enhance efficiency, collaboration, and ultimately, improving compliance testing within the organization.

The proposed centralized test repository architecture aims to address limitations and fulfil key attributes. It focuses on enhanced information availability by encompassing a broader range of security standards and incorporating detailed requirement information and security level/classification integration through labels. Improved presentation is achieved by grouping requirements by standard and establishing clear links between test cases and requirements. Leveraging the existing Jira platform offers cost-effectiveness and familiarity within the organization. Additionally, the architecture facilitates efficient non-compliance identification and generation of compliance reports. Finally, it utilizes Jira's existing access control system for secure information access. This proposed architecture, designed to address limitations and foster knowledge sharing.

To incorporate the developed test repository for RQ2 into the system of IoTCorp, manual population from an Excel spreadsheet into the Jira platform is required. This manual process ensures accurate data transfer and classification due to the specific formatting requirements of Jira and the need for expert interpretation. Following a defined sequence, the population process involves creating "Requirement" and "Test Case" categories with specific attributes like titles, descriptions, labels, and steps. This allows the populated Jira repository to serve as a centralized and comprehensive resource for all testing needs at IoTCorp.

## 5 Development of the Test Repository

This chapter addresses RQ2 (formulated in Chapter 1.3) by developing the test repository which facilitates compliance assessments against various security standards. We first detail the structure of the repository in Chapter 5.1, after which we highlight the foundation for the test cases in Chapter 5.2, and the techniques and dependencies of the test cases in Chapter 5.3. Furthermore, we discuss the distribution of the test cases over the standards in Chapter 5.4. To close this chapter, we address RQ3 by discussing the integration of the test repository into the architecture in Chapter 5.5.

### 5.1 Structure of the Test Repository

The repository of test cases to the requirements of IEC 62443-4-2, IEC 6244-3-3, and EN 303 645 are presented in Appendix B in Table 25, Table 26, and Table 27 respectively.

Additionally, for enhancing user readability, an Excel file containing the repository is published on GitHub under Kes-G/Master-Thesis [37]. Table 22 illustrates one of the requirements of EN 303 645 within the repository and is a sample of the requirements in Table 27.

Requirement Source	Requirement Text	Test Case(s)
Provision 5.5-8	The manufacturer should follow secure management processes for critical security parameters that relate to the device.	Verify secure management processes (e.g. secure key management, firmware updates, boot process, and password management) are followed for critical parameters that relate to the device.

**Table 22:** Columns of the EN 303 645 mapping in the repository.

It is important to note that the requirement texts for ETSI EN 303 645 [2] are publicly available and therefore included in the repository. Conversely, due to copyright restrictions, the requirement texts for IEC 62443 are not included. Therefore, we depict the Requirement Source for those requirements, as illustrated in Table 23.

Requirement Source	Test Case(s)
IEC 62443-4-2 CR 1.7	Verify password complexity policies are configurable and enforceable.
	Verify that, in case a credential is a password, its minimum length is 6 characters.

**Table 23:** Columns of IEC 62443-4-2 mapping in the repository.

To aid readers who have purchased the IEC 62443 standard, the repository references to the requirement codes (e.g., “IEC 62443-3-3 SR 1”) in the standard, which can be used by readers to easily identify the related requirements. Note that some test cases may be used for multiple requirements, and thus not all test cases are unique. As described in Chapter 3.2.2, the initial data collection process involves gathering the security requirements from the chosen standards and their corresponding mapped test cases. This data is initially compiled in an Excel spreadsheet [90] due to its effectiveness in organizing large datasets.

### 5.2 The Foundation for the Test Cases

Security requirements are the foundation upon which effective test cases are built. As Merkow et. al. [95] highlight in their book, these requirements can be broadly categorized into two types:

- **Generic Requirements:** These are broad, high-level statements that “act as the first line of defense”. They outline general security objectives applicable to various systems, laying the groundwork for security but lacking specific implementation details.
- **Application-Specific Requirements:** These delve deeper and tailor themselves to the unique vulnerabilities and needs of a particular system or application. They build upon and refine generic requirements, providing concrete instructions on how to achieve security in a specific context.

This distinction between generic and application-specific requirements becomes evident when examining the two security standards and their corresponding test cases. For instance, Provision 5.5-1 of EN 303 645 states “*The consumer IoT device shall use best practice cryptography to communicate securely*”. This requirement emphasizes best practices, which is a hallmark of generic requirements. One of the test cases mapped to this requirement, “*Verify TLS 1.2 or TLS 1.3 is used for encryption over network for HTTPS*”, also references best practices, demonstrating the alignment between the requirement and the test case.

Conversely, IEC 62443-4-2 NDR 1.6 states, “*A network device supporting wireless access management shall provide the capability to identify and authenticate all users (humans, software processes or devices) engaged in wireless communication.*”. This is an application-specific requirement as it specifies the application context and types of users involved. The corresponding test cases are application-specific as well. For example, one corresponding test case states “*In case of using ZigBee, verify all communication is encrypted with the Network (NWK) key.*”. This test case directly mentions specific technology, in this case ZigBee.

### 5.3 Test Cases Techniques and Dependencies

Each test case within the repository defines an action or verification statement, outlining what needs to be tested. These test cases employ two techniques to ensure comprehensive coverage:

- **Positive Testing** [96]: This focuses on verifying the expected functionalities of the system under normal operating conditions. In the context of firmware updates, a positive test case could be “*Verify that any update to the executables and firmware results in a recalculation and update of this hash.*”
- **Negative Testing** [96]: This deliberately introduces unexpected inputs or conditions to identify potential weaknesses. An example of a test case is “*Verify inputs exceeding the field length are disallowed in user input fields*”.

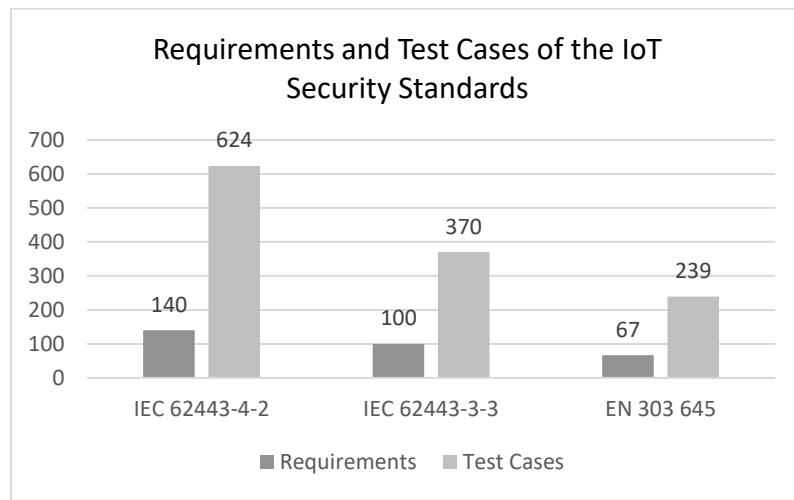
Furthermore, test cases can be either **dependent** or **independent** [97]. Independent test cases, such as “*Verify secure management processes (e.g. secure key management, firmware updates, boot process, and password management) are followed for critical parameters that relate to the device.*” can be executed individually. Dependent test cases require another test case to be completed first. An example of two dependent test cases are:

- **Test Case 1:** “*Enumerate all password storage locations, including text files, databases, and binary objects.*”
- **Test Case 2:** “*Verify that the file, database, or object that is used to maintain passwords is only write-able by the application.*”

These test cases exhibit a dependency relationship. Test Case 1 acts as a prerequisite for Test Case 2. Test Case 1 focuses on identifying password storage locations, while Test Case 2 utilizes this information to verify write permissions on the specific identified location. Without knowing the exact location from Test Case 1, Test Case 2 wouldn't be able to target the correct storage unit. This dependency is crucial because it ensures a comprehensive security evaluation. By first identifying password storage locations, the testing process can then verify that write access to these locations is restricted, a critical security measure.

#### 5.4 Distribution of the Test Cases

Figure 10 shows the distribution of requirements and their corresponding test cases across the three security standards included in this thesis. For example, the left graphs show that the IEC 62443-4-2 has 140 requirements and has 624 test cases mapped to those requirements.



**Figure 10:** Requirements and Test Cases of IoT Security Standards.

The figure also indicates some interesting findings. We note that the IEC 62443 standard translates into 140 requirements, while matching those to nearly 5 times more test cases. This suggests a more granular approach to testing for this standard, ensuring comprehensive coverage of its requirements, which is aligned with the nature of the standard, which focuses on component security. The ratio of requirements to test cases is more similar for IEC 62443-3-3 and EN 303 645.

#### 5.5 Integration of the Test Repository into the Architecture

In this chapter we address RQ3 and detail the transfer of requirements and their associated test cases into the new Jira Repository designed in Chapter 4.

The manual data transfer process from the Excel spreadsheet to Jira was completed successfully. All requirements and their associated test cases are now accessible within the centralized Jira platform. This centralized location simplifies access and retrieval of testing information for users involved in the compliance assessment process. The creation of dedicated "Requirement" and "Test Case" "Issue Types" within Jira facilitates structured data management. Information associated with each requirement (title, description, labels) and test case (title, steps) is organized and easily searchable within the platform. This structure enhances the overall clarity and usability of the repository for testers.

#### 5.6 Summary

This chapter details the development of a test repository to assess compliance with various IoT security standards. The repository stores test cases mapped to requirements from three standards (IEC 62443-4-2, IEC 62443-3-3, and EN 303 645) and is available as an Excel file [37] and in Table 25, Table 26, and Table 27 in Appendix B. Security requirements form the foundation for these test cases, categorized as generic (broad principles) or specific to the application's needs. Positive testing (verifying normal functionality) and negative testing (identifying weaknesses) techniques are employed. Test cases can be independent or dependent on each other, ensuring comprehensive security evaluation. The chapter analyzes the distribution of requirements and test cases across the standards, revealing a more granular testing approach for IEC 62443-4-2. Finally, it details the integration of requirements and test cases into a centralized Jira repository for easier access and management during compliance assessments.

## 6 Discussion

This chapter discusses the architecture to address RQ1 (Chapter 6.1), the test repository to address RQ2 (Chapter 6.2), and the integration to address RQ3 (Chapter 6.3). Please note that the limitations of this thesis are discussed in Chapter 7.

### 6.1 Discussion on the Architecture

This chapter summarizes the key functionalities and implications (Chapters 6.1.1 and 6.1.2) of a centralized test repository architecture designed to address the limitations identified in the case study context (Chapter 4). As explained in Chapter 3.2.1, the attributes are selected according to the principles of Norman [87]. The proposed attributes follow these principles as follows:

- **Information Fields:** The attribute Information Fields aligns with Norman's principle "Use both knowledge in the world and knowledge in the head." As it incorporates "Requirement" as an issue type, which is familiar to the employees in the compliance testing teams. It also includes labels for Security Levels and classification, which leverages the existing knowledge of the security standards.
- **Presentation:** This attribute aligns with the principle "Make things visible." as the grouping of the requirements by using clear labels makes it easier to find relevant information. The proposed architecture also established a clear link between test cases and requirements, which provides a well-structured overview.
- **Platform:** The principle "Get the mappings right." is applied by the decision to leverage the existing Jira platform, which users are already familiar with. This reduces the learning curve and makes the system more intuitive for the compliance testers.
- **Configurations:** The principle "Simply the structure of tasks." is applied as the proposed architecture enables easy identification of non-compliant requirements based on failed tests. Comprehensive compliance reports including both product and requirement perspectives simplify the tasks of understanding test results.
- **Access Control:** The attribute Access Control builds upon the principle "Exploit the power of constraints, both natural and artificial.". The architecture uses Jira's existing RBAC, ensuring data security and restricting unauthorized actions. This aligns with the principle of using constraints to promote user safety.

The architecture, addressing research question 1 (RQ1), offers the following functionalities that enhance testing in the following ways:

- **Multi-Standard Compliance Testing:** Supports multiple security standards by including pre-defined test cases for IEC 62443-4-2, IEC 62443-3-3, and EN 303 645
- **Traceability Mapping:** Provides clear traceability mapping between individual test cases and their corresponding requirements from each standard.
- **Requirement-Based Search:** Enables filtering and searching by requirement to easily identify associated test cases.
- **Compliance Reporting:** Generates compliance assessment reports based on test case execution results. These reports can highlight areas of compliance and identify potential vulnerabilities.
- **Compliance Visualization Dashboard:** Provides a centralized dashboard for visualizing compliance progress across different standards and devices under assessment.

These functionalities address the case study organization's needs for broader standard coverage, improved traceability, and easier access to security compliance insights. Having the architecture in place, the security specialist in the case study organization could leverage the existing Jira platform with its access controls, which in turn promotes cost-effectiveness and efficient implementation.

### 6.1.1 Implications for Practice

The new centralized test repository offers several advantages that enhance compliance testing practices for IoT devices. These are discussed as follows:

- **Broader Standard Coverage:** The scope of the system is expanded as the new repository encompasses a broader range of security standards, including IEC 62443-3-3 and EN 303 645, in addition to the existing focus on IEC 62443-4-2.
- **Improved Traceability:** It creates a clear link between test cases and specific requirements, which facilitates identifying non-compliant areas and simplifies compliance reporting.
- **Increased Accessibility:** The inclusion of full text requirements and grouping by standard enhances accessibility and information organization.
- **Comprehensive Reporting:** The new architecture leverages Jira's functionalities to generate comprehensive compliance reports, providing both product-centric and requirement-specific insights.

Furthermore, the architecture, as implemented in the case study organization, presents opportunities for integration with existing tools and systems, which in turn further helps streamlining the compliance process:

- **Issue Tracking System:** Integrating with a security issue tracking system can streamline the reporting and management of issues identified during testing by establishing bi-directional communication channels for automatic creation of updates within the system. For instance, if a failing test case is detected during testing with the test repository, it could create a defect report in an issue tracking system. This report facilitates issue diagnosis for developers, who can solve the issue in the product and update the report in the issue tracking system. This information is then reflected in the test repository. Such an integration streamlines communication and collaboration for fixing security issues.
- **Test Automation Tools:** Integration with test automation tools could facilitate the execution and management of automated tests by triggering test execution and capturing results into the repository.

### 6.1.2 Implications for Research

The proposed architecture requires development and maintenance efforts to ensure it reflects new security vulnerabilities, best practices, and updated standards and regulations. However, it offers a cost-effective solution for organizations to maintain security according to best practice. The architecture's centralized approach offers several advantages from a scientific perspective:

- **Standardized Architecture:** The proposed architecture provides a well-defined model for centralized test repositories, potentially serving as a reference point for future research and development efforts in this domain. Researchers interested in follow-up empirical studies might start from the model proposed in this thesis and adapt it and possibly refine it in other contexts.
- **Integration Potential:** This master thesis research highlights the value of integrating test repositories with existing tools and systems for a more holistic approach to compliance testing. This can prompt further research on effective integration strategies to optimize the entire compliance testing process. Such research is necessary and useful if the community as a whole would like to come up with end-to-end solutions for which evidence would exist to inform practitioners on what solution to employ in what security certification context.
- **Cost-Effectiveness Analysis:** This research contributes to the discussion of cost-effectiveness in maintaining compliance testing infrastructure. The repository centralizes best practices, eliminating redundancy and streamlining maintenance efforts. This opens doors for further research on cost-benefit analysis of centralized repositories compared to traditional testing methods. Ideally, convincing business cases are prerequisites for practitioners to make investments in technology solutions that enhance security and compliance certifications, and future research on cost-benefit analysis would be instrumental to generating empirical evidence for this.



## 6.2 Discussion of the Test Repository

This section discusses the key findings of the research in relation to RQ2 and explores the implications of the developed test repository for both practice (Chapter 6.2.1) and the scientific community (Chapter 6.2.2). The development of the test repository directly addresses RQ2: identifying test cases for compliance with the security standards EN 303 645, IEC 62443-3-3, and IEC 62443-4-2.

The repository serves as a centralized resource, offering a comprehensive set of test cases mapped to each requirement within the chosen standards (Appendix B). This mapping ensures thorough coverage and simplifies the compliance assessment process for practitioners.

Furthermore, the test cases within the repository leverage both positive and negative testing techniques. Positive testing verifies expected functionalities under normal conditions, while negative testing deliberately introduces unexpected inputs to identify potential vulnerabilities. Both positive and negative testing contribute to a more rigorous evaluation process. This builds confidence that the system can handle both typical and atypical situations, leading to a more robust and secure product.

An additional feature of the repository is the distinction between independent and dependent test cases. The difference between independent and dependent test cases helps gather information in stages, making the testing process faster and more reliable. Independent test cases can be run on their own without needing any extra information beyond the settings already defined for them. This allows for tests to be run at the same time on different devices, making the best use of available resources. For example, a test that checks if a device can resist repeated login attempts (brute-force attack) might only need pre-defined limits on the number of allowed attempts. This independent test can be run alongside another independent test that verifies if the device encrypts sensitive information when it's not being used (data at rest), because the second test doesn't need information about login attempts. On the other hand, dependent test cases need to be run in a specific order to ensure a thorough and logical security evaluation. This controlled order allows for information to be collected in stages. The results of one test case can influence what information needs to be collected for the next dependent test case. For instance, a test that checks secure remote access to a device might rely on a successful vulnerability scan test being run first. The vulnerability scan test would identify potential weaknesses in the device's security that could be used to gain unauthorized access. This information about vulnerabilities is then available for the dependent remote access test case, allowing it to focus on exploiting those specific weaknesses during the remote access attempt. This eliminates the need to repeat the vulnerability scan and makes the testing process more efficient.

The distribution of test cases across the three security standards reveals an interesting finding. The IEC 62443-4-2 standard exhibits a significantly higher ratio of test cases to requirements compared to the other standards. This suggests a more granular testing approach for IEC 62443-4-2, which aligns with its focus on component security and the need for in-depth verification of individual components within a device.

### 6.2.1 Implications for Practice

The developed test repository has multiple practical implications. Below we reflect on those by focusing on two key advantages that the repository brings to those practitioners involved in compliance testing of IoT devices:

- **Reduced Risk and Enhanced Efficiency:** The use of standardized test cases within the repository reduces the likelihood of errors and inconsistencies that might occur during the testing process. The repository provides clear instructions for each test case, ensuring consistency and reducing the chance of misinterpreting requirements.
- **Internal Compliance Testing and Gap Identification:** The clear mapping of test cases to specific requirements within the standards allows testers to identify potential gaps in coverage and ensure thorough testing. More importantly, the repository offers implementation guidance that is currently not available elsewhere. This guidance empowers organizations to conduct internal compliance testing before seeking external

certification. This proactive approach allows for early identification and rectification of security vulnerabilities, significantly enhancing the overall security posture of their products.

### 6.2.2 Implications for Research and Education

The implementation of the test repository and the experiences of our case study organization offer valuable insights for research and education. We now consider the potential for adapting the repository to contexts beyond the specific case study. In other words, could the repository be useful for other organizations working with similar but different security needs? Exploring this question has implications for future research focused on improving the generalizability of the proposed repository. Our reflection concerns contexts that differ in terms of organizational aspects, of devices being used, and of standards targeted for certification.

- **Generalizability regarding the use of the repository in contexts beyond the organization for which it was originally created:** We reflect on the extent to which the developed approach for creating and maintaining test cases could be adapted for other IoT device types and compliance standards. This is an important generalizability threat [85] in any research context where a single case study approach has been adopted as the basis for the research process followed. We think that, despite the evaluation of our solution proposal in the context of one organization, it might well be possible for our solution to be useful to other similar but different contexts. For example, to contexts of other IoT device manufacturing organizations which adopt the same standards as those in this thesis, have similar development practices, follow similar work processes and set similar product-related goals. We think this might be possible based on the argumentation of research methodologists [98] according to which it might be possible to observe similar phenomena in other organizational contexts that share similar contextual characteristics. To this end, more case study driven research could and should be done to shed light into the extent to which the findings of this work are transferable to other similar but different organizational contexts.
- **Generalizability regarding the use of devices different from those included in our case study organization:** The underlying principles of the repository can potentially be applied to other types of embedded or connected devices beyond just IoT devices. The repository structure can be adapted to accommodate the specific functionalities and testing needs of these devices. We consider this an important line of research for the future, as it will lead to better understanding of the adaptability of the approach proposed in this thesis.
- **Generalizability regarding the use of standards different from those included in this thesis:** The modular structure of the repository allows for easy expansion to include additional security requirements as they emerge. For instance, the core functionalities could be adapted to different compliance needs and standards by incorporating new test cases associated with these standards. Similarly, new compliance standards can be mapped to existing ones to identify reusable test cases, while creating new ones for non-overlapping areas. The approach proposed in this thesis seems a viable one to promote efficiency and reduces the cost of developing new test cases. More research is however needed to build up the necessary extensions and the empirical evidence demonstrating the fit.

Finally, the work on the repository has also some implications for teaching of security courses in Computer Science schools. Cybersecurity courses can leverage the repository's standardized test cases to illustrate the practical application of security testing principles. By linking specific tests to corresponding security standards, educators can provide a more concrete understanding of how testing practices ensure compliance.

### 6.3 Discussion of the Repository Integration

This section addresses RQ3 and discusses the successful integration of the test repository into the Jira platform. The data transfer process migrated all requirements and their linked test cases from the

initial Excel spreadsheet, establishing a centralized repository within Jira. As we have seen in Chapter 6.1.1 this readily accessible platform offers significant benefits for testers involved in compliance assessment activities.

Furthermore, the creation of dedicated "Requirement" and "Test Case" "Issue Types" within Jira fosters structured data management. This, in turn, made it possible for the information pertaining to each requirement (title, description, labels) and test case (title, steps) to become now well-organized and searchable within the platform. We consider it important as this structured format enhances the overall clarity and usability of the test repository for testers conducting compliance assessments.

#### 6.4 Summary

This chapter explores the contributions of this thesis by outlining a centralized test repository designed to address limitations in compliance testing of Internet of Things (IoT) devices. The architecture offers functionalities for multi-standard compliance testing, traceability mapping, requirement-based search, compliance reporting, and a visualization dashboard. A test repository for three security standards (IEC 62443-4-2, IEC 62443-3-3, and EN 303 645) was implemented, utilizing positive and negative testing techniques and distinguishing between independent and dependent test cases. An interesting finding is the higher ratio of test cases to requirements in the IEC 62443-4-2 standard, suggesting a more granular testing approach for component security. The research holds implications for practitioners, researchers, and educators. Practitioners benefit from reduced risk and enhanced efficiency through standardized test cases, along with the ability to conduct internal compliance testing before external certification. Researchers can explore the generalizability of the repository to different contexts, analyze cost-effectiveness compared to traditional methods, and investigate integration potential with existing tools. Finally, educators can leverage the repository's standardized test cases to illustrate practical applications of security testing principles. Overall, this thesis contributes to the field of IoT security by proposing a centralized test repository that can enhance compliance testing practices and improve the overall security posture of IoT devices.

## 7 Conclusion, Limitations, and Future Work

This chapter discusses the conclusion (Chapter 7.1) of the research and its limitations (Chapter 7.2). It also explores potential avenues for future work (Chapter 7.3) that can build upon the foundation established in this thesis.

### 7.1 Conclusion

This research proposed a solution that addressed the challenge of achieving efficient and effective compliance testing for IoT security standards. The solution includes an architecture for a centralized test repository and an implementation of this architecture in a real-world organizational context. The proposed architecture for a centralized test repository, as outlined in the case study context of one particular IoT manufacturing organization, directly addressed the limitations experienced in this organization, by providing clear traceability between test cases and corresponding security requirements. The proposed architecture and its implementation were empirically evaluated from the perspective of engineers working in the field. The perception-based evaluation with the help of the participating practitioners from the case study organization let us conclude that the solution is promising and solves the problems for which it was supposed to solve. It was found that the traceability between test cases and corresponding security requirements simplifies the process of pinpointing non-compliant areas and facilitates streamlined reporting that offers valuable insights into both product-level and requirement-level compliance.

Furthermore, this research offers a comprehensive set of test cases meticulously mapped to the specific requirements of the chosen IoT security standards. By leveraging this centralized test repository, manufacturers can significantly reduce uncertainties associated with meeting these standards, streamline compliance testing processes, and ultimately enhance the overall security posture of their IoT devices.

The potential impact of this research extends beyond the immediate benefits for IoT manufacturers. By promoting secure development practices within the IoT industry, this work contributes to a safer and more secure IoT ecosystem for everyone.

In conclusion, this research made a valuable step towards a more secure IoT landscape by providing a comprehensive framework for testing and promoting secure development practices. The proposed centralized test repository empowers manufacturers to navigate the complexities of IoT security compliance more efficiently and effectively, paving the way for a safer and more trustworthy IoT ecosystem. We encourage manufacturers and standardization bodies to consider the potential of this approach in furthering robust IoT security practices.

### 7.2 Limitations

This research acknowledges several limitations that are important to consider when interpreting the findings and their broader applicability.

1. **Focus on a Single Case Study (Chapter 1.4):** As the present research primarily focuses on a single case study company, IoT Corp, this approach enables in-depth exploration and development of a solution tailored to their specific needs. The insights gained are likely to be relevant to other organizations with similar characteristics and operating within the same industry [9], particularly those facing comparable challenges in ensuring IoT security compliance. It is also noteworthy that the test cases prioritize covering the core aspects of security requirements, providing a foundation for broader application.
2. **Test Case Selection:** The test cases included in the mapping (Chapter 5) are selected to ensure they adequately test compliance to the requirements outlined in the IoT security standards. This selection process prioritizes covering the fundamental aspects of each requirement. It is important to acknowledge that additional test cases might exist that could delve deeper into specific nuances of certain requirements. As an example, if a requirement states that a device must implement encryption for all data transmissions, a test case that would test this requirement sufficiently is to verify the device uses recognized and secure encryption

algorithms. An additional test case would be to test the device's behavior under various network conditions, such as high latency, to assess the impact on the effectiveness of the encryption. However, this research prioritizes establishing a solid foundation for compliance assessment by gathering a sufficient number of well-defined test cases, while acknowledging the potential for further exploration and customization based on specific organizational needs and risk profiles.

3. **Standards Selection:** The research specifically focuses on the ETSI EN 303 645, IEC 62443-3-3, and IEC 62443-4-2 standards, which represent crucial aspects of IoT security. While other relevant standards exist, they are not included within the scope of this research due to their specific focus or nature, as explained in Chapter 2.1. An example of such an excluded standard is ISO/IEC 27001 [99], which focuses on broader information management practices and is not specifically tailored to the challenges of IoT security. Further exploration of these excluded standards and their potential integration into the proposed framework could be pursued in future work.
4. **Validation Process:** This thesis employs a structured validation of the repository to ensure completeness and adequacy of the test cases. While specific details are outlined in Chapter 3.2.2, it is important to acknowledge that limitations exist within any validation approach. Future interactions could explore strategies to further enhance the generalizability of the findings.
5. **Employee Capacity:** The capacity of employees to conduct security testing remains crucial for adaptation of a new architecture for testing. While the architecture streamlines testing, employee training and resource allocation are essential for its successful adoption.
6. **Permission Management:** Implementing role-based access control (RBAC) is crucial after deploying the test repository. This mitigates risks associated with unauthorized access to sensitive information, ensuring only authorized personnel can view security-critical details like product status and failing test cases. Granular permission management, ideally per product or department, adheres to the principle of least privilege, granting users only the minimum access level required for their role.

### 7.3 Future Work

This thesis opens up multiple opportunities for future work. New lines of research can be pursued in several areas to expand upon the test repository presented in this master thesis:

- **Investigate Test Case Automation:** An investigation of the options for automating the test cases seems a worthwhile endeavor. This investigation would include the identification of which test cases are possible to automate, and following, which security tools can be utilized for this. Specifically for IoTCorp, future work on automated testing includes the examination of which security tools are already in use by IoTCorp and how they contribute to the test cases. The results of this investigation can be used to determine whether additional tools can be integrated with the test repository to increase efficiency and reduce human error.
- **Feasibility of Automated Testing:** Research can be performed to investigate the feasibility of automating test cases by writing scripts, which may further improve the efficiency and effectiveness of the testing process. While automation can increase efficiency and reduce human error, it is crucial to evaluate whether the time and effort involved in creating these scripts outweigh the benefits of manual testing. Test cases that an organization may want to continuously test may benefit from automation, but test cases that are more of a 'one-time check' may not require a script and can be tested annually during an internal penetration test.
- **Expand Security Standard Coverage:** The test repository can be expanded to include additional IoT security standards, such as NIST 800-53 and ISO 27001 [42]. This will further enhance the value of the test repository to organizations striving to comply with multiple IoT security standards. Collaboration with industry experts and regulatory bodies can be pursued

to ensure that the test repository remains up to date with the latest security standards and best practices.

- **Improve the Generalizability:** The proposed solution demonstrated its viability and usefulness in the context of one specific organization. Our reflection on the transferability of the observations, experiences, and perceptions of the implementation of the solution to other contexts brought us to suggest lines for more empirical research. For example, it is important to evaluate the extent to which our solution is adaptable to the context of using other IoT devices and other standards.

## 8 References

- [1] IEC, "Understanding IEC 62443," 26 02 2021. [Online]. Available: <https://www.iec.ch/blog/understanding-iec-62443>. [Accessed 12 09 2022].
- [2] Dekra, "Cybersecurity ETSI EN 303 645," [Online]. Available: <https://www.dekra.com/media/cybersecurity-etsi-en-303645-dekra.pdf>. [Accessed 27 09 2022].
- [3] IDC, "Worldwide IDC Global Datasphere IoT Device Install Base and Data Generated Forecast, 2022-2026," IDC, 09 09 2022. [Online]. Available: <https://www.idc.com/getdoc.jsp?containerId=US49631322&pageType=PRINTFRIENDLY>. [Accessed 17 02 2023].
- [4] M. J. Paul Brous, "Effects of the Internet of Things (IoT): A systematic review of the benefits and risks," in *The 2015 International Conference on Electronic Business*, Hong Kong, 2015.
- [5] L. K. Ramasamy and S. Kadry, "Internet of things (IoT)," in *Blockchain in the Industrial Internet of Things*, IOP, 2021, pp. 1-16.
- [6] International Society of Automation, "IEC 62443 series of standards," ISA, [Online]. Available: <https://www.isa.org/standards-and-publications/isa-standards/isa-iec-62443-series-of-standards>. [Accessed 13 09 2022].
- [7] European Telecommunications Standards Institute, "Final draft ETSI Cyber Security for Consumer Internet of Things: Requirements Baseline," 06 2020. [Online]. Available: [https://www.etsi.org/deliver/etsi\\_en/303600\\_303699/303645/02.01.01\\_60/en\\_303645v020101p.pdf](https://www.etsi.org/deliver/etsi_en/303600_303699/303645/02.01.01_60/en_303645v020101p.pdf). [Accessed 13 09 2022].
- [8] R. Yin, "Case Study Research," in *Design and Methods 4th edition*, Thousand Oaks, Sage Publications, 2012, p. 240.
- [9] R. Wieringa, "Case study research in information systems engineering," University of Twente, Enschede, 2013.
- [10] Philips Hue, "Explore Hue, how it works," [Online]. Available: <https://www.philips-hue.com/en-gb/explore-hue/how-it-works>. [Accessed 16 02 2023].
- [11] R. Ahmad and I. Alsmadi, "Machine learning approaches to IoT security: A systematic literature review," *Internet of Things*, vol. 14, 2021.
- [12] S. Notra, M. Siddiqi, H. H. Gharakheili, V. Sivaraman and R. Boreli, "Notra, S., Siddiqi, M., Gharakheili, H. H., Sivaraman, V., & Boreli, R. An Experimental Study of Security and Privacy Risks with Emerging Household Appliances," in *IEEE Conference on Communications and Network Security*, 2014.
- [13] OWASP, "OWASP Testing guide v4," [Online]. Available: [https://owasp.org/www-project-web-security-testing-guide/assets/archive/OWASP\\_Testing\\_Guide\\_v4.pdf](https://owasp.org/www-project-web-security-testing-guide/assets/archive/OWASP_Testing_Guide_v4.pdf). [Accessed 17 02 2023].

- [14] Andres Froehlich, "What are the biggest hardware security threats," Techtarget, [Online]. Available: <https://www.techtarget.com/searchsecurity/tip/What-are-the-biggest-hardware-security-threats>. [Accessed 19 10 2022].
- [15] The Guardian, „Ring hackers reportedly watching and talking to strangers via in-home cameras," The Guardian, 13 December 2019. [Online]. Available: <https://www.theguardian.com/technology/2019/dec/13/ring-hackers-reportedly-watching-talking-strangers-in-home-cameras>. [Geopend 02 02 2024].
- [16] C. Musonda, M. Kameba, M. Nyirenda and J. Phiri, "Security, Privacy and Integrity in Internet of Things," in *ICTSZ international conference in ICTs*, Lusaka, Zambia, 2018.
- [17] A. Lindberg, "Penetration testing of current smart thermostats," KTH Royal institute of technology, Stockholm, Sweden, 2023.
- [18] Kaspersky, "Reputation and cybersecurity: from risk to opportunity, and cyber-pride," 2020. [Online]. Available: [https://media.kaspersky.com/pdf/b2b/KES\\_cloud\\_reputation.pdf](https://media.kaspersky.com/pdf/b2b/KES_cloud_reputation.pdf). [Accessed 13 09 2022].
- [19] G. Belding, "Cost of non-compliance: 8 largest data breach fines and penalties," InfoSec, 20 October 2020. [Online]. Available: <https://resources.infosecinstitute.com/topic/cost-of-non-compliance-8-largest-data-breach-fines-and-penalties/>. [Accessed 13 09 2022].
- [20] European Union, "Directive (EU) 2022/2555 of the European Parliament and of the Council of 14 December 2022 on measures for a high common level of cybersecurity across the Union, amending Regulation (EU) No 910/2014 and Directive (EU) 2018/1972, and repealing Directive (E," 14 December 2022. [Online]. Available: <https://eur-lex.europa.eu/eli/dir/2022/2555>. [Accessed 22 02 2024].
- [21] European Union, "Regulation (EU) 2019/881 of the European Parliament and of the Council of 17 April 2019 on ENISA (the European Union Agency for Cybersecurity) and on information and communications technology cybersecurity certification and repealing Regulation (EU) No 52," 17 April 2019. [Online]. Available: <https://eur-lex.europa.eu/eli/reg/2019/881/oj>. [Accessed 22 02 2024].
- [22] European Union, "Proposal for a Regulation of the European Parliament and of the Council," Council of the European Union, Brussels, Belgium, 2023.
- [23] European Union, "Commission Delegated Regulation (EU) 2022/30 of 29 October 2021 supplementing Directive 2014/53/EU of the European Parliament and of the Council with regard to the application of the essential requirements referred to in Article 3(3), points (d), (e) and," 29 October 2021. [Online]. Available: [https://eur-lex.europa.eu/eli/reg\\_del/2022/30/oj](https://eur-lex.europa.eu/eli/reg_del/2022/30/oj). [Accessed 22 02 2024].
- [24] European Union, "Directive (EU) 2016/1148 of the European Parliament and of the Council of 6 July 2016 concerning measures for a high common level of security of network and information systems across the Union," 6 July 2016. [Online]. Available: <https://eur-lex.europa.eu/legal-content/NL/TXT/?uri=celex%3A32016L1148>. [Accessed 22 02 2024].



- [25] G. Kreukniet, "Leverage IEC 62443 for EU NIS2 Directive compliance," DNV, 2023. [Online]. Available: <https://www.dnv.com/cybersecurity/cyber-insights/leverage-iec-62443-for-eu-nis2-directive-compliance.html>. [Accessed 23 02 2024].
- [26] S. Fluchs, "The EU Cyber Resilience Act," Industrial Cyber, 29 September 2022. [Online]. Available: <https://industrialcyber.co/expert/the-eu-cyber-resilience-act/>. [Accessed 23 02 2024].
- [27] B. Hubert, "The EU's new Cyber Resilience Acts is about to tell us how to code," Berthub, 14 March 2023. [Online]. Available: <https://berthub.eu/articles/posts/eu-cra-secure-coding-solution/>. [Accessed 23 02 2024].
- [28] European Union, "Directive 2014/53/EU of the European Parliament and of the Council of 16 April 2014 on the harmonisation of the laws of the Member States relating to the making available on the market of radio equipment and repealing Directive 1999/5/EC," 16 April 2024. [Online]. Available: <https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=celex%3A32014L0053>. [Accessed 22 02 2024].
- [29] UL solutions, "RED Cyber FAQs," 1 June 2023. [Online]. Available: [https://collateral-library-production.s3.amazonaws.com/uploads/asset\\_file/attachment/56315/RED-Cyber-FAQs-Digital.pdf](https://collateral-library-production.s3.amazonaws.com/uploads/asset_file/attachment/56315/RED-Cyber-FAQs-Digital.pdf). [Accessed 23 02 2024].
- [30] DEKRA, "Product Certification and Marks," [Online]. Available: <https://www.dekra-product-safety.com/en/solutions/certification-marks>. [Accessed 17 02 2023].
- [31] S. M. M. F.-B. T. F. S. H. B. W. Markus Fockel, "Designing and Integrating IEC 62443 Compliant Threat Analysis," *Communications in Computer and Information Science*, vol. 1060, pp. 1-12, 2019.
- [32] T. Payne, "What are the CE Certification Costs?," Sunfire Testing, [Online]. Available: <https://www.sunfiretesting.com/What-Are-the-CE-Certification-Costs/>. [Accessed 23 09 2022].
- [33] European Commission, "Notified Bodies," [Online]. Available: [https://single-market-economy.ec.europa.eu/single-market/goods/building-blocks/notified-bodies\\_en](https://single-market-economy.ec.europa.eu/single-market/goods/building-blocks/notified-bodies_en). [Accessed 17 02 2023].
- [34] Testbytes, "What is Compliance Testing?," [Online]. Available: <https://www.testbytes.net/blog/compliance-testing/>. [Accessed 13 09 2022].
- [35] H. Gupta, "Anti virus," Govt. Polytechnic Panchkula, Panchkula, India, 2020.
- [36] R. Bace and P. Mell, "Intrusion Detection Systems," National Institute of Standards and Technology, Gaithersburg, MD, 2001.
- [37] K. Greuter, "Kes-G/Master-Thesis/Public-version-test-cases-mapping," Github, 2 2 2024. [Online]. Available: <https://github.com/Kes-G/Master-Thesis/blob/main/Public%20version%20test%20cases%20mapping.xlsx>. [Accessed 2 2 2024].

- [38] M. Saunders, L. Philip, A. Thornhill and A. Bristow, "Chapter 4: Understanding research philosophy and approaches to theory development.," in *Research Methods for Business Students*, Harlow, Pearson Education Limited, 2019, pp. 128 -165.
- [39] H. Taherdoost, "Understanding Cybersecurity Frameworks and Information Security Standards—A Review and Comprehensive Overview," *Electronics*, vol. 11, no. 2181, pp. 1-20, 2022.
- [40] Synopsys, "What is UL 2900?," [Online]. Available: <https://www.synopsys.com/glossary/what-is-ul-2900.html>. [Accessed 27 09 2022].
- [41] UL solutions, "FDA Recognizes UL 2900-1 Cybersecurity Standard for Medical Devices," 12 09 2017. [Online]. Available: <https://www.ul.com/news/fda-recognizes-ul-2900-1-cybersecurity-standard-medical-devices>. [Accessed 27 09 2022].
- [42] IT governance, "ISO 27000 Series of Standards," June 2020. [Online]. Available: <https://www.itgovernance.co.uk/iso27000-family>. [Accessed 27 09 2022].
- [43] Bundesamt für Sicherheit in der Informationstechnik, "Federal Office for Information Security," [Online]. Available: [https://www.bsi.bund.de/EN/Home/home\\_node.html](https://www.bsi.bund.de/EN/Home/home_node.html). [Accessed 27 09 2022].
- [44] Security Forum, "Standard of Good Practice for Information Security," 2020. [Online]. Available: <https://www.securityforum.org/solutions-and-insights/standard-of-good-practice-for-information-security-2020/>. [Accessed 27 09 2022].
- [45] ISO, "ISO/SAE 21434:2021 Road Vehicles - Cybersecurity engineering," [Online]. Available: <https://www.iso.org/standard/70918.html>. [Accessed 27 09 2022].
- [46] NIST, "FIPS 140-2," [Online]. Available: <https://csrc.nist.gov/publications/detail/fips/140/2/final>. [Accessed 27 09 2022].
- [47] Secura, "IEC 62443," [Online]. Available: <https://www.secura.com/uploads/factsheets/IEC-62443.pdf>. [Accessed 23 09 2022].
- [48] A. A., "What Is The ISA/IEC 62443 Framework?," Tripwire, 2020. [Online]. Available: <https://www.tripwire.com/state-of-security/regulatory-compliance/isa-iec-62443-framework/>. [Accessed 13 09 2022].
- [49] K. O. Greuter and D. K. Sarmah, "The baseline of global consumer cybersecurity standards for IoT: quality evaluation," *Journal of Cyber Security Technology*, vol. 6, no. 4, pp. 175-200, 2022.
- [50] Beyond Security, "How to Use SAST and DAST to meet ISA/IEC 62443 Compliance," 25 05 2020. [Online]. Available: <https://blog.beyondsecurity.com/isa-iec-62443-security-testing/>. [Accessed 27 09 2022].
- [51] Kiwa, "ETSI EN 303 645: beveiliging IoT consumentenelectronica," [Online]. Available: <http://www.kiwa.com/nl/nl/service/etsi-en-303-645-beveiliging-iot-consumentenelectronica/pdf/>. [Accessed 27 09 2022].

- [52] T. Stewart , “Can we Really Prevent Security Vulnerabilities at the Source?,” in *Proceedings of the 29th European Safety and Reliability Conference (ESREL)*, 2019.
- [53] B. Leander, A. D. Causevic and H. Hansson, “Applicability of the IEC 62443 standard in Industry 4.0 / IIoT,” in *the 14th International Conference*, 2019.
- [54] H. L. Hassani, A. Bahnasse, C. Roland, E. Martin, O. Bouattane and M. El Mehdi Diouri, “Vulnerability and security risk assessment in a IIoT environment in compliance with standard IEC 62443,” in *The 18th International Conference on Mobile Systems and Pervasive Computing*, Leuven, 2021.
- [55] A. Mosteiro-Sanchez, M. Barcelo, J. Astorga and A. Urbieta, “Securing IIoT using Defence-in-Depth: Towards an End-to-End Secure Industry 4.0,” *Journal of Manufacturing Systems*, vol. 57, pp. 367-378, 2022.
- [56] A. M. Shaaban, E. Kristen and C. Schmittner, “Application of IEC 62443 for IoT Components,” *Computer Safety, Reliability, and Security*, vol. 11094, pp. 214-223, 2018.
- [57] M. Kon, “How to define zones and conduits,” ISA, [Online]. Available: <https://gca.isa.org/blog/how-to-define-zones-and-conduits#:~:text=Zone%3A%20consists%20of%20the%20grouping,share%20the%20same%20cybersecurity%20requirements>. [Accessed 13 03 2023].
- [58] M. Fockel, S. Merschjohann, M. Fazal-Baqaie, T. Forder, S. Hausmann and B. Waldeck, “Designing and Integrating IEC 62443 Compliant Threat Analysis,” in *European System, Software & Service Process Improvement & Innovation*, Edinburgh, UK, 2019.
- [59] A. M. Shabaan, S. Chlup, N. El-Araby and C. Schmittner, “Towards Optimized Security Attributes for IoT Devices in Smart Agriculture Based on the IEC 62443 Security Standard,” *Applied Sciences*, vol. 12, no. 11, 2022.
- [60] Kuliktomas, “iec62443verification,” Github, 8 01 2018. [Online]. Available: <https://github.com/kuliktomas/iec62443verification/blob/master/vehiclecloud.tla>. [Accessed 3 10 2022].
- [61] Kernelci, “iec-security,” Github, 1 09 2021. [Online]. Available: <https://github.com/kernelci/iec-security>. [Accessed 3 10 2022].
- [62] M. Puys, J.-P. Krimm and R. Collado, “Towards Cybersecurity Act: A Survey on IoT Evaluation Frameworks,” in *The Fourteenth International Conference on Emerging Security Information, Systems and Technologies*, 2020.
- [63] B. Sereda and J. Jaskolka, “An Evaluation of IoT Security Guidance Documents: A Shared Responsibility Perspective,” in *The 13th International Conference on Ambient Systems, Networks and Technologies (ANT)*, Porto, Portugal, 2022.
- [64] F. Catal, S. Hackel, R. Barakat, A. Rennoch and M. A. Schneider , “Towards a certification scheme for IoT security evaluation,” *INFORMATIK*, 2021.

- [65] S. Fischer, "Internet of Things: A Model for Cybersecurity Standards and the Categorisation of Devices (Doctoral dissertation).," Berlin, 2022.
- [66] P. Meulenhoff, W. Westerhof and S. Langkemper, "Essential requirements for securing consumer IoT devices.," Hogeschool van Amsterdam, Amsterdam, 2020.
- [67] F. Djebbar and K. Nordstrom, "A Comparative Analysis of Industrial Cybersecurity Standards," *IEEE Access*, vol. 11, pp. 85315 - 85332, 2023.
- [68] T. Stewart , „Can we Really Prevent Security Vulnerabilities at the Source?," in *Proceedings of the 29th European Safety and Reliability Conference (ESREL)*, 2019.
- [69] B. Leander, A. D. Causevic en H. Hansson, „Applicability of the IEC 62443 standard in Industry 4.0 / IIoT," in *the 14th International Conference*, 2019.
- [70] H. L. Hassani, A. Bahnasse, C. Roland, E. Martin, O. Bouattane en M. El Mehdi Diouri, „Vulnerability and security risk assessment in a IIoT environment in compliance with standard IEC 62443," in *The 18th International Conference on Mobile Systems and Pervasive Computing*, Leuven, 2021.
- [71] A. Mosteiro-Sanchez, M. Barcelo, J. Astorga en A. Urbieto, „Securing IIoT using Defence-in-Depth: Towards an End-to-End Secure Industry 4.0," *Journal of Manufacturing Systems*, vol. 57, pp. 367-378, 2022.
- [72] A. M. Shaaban, E. Kristen en C. Schmittner, „Application of IEC 62443 for IoT Components," *Computer Safety, Reliability, and Security* , vol. 11094, pp. 214-223, 2018.
- [73] Kuliktomas, „iec62443verification," Github, 8 01 2018. [Online]. Available: <https://github.com/kuliktomas/iec62443verification/blob/master/vehiclecloud.tla>. [Geopend 3 10 2022].
- [74] Kernelci, „iec-security," Github, 1 09 2021. [Online]. Available: <https://github.com/kernelci/iec-security>. [Geopend 3 10 2022].
- [75] M. Puys, J.-P. Krimm en R. Collado, „Towards Cybersecurity Act: A Survey on IoT Evaluation Frameworks," in *The Fourteenth International Conference on Emerging Security Information, Systems and Technologies*, 2020.
- [76] B. Sereda en J. Jaskolka, „An Evaluation of IoT Security Guidance Documents: A Shared Responsibility Perspective," in *The 13th International Conference on Ambient Systems, Networks and Technologies (ANT)*, Porto, Portugal, 2022.
- [77] F. Catal, S. Hackel, R. Barakat, A. Rennoch en M. A. Schneider , „Towards a certification scheme for IoT security evaluation," *INFORMATIK*, 2021.
- [78] S. Fischer, „Internet of Things: A Model for Cybersecurity Standards and the Categorisation of Devices (Doctoral dissertation).," Berlin, 2022.
- [79] P. Meulenhoff, W. Westerhof en S. Langkemper, „Essential requirements for securing consumer IoT devices.," Hogeschool van Amsterdam, Amsterdam, 2020.

- [80] K. O. Greuter en D. K. Sarmah, „The baseline of global consumer cybersecurity standards for IoT: quality evaluation,” *Journal of Cyber Security Technology*, vol. 6, nr. 4, pp. 175-200, 2022.
- [81] F. Djebbar en K. Nordstrom, „A Comparative Analysis of Industrial Cybersecurity Standards,” *IEEE Access*, vol. 11, pp. 85315 - 85332, 2023.
- [82] M. Felderer, M. Buchler, M. Johns, A. Brucker, R. Breu and A. Pretschner, “Security Testing: A Survey,” *Advances in Computers*, vol. 101, pp. 1-51, 2026.
- [83] K. Greuter, “Mapping of Security Testing methods,” 13 01 2023. [Online]. Available: <https://github.com/Kes-G/Master-Thesis>. [Accessed 17 02 2023].
- [84] J. W. Creswell, *Research Design: Qualitative, Quantitative, and Mixed Methods Approaches*, Nebraska, USA: SAGE Publications, 2009.
- [85] R. Wieringa, *Design Science Methodology for Information Systems and Software Engineering*, Heidelberg: Springer Berlin, 2014.
- [86] M. S. Rahman, “The Advantages and Disadvantages of Using Qualitative and Quantitative Approaches and Methods in Language “Testing and Assessment” Research: A Literature Review,” *Journal of Education and Learning*, vol. 6, no. 1, pp. 102 - 112, 2017.
- [87] D. A. Norman, *The Design of Everyday Things*, Basic Books, 1988.
- [88] K. Charmaz, „Grounded theory as an emergent method,” in *Handbook of emergent methods*, The Guilford Press., 2008, pp. 155-170.
- [89] K. Greuter, “Similar Requirements EN 303 645 and IEC 62443-3-3,” Github, 16 June 2020. [Online]. Available: <https://github.com/Kes-G/Bachelor-thesis/blob/master/RQ-1%20Similar%20requirements.xlsx>. [Accessed 2024 02 05].
- [90] Microsoft, “Excel,” Microsoft, [Online]. Available: <https://www.microsoft.com/nl-nl/microsoft-365/excel/?market=nl>. [Accessed 2024 02 29].
- [91] B. Oosterveld, “Linkedin page,” [Online]. Available: <https://www.linkedin.com/in/barbaraosterveld/>. [Accessed 17 02 2023].
- [92] D. Papadopoulos, “LinkedIn page,” [Online]. Available: <https://www.linkedin.com/in/dimitrios-papadopoulos-82508a62/>. [Accessed 17 02 2023].
- [93] Atlassian, “Jira,” Atlassian, [Online]. Available: [https://www.atlassian.com/software/jira?gclid=4e0a7cb1d17d1b5fc59d1971663cf028&gclid=3p.ds&&aceid={aceid}&adposition=&adgroup=1308419274399724&campaign=470096596&creative=&device=c&keyword=jira%20com&matchtype=p&network=o&placement=&ds\\_kids=p74740116894&](https://www.atlassian.com/software/jira?gclid=4e0a7cb1d17d1b5fc59d1971663cf028&gclid=3p.ds&&aceid={aceid}&adposition=&adgroup=1308419274399724&campaign=470096596&creative=&device=c&keyword=jira%20com&matchtype=p&network=o&placement=&ds_kids=p74740116894&). [Accessed 2024 02 05].
- [94] E. Goodman, M. Kuniavsky and A. Moed, *Observing the User Experience*, Waltham: Morgan Kaufmann, 2012.

- [95] M. Merkow and L. Raghavan, *Secure and Resilient Software: Requirements, Test Cases, and Testing Methods*, Boca Raton: Taylor & Francis Group, 2011.
- [96] P. Jorgensen, *Software Testing: A Craftsman's Approach*, CRC Press: Boca Raton, 2021.
- [97] R. Black, E. Van Veenendaal en D. Graham, *Foundations of Software Testing*, London: Cengage Learning, 2019.
- [98] P. Seddon and R. Scheepers, "Towards the improved treatment of generalization of knowledge claims in IS research: drawing general conclusions from samples.," *European Journal of Information Systems*, vol. 21, no. 1, pp. 6-21, 2012.
- [99] International Standardization Organization, *ISO/IEC 27001:2022*, Geneva, Switzerland: ISO, 2022.
- [100] The Guardian, "Ring hackers reportedly watching and talking to strangers via in-home cameras," *The Guardian*, 13 December 2019. [Online]. Available: <https://www.theguardian.com/technology/2019/dec/13/ring-hackers-reportedly-watching-talking-strangers-in-home-cameras>. [Accessed 02 02 2024].
- [101] A. Di Felice and Z. Stambolliu, "Building blocks for a scalable Cyber Resilience Act," *Digital Europe*, Brussel, 2020.
- [102] M. Felderer, M. Buchler, M. Johns, A. Brucker, R. Breu and A. Pretschner, "Security Testing: A Survey," in *Advances in Computers*, Elsevier, 2016, pp. 1-51.
- [103] S. Singh, S. Anand and M. Satyarthi, "A Comprehensive Review of Smart Home Automation Systems.," *Advances in Computer Science and Information Technology*, vol. 10, no. 2, pp. 61-66, 2023.
- [104] I. Burnstein, *Practical Software Testing*, New York: Springer-Verlag, 2003.

## Appendix A

**Table 24:** Security testing methods and the SDLC phase they belong to, along with their description and examples of requirements that they test.

SDLC phase	Testing method	Description	Matching requirement topic
Analysis	SDLC Process Review	Ensure an adequate SDLC is defined where security is inherent at each stage.	Requirements regarding security within SDLC, Security by Design.  <b>Example of a fitting requirement:</b> <i>The manufacturer should follow secure development processes for software deployed on the device.</i>
Analysis	Policy and Standards Review	Test whether appropriate policies, standards, and documentation are in place.	Requirements regarding policies, standards, processes, and documentation.  <b>Example of a fitting requirement:</b> <i>The manufacturer shall publish, in an accessible way that is clear and transparent to the user, the defined support period.</i>
Analysis / Design	Model-Based Security Testing Approach for Web Applications	Test whether the system will violate a security property once given an attack trace for a web-based attack.	Requirements regarding protection against web application attacks using attack traces.  <b>Example of a fitting requirement:</b> <i>When the device is not constrained, it shall have a mechanism available that makes brute force attacks on authentication mechanisms via network interfaces impracticable.</i>
Design	Security Requirements Review	The assumptions made in the security requirements are tested, and it is checked whether there are gaps in the requirements definitions, such as ambiguity	Requirements regarding requirement evaluation of clarity and completeness.  <b>Example of a fitting requirement:</b> <i>The security requirements shall be reviewed by both the design and development parties before implementation.</i>
Design	Design and Architecture Review	Test if the design and architecture enforce the appropriate level of security as defined in the requirements.	Requirements regarding the design of the system and the architecture of the system.  <b>Example of a fitting requirement:</b> <i>The consumer IoT device shall protect the confidentiality of critical security parameters that are communicated via remotely accessible network interfaces.</i>
Design	UML Models Review	Test if the system does not contain weaknesses in the	Requirements regarding the data flow of the application, infrastructure, or access policies.

		UML models that describe the working of the system.	<b>Example of a fitting requirement:</b> <i>When the device is not constrained, it shall have a mechanism available that makes brute force attacks on authentication mechanisms via network interfaces impracticable.</i>
Design	Threat Models Review	Test, using threat scenarios, whether the design and architecture mitigate the threats, accept the threats, or delegate the threats to third parties.	Requirements regarding attacks, security threats, and risk management.  <b>Example of a fitting requirement:</b> <i>Where pre-installed unique per-device passwords are used, these shall be generated with a mechanism that reduces the risk of automated attacks against a class or type of device.</i>
Development	Code Review	Examine the code line-by-line for security defects.	Requirements regarding the code, vulnerabilities, and insecure configurations within the code.  <b>Example of fitting requirement:</b> <i>Where pre-installed unique per-device passwords are used, these shall be generated with a mechanism that reduces the risk of automated attacks against a class or type of device.</i>
Development	Static Application Security Testing	Automatically analyze software component code for insecure configurations or analyze data flow or control flow.	Requirements regarding configurations, data flow, or control flow.  <b>Example of fitting requirement:</b> <i>Authentication mechanisms used to authenticate users against a device shall use best-practice cryptography, appropriate to the properties of the technology, risk, and usage.</i>
Development	Code Walkthrough	Understand the logic and flow of implemented code that makes up the application.	Requirements regarding the flow and (security) logic of the code.  <b>Example of fitting requirement:</b> <i>When the device is not constrained, it shall have a mechanism available that makes brute force attacks on authentication mechanisms via network interfaces impracticable.</i>
Development/Deployment	Unit and System testing	Test whether the unit or system adheres to the set security requirements.	Requirements regarding adherence to security requirements on the component level and system level.  <b>Example of fitting requirement:</b> <i>For constrained devices that cannot have their software updated, the product should be isolable and the hardware replaceable.</i>



Deployment	Penetration Testing	Test the application after deployment to ensure no issues are missed.	Requirements regarding vulnerabilities and attacks that can be simulated using penetration testing.  <b>Example of a fitting requirement:</b> <i>When the device is not constrained, it shall have a mechanism available that makes brute force attacks on authentication mechanisms via network interfaces impracticable.</i>
Deployment	Configuration Management Review	Test how the infrastructure was deployed and secured, and whether there are still default configurations that can be vulnerable to exploitation.	Requirements regarding configurations and default values.  <b>Example of a fitting requirement:</b> <i>Where passwords are used and, in any state, other than the factory default, all consumer IoT device passwords shall be unique per device or defined by the user.</i>
Deployment	Acceptance Test	Test whether the application adheres to the security goals using user stories containing security requirements.	Requirements regarding the testing of security requirements from a user point of view.  <b>Example of a fitting requirement:</b> <i>An update shall be simple for the user to apply.</i>
Deployment/Maintenance	Vulnerability Scanning	Identify security issues in the system using tools that input a set of pre-defined attack payloads and analyze the system's output.	Requirements regarding protection against attacks and identification of vulnerabilities that can be detected using an attack payload.  <b>Example of a fitting requirement:</b> <i>If telemetry data is collected from consumer IoT devices and services, such as usage and measurement data, it should be examined for security anomalies.</i>
Deployment/Maintenance	Dynamic Taint Analysis	Track the flow of sensitive information during the execution of the program.	Requirements regarding sensitive information flow.  <b>Example of a fitting requirement:</b> <i>The confidentiality of sensitive personal data communicated between the device and associated services shall be protected, with cryptography appropriate to the properties of the technology and usage.</i>
Deployment/Maintenance	Fuzzing	Test if the system crashes or behaves unexpectedly when given random data as input. This finds software defects or vulnerabilities.	Requirements regarding input testing and input sanitation.  <b>Example of a fitting requirement:</b> <i>The consumer IoT device software shall validate data input via user interfaces or transferred via Application Programming Interfaces (APIs) or between networks in services and devices.</i>

Maintenance	Change Verification	After every change is approved and tested, test whether the level of security has not been affected by the change.	Requirements regarding change management and security evaluations after implementation.  <b>Example of a fitting requirement:</b> <i>After a security-relevant change, the security risk level of the system or component altered must be re-evaluated.</i>
Maintenance	Health Checks	Monthly or quarterly health checks need to be performed on the application and infrastructure to test if no new security risks have been introduced and the level of security is still intact.	Requirements regarding processes for regular security evaluations such as vulnerability scanning.  <b>Example of a fitting requirement:</b> <i>Manufacturers should continually monitor for, identify, and rectify security vulnerabilities within products and services they sell, produce, have produced, and services they operate during the defined support period.</i>
Maintenance	Operational Management Reviews	Test how the operational side of both the application and infrastructure is managed.	Requirements regarding operational security processes such as monitoring, patch management audits, and logs.  <b>Example of a fitting requirement:</b> <i>Manufacturers should continually monitor for, identify, and rectify security vulnerabilities within products and services they sell, produce, have produced, and services they operate during the defined support period.</i>
Maintenance	Regression Tests	After every change is approved and tested, test if the change caused any unexpected side effects.	Requirements regarding change management and side effects of implementations.  <b>Example of a fitting requirement:</b> <i>After a security-relevant change, the security risk level of the systems or components related to the altered system or component must be re-evaluated.</i>

## Appendix B

**Table 25:** Test cases mapped to requirements of IEC 62443-4-2.

Requirement Source	Test Case(s)
IEC 62443-4-2 CR 1.1	Verify user authentication mechanism is implemented on all network interfaces that exposes sensitive information or admin level functionality.
	Enumerate all authentication and authorization methods.
IEC 62443-4-2 CR 1.1 RE 1	Verify users are uniquely identified.
IEC 62443-4-2 CR 1.1 RE 2	Verify at least two-factor authentication is implemented wherever applicable and required.
IEC 62443-4-2 CR 1.2	Verify that the system is capable of uniquely identifying and authenticating device(s) before establishing network connections.
	Verify that Organizational Authentication solutions are used to identify and authenticate devices on local and/or wide area networks.
	Verify that device-to-device identification and authentication when Standard Industrial protocols do not support cryptographic identification and authentication.
IEC 62443-4-2 CR 1.2 RE 1	Verify that the system is capable of uniquely identifying and authenticating device(s) before establishing network connections.
	Verify that Organizational Authentication solutions are used to identify and authenticate devices on local and/or wide area networks.
	Verify that device-to-device identification and authentication when Standard Industrial protocols do not support cryptographic identification and authentication.
IEC 62443-4-2 CR 1.3	Verify that each account within a system is tied to an individual user for proper auditing, management, and tracking, and that shared accounts are not used.

	Verify that Super User accounts, such as Administrator or Root, are disabled or removed wherever possible.
	Verify that the identity requirements for user registration are aligned with business and security requirements.
	Verify that the proper identification and authorization process is followed while provisioning of accounts.
IEC 62443-4-2 CR 1.4	Enumerate the users and roles of all resources.
IEC 62443-4-2 CR 1.5	Verify stored passwords are encrypted/hashed.
	Verify that all accounts require a password change prior to continuing with the install.
	Verify sensitive password files /etc/shadow and /etc/passwd are accessible only by the Root account.
	Verify that passwords are not hardcoded in the source code, firmware, configuration tools, registry keys, scripts, or any other system components.
	Verify the password can be changed for a default account.
	Enumerate all password storage locations, including text files, databases, and binary objects.
	Verify that the file, database, or object that is used to maintain passwords is only write-able by the application.
	Verify that a password change requires the current password to be entered.
	Go through the installation process and document all accounts and passwords used during the process, if applicable.
	Verify that the default account's username cannot be changed.
	Verify default credentials are not used.
	Verify passwords are unique and not resettable to any universal factory default value.
IEC 62443-4-2 CR 1.5 RE 1	Verify Hardware Security Modules or Trusted Platform Modules can be used for user identification and authentication.
IEC 62443-4-2 CR 1.6	Verify the device uses encryption for wireless access.

	In case of using WiFi, verify the WiFi client supports modern authentication protocols.
	For using other wireless protocols than WiFi, verify proper security measures are implemented as per protocol.
	In case of using ZigBee, verify all communication is encrypted with the Network (NWK) key.
	In case of using Bluetooth Low Energy (BLE), verify the application uses BLE 5.3 or later release.
	In case of using Bluetooth Low Energy (BLE), verify the application uses best feasible security mode and level based on i/o capabilities of devices.
	In case of using Bluetooth Low Energy (BLE), verify all keys and data are sent over encrypted link only and encryption key size is configured to maximum allowable.
	In case of using Bluetooth Low Energy (BLE) mesh, verify, on node removal, that the application blacklists the device and refreshes the key.
	In case of using ZigBee, verify no publicly known keys are used in the network.
	In case of using ZigBee, verify rejoin requests are securely handled.
	In case of using ZigBee, verify insecure rejoin requests are disabled or not responded to.
	In case of using ZigBee, verify network communication is not permitted with a modified Network (NWK) key.
	In case of using ZigBee, verify network communication is not permitted with a modified Network (NWK) key.
IEC 62443-4-2 CR 1.7	Verify password complexity policies are configurable and enforceable.
	Verify that, in case a credential is a password, its minimum length is 6 characters.
IEC 62443-4-2 CR 1.7 RE 1	Verify that passwords can either be expired based on a specific time interval or restricted from reuse.
	Verify password complexity policies are configurable and enforceable.

IEC 62443-4-2 CR 1.7 RE 2	Verify that passwords can either be expired based on a specific time interval or restricted from reuse.
IEC 62443-4-2 CR 1.8	Verify that certificates for PKI-based authentication are validated against a trust anchor.
IEC 62443-4-2 CR 1.9	Verify that certificates for PKI-based authentication are validated against a trust anchor.
	Verify that authorized access is enforced to access or use the private key for PKI-based authentication.
	Verify that the authenticated identity is mapped to the corresponding user account for PKI-based authentication.
IEC 62443-4-2 CR 1.9 RE 1	Verify that Hardware Security Modules (HSMs) or Trusted Platform Modules (TPMs) in use are physically protected.
	Verify that the failure of the cryptographic module authentication fails secure.
	Verify the system protects "Root of trust" data via hardware mechanisms, preventing any modification of the data during normal operations of the component.
IEC 62443-4-2 CR 1.10	Verify error messages do not specify whether a user exists.
	Verify error messages do not specify a locked out account.
	Verify error messages do not specify a maximum number of failed attempts tried before locking.
	Verify the system mitigates account enumeration and guessable user accounts.
IEC 62443-4-2 CR 1.11	Verify unsuccessful login attempts are limited to a maximum number of tries within a time period.
	Verify an unsuccessful login attempt automatically locks the account for a specified period of time.
	Verify additional login attempts are delayed.
	Verify only an administrator can unlock an account when the maximum number of unsuccessful attempts is exceeded.

IEC 62443-4-2 CR 1.12	Verify that the device or system displays an 'Appropriate Use' warning banner to warn-off unauthorized users when required.
	Verify appropriate warning messages are displayed.
IEC 62443-4-2 CR 1.13	Verify access is restricted to privileged functions such as hardware resets and sudo commands.
	Verify access is restricted to security information.
	Verify that the device interface and ports are protected from unauthorized access.
	Verify that Field Tools require access control to utilize the tool(s).
IEC 62443-4-2 CR 1.14	Verify separate and unique keys are used for different functions such as transmitting data between devices, communicating with servers, encrypting files, and generating digital signatures.
	Verify that there exists a crypto-log in product documentation with information about each key or certificate, key-lengths, usage, location, and responsible party.
	Confirm that all cryptographic algorithms utilized are approved by NIST or FIPS.
	Verify keys are stored and managed securely.
IEC 62443-4-2 CR 1.14 RE 1	Verify Hardware Security Modules or Trusted Platform Modules can be used for user identification and authentication.
IEC 62443-4-2 CR 2.1	Verify that the level of privileges assigned to all processes/services are identified by using the list of enumerated processes/services. This includes determining the service level account allocated to each process/service, such as local service, network service, or local system.
	Verify that all non-root or non-admin accounts have read-only permissions or less for important files, such as /etc/sunders, /etc/ssh/sshd_config, /etc/group, /etc/shadow, /etc/passwd, Program Data etc.
	Verify users have a minimal write permissions.

	Verify that protection against directory traversal/file include attacks is implemented.
	Verify the system mitigates unauthorized access to admin functions.
	Verify that privileges for Set User ID accounts are raised as late as possible and released as soon as possible. This may require access to code to confirm, or reverse engineer using a tool such as Interactive Disassembler (IDA).
IEC 62443-4-2 CR 2.1 RE 1	Verify all web pages and resources by default require authentication, except those specifically intended to be public.
	Verify all authentication controls are enforced on the server side, and not on the client.
	Verify access is restricted to privileged functions such as hardware resets and sudo commands.
	Verify access is restricted to security information.
	Verify access is restricted based on user, role, and attributes.
	Verify that there is no method to bypass the Role Based Access Control or Access Control List.
	Verify that the device interface and ports are protected from unauthorized access.
	Verify that Field Tools require access control to utilize the tool(s).
	Verify a mechanism for device or user authentication is implemented.
	Verify device/user authentication mechanism is implemented on all network interfaces that exposes sensitive information or admin level functionality.
	Verify that guest and training accounts have limited access permissions and cannot perform any actions that compromise the security or integrity of the application or its users.
	Enumerate all the processes and services running in the device or system using a root shell.



	<p>Verify that the level of privileges assigned to all processes/services are identified by using the list of enumerated processes/services. This includes determining the service level account allocated to each process/service, such as local service, network service, or local system.</p>
	<p>Verify each process and service is not running as root for Linux systems.</p>
	<p>Verify each process and service is not running as LOCAL SYSTEM for Windows systems.</p>
	<p>Verify that administrative interfaces are not accessible to untrusted parties.</p>
	<p>Verify that non-root accounts are used for network services.</p>
	<p>Verify that all service accounts have shell access disabled by verifying that the shell is set to "/dev/null" in the passwd file</p>
IEC 62443-4-2 CR 2.1 RE 2	<p>Verify access is restricted based on user, role, and attributes.</p>
	<p>Verify that there is no method to bypass the Role Based Access Control or Access Control List.</p>
	<p>Verify that the system has defined roles for admin and regular users, and that the permission matrix is mapped according to the hierarchy of the roles.</p>
	<p>Verify device/user authentication mechanism is implemented on all network interfaces that exposes sensitive information or admin level functionality.</p>
IEC 62443-4-2 CR 2.1 RE 3	<p>Verify that the system supports supervisor manual override of current authorizations.</p>
IEC 62443-4-2 CR 2.1 RE 4	<p>Verify if the system supports dual approval checks for critical actions.</p>
IEC 62443-4-2 CR 2.2	<p>In case of using Bluetooth Low Energy (BLE), verify the application uses BLE 5.3 or later release.</p>
	<p>In case of using Bluetooth Low Energy, verify advertisements are transmitted evenly across all three advertising channels and advertisement data is encrypted.</p>

	In case of using Bluetooth Low Energy (BLE), verify the application uses best feasible security mode and level based on i/o capabilities of devices.
	In case of using Bluetooth Low Energy, verify unneeded and unapproved services and profiles are disabled.
	In case of using Bluetooth Low Energy (BLE) mesh, verify, on node removal, that the application blacklists the device and refreshes the key.
	Enumerate all remote interfaces, including network ports and services.
	Enumerate all the local interfaces, including physical ones at board level (such as JTAG, SPI, I2C, FTDI, OBDII) and device level (such as USB, Serial).
	Enumerate all wireless interfaces (such as WiFi, ZigBee, Bluetooth, etc.).
	Verify that each enumerated interface either has authentication or has been rendered inoperable.
	Verify that no insecure services such as HTTP, FTP, or Telnet are being used and that they are disabled by default at minimum.
	Verify only secure services are used, such as HTTPS, SFTP, and SSH.
	Verify that wireless radios such as WiFi and Bluetooth are disabled by default.
	Verify Secure DNS is used instead of DNS.
	Verify Secure DNP3 is utilized if DNP3 is supported.
	Verify Secure Tunneling is used for unsecured legacy protocols.
	If continued developer access is necessary, verify that any developer-level debugging interfaces are appropriately protected to limit access to authorized privileged users.
IEC 62443-4-2 CR 2.3	Verify portable devices and media are disabled by default.
	Verify portable devices and media can be enabled only by specifically authorized users.
IEC 62443-4-2 CR 2.4	Verify automated execution of code from portable devices or media is disabled.

	Enumerate all the code that is downloaded and active on mobile devices during runtime, such as JavaScript, ActiveX, Flash, Java Applets, and others.
	Verify that mobile and active code technologies, including JavaScript, ActiveX, Flash, and Java Applets, are only permitted to run or be accepted from components deployed on local servers or user networks, and not directly from the internet.
	Verify the device controls which users (human, software process, or device) are allowed to transfer mobile code to and from the application.
IEC 62443-4-2 CR 2.5	Verify configurations cannot be changed while the session is locked.
	Identify all authentication methods and session management techniques.
	Verify a session can be re-established only after completing the identification and authentication procedures.
	Verify the Session Management Schema can not be bypassed.
	Verify adequate logout functionality is implemented.
	Verify a session is terminated on server side, and not on the client side.
	Verify a local session is automatically terminated after an appropriate set time of inactivity.
IEC 62443-4-2 CR 2.6	Verify that if remote access, for example, SSH, SFTP, etc., is provisioned, remote access sessions are encrypted.
	Verify the cryptographic libraries used for encryption are FIPS-140-2 compliant.
	Verify remote access is logged.
	Verify remote administrative activities are logged. Examples are configurations changes, settings, upgrades, etc.
	Verify remote maintenance activities are logged. Examples are diagnostics, reading logs, preventive maintenance, etc.
	Verify a remote session is automatically terminated after a set period of time of inactivity.

	If SSH/Remote Access is enabled, then verify that PermitRootLogin is disabled or configured to Without-Password.
IEC 62443-4-2 CR 2.7	Verify concurrent logins are disallowed by default.
IEC 62443-4-2 CR 2.8	Verify that the system or the device has an audit log that can be exported, in a business readable format.
	Verify that audit events including logon and logoff attempts, configuration changes, upgrades, and other related activities are logged with the date and time stamp using the WHEN parameter.
	Verify that audit events including logon and logoff attempts, configuration changes, upgrades, and other related activities are logged with the user-identity using the WHO parameter.
	Verify that audit events including logon and logoff attempts, configuration changes, upgrades, and other related activities are logged with the event type using the WHAT parameter.
	Verify that audit events including logon and logoff attempts, configuration changes, upgrades, and other related activities are logged with the location of the event using the WHERE parameter.
	Verify that appropriate access controls are in place for event log data and that access to logs is restricted based on a need-to-know basis.
	Verify that the event of deleting logs is recorded, either in the system or at the beginning of a new log.
IEC 62443-4-2 CR 2.9	Verify that logging cannot be used to deplete system resources, for example by filling up disk space or exceeding database log space, leading to Denial of Service (DoS).
IEC 62443-4-2 CR 2.9 RE 1	Verify the device gives a warning when the audit record storage capacity threshold is reached.
IEC 62443-4-2 CR 2.10	Verify logs are retained after a local power outage.
	Verify logs are retained after a device or system reboot.

	<p>Verify the application continues to function even when audit logging failed.</p> <p>Verify appropriate action is taken when audit logging failed according to commonly accepted industry practices and recommendations.</p>
IEC 62443-4-2 CR 2.11	<p>Verify that audit events including logon and logoff attempts, configuration changes, upgrades, and other related activities are logged with the date and time stamp using the WHEN parameter.</p> <p>Verify periodic time-shift events, such as daylight savings time in some locations, are considered and reflect in audit logs.</p>
IEC 62443-4-2 CR 2.11 RE 1	<p>Verify components shall provide the capability to create timestamps that are synchronized with a system wide time source e.g. NTP server.</p> <p>Verify the network device compares internal information system clocks at a configurable time with an authoritative time server.</p>
IEC 62443-4-2 CR 2.11 RE 2	<p>Verify that logs are restricted to authorized users only and are immutable.</p> <p>Verify periodic time-shift events, such as daylight savings time in some locations, are considered and reflect in audit logs.</p> <p>Verify that the time synchronization mechanism provides the capability to detect unauthorized alteration and causes an audit event upon alteration.</p>
IEC 62443-4-2 CR 2.12	<p>Verify that audit events including logon and logoff attempts, configuration changes, upgrades, and other related activities are logged with the user-identity using the WHO parameter.</p> <p>Verify that audit events including logon and logoff attempts, configuration changes, upgrades, and other related activities are logged with the event type using the WHAT parameter.</p>
IEC 62443-4-2 CR 2.12 RE 1	<p>Verify that audit events including logon and logoff attempts, configuration changes, upgrades, and other related activities are logged with the user-identity using the WHO parameter.</p>

	Verify that audit events including logon and logoff attempts, configuration changes, upgrades, and other related activities are logged with the event type using the WHAT parameter.
IEC 62443-4-2 CR 2.13	N.A.
IEC 62443-4-2 CR 3.1	Verify message authentication is implemented at the protocol level, if the protocol supports it.
	Using a enumerated list of services, for each authentication request, intercept at least one sample.
	Verify by sniffing that credentials are transported using a suitable encrypted link and that all pages and functions that require a user to enter credentials are done so using an encrypted link.
	Verify that the device adheres to the relevant security standards for each protocol in case message authentication is not implemented at the protocol level.
	Verify TLS 1.2 or TLS 1.3 is used for encryption over network for HTTPS.
	Verify HTTP Strict Transport Security (HSTS) implementation by ensuring the server has a valid SSL/TLS certificate and confirming the presence of the HSTS header.
	Verify that adequate mechanisms are implemented to ensure communication integrity and confidentiality on all protocols that do not support cryptography.
	Verify that there exists a mechanism to preserve and check data integrity on a Controller Area Network (CAN) Bus or any other similar network.
IEC 62443-4-2 CR 3.1 RE 1	Verify message authentication is implemented at the protocol level, if the protocol supports it.
	Using a enumerated list of services, for each authentication request, intercept at least one sample.

	<p>Verify by sniffing that credentials are transported using a suitable encrypted link and that all pages and functions that require a user to enter credentials are done so using an encrypted link.</p>
	<p>Verify that the device adheres to the relevant security standards for each protocol in case message authentication is not implemented at the protocol level.</p>
	<p>Verify TLS 1.2 or TLS 1.3 is used for encryption over network for HTTPS.</p>
	<p>Verify HTTP Strict Transport Security (HSTS) implementation by ensuring the server has a valid SSL/TLS certificate and confirming the presence of the HSTS header.</p>
IEC 62443-4-2 CR 3.2	N.A.
IEC 62443-4-2 CR 3.3	<p>Verify that the integrity check cannot be bypassed by any means.</p>
	<p>Verify that software is verified using secure boot mechanisms, which require a hardware root of trust.</p>
IEC 62443-4-2 CR 3.3 RE 1	<p>Verify that the integrity check cannot be bypassed by any means.</p>
	<p>Verify that software is verified using secure boot mechanisms, which require a hardware root of trust.</p>
IEC 62443-4-2 CR 3.4	<p>Enumerate all external file inputs to the device.</p>
	<p>Verify that unsupported file formats can not be uploaded.</p>
	<p>Verify input files include an integrity check, such as a Message Authentication Code (MAC) or Signature.</p>
	<p>Verify, if a digital signature is used, the Message Authentication Code (MAC) is either SHA, RSA, DSA, or ECDSA.</p>
	<p>Verify that the minimum key length for the digital signature is 2048-bits for asymmetric RSA and DSA, and 224 for ECDSA.</p>
	<p>Verify that the hashing algorithm used for the digital signature is SHA256.</p>
	<p>Verify the integrity check status is logged.</p>

	Verify that only read-only configuration files can be uploaded by the user in case there is no integrity check mechanism for configuration files.
	Verify that the configuration file integrity mechanism cannot be bypassed in any manner.
	Verify that applications can undergo an integrity check by comparing them to a known source, either a full copy or a hash.
	Verify that the device or system has the capability to generate and maintain a hash of the currently installed executables and firmwares.
	Verify the hash used is SHA2 and stored at secure location, which is not accessible to users.
	Verify the integrity check is logged.
	Verify that the device or system maintains a complete image of all currently deployed software.
	Verify that the device maintains a hash of the currently installed software and firmware, including patches.
	Verify that any update to the executables and firmwares results in a recalculation and update of this hash.
	Verify that an integrity check of all executables and firmwares can be performed by comparing the hash of the component to the hash in the repository.
	Verify that the integrity check is performed by the device at least once every 30 days.
IEC 62443-4-2 CR 3.4 RE 1	Verify firmware executables contain a Digital Signature from a trusted CA and are thus not self-signed.
	Verify that the Digital Signature Public Key Algorithm is RSA, DSA or ECDSA.
	Verify that the minimum key length for the digital signature is 2048-bits for asymmetric RSA and DSA, and 224 for ECDSA.
	Verify that the hashing algorithm used for the digital signature is SHA256.



	<p>Verifying firmware with a bad digital signature will result in an error being logged and the default firmware being employed.</p> <p>Verify that the authenticity check is implemented properly in the device. The device should not accept unsigned firmware as well as firmware signed with a self-signed certificate on any interface.</p>
IEC 62443-4-2 CR 3.4 RE 2	<p>Verify that any update to the executables and firmwares results in a recalculation and update of this hash.</p> <p>Verify that an integrity check of all executables and firmwares can be performed by comparing the hash of the component to the hash in the repository.</p>
IEC 62443-4-2 CR 3.5	<p>Verify that user supplied data is encoded prior to viewing in intended log viewing interface, to prevent injection attacks.</p> <p>Verify all data inputs via user interfaces and transferred via Application Programming Interfaces (APIs) or between networks in services and devices are validated.</p> <p>Verify that all data inputs from web applications, mobile apps, and PC tools undergo validation.</p> <p>Verify inputs outside the character set are disallowed in user input fields.</p> <p>Verify inputs exceeding the field length are disallowed in user input fields.</p> <p>Verify inputs outside the allowed range are disallowed in user input fields</p> <p>Verify inputs are handled as expected when fuzzed.</p> <p>Verify the device mitigates vulnerabilities in HTTP Methods.</p> <p>Verify the application only accepts logically valid data.</p> <p>Verify the application mitigates the ability to forge requests.</p> <p>Verify the application mitigates process timing vulnerabilities.</p> <p>Verify the application or system mitigates function limit vulnerabilities.</p> <p>Verify the workflows of the application cannot be circumvented.</p>

	Verify the device has protection mechanisms against application mis-uses.
	Verify only approved file types can be uploaded in the application.
	Verify the device mitigates the upload of malicious files.
	Verify if the device mitigates JavaScript Execution.
	Verify the application mitigates CSS Injection.
	Verify the device mitigates Client Side Resource Manipulation.
	Validate the device mitigates Cross Origin Resource Sharing.
	Verify the device mitigates Cross Site Flashing.
	Verify the application mitigates Clickjacking.
	Verify the device establishes a secure WebSocket connection.
	Verify the message's origin is secure and that safe methods are used to process data and validate input.
	Verify the device mitigates Local Storage vulnerabilities.
	Verify that proper character sets, such as UTF-8 have been enforced for all user input, to prevent alternate character sets, for example allowing foreign languages that may issue injection attacks.
	Verify that saved files are not accessible through the application's web context and are instead stored securely on a content server or database.
	Verify that all output to other system components is sanitized and encoded properly before sending outside of the application trust boundary.
IEC 62443-4-2 CR 3.6	Verify that upon a failure the product's functionality is limited only to allow maintenance of the device.
	Verify upon a successful startup, only necessary ports, services, and settings are enabled by default.
IEC 62443-4-2 CR 3.7	Verify that detailed error messages are only accessible to authorized users, such as admins or maintenance personnel.

	Verify error messages do not contain exploitable information.
	Verify error messages do not contain device or system component information.
	Verify error messages do not contain information for correction.
	Verify error messages do not contain failure details.
	Verify error messages do not contain stack traces.
	Verify error messages do not contain memory locations.
	Verify error messages do not contain information outside the user's scope.
IEC 62443-4-2 CR 3.8	Identify all authentication methods and session management techniques.
	Verify a session can be re-established only after completing the identification and authentication procedures.
	Verify that user input is transmitted using secure protocols.
	Verify the Session Management Schema can not be bypassed.
	Verify secure cookie handling.
	Verify the system mitigates session hijacking.
	Verify the protection of session variables from eavesdropping and reuse of session tokens vulnerabilities.
	Verify the system mitigates Cross Site Request Forgery.
	Verify the server framework is used for session management and that the developer did not create their own.
	Verify the system is mitigates session puzzling.
	Verify the protection of sensitive data during network transmission.
IEC 62443-4-2 CR 3.9	Verify logs are retained after a local power outage.
	Verify logs are retained after a device or system reboot.
	Verify that logs are restricted to authorized users only and are immutable.

	Verify that appropriate access controls are in place for event log data and that access to logs is restricted based on a need-to-know basis.
IEC 62443-4-2 CR 3.9 RE 1	Verify that audit records are being produced on hardware-enforced write-once media, such as WORM drives or Non-Volatile Memory.
IEC 62443-4-2 CR 3.10	Verify that the product team has a mechanism for device upgrades in field such as for firmware and hardware.
	Verify that the system can update and patch the latest updates.
	Perform an initial configuration sequence from factory default settings.
	Verify that system components check for system updates prior to final activation and operation.
	Verify that the user is at least given the option to apply new updates, prior to operation.
	Verify that the product team has a mechanism for patching devices in field, in the event of a critical vulnerability discovery.
	Verify that any Over-The-Air upgrade process is secure.
	Verify the product supports security updates to the product's software.
	Verify security updates are possible in both online and offline network modes.
	Verify that product supports reverting to previously installed version if an update is unsuccessful.
	Verify that before deployment of the software or firmware components to the product, the download of the software or firmware components is completed.
	Verify that download of the software or firmware components to the product does not interrupt the continued operation of the product as intended.
IEC 62443-4-2 CR 3.11	Verify access to the electronics are not easily compromised, for example ensure there are no exposed fasteners, screws, or other compromisable components.

	<p>Visually inspect and enumerate any tamper resistant measures.</p> <p>Verify the effectiveness of each physical tamper resistant measure to deter and/or alert the system owner to tampering.</p> <p>Verify that hardware supported tamper resistance mechanisms are effective.</p>
IEC 62443-4-2 CR 3.12	<p>Verify the system protects “Root of trust” data via hardware mechanisms, preventing any modification of the data during normal operations of the component.</p> <p>Verify that the device or system has the capability to generate and maintain a hash of the currently installed executables and firmwares.</p> <p>Verify the hash used is SHA2 and stored at secure location, which is not accessible to users.</p> <p>Verify the integrity check is logged.</p> <p>Verify that the device maintains a hash of the currently installed software and firmware, including patches.</p> <p>Verify that any update to the executables and firmwares results in a recalculation and update of this hash.</p> <p>Verify that an integrity check of all executables and firmwares can be performed by comparing the hash of the component to the hash in the repository.</p> <p>Verify that the integrity check is performed by the device at least once every 30 days.</p> <p>Verify that the integrity check cannot be bypassed by any means.</p> <p>Verify that software is verified using secure boot mechanisms, which require a hardware root of trust.</p>
IEC 62443-4-2 CR 3.13	<p>Verify the system protects “Root of trust” data via hardware mechanisms, preventing any modification of the data during normal operations of the component.</p> <p>Verify that the device or system has the capability to generate and maintain a hash of the currently installed executables and firmwares.</p>

	Verify the hash used is SHA2 and stored at secure location, which is not accessible to users.
	Verify the integrity check is logged.
	Verify that the device maintains a hash of the currently installed software and firmware, including patches.
	Verify that any update to the executables and firmwares results in a recalculation and update of this hash.
	Verify that an integrity check of all executables and firmwares can be performed by comparing the hash of the component to the hash in the repository.
	Verify that the integrity check is performed by the device at least once every 30 days.
	Verify that the integrity check cannot be bypassed by any means.
IEC 62443-4-2 CR 3.14	Verify that applications can undergo an integrity check by comparing them to a known source, either a full copy or a hash.
	Verify that the device or system has the capability to generate and maintain a hash of the currently installed executables and firmwares.
	Verify the hash used is SHA2 and stored at secure location, which is not accessible to users.
	Verify that the device maintains a hash of the currently installed software and firmware, including patches.
	Verify that any update to the executables and firmwares results in a recalculation and update of this hash.
	Verify that an integrity check of all executables and firmwares can be performed by comparing the hash of the component to the hash in the repository.
	Verify that the integrity check cannot be bypassed by any means.
IEC 62443-4-2 CR 4.1	Identify all the sensitive and personal information generated, stored, used or communicated by the system.
	Enumerate all cryptographic functions of the application.

	Verify that the developer did not 'roll their own' encryption and that best practice libraries are used for all cryptographic functions.
	Verify that all sensitive or personal data is encrypted when at rest. This includes all forms of sensitive information, for example cached data in web browsers, temporary files, logs, sensitive secrets in configuration files, etc.
	Verify that device manufacturers and service providers provide consumers with clear and transparent information about how their personal data is being used.
	Verify that where personal data is processed on the basis of consumers' consent, this consent is obtained in a valid way.
	Verify that customers who gave consent for the processing of their personal data are given the opportunity to withdraw it at any time.
	Verify by sniffing that credentials are transported using a suitable encrypted link and that all pages and functions that require a user to enter credentials are done so using an encrypted link.
	Verify that adequate mechanisms are implemented to ensure communication integrity and confidentiality on all protocols that do not support cryptography.
	Verify that there exists a mechanism to preserve and check data confidentiality on a Controller Area Network (CAN) Bus or any other similar network.
IEC 62443-4-2 CR 4.2	Identify all the sensitive and personal information generated, stored, used or communicated by the system.
	Verify the product documentation details a secure decommissioning of the device.
	Verify that the device sanitizes sensitive information as per the device documentation.

	<p>Verify that recovering of any sensitive information from the decommissioned device is difficult and requires significant effort using specialized tools and skill sets.</p>
	<p>Verify personal data can easily be removed.</p>
	<p>Verify consumers are given clear instructions on how to delete their personal data, especially for IoT systems.</p>
	<p>Verify consumers are provided with clear confirmation that personal data has been deleted from services, devices and applications, especially for IoT systems.</p>
IEC 62443-4-2 CR 4.2 RE 1	<p>Verify portable devices and media can be enabled only by specifically authorized users.</p>
	<p>Verify automated execution of code from portable devices or media is disabled.</p>
	<p>Remove the removable memory and inspect the contents, verify that no sensitive data is stored on the removable memory.</p>
	<p>Visually inspect and enumerate any removable memory, such as SD cards.</p>
IEC 62443-4-2 CR 4.2 RE 2	<p>Verify that the devices can verify whether the erasure of information occurred.</p>
IEC 62443-4-2 CR 4.3	<p>Enumerate all cryptographic functions of the application.</p>
	<p>Verify only recommended ciphers are enabled on the server.</p>
	<p>Verify TLS 1.2 or TLS 1.3 is used for encryption over network for HTTPS.</p>
	<p>Verify SSL certificates are signed using SHA-2 (with SHA-256 or higher) hashing algorithm.</p>
	<p>Verify that the minimum key length for the digital signature is 2048-bits for asymmetric RSA and DSA, and 224 for ECDSA.</p>
	<p>Verify the cryptography does not degrade the performance of the system.</p>
	<p>Verify the failure of the cryptography does not result in a Denial of Service (DoS).</p>
	<p>Verify used encryption does not support or fallback to insecure ciphers.</p>
	<p>Verify that any connection does not accept invalid certificates.</p>



IEC 62443-4-2 CR 5.1	Verify that the system has the capability to logically segment system networks.
IEC 62443-4-2 CR 5.2	N.A.
IEC 62443-4-2 CR 5.3	N.A.
IEC 62443-4-2 CR 5.4	N.A.
IEC 62443-4-2 CR 6.1	Verify that logs are restricted to authorized users only and are immutable.
IEC 62443-4-2 CR 6.1 RE 1	Verify that the system or the device has an audit log that can be exported, in a business readable format.
IEC 62443-4-2 CR 6.2	Verify that the system supports security monitoring mechanisms.
IEC 62443-4-2 CR 7.1	Verify a single user cannot overload a system with certain requests, such as the recording and sending of the same packets repetitively.
IEC 62443-4-2 CR 7.1 RE 1	Verify that the device has measures in place to prevent or minimize the impact of denial-of-service attacks, such as excessive network traffic, log flooding, and application/protocol traffic.
	Verify that request throttling is in place to prevent automated attacks against common authentication attacks such as brute force attacks or denial of service attacks.
IEC 62443-4-2 CR 7.2	Verify that the device has measures in place to prevent or minimize the impact of denial-of-service attacks, such as excessive network traffic, log flooding, and application/protocol traffic.
	Verify that there exists some protective mechanism that can prevent DoS attacks, which involve flooding the network with excessive data or using unauthorized applications from remote devices to disrupt the system.
IEC 62443-4-2 CR 7.3	Verify there are backup mechanisms implemented for system components, such as application servers, database servers, and others.
	Verify that the backup mechanism covers all critical system components.
	Verify the backup mechanism is fully documented.

	Verify the recovery mechanism is fully documented.
	Verify the recovery mechanism has been tested.
IEC 62443-4-2 CR 7.3 RE 1	Verify that any backups containing Personal Identifiable Information (PII) or sensitive data are encrypted.
	Verify that the backups are prevented from unauthorized access.
IEC 62443-4-2 CR 7.4	Verify that product performs the software integrity test before loading any applications or performing any functions of the product.
	Simulate a product failure by tampering one of the software components in the data store and verify that it fails the software integrity test on initial power up.
	Verify that upon a failure the product enters a failure mode which clearly indicates to the user that the product has failed to start up successfully.
	Verify the system starts up in a defined known state.
	Verify the system has a defined known secure state.
	Verify the system fails to the known secure state.
	Verify a device failed to the known state does not allow access without verifying credentials.
	Verify that a control system has been successfully recovered and reconstituted to a known secure state by checking that all system parameters are set to secure values, security-critical patches are reinstalled, security-related configuration settings are reestablished, system documentation and operating procedures are available, application and system software is reinstalled and configured with secure settings, information from the most recent, known secure backups is loaded, and the system is fully tested and functional.
IEC 62443-4-2 CR 7.5	Perform a power failure simulation test.

IEC 62443-4-2 CR 7.6	Enumerate all services, components, and operating systems.
	Verify all enumerated components are compliant with standard secure baseline configurations such as NIST, CIS, DISA, and others.
	Verify the system provides the capability to be configured according to recommended network and security configurations as described in guidelines provided by the control system supplier.
	Verify the system has an interface to the currently deployed network and security configuration settings.
IEC 62443-4-2 CR 7.6 RE 1	Verify the system can generate a report listing the currently deployed security settings in a machine-readable format.
IEC 62443-4-2 CR 7.7	Enumerate additional services that are installed, but not enabled.
	Verify that all additional services which are installed, but not required for operations are removed.
	If continued developer access is necessary, verify that any developer-level debugging interfaces are appropriately protected to limit access to authorized privileged users.
	Verify there are not any unnecessary network services.
IEC 62443-4-2 CR 7.8	Verify that there is a designated system capable of retrieving and storing data from all system components either daily or upon request.
	Verify that the data retrieved from all system components includes information such as current version numbers, installation dates, configuration settings, and patch levels.
IEC 62443-4-2 SAR 2.4	Verify automated execution of code from portable devices or media is disabled.
	Enumerate all the code that is downloaded and active on mobile devices during runtime, such as JavaScript, ActiveX, Flash, Java Applets, and others.

	<p>Verify that mobile and active code technologies, including JavaScript, ActiveX, Flash, and Java Applets, are only permitted to run or be accepted from components deployed on local servers or user networks, and not directly from the internet.</p>
	<p>Verify the device controls which users (human, software process, or device) are allowed to transfer mobile code to and from the application.</p>
IEC 62443-4-2 SAR 2.4 RE 1	<p>Verify the device controls the execution of mobile code based on the results of an integrity check prior to the code being executed.</p>
IEC 62443-4-2 SAR 3.2	<p>Verify portable devices and media are disabled by default.</p>
	<p>Verify portable devices and media can be enabled only by specifically authorized users.</p>
	<p>Verify that mobile and active code technologies, including JavaScript, ActiveX, Flash, and Java Applets, are only permitted to run or be accepted from components deployed on local servers or user networks, and not directly from the internet.</p>
	<p>Verify the device controls the execution of mobile code based on the results of an integrity check prior to the code being executed.</p>
IEC 62443-4-2 EDR 2.4	<p>Verify automated execution of code from portable devices or media is disabled.</p>
	<p>Enumerate all the code that is downloaded and active on mobile devices during runtime, such as JavaScript, ActiveX, Flash, Java Applets, and others.</p>
	<p>Verify that mobile and active code technologies, including JavaScript, ActiveX, Flash, and Java Applets, are only permitted to run or be accepted from components deployed on local servers or user networks, and not directly from the internet.</p>

	Verify the device controls which users (human, software process, or device) are allowed to transfer mobile code to and from the application.
IEC 62443-4-2 EDR 2.4 RE 1	Verify the device controls the execution of mobile code based on the results of an integrity check prior to the code being executed.
IEC 62443-4-2 EDR 2.13	Enumerate all the local interfaces, including physical ones at board level (such as JTAG, SPI, I2C, FTDI, OBDII) and device level (such as USB, Serial).
	Verify that each enumerated interface either has authentication or has been rendered inoperable.
	Verify all interfaces and services not required for operations are disabled by default.
	Verify that no insecure services such as HTTP, FTP, or Telnet are being used and that they are disabled by default at minimum.
	If continued developer access is necessary, verify that any developer-level debugging interfaces are appropriately protected to limit access to authorized privileged users.
IEC 62443-4-2 EDR 2.13 RE 1	Verify active monitoring of diagnostic and test interfaces are implemented.
IEC 62443-4-2 EDR 3.2	Verify firmware executables contain a Digital Signature from a trusted CA and are thus not self-signed.
	Verify that the Digital Signature Public Key Algorithm is RSA, DSA or ECDSA.
	Verify that the minimum key length for the digital signature is 2048-bits for asymmetric RSA and DSA, and 224 for ECDSA.
	Verify that the hashing algorithm used for the digital signature is SHA256.
	Verifying firmware with a bad digital signature will result in an error being logged and the default firmware being employed.

	Verify that the authenticity check is implemented properly in the device. The device should not accept unsigned firmware as well as firmware signed with a self-signed certificate on any interface.
IEC 62443-4-2 EDR 3.10	Verify that the product team has a mechanism for device upgrades in field such as for firmware and hardware.
	Verify that the system can update and patch the latest updates.
	Perform an initial configuration sequence from factory default settings.
	Verify that system components check for system updates prior to final activation and operation.
	Verify that the user is at least given the option to apply new updates, prior to operation.
	Verify that the product team has a mechanism for patching devices in field, in the event of a critical vulnerability discovery.
	Verify that any Over-The-Air upgrade process is secure.
	Verify the product supports security updates to the product's software.
	Verify security updates are possible in both online and offline network modes.
	Verify that product supports reverting to previously installed version if an update is unsuccessful.
	Verify that before deployment of the software or firmware components to the product, the download of the software or firmware components is completed.
Verify that download of the software or firmware components to the product does not interrupt the continued operation of the product as intended.	
IEC 62443-4-2 EDR 3.10 RE 1	Verify that the authenticity check is implemented properly in the device. The device should not accept unsigned firmware as well as firmware signed with a self-signed certificate on any interface.

IEC 62443-4-2 EDR 3.11	Verify access to the electronics are not easily compromised, for example ensure there are no exposed fasteners, screws, or other compromisable components.
	Visually inspect and enumerate any tamper resistant measures.
	Verify the effectiveness of each physical tamper resistant measure to deter and/or alert the system owner to tampering.
	Verify that hardware supported tamper resistance mechanisms are effective.
IEC 62443-4-2 EDR 3.11 RE 1	Verify that if an unauthorized change is detected to the software, the device should alert the consumer and/or administrator to an issue and should not connect to wider networks than those necessary to perform the alerting function.
	Verify the effectiveness of each physical tamper resistant measure to deter and/or alert the system owner to tampering.
IEC 62443-4-2 EDR 3.12	Verify the system protects “Root of trust” data via hardware mechanisms, preventing any modification of the data during normal operations of the component.
	Verify that the device or system has the capability to generate and maintain a hash of the currently installed executables and firmwares.
	Verify the hash used is SHA2 and stored at secure location, which is not accessible to users.
	Verify the integrity check is logged.
	Verify that the device maintains a hash of the currently installed software and firmware, including patches.
	Verify that any update to the executables and firmwares results in a recalculation and update of this hash.
	Verify that an integrity check of all executables and firmwares can be performed by comparing the hash of the component to the hash in the repository.
	Verify that the integrity check is performed by the device at least once every 30 days.
	Verify that the integrity check cannot be bypassed by any means.

	Verify that software is verified using secure boot mechanisms, which require a hardware root of trust.
IEC 62443-4-2 EDR 3.13	Verify the system protects “Root of trust” data via hardware mechanisms, preventing any modification of the data during normal operations of the component.
	Verify that the device or system has the capability to generate and maintain a hash of the currently installed executables and firmwares.
	Verify the hash used is SHA2 and stored at secure location, which is not accessible to users.
	Verify the integrity check is logged.
	Verify that the device maintains a hash of the currently installed software and firmware, including patches.
	Verify that any update to the executables and firmwares results in a recalculation and update of this hash.
	Verify that an integrity check of all executables and firmwares can be performed by comparing the hash of the component to the hash in the repository.
	Verify that the integrity check is performed by the device at least once every 30 days.
	Verify that the integrity check cannot be bypassed by any means.
IEC 62443-4-2 EDR 3.14	Verify that applications can undergo an integrity check by comparing them to a known source, either a full copy or a hash.
	Verify that the device or system has the capability to generate and maintain a hash of the currently installed executables and firmwares.
	Verify the hash used is SHA2 and stored at secure location, which is not accessible to users.
	Verify that the device maintains a hash of the currently installed software and firmware, including patches.
	Verify that any update to the executables and firmwares results in a recalculation and update of this hash.



	<p>Verify that an integrity check of all executables and firmwares can be performed by comparing the hash of the component to the hash in the repository.</p>
	<p>Verify that the integrity check cannot be bypassed by any means.</p>
IEC 62443-4-2 EDR 3.14 RE 1	<p>Verify that the authenticity check is implemented properly in the device. The device should not accept unsigned firmware as well as firmware signed with a self-signed certificate on any interface.</p>
IEC 62443-4-2 HDR 2.4	<p>Verify automated execution of code from portable devices or media is disabled.</p>
	<p>Enumerate all the code that is downloaded and active on mobile devices during runtime, such as JavaScript, ActiveX, Flash, Java Applets, and others.</p>
	<p>Verify that mobile and active code technologies, including JavaScript, ActiveX, Flash, and Java Applets, are only permitted to run or be accepted from components deployed on local servers or user networks, and not directly from the internet.</p>
	<p>Verify the device controls which users (human, software process, or device) are allowed to transfer mobile code to and from the application.</p>
IEC 62443-4-2 HDR 2.4 RE 1	<p>Verify the device controls the execution of mobile code based on the results of an integrity check prior to the code being executed.</p>
IEC 62443-4-2 HDR 2.13	<p>Enumerate all the local interfaces, including physical ones at board level (such as JTAG, SPI, I2C, FTDI, OBDII) and device level (such as USB, Serial).</p>
	<p>Verify that each enumerated interface either has authentication or has been rendered inoperable.</p>
	<p>Verify all interfaces and services not required for operations are disabled by default.</p>
	<p>Verify that no insecure services such as HTTP, FTP, or Telnet are being used and that they are disabled by default at minimum.</p>

	If continued developer access is necessary, verify that any developer-level debugging interfaces are appropriately protected to limit access to authorized privileged users.
IEC 62443-4-2 HDR 2.13 RE 1	Verify active monitoring of diagnostic and test interfaces are implemented.
IEC 62443-4-2 HDR 3.2	Verify portable devices and media are disabled by default.
	Verify portable devices and media can be enabled only by specifically authorized users.
	Verify that mobile and active code technologies, including JavaScript, ActiveX, Flash, and Java Applets, are only permitted to run or be accepted from components deployed on local servers or user networks, and not directly from the internet.
	Verify the device controls the execution of mobile code based on the results of an integrity check prior to the code being executed.
IEC 62443-4-2 HDR 3.2 RE 1	Verify that there is a log entry of the product firmware or software update process.
	In case the product update erases the audit log during the update process, verify that the product should start the new log with a record of the log erasure including the timestamp, authenticated means, and account.
IEC 62443-4-2 HDR 3.10	Verify that the product team has a mechanism for device upgrades in field such as for firmware and hardware.
	Verify that the system can update and patch the latest updates.
	Perform an initial configuration sequence from factory default settings.
	Verify that system components check for system updates prior to final activation and operation.
	Verify that the user is at least given the option to apply new updates, prior to operation.

	<p>Verify that the product team has a mechanism for patching devices in field, in the event of a critical vulnerability discovery.</p>
	<p>Verify that any Over-The-Air upgrade process is secure.</p>
	<p>Verify the product supports security updates to the product's software.</p>
	<p>Verify security updates are possible in both online and offline network modes.</p>
	<p>Verify that product supports reverting to previously installed version if an update is unsuccessful.</p>
	<p>Verify that before deployment of the software or firmware components to the product, the download of the software or firmware components is completed.</p>
	<p>Verify that download of the software or firmware components to the product does not interrupt the continued operation of the product as intended.</p>
IEC 62443-4-2 HDR 3.10 RE 1	<p>Verify that the authenticity check is implemented properly in the device. The device should not accept unsigned firmware as well as firmware signed with a self-signed certificate on any interface.</p>
IEC 62443-4-2 HDR 3.11	<p>Verify access to the electronics are not easily compromised, for example ensure there are no exposed fasteners, screws, or other compromisable components.</p>
	<p>Visually inspect and enumerate any tamper resistant measures.</p>
	<p>Verify the effectiveness of each physical tamper resistant measure to deter and/or alert the system owner to tampering.</p>
	<p>Verify that hardware supported tamper resistance mechanisms are effective.</p>
IEC 62443-4-2 HDR 3.11 RE 1	<p>Verify that if an unauthorized change is detected to the software, the device should alert the consumer and/or administrator to an issue and should not connect to wider networks than those necessary to perform the alerting function.</p>
	<p>Verify the effectiveness of each physical tamper resistant measure to deter and/or alert the system owner to tampering.</p>

IEC 62443-4-2 HDR 3.12	Verify the system protects “Root of trust” data via hardware mechanisms, preventing any modification of the data during normal operations of the component.
	Verify that the device or system has the capability to generate and maintain a hash of the currently installed executables and firmwares.
	Verify the hash used is SHA2 and stored at secure location, which is not accessible to users.
	Verify the integrity check is logged.
	Verify that the device maintains a hash of the currently installed software and firmware, including patches.
	Verify that any update to the executables and firmwares results in a recalculation and update of this hash.
	Verify that an integrity check of all executables and firmwares can be performed by comparing the hash of the component to the hash in the repository.
	Verify that the integrity check is performed by the device at least once every 30 days.
	Verify that the integrity check cannot be bypassed by any means.
	Verify that software is verified using secure boot mechanisms, which require a hardware root of trust.
IEC 62443-4-2 HDR 3.13	Verify the system protects “Root of trust” data via hardware mechanisms, preventing any modification of the data during normal operations of the component.
	Verify that the device or system has the capability to generate and maintain a hash of the currently installed executables and firmwares.
	Verify the hash used is SHA2 and stored at secure location, which is not accessible to users.
	Verify the integrity check is logged.
	Verify that the device maintains a hash of the currently installed software and firmware, including patches.

	Verify that any update to the executables and firmwares results in a recalculation and update of this hash.
	Verify that an integrity check of all executables and firmwares can be performed by comparing the hash of the component to the hash in the repository.
	Verify that the integrity check is performed by the device at least once every 30 days.
	Verify that the integrity check cannot be bypassed by any means.
IEC 62443-4-2 HDR 3.14	Verify that applications can undergo an integrity check by comparing them to a known source, either a full copy or a hash.
	Verify that the device or system has the capability to generate and maintain a hash of the currently installed executables and firmwares.
	Verify the hash used is SHA2 and stored at secure location, which is not accessible to users.
	Verify that the device maintains a hash of the currently installed software and firmware, including patches.
	Verify that any update to the executables and firmwares results in a recalculation and update of this hash.
	Verify that an integrity check of all executables and firmwares can be performed by comparing the hash of the component to the hash in the repository.
	Verify that the integrity check cannot be bypassed by any means.
IEC 62443-4-2 HDR 3.14 RE 1	Verify firmware executables contain a Digital Signature from a trusted CA and are thus not self-signed.
	Verify that the Digital Signature Public Key Algorithm is RSA, DSA or ECDSA.
	Verify that the minimum key length for the digital signature is 2048-bits for asymmetric RSA and DSA, and 224 for ECDSA.
	Verify that the hashing algorithm used for the digital signature is SHA256.

	<p>Verifying firmware with a bad digital signature will result in an error being logged and the default firmware being employed.</p>
	<p>Verify that the authenticity check is implemented properly in the device. The device should not accept unsigned firmware as well as firmware signed with a self-signed certificate on any interface.</p>
<p>IEC 62443-4-2 NDR 1.6</p>	<p>Verify the device uses encryption for wireless access.</p>
	<p>In case of using WiFi, verify the WiFi client supports modern authentication protocols.</p>
	<p>For using other wireless protocols than WiFi, verify proper security measures are implemented as per protocol.</p>
	<p>In case of using ZigBee, verify all communication is encrypted with the Network (NWK) key.</p>
	<p>In case of using Bluetooth Low Energy (BLE), verify the application uses BLE 5.3 or later release.</p>
	<p>In case of using Bluetooth Low Energy (BLE), verify the application uses best feasible security mode and level based on i/o capabilities of devices.</p>
	<p>In case of using Bluetooth Low Energy (BLE), verify all keys and data are sent over encrypted link only and encryption key size is configured to maximum allowable.</p>
	<p>In case of using Bluetooth Low Energy (BLE) mesh, verify, on node removal, that the application blacklists the device and refreshes the key.</p>
	<p>In case of using ZigBee, verify no publicly known keys are used in the network.</p>
	<p>In case of using ZigBee, verify rejoin requests are securely handled.</p>
	<p>In case of using ZigBee, verify insecure rejoin requests are disabled or not responded to.</p>
	<p>In case of using ZigBee, verify network communication is not permitted with a modified Network (NWK) key.</p>

	In case of using ZigBee, verify previously captured network traffic cannot be replayed.
IEC 62443-4-2 NDR 1.6 RE 1	Verify users are uniquely identified.
	Verify that the system is capable of uniquely identifying and authenticating device(s) before establishing network connections.
	Verify that Organizational Authentication solutions are used to identify and authenticate devices on local and/or wide area networks.
	Verify that device-to-device identification and authentication when Standard Industrial protocols do not support cryptographic identification and authentication.
IEC 62443-4-2 NDR 1.13	Verify access is restricted to privileged functions such as hardware resets and sudo commands.
	Verify access is restricted to security information.
	Verify that the device interface and ports are protected from unauthorized access.
	Verify that Field Tools require access control to utilize the tool(s).
IEC 62443-4-2 NDR 1.13 RE 1	Verify access is restricted to privileged functions such as hardware resets and sudo commands.
	Verify access is restricted to security information.
	Verify that there is no method to bypass the Role Based Access Control or Access Control List.
	Verify that the device interface and ports are protected from unauthorized access.
	Verify that Field Tools require access control to utilize the tool(s).
	Verify device/user authentication mechanism is implemented on all network interfaces that exposes sensitive information or admin level functionality.
IEC 62443-4-2 NDR 2.4	Verify automated execution of code from portable devices or media is disabled.

	<p>Enumerate all the code that is downloaded and active on mobile devices during runtime, such as JavaScript, ActiveX, Flash, Java Applets, and others.</p>
	<p>Verify that mobile and active code technologies, including JavaScript, ActiveX, Flash, and Java Applets, are only permitted to run or be accepted from components deployed on local servers or user networks, and not directly from the internet.</p>
	<p>Verify the device controls which users (human, software process, or device) are allowed to transfer mobile code to and from the application.</p>
IEC 62443-4-2 NDR 2.4 RE 1	<p>Verify the device controls the execution of mobile code based on the results of an integrity check prior to the code being executed.</p>
IEC 62443-4-2 NDR 2.13	<p>Enumerate all the local interfaces, including physical ones at board level (such as JTAG, SPI, I2C, FTDI, OBDII) and device level (such as USB, Serial).</p>
	<p>Verify that each enumerated interface either has authentication or has been rendered inoperable.</p>
	<p>Verify all interfaces and services not required for operations are disabled by default.</p>
	<p>Verify that no insecure services such as HTTP, FTP, or Telnet are being used and that they are disabled by default at minimum.</p>
	<p>If continued developer access is necessary, verify that any developer-level debugging interfaces are appropriately protected to limit access to authorized privileged users.</p>
IEC 62443-4-2 NDR 2.13 RE 1	<p>Verify active monitoring of diagnostic and test interfaces are implemented.</p>
IEC 62443-4-2 NDR 3.2	<p>Verify portable devices and media are disabled by default.</p>
	<p>Verify portable devices and media can be enabled only by specifically authorized users.</p>



	<p>Verify that mobile and active code technologies, including JavaScript, ActiveX, Flash, and Java Applets, are only permitted to run or be accepted from components deployed on local servers or user networks, and not directly from the internet.</p>
	<p>Verify the device controls the execution of mobile code based on the results of an integrity check prior to the code being executed.</p>
<p>IEC 62443-4-2 NDR 3.10</p>	<p>Verify that the product team has a mechanism for device upgrades in field such as for firmware and hardware.</p>
	<p>Verify that the system can update and patch the latest updates.</p>
	<p>Perform an initial configuration sequence from factory default settings.</p>
	<p>Verify that system components check for system updates prior to final activation and operation.</p>
	<p>Verify that the user is at least given the option to apply new updates, prior to operation.</p>
	<p>Verify that the product team has a mechanism for patching devices in field, in the event of a critical vulnerability discovery.</p>
	<p>Verify that any Over-The-Air upgrade process is secure.</p>
	<p>Verify the product supports security updates to the product's software.</p>
	<p>Verify security updates are possible in both online and offline network modes.</p>
	<p>Verify that product supports reverting to previously installed version if an update is unsuccessful.</p>
	<p>Verify that before deployment of the software or firmware components to the product, the download of the software or firmware components is completed.</p>
	<p>Verify that download of the software or firmware components to the product does not interrupt the continued operation of the product as intended.</p>

IEC 62443-4-2 NDR 3.10 RE 1	Verify that the authenticity check is implemented properly in the device. The device should not accept unsigned firmware as well as firmware signed with a self-signed certificate on any interface.
IEC 62443-4-2 NDR 3.11	<p>Verify access to the electronics are not easily compromised, for example ensure there are no exposed fasteners, screws, or other compromisable components.</p> <p>Visually inspect and enumerate any tamper resistant measures.</p> <p>Verify the effectiveness of each physical tamper resistant measure to deter and/or alert the system owner to tampering.</p> <p>Verify that hardware supported tamper resistance mechanisms are effective.</p>
IEC 62443-4-2 NDR 3.11 RE 1	<p>Verify that if an unauthorized change is detected to the software, the device should alert the consumer and/or administrator to an issue and should not connect to wider networks than those necessary to perform the alerting function.</p> <p>Verify the effectiveness of each physical tamper resistant measure to deter and/or alert the system owner to tampering.</p>
IEC 62443-4-2 NDR 3.12	<p>Verify the system protects “Root of trust” data via hardware mechanisms, preventing any modification of the data during normal operations of the component.</p> <p>Verify that the device or system has the capability to generate and maintain a hash of the currently installed executables and firmwares.</p> <p>Verify the hash used is SHA2 and stored at secure location, which is not accessible to users.</p> <p>Verify the integrity check is logged.</p> <p>Verify that the device maintains a hash of the currently installed software and firmware, including patches.</p> <p>Verify that any update to the executables and firmwares results in a recalculation and update of this hash.</p>

	<p>Verify that an integrity check of all executables and firmwares can be performed by comparing the hash of the component to the hash in the repository.</p>
	<p>Verify that the integrity check is performed by the device at least once every 30 days.</p>
	<p>Verify that the integrity check cannot be bypassed by any means.</p>
	<p>Verify that software is verified using secure boot mechanisms, which require a hardware root of trust.</p>
IEC 62443-4-2 NDR 3.13	<p>Verify the system protects “Root of trust” data via hardware mechanisms, preventing any modification of the data during normal operations of the component.</p>
	<p>Verify that the device or system has the capability to generate and maintain a hash of the currently installed executables and firmwares.</p>
	<p>Verify the hash used is SHA2 and stored at secure location, which is not accessible to users.</p>
	<p>Verify the integrity check is logged.</p>
	<p>Verify that the device maintains a hash of the currently installed software and firmware, including patches.</p>
	<p>Verify that any update to the executables and firmwares results in a recalculation and update of this hash.</p>
	<p>Verify that an integrity check of all executables and firmwares can be performed by comparing the hash of the component to the hash in the repository.</p>
	<p>Verify that the integrity check is performed by the device at least once every 30 days.</p>
	<p>Verify that the integrity check cannot be bypassed by any means.</p>
IEC 62443-4-2 NDR 3.14	<p>Verify that applications can undergo an integrity check by comparing them to a known source, either a full copy or a hash.</p>
	<p>Verify that the device or system has the capability to generate and maintain a hash of the currently installed executables and firmwares.</p>

	Verify the hash used is SHA2 and stored at secure location, which is not accessible to users.
	Verify that the device maintains a hash of the currently installed software and firmware, including patches.
	Verify that any update to the executables and firmwares results in a recalculation and update of this hash.
	Verify that an integrity check of all executables and firmwares can be performed by comparing the hash of the component to the hash in the repository.
	Verify that the integrity check cannot be bypassed by any means.
IEC 62443-4-2 NDR 3.14 RE 1	Verify that the device or system has the capability to generate and maintain a hash of the currently installed executables and firmwares.
	Verify the hash used is SHA2 and stored at secure location, which is not accessible to users.
	Verify that the device or system maintains a complete image of all currently deployed software.
	Verify that the device maintains a hash of the currently installed software and firmware, including patches.
	Verify that any update to the executables and firmwares results in a recalculation and update of this hash.
	Verify that an integrity check of all executables and firmwares can be performed by comparing the hash of the component to the hash in the repository.
	Verify that the integrity check cannot be bypassed by any means.
	Verify firmware executables contain a Digital Signature from a trusted CA and are thus not self-signed.
	Verify that the Digital Signature Public Key Algorithm is RSA, DSA or ECDSA.
	Verify that the minimum key length for the digital signature is 2048-bits for asymmetric RSA and DSA, and 224 for ECDSA.

	<p>Verify that the hashing algorithm used for the digital signature is SHA256.</p> <p>Verifying firmware with a bad digital signature will result in an error being logged and the default firmware being employed.</p> <p>Verify that the authenticity check is implemented properly in the device. The device should not accept unsigned firmware as well as firmware signed with a self-signed certificate on any interface.</p>
IEC 62443-4-2 NDR 5.2	Verify the system supports security controls to control and monitor zone boundary communication.
IEC 62443-4-2 NDR 5.2 RE 1	<p>Verify that the system controls and filters all traffic passing between network segments using 'deny unless specifically permitted' policies.</p> <p>Verify that permit rules are restricted to the smallest number of endpoints, workstations, devices, and services possible.</p>
IEC 62443-4-2 NDR 5.2 RE 2	<p>Verify that the network component provides the capability to protect against any communication through the control system boundary when there is an operational failure of the boundary protection mechanisms, using the fail close approach.</p> <p>Verify that the network component provides the capability to protect against any communication through the control system boundary (also termed island mode).</p>
IEC 62443-4-2 NDR 5.2 RE 3	Verify that the network component provides the capability to protect against any communication through the control system boundary when there is an operational failure of the boundary protection mechanisms, using the fail close approach.
IEC 62443-4-2 NDR 5.3	Verify that the network device at a zone boundary provides the capability to protect against general purpose, person-to-person messages from being received from users or systems external to the control system.

**Table 26: Mapping of test cases to requirements of IEC 62443-3-3.**

Requirement Source	Test Case(s)
IEC 62443-3-3 SR 1.1	Verify user authentication mechanism is implemented on all network interfaces that exposes sensitive information or admin level functionality.
	Enumerate all authentication and authorization methods.
IEC 62443-3-3 SR 1.1 RE 1	Verify users are uniquely identified.
IEC 62443-3-3 SR 1.1 RE 2	Verify at least two-factor authentication is implemented wherever applicable and required.
IEC 62443-3-3 SR 1.1 RE 3	Verify at least two-factor authentication is implemented wherever applicable and required.
IEC 62443-3-3 SR 1.2	Verify that the system is capable of uniquely identifying and authenticating device(s) before establishing network connections.
	Verify that Organizational Authentication solutions are used to identify and authenticate devices on local and/or wide area networks.
	Verify that device-to-device identification and authentication when Standard Industrial protocols do not support cryptographic identification and authentication.
IEC 62443-3-3 SR 1.2 RE 1	Verify that the system is capable of uniquely identifying and authenticating device(s) before establishing network connections.
	Verify that Organizational Authentication solutions are used to identify and authenticate devices on local and/or wide area networks.
	Verify that device-to-device identification and authentication when Standard Industrial protocols do not support cryptographic identification and authentication.
IEC 62443-3-3 SR 1.3	Verify that each account within a system is tied to an individual user for proper auditing, management, and tracking, and that shared accounts are not used.
	Verify that Super User accounts, such as Administrator or Root, are disabled or removed wherever possible.
	Verify that the identity requirements for user registration are aligned with business and security requirements.
	Verify that the proper identification and authorization process is followed while provisioning of accounts.

IEC 62443-3-3 SR 1.3 RE 1	Verify that consistent account management mechanisms are employed across the system.
	Verify that the management of accounts is deployed locally in the relevant components of control system
IEC 62443-3-3 SR 1.4	Enumerate the users and roles of all resources.
IEC 62443-3-3 SR 1.5	Verify stored passwords are encrypted/hashed.
	Verify that all accounts require a password change prior to continuing with the install.
	Verify sensitive password files /etc/shadow and /etc/passwd are accessible only by the Root account.
	Verify that passwords are not hardcoded in the source code, firmware, configuration tools, registry keys, scripts, or any other system components.
	Verify the password can be changed for a default account.
	Enumerate all password storage locations, including text files, databases, and binary objects.
	Verify that the file, database, or object that is used to maintain passwords is only write-able by the application.
	Verify that a password change requires the current password to be entered.
	Go through the installation process and document all accounts and passwords used during the process, if applicable.
	Verify that the default account's username cannot be changed.
	Verify default credentials are not used.
Verify passwords are unique and not resettable to any universal factory default value.	
IEC 62443-3-3 SR 1.5 RE 1	Verify Hardware Security Modules or Trusted Platform Modules can be used for user identification and authentication.
IEC 62443-3-3 SR 1.6	Verify the device uses encryption for wireless access.
	In case of using WiFi, verify the WiFi client supports modern authentication protocols.
	For using other wireless protocols than WiFi, verify proper security measures are implemented as per protocol.
	In case of using ZigBee, verify all communication is encrypted with the Network (NWK) key.
	In case of using Bluetooth Low Energy (BLE), verify the application uses BLE 5.3 or later release.
	In case of using Bluetooth Low Energy (BLE), verify the application uses best feasible security mode and level based on i/o capabilities of devices.

	In case of using Bluetooth Low Energy (BLE), verify all keys and data are sent over encrypted link only and encryption key size is configured to maximum allowable.
	In case of using Bluetooth Low Energy (BLE) mesh, verify, on node removal, that the application blacklists the device and refreshes the key.
	In case of using ZigBee, verify no publicly known keys are used in the network.
	In case of using ZigBee, verify rejoin requests are securely handled.
	In case of using ZigBee, verify insecure rejoin requests are disabled or not responded to.
	In case of using ZigBee, verify network communication is not permitted with a modified Network (NWK) key.
	In case of using ZigBee, verify previously captured network traffic cannot be replayed.
IEC 62443-3-3 SR 1.6 RE 1	Verify users are uniquely identified.
	Verify that the system is capable of uniquely identifying and authenticating device(s) before establishing network connections.
	Verify that Organizational Authentication solutions are used to identify and authenticate devices on local and/or wide area networks.
	Verify that device-to-device identification and authentication when Standard Industrial protocols do not support cryptographic identification and authentication.
IEC 62443-3-3 SR 1.7	Verify password complexity policies are configurable and enforceable.
	Verify that, in case a credential is a password, its minimum length is 6 characters.
IEC 62443-3-3 SR 1.7 RE 1	Verify that passwords can either be expired based on a specific time interval or restricted from reuse.
	Verify password complexity policies are configurable and enforceable.
IEC 62443-3-3 SR 1.7 RE 2	Verify that passwords can either be expired based on a specific time interval or restricted from reuse.
IEC 62443-3-3 SR 1.8	Verify that certificates for PKI-based authentication are validated against a trust anchor.
IEC 62443-3-3 SR 1.9	Verify that certificates for PKI-based authentication are validated against a trust anchor.
	Verify that authorized access is enforced to access or use the private key for PKI-based authentication.
	Verify that the authenticated identity is mapped to the corresponding user account for PKI-based authentication.



IEC 62443-3-3 SR 1.9 RE 1	Verify that Hardware Security Modules (HSMs) or Trusted Platform Modules (TPMs) in use are physically protected.
	Verify that the failure of the cryptographic module authentication fails secure.
	Verify the system protects "Root of trust" data via hardware mechanisms, preventing any modification of the data during normal operations of the component.
IEC 62443-3-3 SR 1.10	Verify error messages do not specify whether a user exists.
	Verify error messages do not specify a locked out account.
	Verify error messages do not specify a maximum number of failed attempts tried before locking.
	Verify the system mitigates account enumeration and guessable user accounts.
IEC 62443-3-3 SR 1.11	Verify unsuccessful login attempts are limited to a maximum number of tries within a time period.
	Verify an unsuccessful login attempt automatically locks the account for a specified period of time.
	Verify additional login attempts are delayed.
	Verify only an administrator can unlock an account when the maximum number of unsuccessful attempts is exceeded.
IEC 62443-3-3 SR 1.12	Verify that the device or system displays an 'Appropriate Use' warning banner to warn-off unauthorized users when required.
	Verify appropriate warning messages are displayed.
IEC 62443-3-3 SR 1.13	Verify access is restricted to privileged functions such as hardware resets and sudo commands.
	Verify access is restricted to security information.
	Verify that the device interface and ports are protected from unauthorized access.
	Verify that Field Tools require access control to utilize the tool(s).
IEC 62443-3-3 SR 1.13 RE 1	Verify access is restricted to privileged functions such as hardware resets and sudo commands.
	Verify access is restricted to security information.
	Verify that there is no method to bypass the Role Based Access Control or Access Control List.
	Verify that the device interface and ports are protected from unauthorized access.
	Verify that Field Tools require access control to utilize the tool(s).
	Verify device/user authentication mechanism is implemented on all network interfaces that exposes sensitive information or admin level functionality.

<p>IEC 62443-3-3 SR 2.1</p>	<p>Verify that the level of privileges assigned to all processes/services are identified by using the list of enumerated processes/services. This includes determining the service level account allocated to each process/service, such as local service, network service, or local system.</p> <p>Verify that all non-root or non-admin accounts have read-only permissions or less for important files, such as /etc/sunders, /etc/ssh/sshd_config, /etc/group, /etc/shadow, /etc/passwd, Program Data etc.</p> <p>Verify users have a minimal write permissions.</p> <p>Verify that protection against directory traversal/file include attacks is implemented.</p> <p>Verify the system mitigates unauthorized access to admin functions.</p> <p>Verify that privileges for Set User ID accounts are raised as late as possible and released as soon as possible. This may require access to code to confirm, or reverse engineer using a tool such as Interactive Disassembler (IDA).</p>
<p>IEC 62443-3-3 SR 2.1 RE 1</p>	<p>Verify all web pages and resources by default require authentication, except those specifically intended to be public.</p> <p>Verify all authentication controls are enforced on the server side, and not on the client.</p> <p>Verify access is restricted to privileged functions such as hardware resets and sudo commands.</p> <p>Verify access is restricted to security information.</p> <p>Verify access is restricted based on user, role, and attributes.</p> <p>Verify that there is no method to bypass the Role Based Access Control or Access Control List.</p> <p>Verify that the device interface and ports are protected from unauthorized access.</p> <p>Verify that Field Tools require access control to utilize the tool(s).</p> <p>Verify a mechanism for device or user authentication is implemented.</p> <p>Verify device/user authentication mechanism is implemented on all network interfaces that exposes sensitive information or admin level functionality.</p> <p>Verify that guest and training accounts have limited access permissions and cannot perform any actions that compromise the security or integrity of the application or its users.</p> <p>Enumerate all the processes and services running in the device or system using a root shell.</p>

	<p>Verify that the level of privileges assigned to all processes/services are identified by using the list of enumerated processes/services. This includes determining the service level account allocated to each process/service, such as local service, network service, or local system.</p>
	<p>Verify each process and service is not running as root for Linux systems.</p>
	<p>Verify each process and service is not running as LOCAL SYSTEM for Windows systems.</p>
	<p>Verify that administrative interfaces are not accessible to untrusted parties.</p>
	<p>Verify that non-root accounts are used for network services.</p>
	<p>Verify that all service accounts have shell access disabled by verifying that the shell is set to "/dev/null" in the passwd file</p>
IEC 62443-3-3 SR 2.1 RE 2	<p>Verify access is restricted based on user, role, and attributes.</p>
	<p>Verify that there is no method to bypass the Role Based Access Control or Access Control List.</p>
	<p>Verify that the system has defined roles for admin and regular users, and that the permission matrix is mapped according to the hierarchy of the roles.</p>
	<p>Verify device/user authentication mechanism is implemented on all network interfaces that exposes sensitive information or admin level functionality.</p>
IEC 62443-3-3 SR 2.1 RE 3	<p>Verify that the system supports supervisor manual override of current authorizations.</p>
IEC 62443-3-3 SR 2.1 RE 4	<p>Verify if the system supports dual approval checks for critical actions.</p>
IEC 62443-3-3 SR 2.2	<p>In case of using Bluetooth Low Energy (BLE), verify the application uses BLE 5.3 or later release.</p>
	<p>In case of using Bluetooth Low Energy, verify advertisements are transmitted evenly across all three advertising channels and advertisement data is encrypted.</p>
	<p>In case of using Bluetooth Low Energy (BLE), verify the application uses best feasible security mode and level based on i/o capabilities of devices.</p>
	<p>In case of using Bluetooth Low Energy, verify unneeded and unapproved services and profiles are disabled.</p>
	<p>In case of using Bluetooth Low Energy (BLE) mesh, verify, on node removal, that the application blacklists the device and refreshes the key.</p>
	<p>Enumerate all remote interfaces, including network ports and services.</p>

	Enumerate all the local interfaces, including physical ones at board level (such as JTAG, SPI, I2C, FTDI, OBDII) and device level (such as USB, Serial).
	Enumerate all wireless interfaces (such as WiFi, ZigBee, Bluetooth, etc.).
	Verify that each enumerated interface either has authentication or has been rendered inoperable.
	Verify that no insecure services such as HTTP, FTP, or Telnet are being used and that they are disabled by default at minimum.
	Verify only secure services are used, such as HTTPS, SFTP, and SSH.
	Verify that wireless radios such as WiFi and Bluetooth are disabled by default.
	Verify Secure DNS is used instead of DNS.
	Verify Secure DNP3 is utilized if DNP3 is supported.
	Verify Secure Tunneling is used for unsecured legacy protocols.
	If continued developer access is necessary, verify that any developer-level debugging interfaces are appropriately protected to limit access to authorized privileged users.
IEC 62443-3-3 SR 2.2 RE 1	Verify if the system supports detection and reporting of unauthorized devices transmitting within the system physical environment.
IEC 62443-3-3 SR 2.3	Verify portable devices and media are disabled by default.
	Verify portable devices and media can be enabled only by specifically authorized users.
IEC 62443-3-3 SR 2.3 RE 1	Verify access control is in place to restrict devices that do not adhere to the security requirements of a zone.
IEC 62443-3-3 SR 2.4	Verify automated execution of code from portable devices or media is disabled.
	Enumerate all the code that is downloaded and active on mobile devices during runtime, such as JavaScript, ActiveX, Flash, Java Applets, and others.
	Verify that mobile and active code technologies, including JavaScript, ActiveX, Flash, and Java Applets, are only permitted to run or be accepted from components deployed on local servers or user networks, and not directly from the internet.
	Verify the device controls which users (human, software process, or device) are allowed to transfer mobile code to and from the application.
IEC 62443-3-3 SR 2.4 RE 1	Verify the device controls the execution of mobile code based on the results of an integrity check prior to the code being executed.

IEC 62443-3-3 SR 2.5	Verify configurations cannot be changed while the session is locked.
	Identify all authentication methods and session management techniques.
	Verify a session can be re-established only after completing the identification and authentication procedures.
	Verify the Session Management Schema can not be bypassed.
	Verify adequate logout functionality is implemented.
	Verify a session is terminated on server side, and not on the client side.
	Verify a local session is automatically terminated after an appropriate set time of inactivity.
IEC 62443-3-3 SR 2.6	Verify that if remote access, for example, SSH, SFTP, etc., is provisioned, remote access sessions are encrypted.
	Verify the cryptographic libraries used for encryption are FIPS-140-2 compliant.
	Verify remote access is logged.
	Verify remote administrative activities are logged. Examples are configurations changes, settings, upgrades, etc.
	Verify remote maintenance activities are logged. Examples are diagnostics, reading logs, preventive maintenance, etc.
	Verify a remote session is automatically terminated after a set period of time of inactivity.
	If SSH/Remote Access is enabled, then verify that PermitRootLogin is disabled or configured to Without-Password.
IEC 62443-3-3 SR 2.7	Verify concurrent logins are disallowed by default.
IEC 62443-3-3 SR 2.8	Verify that the system or the device has an audit log that can be exported, in a business readable format.
	Verify that audit events including logon and logoff attempts, configuration changes, upgrades, and other related activities are logged with the date and time stamp using the WHEN parameter.
	Verify that audit events including logon and logoff attempts, configuration changes, upgrades, and other related activities are logged with the user-identity using the WHO parameter.
	Verify that audit events including logon and logoff attempts, configuration changes, upgrades, and other related activities are logged with the event type using the WHAT parameter.

	Verify that audit events including logon and logoff attempts, configuration changes, upgrades, and other related activities are logged with the location of the event using the WHERE parameter.
	Verify that appropriate access controls are in place for event log data and that access to logs is restricted based on a need-to-know basis.
	Verify that the event of deleting logs is recorded, either in the system or at the beginning of a new log.
IEC 62443-3-3 SR 2.8 RE 1	Verify the system can centrally manage audit events.
IEC 62443-3-3 SR 2.9	Verify that logging cannot be used to deplete system resources, for example by filling up disk space or exceeding database log space, leading to Denial of Service (DoS).
IEC 62443-3-3 SR 2.9 RE 1	Verify the device gives a warning when the audit record storage capacity threshold is reached.
IEC 62443-3-3 SR 2.10	Verify logs are retained after a local power outage.
	Verify logs are retained after a device or system reboot.
	Verify the application continues to function even when audit logging failed.
	Verify appropriate action is taken when audit logging failed according to commonly accepted industry practices and recommendations.
IEC 62443-3-3 SR 2.11	Verify that audit events including logon and logoff attempts, configuration changes, upgrades, and other related activities are logged with the date and time stamp using the WHEN parameter.
	Verify periodic time-shift events, such as daylight savings time in some locations, are considered and reflect in audit logs.
IEC 62443-3-3 SR 2.11 RE 1	Verify components shall provide the capability to create timestamps that are synchronized with a system wide time source e.g. NTP server.
	Verify the network device compares internal information system clocks at a configurable time with an authoritative time server.
IEC 62443-3-3 SR 2.11 RE 2	Verify that logs are restricted to authorized users only and are immutable.
	Verify periodic time-shift events, such as daylight savings time in some locations, are considered and reflect in audit logs.
	Verify that the time synchronization mechanism provides the capability to detect unauthorized alteration and causes an audit event upon alteration.

IEC 62443-3-3 SR 2.12	Verify that audit events including logon and logoff attempts, configuration changes, upgrades, and other related activities are logged with the user-identity using the WHO parameter.
	Verify that audit events including logon and logoff attempts, configuration changes, upgrades, and other related activities are logged with the event type using the WHAT parameter.
IEC 62443-3-3 SR 2.12 RE 1	Verify that audit events including logon and logoff attempts, configuration changes, upgrades, and other related activities are logged with the user-identity using the WHO parameter.
	Verify that audit events including logon and logoff attempts, configuration changes, upgrades, and other related activities are logged with the event type using the WHAT parameter.
IEC 62443-3-3 SR 3.1	Verify message authentication is implemented at the protocol level, if the protocol supports it.
	Using a enumerated list of services, for each authentication request, intercept at least one sample.
	Verify by sniffing that credentials are transported using a suitable encrypted link and that all pages and functions that require a user to enter credentials are done so using an encrypted link.
	Verify that the device adheres to the relevant security standards for each protocol in case message authentication is not implemented at the protocol level.
	Verify TLS 1.2 or TLS 1.3 is used for encryption over network for HTTPS.
	Verify HTTP Strict Transport Security (HSTS) implementation by ensuring the server has a valid SSL/TLS certificate and confirming the presence of the HSTS header.
	Verify that adequate mechanisms are implemented to ensure communication integrity and confidentiality on all protocols that do not support cryptography.
	Verify that there exists a mechanism to preserve and check data integrity on a Controller Area Network (CAN) Bus or any other similar network.
IEC 62443-3-3 SR 3.1 RE 1	Verify message authentication is implemented at the protocol level, if the protocol supports it.
	Using a enumerated list of services, for each authentication request, intercept at least one sample.
	Verify by sniffing that credentials are transported using a suitable encrypted link and that all pages and functions that require a user to enter credentials are done so using an encrypted link.

	Verify that the device adheres to the relevant security standards for each protocol in case message authentication is not implemented at the protocol level.
	Verify TLS 1.2 or TLS 1.3 is used for encryption over network for HTTPS.
	Verify HTTP Strict Transport Security (HSTS) implementation by ensuring the server has a valid SSL/TLS certificate and confirming the presence of the HSTS header.
IEC 62443-3-3 SR 3.2	Verify portable devices and media are disabled by default.
	Verify portable devices and media can be enabled only by specifically authorized users.
	Verify that mobile and active code technologies, including JavaScript, ActiveX, Flash, and Java Applets, are only permitted to run or be accepted from components deployed on local servers or user networks, and not directly from the internet.
	Verify the device controls the execution of mobile code based on the results of an integrity check prior to the code being executed.
IEC 62443-3-3 SR 3.2 RE 1	Verify that the system supports malicious code mechanisms at all entry and exit points
IEC 62443-3-3 SR 3.2 RE 2	Verify if management and reporting is in place for malicious code protection mechanisms
IEC 62443-3-3 SR 3.3	Verify that the integrity check cannot be bypassed by any means.
	Verify that software is verified using secure boot mechanisms, which require a hardware root of trust.
IEC 62443-3-3 SR 3.3 RE 1	Verify that the system or the device has an audit log that can be exported, in a business readable format.
	Verify that the integrity check cannot be bypassed by any means.
	Verify that automated mechanisms to support management of security verification are in place.
IEC 62443-3-3 SR 3.3 RE 2	Verify that the integrity check cannot be bypassed by any means.
	Verify that software is verified using secure boot mechanisms, which require a hardware root of trust.
IEC 62443-3-3 SR 3.4	Enumerate all external file inputs to the device.
	Verify that unsupported file formats can not be uploaded.
	Verify input files include an integrity check, such as a Message Authentication Code (MAC) or Signature.
	Verify, if a digital signature is used, the Message Authentication Code (MAC) is either SHA, RSA, DSA, or ECDSA.



	Verify that the minimum key length for the digital signature is 2048-bits for asymmetric RSA and DSA, and 224 for ECDSA.
	Verify that the hashing algorithm used for the digital signature is SHA256.
	Verify the integrity check status is logged.
	Verify that only read-only configuration files can be uploaded by the user in case there is no integrity check mechanism for configuration files.
	Verify that the configuration file integrity mechanism cannot be bypassed in any manner.
	Verify that applications can undergo an integrity check by comparing them to a known source, either a full copy or a hash.
	Verify that the device or system has the capability to generate and maintain a hash of the currently installed executables and firmwares.
	Verify the hash used is SHA2 and stored at secure location, which is not accessible to users.
	Verify the integrity check is logged.
	Verify that the device or system maintains a complete image of all currently deployed software.
	Verify that the device maintains a hash of the currently installed software and firmware, including patches.
	Verify that any update to the executables and firmwares results in a recalculation and update of this hash.
	Verify that an integrity check of all executables and firmwares can be performed by comparing the hash of the component to the hash in the repository.
	Verify that the integrity check is performed by the device at least once every 30 days.
IEC 62443-3-3 SR 3.4 RE 1	Verify that any update to the executables and firmwares results in a recalculation and update of this hash.
	Verify that an integrity check of all executables and firmwares can be performed by comparing the hash of the component to the hash in the repository.
IEC 62443-3-3 SR 3.5	Verify that user supplied data is encoded prior to viewing in intended log viewing interface, to prevent injection attacks.
	Verify all data inputs via user interfaces and transferred via Application Programming Interfaces (APIs) or between networks in services and devices are validated.
	Verify that all data inputs from web applications, mobile apps, and PC tools undergo validation.
	Verify inputs outside the character set are disallowed in user input fields.

	Verify inputs exceeding the field length are disallowed in user input fields.
	Verify inputs outside the allowed range are disallowed in user input fields
	Verify inputs are handled as expected when fuzzed.
	Verify the device mitigates vulnerabilities in HTTP Methods.
	Verify the application only accepts logically valid data.
	Verify the application mitigates the ability to forge requests.
	Verify the application mitigates process timing vulnerabilities.
	Verify the application or system mitigates function limit vulnerabilities.
	Verify the workflows of the application cannot be circumvented.
	Verify the device has protection mechanisms against application mis-uses.
	Verify only approved file types can be uploaded in the application.
	Verify the device mitigates the upload of malicious files.
	Verify if the device mitigates JavaScript Execution.
	Verify the application mitigates CSS Injection.
	Verify the device mitigates Client Side Resource Manipulation.
	Validate the device mitigates Cross Origin Resource Sharing.
	Verify the device mitigates Cross Site Flashing.
	Verify the application mitigates Clickjacking.
	Verify the device establishes a secure WebSocket connection.
	Verify the message's origin is secure and that safe methods are used to process data and validate input.
	Verify the device mitigates Local Storage vulnerabilities.
	Verify that proper character sets, such as UTF-8 have been enforced for all user input, to prevent alternate character sets, for example allowing foreign languages that may issue injection attacks.
	Verify that saved files are not accessible through the application's web context and are instead stored securely on a content server or database.
	Verify that all output to other system components is sanitized and encoded properly before sending outside of the application trust boundary.
IEC 62443-3-3 SR 3.6	Verify that upon a failure the product's functionality is limited only to allow maintenance of the device.

	Verify upon a successful startup, only necessary ports, services, and settings are enabled by default.
IEC 62443-3-3 SR 3.7	Verify that detailed error messages are only accessible to authorized users, such as admins or maintenance personnel.
	Verify error messages do not contain exploitable information.
	Verify error messages do not contain device or system component information.
	Verify error messages do not contain information for correction.
	Verify error messages do not contain failure details.
	Verify error messages do not contain stack traces.
	Verify error messages do not contain memory locations.
	Verify error messages do not contain information outside the user's scope.
IEC 62443-3-3 SR 3.8	Identify all authentication methods and session management techniques.
	Verify a session can be re-established only after completing the identification and authentication procedures.
	Verify that user input is transmitted using secure protocols.
	Verify the Session Management Schema can not be bypassed.
	Verify secure cookie handling.
	Verify the system mitigates session hijacking.
	Verify the protection of session variables from eavesdropping and reuse of session tokens vulnerabilities.
	Verify the system mitigates Cross Site Request Forgery.
	Verify the server framework is used for session management and that the developer did not create their own.
	Verify the system is mitigates session puzzling.
Verify the protection of sensitive data during network transmission.	
IEC 62443-3-3 SR 3.8 RE 1	Verify a local session is automatically terminated after an appropriate set time of inactivity.
	Verify the server framework is used for session management and that the developer did not create their own.
IEC 62443-3-3 SR 3.8 RE 2	Verify the protection of session variables from eavesdropping and reuse of session tokens vulnerabilities.

	Verify the server framework is used for session management and that the developer did not create their own.
IEC 62443-3-3 SR 3.8 RE 3	Verify the server framework is used for session management and that the developer did not create their own.
IEC 62443-3-3 SR 3.9	Verify logs are retained after a local power outage.
	Verify logs are retained after a device or system reboot.
	Verify that logs are restricted to authorized users only and are immutable.
	Verify that appropriate access controls are in place for event log data and that access to logs is restricted based on a need-to-know basis.
IEC 62443-3-3 SR 3.9 RE 1	Verify that audit records are being produced on hardware-enforced write-once media, such as WORM drives or Non-Volatile Memory.
IEC 62443-3-3 SR 4.1	Identify all the sensitive and personal information generated, stored, used or communicated by the system.
	Enumerate all cryptographic functions of the application.
	Verify that the developer did not 'roll their own' encryption and that best practice libraries are used for all cryptographic functions.
	Verify that all sensitive or personal data is encrypted when at rest. This includes all forms of sensitive information, for example cached data in web browsers, temporary files, logs, sensitive secrets in configuration files, etc.
	Verify that device manufacturers and service providers provide consumers with clear and transparent information about how their personal data is being used.
	Verify that where personal data is processed on the basis of consumers' consent, this consent is obtained in a valid way.
	Verify that customers who gave consent for the processing of their personal data are given the opportunity to withdraw it at any time.
	Verify by sniffing that credentials are transported using a suitable encrypted link and that all pages and functions that require a user to enter credentials are done so using an encrypted link.
	Verify that adequate mechanisms are implemented to ensure communication integrity and confidentiality on all protocols that do not support cryptography.

	Verify that there exists a mechanism to preserve and check data confidentiality on a Controller Area Network (CAN) Bus or any other similar network.
IEC 62443-3-3 SR 4.1 RE 1	Verify the confidentiality of information in remote access sessions is protected.
	Verify the confidentiality of information at rest is protected.
IEC 62443-3-3 SR 4.1 RE 2	Verify the confidentiality of information traversing the zone boundaries is protected.
IEC 62443-3-3 SR 4.2	Identify all the sensitive and personal information generated, stored, used or communicated by the system.
	Verify the product documentation details a secure decommissioning of the device.
	Verify that the device sanitizes sensitive information as per the device documentation.
	Verify that recovering of any sensitive information from the decommissioned device is difficult and requires significant effort using specialized tools and skill sets.
	Verify personal data can easily be removed.
	Verify consumers are given clear instructions on how to delete their personal data, especially for IoT systems.
	Verify consumers are provided with clear confirmation that personal data has been deleted from services, devices and applications, especially for IoT systems.
IEC 62443-3-3 SR 4.2 RE 1	Verify portable devices and media can be enabled only by specifically authorized users.
	Verify automated execution of code from portable devices or media is disabled.
	Remove the removable memory and inspect the contents, verify that no sensitive data is stored on the removable memory.
	Visually inspect and enumerate any removable memory, such as SD cards.
IEC 62443-3-3 SR 4.3	Enumerate all cryptographic functions of the application.
	Verify only recommended ciphers are enabled on the server.
	Verify TLS 1.2 or TLS 1.3 is used for encryption over network for HTTPS.
	Verify SSL certificates are signed using SHA-2 (with SHA-256 or higher) hashing algorithm.
	Verify that the minimum key length for the digital signature is 2048-bits for asymmetric RSA and DSA, and 224 for ECDSA.
	Verify the cryptography does not degrade the performance of the system.

	Verify the failure of the cryptography does not result in a Denial of Service (DoS).
	Verify used encryption does not support or fallback to insecure ciphers.
	Verify that any connection does not accept invalid certificates.
IEC 62443-3-3 SR 5.1	Verify that the system has the capability to logically segment system networks.
IEC 62443-3-3 SR 5.1 RE 1	Verify that the system has the capability to physically segment system networks.
IEC 62443-3-3 SR 5.1 RE 2	Verify that the system supports stand-alone network services.
	Verify the system is isolated by using a network scanning tool to scan the system networks and verify that no non-system connections or services are detected.
IEC 62443-3-3 SR 5.1 RE 3	Verify that networks can be logically and physically isolated.
	Verify that critical system networks are only accessible by authorized networks.
IEC 62443-3-3 SR 5.2	Verify the system supports security controls to control and monitor zone boundary communication.
IEC 62443-3-3 SR 5.2 RE 1	Verify that the system controls and filters all traffic passing between network segments using 'deny unless specifically permitted' policies.
	Verify that permit rules are restricted to the smallest number of endpoints, workstations, devices, and services possible.
IEC 62443-3-3 SR 5.2 RE 2	Verify that the network component provides the capability to protect against any communication through the control system boundary when there is an operational failure of the boundary protection mechanisms, using the fail close approach.
	Verify that the network component provides the capability to protect against any communication through the control system boundary (also termed island mode).
IEC 62443-3-3 SR 5.2 RE 3	Verify that the network component provides the capability to protect against any communication through the control system boundary when there is an operational failure of the boundary protection mechanisms, using the fail close approach.
IEC 62443-3-3 SR 5.3	Verify that the network device at a zone boundary provides the capability to protect against general purpose, person-to-person messages from being received from users or systems external to the control system.

IEC 62443-3-3 SR 5.3 RE 1	Verify that the network device at a zone boundary provides the capability to protect against general purpose, person-to-person messages from being received from users or systems external to the control system.
	Verify the system supports security controls to control and monitor zone boundary communication.
IEC 62443-3-3 SR 5.4	Verify that applications and services can be physically or logically isolated.
IEC 62443-3-3 SR 6.1	Verify that logs are restricted to authorized users only and are immutable.
IEC 62443-3-3 SR 6.1 RE 1	Verify that the system or the device has an audit log that can be exported, in a business readable format.
IEC 62443-3-3 SR 6.2	Verify that the system supports security monitoring mechanisms.
IEC 62443-3-3 SR 7.1	Verify a single user cannot overload a system with certain requests, such as the recording and sending of the same packets repetitively.
IEC 62443-3-3 SR 7.1 RE 1	Verify that the device has measures in place to prevent or minimize the impact of denial-of-service attacks, such as excessive network traffic, log flooding, and application/protocol traffic.
	Verify that request throttling is in place to prevent automated attacks against common authentication attacks such as brute force attacks or denial of service attacks.
IEC 62443-3-3 SR 7.1 RE 2	Verify that there exists some mechanism to prevent DoS attacks such as CANBus Flooding, Overloading, and others.
	Verify that there exists some protective mechanism that can prevent DoS attacks, which involve flooding the network with excessive data or using unauthorized applications from remote devices to disrupt the system.
IEC 62443-3-3 SR 7.2	Verify that the device has measures in place to prevent or minimize the impact of denial-of-service attacks, such as excessive network traffic, log flooding, and application/protocol traffic.
	Verify that there exists some protective mechanism that can prevent DoS attacks, which involve flooding the network with excessive data or using unauthorized applications from remote devices to disrupt the system.
IEC 62443-3-3 SR 7.3	Verify there are backup mechanisms implemented for system components, such as application servers, database servers, and others.
	Verify that the backup mechanism covers all critical system components.
	Verify the backup mechanism is fully documented.

	Verify the recovery mechanism is fully documented.
	Verify the recovery mechanism has been tested.
IEC 62443-3-3 SR 7.3 RE 1	Verify that any backups containing Personal Identifiable Information (PII) or sensitive data are encrypted.
	Verify that the backups are prevented from unauthorized access.
IEC 62443-3-3 SR 7.3 RE 2	Verify that a backup frequency is defined for the system.
IEC 62443-3-3 SR 7.4	Verify that product performs the software integrity test before loading any applications or performing any functions of the product.
	Simulate a product failure by tampering one of the software components in the data store and verify that it fails the software integrity test on initial power up.
	Verify that upon a failure the product enters a failure mode which clearly indicates to the user that the product has failed to start up successfully.
	Verify the system starts up in a defined known state.
	Verify the system has a defined known secure state.
	Verify the system fails to the known secure state.
	Verify a device failed to the known state does not allow access without verifying credentials.
	Verify that a control system has been successfully recovered and reconstituted to a known secure state by checking that all system parameters are set to secure values, security-critical patches are reinstalled, security-related configuration settings are reestablished, system documentation and operating procedures are available, application and system software is reinstalled and configured with secure settings, information from the most recent, known secure backups is loaded, and the system is fully tested and functional.
IEC 62443-3-3 SR 7.5	Perform a power failure simulation test.
IEC 62443-3-3 SR 7.6	Enumerate all services, components, and operating systems.
	Verify all enumerated components are compliant with standard secure baseline configurations such as NIST, CIS, DISA, and others.
	Verify the system provides the capability to be configured according to recommended network and security configurations as described in guidelines provided by the control system supplier.
	Verify the system has an interface to the currently deployed network and security configuration settings.



IEC 62443-3-3 SR 7.6 RE 1	Verify the system can generate a report listing the currently deployed security settings in a machine-readable format.
IEC 62443-3-3 SR 7.7	Enumerate additional services that are installed, but not enabled.
	Verify that all additional services which are installed, but not required for operations are removed.
	If continued developer access is necessary, verify that any developer-level debugging interfaces are appropriately protected to limit access to authorized privileged users.
	Verify there are not any unnecessary network services.
IEC 62443-3-3 SR 7.8	Verify that there is a designated system capable of retrieving and storing data from all system components either daily or upon request.
	Verify that the data retrieved from all system components includes information such as current version numbers, installation dates, configuration settings, and patch levels.

**Table 27: Mapping of test cases to requirements of EN 303 645 [7].**

Requirement Source	Requirement Text	Test Cases
Provision 5.1-1	Where passwords are used and in any state other than the factory default, all consumer IoT device passwords shall be unique per device or defined by the user.	Verify that all accounts require a password change prior to continuing with the install.
		Verify users are uniquely identified.
		Verify passwords are unique and not resettable to any universal factory default value.
		Verify that the default account's username cannot be changed.
		Verify the password can be changed for a default account.
Provision 5.1-2	Where pre-installed unique per device passwords are used, these shall be generated with a mechanism that reduces the risk of automated attacks against a class or type of device.	Verify password complexity policies are configurable and enforceable.
		Verify that, in case a credential is a password, its minimum length is 6 characters.
		Verify that the file, database, or object that is used to maintain passwords is only write-able by the application.
		Verify default passwords are sufficiently randomized.
		Go through the installation process and document all accounts and passwords used during the process, if applicable.
Provision 5.1-3	Authentication mechanisms used to <b>authenticate</b> users against a device shall use <b>best practice cryptography</b> , appropriate to the properties of the technology, risk and usage.	Verify user authentication mechanism is implemented on all network interfaces that exposes sensitive information or admin level functionality.
		Enumerate all authentication and authorization methods.
		Verify stored passwords are encrypted/hashed.
		Verify sensitive password files /etc/shadow and /etc/passwd are accessible only by the Root account.
		Verify message authentication is implemented at the protocol level, if the protocol supports it.
		Using a enumerated list of services, for each authentication request, intercept at least one sample.

		Verify by sniffing that credentials are transported using a suitable encrypted link and that all pages and functions that require a user to enter credentials are done so using an encrypted link.
		Verify that the device adheres to the relevant security standards for each protocol in case message authentication is not implemented at the protocol level.
		Verify TLS 1.2 or TLS 1.3 is used for encryption over network for HTTPS.
		Verify HTTP Strict Transport Security (HSTS) implementation by ensuring the server has a valid SSL/TLS certificate and confirming the presence of the HSTS header.
		Verify that adequate mechanisms are implemented to ensure communication integrity and confidentiality on all protocols that do not support cryptography.
Provision 5.1-4	Where a user can authenticate against a device, the device shall provide to the user or an administrator a simple mechanism to change the authentication value used.	Verify the password can be changed for a default account.
		Verify that a password change requires the current password to be entered.
		Verify secure password change and reset functionalities.
		Verify secure use of security question and answer.
		Verify that all accounts require a password change prior to continuing with the install.
Provision 5.1-5	When the device is not a constrained device, it shall have a mechanism available which makes bruteforce attacks on authentication mechanisms via network interfaces impracticable.	Verify unsuccessful login attempts are limited to a maximum number of tries within a time period.
		Verify an unsuccessful login attempt automatically locks the account for a specified period of time.
		Verify additional login attempts are delayed.
		Verify only an administrator can unlock an account when the maximum number of unsuccessful attempts is exceeded.
		Verify default credentials are not used.
		Enumerate all password storage locations, including text files, databases, and binary objects.
		Using enumerated list of services, for each authenticated service, brute force the credentials using automated tools.

		Verify there is no hardware bypass of passwords, for example jumpers or switches.
		Verify there is no mechanism to defeat or circumvent the ID/password control.
		Verify that passwords are not hardcoded in the source code, firmware, configuration tools, registry keys, scripts, or any other system components.
		Verify that passwords can either be expired based on a specific time interval or restricted from reuse.
Provision 5.2-1	The manufacturer shall make a vulnerability disclosure policy publicly available. This policy shall include, at a minimum: • contact information for the reporting of issues; and • information on timelines for: 1) initial acknowledgement of receipt; and 2) status updates until the resolution of the reported issues.	Verify for internet-connected devices and services, a public point of contact as part of a vulnerability disclosure policy is provided in order for security researchers and others are to report issues.
		Verify that the publicly available vulnerability disclosure policy includes information on timelines for initial acknowledgement of receipt and status updates until the resolution of the reported issues.
Provision 5.2-2	Disclosed vulnerabilities should be acted on in a timely manner.	Verify disclosed vulnerabilities are acted on in a timely manner.
Provision 5.2-3	Manufacturers should continually monitor for, identify and rectify security vulnerabilities within products and services they sell, produce, have produced and services they operate during the defined support period.	Verify vulnerability scanning mechanisms are in place for the software and hardware of products and services and third parties.
		Verify disclosed vulnerabilities are acted on in a timely manner.
Provision 5.3-1	All software components in consumer IoT devices should be securely updateable.	Verify that all software components in the devices are securely updateable.
		Verify that any Over-The-Air upgrade process is secure.
Provision 5.3-2	When the device is not a constrained device, it shall have an update mechanism for the secure installation of updates.	Verify that all software components in the devices are securely updateable.
		Verify firmware executables contain a Digital Signature from a trusted CA and are thus not self-signed.
		Verify that the Digital Signature Public Key Algorithm is RSA, DSA or ECDSA.
		Verify that the minimum key length for the digital signature is 2048-bits for asymmetric RSA and DSA, and 224 for ECDSA.

		<p>Verify that the hashing algorithm used for the digital signature is SHA256.</p> <p>Verifying firmware with a bad digital signature will result in an error being logged and the default firmware being employed.</p> <p>Verify that the authenticity check is implemented properly in the device. The device should not accept unsigned firmware as well as firmware signed with a self-signed certificate on any interface.</p>
Provision 5.3-3	An update shall be simple for the user to apply.	Verify that when software components are updateable, the need for each update is made clear to consumers and an update is easy to implement.
Provision 5.3-4	Automatic mechanisms should be used for software updates.	<p>Verify that any Over-The-Air upgrade process is secure.</p> <p>Verify that the product team has a mechanism for device upgrades in field such as for firmware and hardware.</p> <p>Verify that the system can update and patch the latest updates.</p> <p>Verify that system components check for system updates prior to final activation and operation.</p>
Provision 5.3-5	The device should check after initialization, and then periodically, whether security updates are available.	<p>Verify that the system can update and patch the latest updates.</p> <p>Perform an initial configuration sequence from factory default settings.</p> <p>Verify that system components check for system updates prior to final activation and operation.</p>
Provision 5.3-6	Verify that user is at least given the option to apply new updates, prior to operation. (If applicable)	Verify that the user is at least given the option to apply new updates, prior to operation.
Provision 5.3-7	The device shall use best practice cryptography to facilitate secure update mechanisms.	<p>Enumerate all cryptographic functions of the application.</p> <p>Verify only recommended ciphers are enabled on the server.</p> <p>Verify TLS 1.2 or TLS 1.3 is used for encryption over network for HTTPS.</p> <p>Verify SSL certificates are signed using SHA-2 (with SHA-256 or higher) hashing algorithm.</p> <p>Verify that the minimum key length for the digital signature is 2048-bits for asymmetric RSA and DSA, and 224 for ECDSA.</p> <p>Verify the cryptography does not degrade the performance of the system.</p>

		Verify the failure of the cryptography does not result in a Denial of Service (DoS).
		Verify used encryption does not support or fallback to insecure ciphers.
		Verify that any connection does not accept invalid certificates.
Provision 5.3-8	Security updates shall be timely.	Verify that when software components are updateable, updates are timely.
Provision 5.3-9	The device should verify the authenticity and integrity of software updates.	Verify that when software components are updateable, the provenance of software updates is assured and security patches are delivered over a secure channel.
		Verify firmware executables contain a Digital Signature from a trusted CA and are thus not self-signed.
		Verifying firmware with a bad digital signature will result in an error being logged and the default firmware being employed.
		Verify that the authenticity check is implemented properly in the device. The device should not accept unsigned firmware as well as firmware signed with a self-signed certificate on any interface.
Provision 5.3-10	Where updates are delivered over a network interface, the device shall verify the authenticity and integrity of each update via a trust relationship.	Verify that when software components are updateable, the provenance of software updates is assured and security patches are delivered over a secure channel.
Provision 5.3-11	The manufacturer should inform the user in a recognizable and apparent manner that a security update is required together with information on the risks mitigated by that update.	Verify that the consumer is informed by the appropriate entity, such as the manufacturer or service provider, that an update is required.
Provision 5.3-12	The device should notify the user when the application of a software update will disrupt the basic functioning of the device.	Verify that when software components are updateable, updates, where possible, maintain the basic functioning of the device, which can be critical to remain available during an update.
Provision 5.3-13	The manufacturer shall publish, in an accessible way that is clear and transparent to the user, the defined support period.	In case the system includes updateable software components, an end-of-life policy is published.

Provision 5.3-14	For constrained devices that cannot have their software updated, the rationale for the absence of software updates, the period and method of hardware replacement support and a defined support period should be published by the manufacturer in an accessible way that is clear and transparent to the user.	Verify for constrained devices that cannot have their software updated, the rationale regarding software updates, hardware replacement support, and end-of-life policy.
Provision 5.3-15	For constrained devices that cannot have their software updated, the product should be isolable and the hardware replaceable.	Verify that for constrained devices that cannot have their software updated, the product is isolable and the hardware replaceable.
Provision 5.3-16	The model designation of the consumer IoT device shall be clearly recognizable, either by labelling on the device or via a physical interface.	Enumerate all the local interfaces, including physical ones at board level (such as JTAG, SPI, I2C, FTDI, OBDII) and device level (such as USB, Serial).
		Verify the model designation of the product is clearly recognizable.
Provision 5.4-1	Sensitive security parameters in persistent storage shall be stored securely by the device.	Verify stored passwords are encrypted/hashed.
		Verify sensitive password files /etc/shadow and /etc/passwd are accessible only by the Root account.
		Enumerate all password storage locations, including text files, databases, and binary objects.
		Verify that the file, database, or object that is used to maintain passwords is only write-able by the application.
		Verify the cryptographic libraries used for encryption are FIPS-140-2 compliant.
		Verify that if the software or firmware contains any Personal Identifiable Information or sensitive data, then it must be encrypted.
		Verify keys are stored and managed securely.
Provision 5.4-2	Where a hard-coded unique per device identity is used in a device for security purposes, it shall be implemented in such a way that it resists tampering by means such as physical, electrical or software.	Simulate a product failure by tampering one of the software components in the data store and verify that it fails the software integrity test on initial power up.
		Verify access to the electronics are not easily compromised, for example ensure there are no exposed fasteners, screws, or other compromisable components.

		<p>Visually inspect and enumerate any tamper resistant measures.</p>
		<p>Verify the effectiveness of each physical tamper resistant measure to deter and/or alert the system owner to tampering.</p>
		<p>Verify that hardware supported tamper resistance mechanisms are effective.</p>
		<p>Verify that if an unauthorized change is detected to the software, the device should alert the consumer and/or administrator to an issue and should not connect to wider networks than those necessary to perform the alerting function.</p>
<p>Provision 5.4-3</p>	<p>Hard-coded critical security parameters in device software source code shall not be used.</p>	<p>Verify that passwords are not hardcoded in the source code, firmware, configuration tools, registry keys, scripts, or any other system components.</p>
<p>Provision 5.4-4</p>	<p>Any <b>critical security parameters</b> used for integrity and authenticity checks of software updates and for protection of communication with associated services in device software <b>shall be unique per device</b> and shall be <b>produced with a mechanism that reduces the risk of automated attacks</b> against classes of devices.</p>	<p>Verify that the system is capable of uniquely identifying and authenticating device(s) before establishing network connections.</p> <p>Verify that Organizational Authentication solutions are used to identify and authenticate devices on local and/or wide area networks.</p> <p>Verify that device-to-device identification and authentication when Standard Industrial protocols do not support cryptographic identification and authentication.</p> <p>Verify firmware executables contain a Digital Signature from a trusted CA and are thus not self-signed.</p> <p>Verify that the Digital Signature Public Key Algorithm is RSA, DSA or ECDSA.</p> <p>Verify that the minimum key length for the digital signature is 2048-bits for asymmetric RSA and DSA, and 224 for ECDSA.</p> <p>Verify that the hashing algorithm used for the digital signature is SHA256.</p> <p>Verifying firmware with a bad digital signature will result in an error being logged and the default firmware being employed.</p> <p>Verify separate and unique keys are used for different functions such as transmitting data between devices, communicating with servers, encrypting files, and generating digital signatures.</p>



Provision 5.5-1	The consumer IoT device shall use best practice cryptography to communicate securely	Verify message authentication is implemented at the protocol level, if the protocol supports it.
		Using a enumerated list of services, for each authentication request, intercept at least one sample.
		Verify by sniffing that credentials are transported using a suitable encrypted link and that all pages and functions that require a user to enter credentials are done so using an encrypted link.
		Verify that the device adheres to the relevant security standards for each protocol in case message authentication is not implemented at the protocol level.
		Verify TLS 1.2 or TLS 1.3 is used for encryption over network for HTTPS.
		Verify HTTP Strict Transport Security (HSTS) implementation by ensuring the server has a valid SSL/TLS certificate and confirming the presence of the HSTS header.
		Enumerate all cryptographic functions of the application.
		Verify only recommended ciphers are enabled on the server.
		Verify SSL certificates are signed using SHA-2 (with SHA-256 or higher) hashing algorithm.
		Verify that the minimum key length for the digital signature is 2048-bits for asymmetric RSA and DSA, and 224 for ECDSA.
		Verify the cryptography does not degrade the performance of the system.
		Verify the failure of the cryptography does not result in a Denial of Service (DoS).
		Verify used encryption does not support or fallback to insecure ciphers.
Verify that any connection does not accept invalid certificates.		
Provision 5.5-2	The consumer IoT device should use reviewed or evaluated implementations to deliver network and security functionalities, particularly in the field of cryptography.	Verify that the developer did not 'roll their own' encryption and that best practice libraries are used for all cryptographic functions.
		Verify that there exists a crypto-log in product documentation with information about each key or certificate, key-lengths, usage, location, and responsible party.

		Confirm that all cryptographic algorithms utilized are approved by NIST or FIPS.
Provision 5.5-3	Cryptographic algorithms and primitives should be updateable.	Verify that the developer did not 'roll their own' encryption and that best practice libraries are used for all cryptographic functions.
		Confirm that all cryptographic algorithms utilized are approved by NIST or FIPS.
		Verify that there exists a crypto-log in product documentation with information about each key or certificate, key-lengths, usage, location, and responsible party.
		Verify that any Over-The-Air upgrade process is secure.
Provision 5.5-4	Access to device functionality via a network interface in the initialized state should only be possible after authentication on that interface.	Verify that logs are restricted to authorized users only and are immutable.
		Verify all web pages and resources by default require authentication, except those specifically intended to be public.
		Verify all authentication controls are enforced on the server side, and not on the client.
		Verify access is restricted to privileged functions such as hardware resets and sudo commands.
		Verify access is restricted to security information.
		Verify access is restricted based on user, role, and attributes.
		Verify that there is no method to bypass the Role Based Access Control or Access Control List.
		Verify that the device interface and ports are protected from unauthorized access.
		Verify that Field Tools require access control to utilize the tool(s).
		Verify device/user authentication mechanism is implemented on all network interfaces that exposes sensitive information or admin level functionality.
		Verify a mechanism for device or user authentication is implemented.
Verify device/user authentication mechanism is implemented on all network interfaces that exposes sensitive information or admin level functionality.		

Provision 5.5-5	Device functionality that allows security-relevant changes in configuration via a network interface shall only be accessible after authentication. The exception is for network service protocols that are relied upon by the device and where the manufacturer cannot guarantee what configuration will be required for the device to operate.	Verify that all non-root or non-admin accounts have read-only permissions or less for important files, such as /etc/sunders, /etc/ssh/sshd_config, /etc/group, /etc/shadow, /etc/passwd, Program Data etc.
		Verify that only read-only configuration files can be uploaded by the user in case there is no integrity check mechanism for configuration files.
		Verify that the configuration file integrity mechanism cannot be bypassed in any manner.
		Enumerate all external file inputs to the device.
		Verify that unsupported file formats can not be uploaded.
		Verify input files include an integrity check, such as a Message Authentication Code (MAC) or Signature.
		Verify, if a digital signature is used, the Message Authentication Code (MAC) is either SHA, RSA, DSA, or ECDSA.
Provision 5.5-6	Critical security parameters should be encrypted in transit, with such encryption appropriate to the properties of the technology, risk and usage.	Verify by sniffing that credentials are transported using a suitable encrypted link and that all pages and functions that require a user to enter credentials are done so using an encrypted link.
		Verify there is an explicit policy for how cryptographic keys are managed, and the lifecycle of cryptographic keys is enforced. For example, by following a key management standard such as NIST SP 800-57.
		Verify that if the software or firmware contains any Personal Identifiable Information or sensitive data, then it must be encrypted.
Provision 5.5-7	The consumer IoT device shall protect the confidentiality of critical security parameters that are communicated via remotely accessible network interfaces.	Verify passwords are protected from unauthorized disclosure and modification when transmitted.
		Identify all potential safety impacts of the information generated, stored, used or communicated by the product.
		Assess the protection mechanisms deployed to secure the safety critical data.

		Verify that safety critical data is traversing through segmented or segregated communication channels only.
		Verify that data passed over common or shared communication channels employ data integrity mechanisms like message authentication scheme to limit the possibility of message spoofing.
Provision 5.5-8	The manufacturer should follow secure management processes for critical security parameters that relate to the device.	Verify that critical parameters related to the device are being managed securely, including secure key management, firmware updates, boot process, and password management.
Provision 5.6-1	All unused network and logical interfaces shall be closed.	Verify all interfaces and services not required for operations are disabled by default.
		Verify that all additional services which are installed, but not required for operations are removed.
		Verify there are not any unnecessary network services.
Provision 5.6-2	In the initialized state, the network interfaces of the device should minimize the unauthenticated exposure of security-relevant information.	Verify all interfaces and services not required for operations are disabled by default.
		Verify there are not any unnecessary network services.
		Verify that the device interface and ports are protected from unauthorized access.
Provision 5.6-3	Device hardware should not unnecessarily expose physical interfaces to attack	Enumerate all the local interfaces, including physical ones at board level (such as JTAG, SPI, I2C, FTDI, OBDII) and device level (such as USB, Serial).
		Verify that each enumerated interface either has authentication or has been rendered inoperable.
		Verify all interfaces and services not required for operations are disabled by default.
		If continued developer access is necessary, verify that any developer-level debugging interfaces are appropriately protected to limit access to authorized privileged users.
		Verify access to the electronics are not easily compromised, for example ensure there are no exposed fasteners, screws, or other compromisable components.

		<p>Visually inspect and enumerate any tamper resistant measures.</p> <p>Verify the effectiveness of each physical tamper resistant measure to deter and/or alert the system owner to tampering.</p> <p>Verify that hardware supported tamper resistance mechanisms are effective.</p> <p>Verify that Hardware Security Modules (HSMs) or Trusted Platform Modules (TPMs) in use are physically protected.</p>
Provision 5.6-4	Where a debug interface is physically accessible, it shall be disabled in software.	Verify that when a debug interface is physical, it is disabled in software.
Provision 5.6-5	The manufacturer should only enable software services that are used or required for the intended use or operation of the device.	<p>Enumerate additional services that are installed, but not enabled.</p> <p>Verify that all additional services which are installed, but not required for operations are removed.</p> <p>If continued developer access is necessary, verify that any developer-level debugging interfaces are appropriately protected to limit access to authorized privileged users.</p> <p>Verify there are not any unnecessary network services.</p>
Provision 5.6-6	Code should be minimized to the functionality necessary for the service/device to operate.	Verify that dead code or code not needed for functionality is removed periodically.
Provision 5.6-7	Software should run with least necessary privileges, taking account of both security and functionality.	<p>Enumerate all the processes and services running in the device or system using a root shell.</p> <p>Verify that the level of privileges assigned to all processes/services are identified by using the list of enumerated processes/services. This includes determining the service level account allocated to each process/service, such as local service, network service, or local system.</p> <p>Verify each process and service is not running as root for Linux systems.</p> <p>Verify each process and service is not running as LOCAL SYSTEM for Windows systems.</p> <p>Verify that all non-root or non-admin accounts have read-only permissions or less for important files, such as /etc/sunders, /etc/ssh/sshd_config, /etc/group, /etc/shadow, /etc/passwd, Program Data etc.</p>

		<p>Verify users have a minimal write permissions.</p> <p>Verify that protection against directory traversal/file include attacks is implemented.</p> <p>Verify the system mitigates unauthorized access to admin functions.</p> <p>Verify that administrative interfaces are not accessible to untrusted parties.</p> <p>Verify that non-root accounts are used for network services.</p> <p>Verify that all service accounts have shell access disabled by verifying that the shell is set to "/dev/null" in the passwd file</p> <p>Verify that privileges for Set User ID accounts are raised as late as possible and released as soon as possible. This may require access to code to confirm, or reverse engineer using a tool such as Interactive Disassembler (IDA).</p>
Provision 5.6-8	The device should include a hardware level access control mechanism for memory	Verify the device has a hardware level access control mechanism for memory.
Provision 5.6-9	The manufacturer should follow secure development processes for software deployed on the device	Verify that the Secure Development Lifecycle is followed during development of the software deployed on the device.
Provision 5.7-1	The consumer IoT device should verify its software using secure boot mechanisms.	Verify that software is verified using secure boot mechanisms, which require a hardware root of trust.
Provision 5.7-2	If an unauthorized change is detected to the software, the device should alert the consumer and/or administrator to the issue and should not connect to wider networks than those necessary to perform the alerting function.	Verify that if an unauthorized change is detected to the software, the device should alert the consumer and/or administrator to an issue and should not connect to wider networks than those necessary to perform the alerting function.
Provision 5.8-1	The confidentiality of personal data transiting between a device and a service, especially associated services, should be protected, with best practice cryptography.	<p>Verify message authentication is implemented at the protocol level, if the protocol supports it.</p> <p>Using a enumerated list of services, for each authentication request, intercept at least one sample.</p> <p>Verify by sniffing that credentials are transported using a suitable encrypted link and that all pages and functions that require a user to enter credentials are done so using an encrypted link.</p>

		Verify that the device adheres to the relevant security standards for each protocol in case message authentication is not implemented at the protocol level.
		Verify TLS 1.2 or TLS 1.3 is used for encryption over network for HTTPS.
		Verify HTTP Strict Transport Security (HSTS) implementation by ensuring the server has a valid SSL/TLS certificate and confirming the presence of the HSTS header.
		Identify all the sensitive and personal information generated, stored, used or communicated by the system.
		Enumerate all cryptographic functions of the application.
		Verify that the developer did not 'roll their own' encryption and that best practice libraries are used for all cryptographic functions.
		Verify that all sensitive or personal data is encrypted when at rest. This includes all forms of sensitive information, for example cached data in web browsers, temporary files, logs, sensitive secrets in configuration files, etc.
Provision 5.8-2	The confidentiality of sensitive personal data communicated between the device and associated services shall be protected, with cryptography appropriate to the properties of the technology and usage.	Enumerate all cryptographic functions of the application.
		Verify only recommended ciphers are enabled on the server.
		Verify TLS 1.2 or TLS 1.3 is used for encryption over network for HTTPS.
		Verify SSL certificates are signed using SHA-2 (with SHA-256 or higher) hashing algorithm.
		Verify that the minimum key length for the digital signature is 2048-bits for asymmetric RSA and DSA, and 224 for ECDSA.
		Verify the cryptography does not degrade the performance of the system.
		Verify the failure of the cryptography does not result in a Denial of Service (DoS).
		Verify used encryption does not support or fallback to insecure ciphers.
		Verify that any connection does not accept invalid certificates.
		Verify that if remote access, for example, SSH, SFTP, etc., is provisioned, remote access sessions are encrypted.

Provision 5.8-3	All external sensing capabilities of the device shall be documented in an accessible way that is clear and transparent for the user.	Verify that all external sensing capabilities of the device are documented in an accessible way that is clear and transparent for the user.
Provision 5.9-1	Resilience should be built in to consumer IoT devices and services, taking into account the possibility of outages of data networks and power.	Verify that product performs the software integrity test before loading any applications or performing any functions of the product.
		Simulate a product failure by tampering one of the software components in the data store and verify that it fails the software integrity test on initial power up.
		Verify that upon a failure the product enters a failure mode which clearly indicates to the user that the product has failed to start up successfully.
		Verify the system starts up in a defined known state.
		Verify the system has a defined known secure state.
		Verify the system fails to the known secure state.
		Verify a device failed to the known state does not allow access without verifying credentials.
Provision 5.9-2	Consumer IoT devices should remain operating and locally functional in the case of a loss of network and should recover cleanly in the case of restoration of a loss of power.	Verify that a control system has been successfully recovered and reconstituted to a known secure state by checking that all system parameters are set to secure values, security-critical patches are reinstalled, security-related configuration settings are reestablished, system documentation and operating procedures are available, application and system software is reinstalled and configured with secure settings, information from the most recent, known secure backups is loaded, and the system is fully tested and functional.
		Perform a power failure simulation test.
Provision 5.9-3	The consumer IoT device should connect to networks in an expected, operational and stable	Verify normal functionality in device is restored in an expected, operational and stable state.



	state and in an orderly fashion, taking the capability of the infrastructure into consideration.	Verify devices return to a network in an orderly fashion, rather than in a massive-scale reconnect.
Provision 5.10-1	If telemetry data is collected from consumer IoT devices and services, such as usage and measurement data, it should be examined for security anomalies.	Verify that collected telemetry data (e.g. usage and measurement data) is examined for security anomalies.
Provision 5.11-1	The user shall be provided with functionality such that user data can be erased from the device in a simple manner.	Verify personal data can easily be removed.
Provision 5.11-2	The consumer should be provided with functionality on the device such that personal data can be removed from associated services in a simple manner.	Verify personal data can easily be removed from associated services.
Provision 5.11-3	Consumers should be given clear instructions on how to delete their personal data.	Verify consumers are given clear instructions on how to delete their personal data, especially for IoT systems.
Provision 5.11-4	Consumers should be provided with clear confirmation that personal data has been deleted from services, devices and applications.	Verify consumers are provided with clear confirmation that personal data has been deleted from services, devices and applications, especially for IoT systems.
Provision 5.12-1	Installation and maintenance of consumer IoT should employ minimal steps and should follow security best practice on usability.	Verify installation and maintenance of the system employ minimal steps and follow security best practice on usability.
Provision 5.12-2	The manufacturer should provide consumers with guidance on how to securely set up their device.	Verify there is guidance available for consumers to securely set up their device. (e.g. guidance on how to validate the device's capability to establish a secure communication channel)
Provision 5.12-3	The manufacturer should provide consumers with guidance on how to check whether their device is securely set up.	Verify there is guidance available for consumers to check if their device is securely set up. (e.g. guidance on how to validate the device's capability to establish a secure communication channel)
Provision 5.13-1	The consumer IoT device software shall validate data input via user interfaces or transferred via application programming interfaces (APIs) or between networks in services and devices.	Verify all data inputs via user interfaces and transferred via Application Programming Interfaces (APIs) or between networks in services and devices are validated.

Provision 6.1	The manufacturer shall provide consumers with clear and transparent information about what personal data is processed, how it is being used, by whom, and for what purposes, for each device and service. This also applies to third parties that can be involved, including advertisers	Verify that device manufacturers and service providers provide consumers with clear and transparent information about how their personal data is being used.
Provision 6.2	Where personal data is processed on the basis of consumers' consent, this consent shall be obtained in a valid way.	Verify that where personal data is processed on the basis of consumers' consent, this consent is obtained in a valid way.
Provision 6.3	Consumers who gave consent for the processing of their personal data shall have the capability to withdraw it at any time.	Verify that customers who gave consent for the processing of their personal data are given the opportunity to withdraw it at any time.
Provision 6.4	If telemetry data is collected from consumer IoT devices and services, the processing of personal data should be kept to the minimum necessary for the intended functionality.	If telemetry data is collected, verify the processing of personal data is kept to the minimum necessary for the intended functionality.
Provision 6.5	If telemetry data is collected from consumer IoT devices and services, consumers shall be provided with information on what telemetry data is collected, how it is being used, by whom, and for what purposes	If telemetry data is collected from consumer devices and services, verify consumers are provided with information on what telemetry data is collected, how it is being used, by whom and for what purposes.