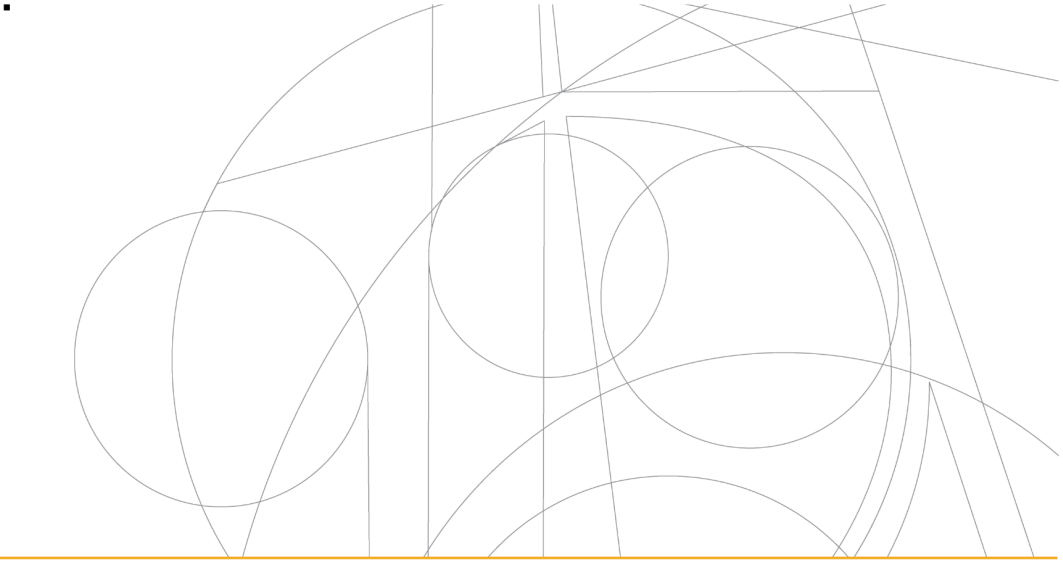
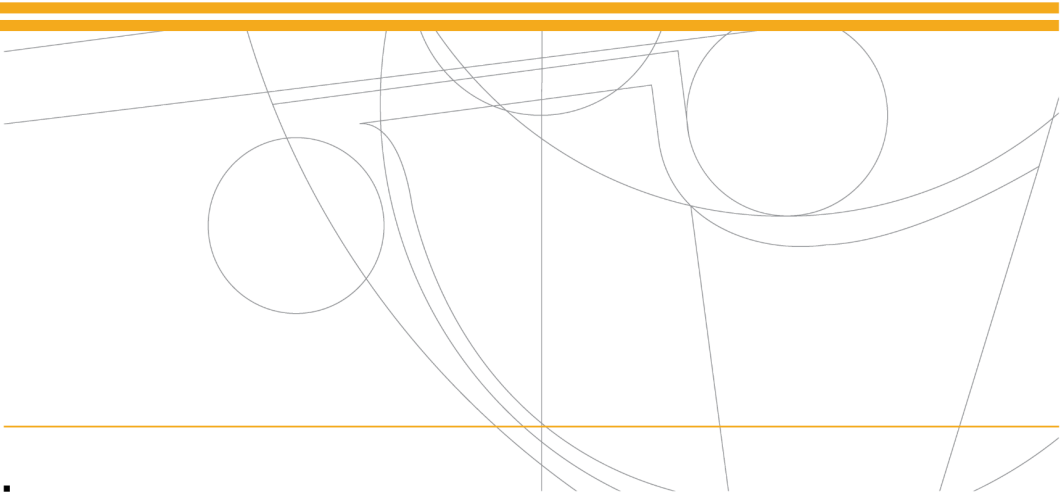


INSTITUTO POLITÉCNICO
DE VIANA DO CASTELO

ESTG

A COMPREHENSIVE EXAMINATION OF CYBERSECURITY IN COMMUNICATION PROTOCOLS UTILIZED BY HOME AUTOMATION TECHNOLOGIES: AN IN-DEPTH EXPLORATION OF POTENTIAL VULNERABILITIES AND MITIGATIVE STRATEGIES

Joel Dinis Pereira Alves



A COMPREHENSIVE EXAMINATION OF CYBERSECURITY IN COMMUNICATION PROTOCOLS UTILIZED BY HOME AUTOMATION TECHNOLOGIES: AN IN-DEPTH EXPLORATION OF POTENTIAL VULNERABILITIES AND MITIGATIVE STRATEGIES | Joel Alves

2025

Escola Superior de Tecnologia e Gestão

Instituto Politécnico
de Viana do Castelo

A Comprehensive Examination of Cybersecurity in Communication Protocols Utilized by Home Automation Technologies: An In-Depth Exploration of Potential Vulnerabilities and Mitigative Strategies

Autor(a)

Joel Dinis Pereira Alves

Trabalho orientado por

Professor Silvestre Malta

Professor Paulo Costa

Mestrado em Cibersegurança

30 de Novembro de 2025

Mestrado em Cibersegurança
Master in Cybersecurity

A Comprehensive Examination of Cybersecurity in Communication Protocols Utilized by Home Automation Technologies: An In-Depth Exploration of Potential Vulnerabilities and Mitigative Strategies

a master's thesis authored by

Joel Dinis Pereira Alves

and supervised by

Silvestre Lomba Malta

Professor Adjunto Convidado, IPVC

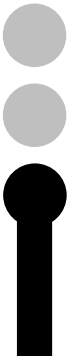
Paulo Costa

Professor Adjunto, IPVC

This thesis was submitted in partial fulfilment of the requirements for the Master's degree in Cybersecurity at the Instituto Politécnico de Viana do Castelo

ipvc

5 of December, 2025



Abstract

The rapid expansion of smart-home ecosystems has widened the attack surface of domestic networks, making the security of Internet of Things (IoT) communication protocols increasingly critical. This thesis evaluates how the main wireless technologies used in home automation Zigbee, Z-Wave, Bluetooth Low Energy (BLE), and Wi-Fi comply with the security requirements of European Telecommunications Standards Institute (ETSI) EN 303 645.The research analyses protocol-level and device-level vulnerabilities using an experimental approach that includes packet capture, adversarial testing, cloud-domain inspection, and an ETSI aligned scoring method. Tests were performed on two representative devices per protocol in a controlled laboratory environment. Results show significant security discrepancies across protocols and products, with weaknesses in pairing, key management, replay protection, and Denial-of-service (DoS) resilience. Several devices failed essential ETSI requirements related to secure communication and update integrity. The thesis delivers a reproducible evaluation methodology, a quantitative ETSI based scoring model, and practical recommendations to strengthen IoT security from device to cloud.

Keywords: IoT, Smart Home, Cybersecurity, Zigbee, Z-Wave, BLE, Wi-Fi, ETSI EN 303 645, Vulnerability Analysis.

Resumo

A crescente adoção de dispositivos inteligentes no ambiente doméstico aumentou significativamente a superfície de ataque das redes residenciais, tornando a segurança dos protocolos de comunicação IoT um tema crítico. Esta dissertação avalia de que forma os principais protocolos sem fios utilizados na automação doméstica Zigbee, Z-Wave, BLE e Wi-Fi cumprem os requisitos de cibersegurança definidos na norma ETSI EN 303 645. A investigação analisa vulnerabilidades ao nível do protocolo e do dispositivo através de uma abordagem experimental que inclui captura de tráfego, testes adversariais, análise de domínios cloud e um modelo de pontuação alinhado com a ETSI. Foram testados dois dispositivos representativos por protocolo num ambiente laboratorial isolado. Os resultados revelam discrepâncias significativas de resiliência entre protocolos e fabricantes, identificando fragilidades no emparelhamento, gestão de chaves, proteção contra replay e resistência a ataques DoS. Vários dispositivos falharam requisitos essenciais da ETSI relacionados com comunicação segura e integridade de atualizações. A dissertação apresenta uma metodologia reprodutível de avaliação, um modelo quantitativo de mapeamento à norma ETSI EN 303 645 e recomendações práticas para reforçar a segurança IoT desde o dispositivo até à infraestrutura cloud.

Palavras-chave: IoT, Casa inteligente, Cibersegurança, Zigbee, Z-Wave, BLE, Wi-Fi, ETSI EN 303 645, Análise de Vulnerabilidades

Aknowledgements

I express my deepest gratitude to all who contributed to the completion of this dissertation. First of all, I thank my family for their unconditional support, patience, and encouragement throughout this journey. Their belief in me has been a constant source of strength and motivation. I extend my sincere appreciation to the Instituto Politécnico de Viana do Castelo (IPVC) and to all the professors of the Master's programmer in Cybersecurity. Their dedication, expertise, and commitment to teaching provided the solid academic foundation that made this work possible. I would especially like to acknowledge teacher Silvestre Malta and teacher Paulo Costa, whose guidance, availability, and insightful feedback were invaluable throughout the development of this thesis. Their mentorship and technical orientation played a crucial role in shaping the direction and quality of this research. Finally, I thank all colleagues, friends, and professionals who, directly or indirectly, contributed through discussions, shared knowledge, or offered support during challenging moments. Each contribution helped make this thesis a reality.

Contents

List of Figures

List of Tables

List of Abbreviations

AAuth authentication denial-of-service

ACL Access Control Lists

AES Advanced Encryption Standard

AP Access Points

API Application Programming Interface

APS Application Support

AWS Amazon Web Services

BFlood beacon flooding

BLE Bluetooth Low Energy

BSSID Basic Service Set Identifier

CCM Counter with CBC-MAC

CDN Content Delivery Network

CIA Confidentiality, Integrity, and Availability

CoAP Constrained Application Protocol

CRA Cyber Resilience Act

CVD coordinated vulnerability disclosure

CVE Common Vulnerabilities and Exposures

DAuth deauthentication flooding

DCRS Device Cloud-Risk Score

DDS Data Distribution Service

DoS Denial-of-service

DSK Device-Specific Key

DTLS Datagram Transport Layer Security

ECDH Elliptic Curve Diffie-Hellman

ETSI European Telecommunications Standards Institute

EU European Union

GATT Generic Attribute Profile

GDPR General Data Protection Regulation

HTML HyperText Markup Language

HTTP Hypertext Transfer Protocol

HTTPS Hypertext Transfer Protocol Secure

IAS Intruder Alarm System

ID Identifier

IoT Internet of Things

IP Internet Protocol

IPVC Instituto Politécnico de Viana do Castelo

JS JavaScript

JSON JavaScript Object Notation

KEX Key Exchange

LAN Local Area Network

MAC Medium Access Control

MCyber Master in Cybersecurity

MFP Management Frame Protection

MGMT Management

MitM Man-in-the-Middle

MQTT Message Queuing Telemetry Transport

NIC Wireless Network Interface Controller

NWK Network

ONVIF Open Network Video Interface Forum

OOB Out-of-Band

OS Operation System

OTA Over-The-Air

OTW Over-The-Wire

OWASP Open Worldwide Application Security Project

PAN Personal Area Network

PCAP Packet Capture

PCAPNG Packet Capture Next Generation

PHY Physical Layer

PID Product ID

PIN Personal Identification Number

PMF Protected Management Frames

PMKID Pairwise Master Key Identifier

PSK Pre-Shared Key

PTI Packet Trace Interface

RF Radio Frequency

RNG Random Number Generator

RTSP Real-Time Streaming Protocol

SAE Simultaneous Authentication of Equals

SDK Software Development Kit

SDR Software-Defined Radios

SMA SubMiniature version A

SSID Service Set Identifier

SSL Secure Sockets Layer

TCP Transmission Control Protocol

TLS Transport Layer Security

UI User Interface

URL Uniform Resource Locator

UUID Universally Unique Identifier

VID Vendor ID

VLAN Virtual Local Area Network

WEP Wired Equivalent Privacy

Wi-Fi Wireless network

WPA Wi-Fi Protected Access

WSTK Wireless Starter Kit

ZHA Zigbee Home Automation

Chapter 1

Introduction

The proliferation of the Internet of Things (Internet of Things (IoT)) has fundamentally reshaped how devices communicate and interact, enabling integrated smart-home ecosystems that deliver automation, energy efficiency, and enhanced user experience. These technological advances, however, have expanded the cybersecurity attack surface, exposing home-automation environments to a diverse range of threats. The widespread use of wireless communication protocols such as Zigbee, Z-Wave, Bluetooth Low Energy (Bluetooth Low Energy (BLE)), and Wi-Fi further amplifies these vulnerabilities due to their heterogeneity, design constraints, and uneven security adoption across vendors.

Ensuring the security and reliability of these interconnected systems requires a thorough understanding of the communication protocols that underpin device interactions and the vulnerabilities arising from protocol design choices, implementation weaknesses, or misalignment with established security standards. The European Telecommunications Standards Institute (ETSI) EN 303 645 standard, currently the most influential global baseline for consumer IoT cybersecurity, provides a structured set of requirements for secure-by-design device development. Despite its growing acceptance, numerous commercial devices still fall short of full compliance, resulting in risks such as unauthorized access, privacy breaches, data manipulation, and persistent exploitation through insecure update mechanisms.

This thesis addresses these challenges by performing both theoretical and empirical analyses of wireless communication protocols used in home automation. Two representative devices per protocol are evaluated in a controlled testbed, enabling systematic

identification of protocol-level and implementation-level vulnerabilities. These findings are subsequently mapped to the relevant requirements of ETSI EN 303 645, providing a standardized view of compliance and highlighting practical gaps in current consumer IoT security practices.

The remainder of this chapter is structured as follows:

- **Context:** Presents the evolution of the IoT ecosystem, its growing relevance in home automation, and the implications of protocol security for user trust and safety.
- **Problem Statement and Motivation:** Defines current cybersecurity challenges in smart-home communication layers and motivates the need for alignment with international standards such as ETSI EN 303 645.
- **Objectives:** Outlines the scientific and technical goals of the research, including vulnerability discovery, standard-based evaluation, and development of mitigation strategies.
- **Organization of the Thesis:** Describes the structure and scope of each chapter.

1.1 Context

The rapid evolution of home automation technologies has transformed modern living, delivering enhanced efficiency, personalized control, and seamless interoperability between devices [4]. These devices thermostats, lighting systems, smart plugs, cameras, and motion sensors depend on communication protocols such as Zigbee [41], Z-Wave [54], Wi-Fi, and Bluetooth to establish functional smart-home ecosystems [30].

However, the exponential growth of consumer IoT has also broadened the attack surface significantly. Many devices operate under stringent resource constraints that limit the implementation of strong cryptographic primitives or robust security controls [7, 43]. As a result, smart homes become susceptible to privacy-invasive surveillance, unauthorized actuation, network compromise, and persistent malware propagation [18, 11]. Investigations continue to expose critical weaknesses including default credentials, unencrypted communications, insecure firmware update mechanisms, and improper key management [36, 48, 41].

To address these issues, the ETSI introduced the EN 303 645 standard [25], which defines baseline cybersecurity requirements for consumer IoT. These include secure credential handling, protected update channels, minimized attack surfaces, strong communication confidentiality, and mechanisms for responsible vulnerability disclosure [39, 24]. Although adoption has improved, vendor implementations remain heterogeneous, especially in low-cost or legacy devices [30].

This thesis therefore evaluates the protocol-level and cloud-level security posture of Zigbee, Z-Wave, BLE, and Wi-Fi, mapping all observed behaviors to applicable EN 303 645 clauses. Testing includes packet-capture analysis, replay experiments, management-frame manipulation, BLE fuzzing, Z-Wave S2 inspection, and cloud-domain interception through Charles Proxy. By focusing specifically on communication layers, this work offers a standardized, reproducible framework for assessing protocol compliance with EN 303 645.

1.2 Problem Statement and Motivation

As smart-home environments become ubiquitous, their underlying communication protocols inherit persistent cybersecurity weaknesses many of which stem from legacy design assumptions, inconsistent vendor implementations, or the resource limitations of embedded hardware. Issues such as weak or unauthenticated pairing, insecure update mechanisms, fallback to deprecated security modes (e.g., Z-Wave S0), and susceptibility to replay, eavesdropping, or denial-of-service attacks are regularly documented [40, 29].

Although ETSI EN 303 645 defines comprehensive security requirements for consumer IoT, real-world compliance remains uneven. Fragmented device ecosystems, limited vendor maturity, and diverging interpretations of "secure-by-design" principles hinder uniform adoption [56]. Moreover, constrained devices often struggle to implement heavyweight defenses, necessitating lightweight but effective mechanisms aligned with the standard.

The core research problem is therefore:

- There is a persistent gap between the security guarantees required for trustworthy home-automation systems and the protections implemented in protocol specifications, device firmware, and deployment configurations.

- Addressing this gap requires standardized methods for identifying vulnerabilities, mapping them to EN 303 645, and proposing actionable improvements that remain feasible for constrained consumer hardware.

1.3 Objectives

The widespread adoption of home automation technologies has brought substantial convenience and innovation to modern living but has also amplified cybersecurity risks in the communication protocols that interconnect IoT devices [61, 6]. Despite the establishment of ETSI EN 303 645 as a foundational standard defining baseline cybersecurity requirements for consumer IoT products [26], real-world implementations frequently exhibit partial or inconsistent adherence. This non-uniform application leaves devices exposed to unauthorized access, data ex-filtration, replay, and Denial-of-service (DoS) attacks, ultimately undermining user trust and system reliability [48, 19, 12].

The persistence of these issues highlights the need for a structured investigation into how IoT communication protocols Zigbee, Z-Wave, BLE, and Wi-Fi, align with established standards and how they can be improved to achieve robust, scalable, and standards-compliant security. The overarching aim of this research is to examine, classify, and mitigate protocol-level vulnerabilities while ensuring alignment with ETSI EN 303 645, thus contributing to the development of secure and trustworthy home automation ecosystems.

To achieve this aim, the study is guided by the following specific objectives:

- **Identify and Categorize Vulnerabilities:** Conduct a detailed analysis of inherent and implementation-based vulnerabilities across widely used IoT communication protocols, focusing on issues such as weak encryption, insecure pairing, default credentials, replay susceptibility, and inadequate update mechanisms [41, 54].

- **Assess Compliance with ETSI EN 303 645:** Evaluate the extent to which the selected protocols meet or deviate from the ETSI EN 303 645 requirements, emphasizing data protection, software update integrity, access control, and secure onboarding processes [26, 56].

- **Develop a Theoretical Framework for Security Solutions:** Propose and an-

alyze a framework of mitigation strategies to address the identified vulnerabilities, ensuring that the proposed measures align with ETSI EN 303 645 principles while remaining compatible with the resource constraints of IoT devices [66, 42].

- **Investigate Practical Implementation Barriers** Examine the technical and operational limitations that hinder the adoption of ETSI EN 303 645-compliant practices across different communication stacks, including challenges related to hardware constraints, interoperability, scalability, and legacy system integration [6, 61].
- **Contribute to IoT Security Knowledge:** Provide actionable insights, vulnerability-to-standard mapping tables, and recommendations that can support manufacturers, researchers, and policymakers in strengthening the security posture of home automation systems [63, 30].

By fulfilling these objectives, this research aims to bridge the gap between theoretical security standards and practical implementation, ensuring that home automation systems are not only functionally efficient, but also resilient against evolving cyber threats. The results are expected to inform industry best practices, guide developers in protocol selection and design, and promote end-user confidence through demonstrable compliance with recognized cybersecurity standards.

1.4 Contributions

The increasing reliance on interconnected home automation systems underscores the urgent need for rigorous, standards-aligned approaches to IoT cybersecurity. While numerous studies have examined isolated vulnerabilities or proposed general security frameworks, few have conducted a comparative, protocol-specific analysis explicitly mapped to ETSI EN 303 645. This research fills that gap by combining theoretical investigation with structured vulnerability mapping and compliance assessment across multiple communication technologies Zigbee, Z-Wave, BLE, and Wi-Fi, which collectively represent the foundation of modern smart-home ecosystems [61, 6, 26].

The primary contributions of this work can be summarized as follows:

- **Comprehensive Protocol Security Assessment:** A systematic evaluation of the security posture of the Zigbee, Z-Wave, BLE, Wi-Fi communication protocols used in home automation, identifying both protocol-level and implementation-level vulnerabilities such as weak key management, insecure pairing, replay susceptibility, and exposure to denial-of-service conditions.

- **ETSI EN 303 645 Compliance Mapping:** Since the ETSI EN 303 645 standard does not provide a quantitative severity model, this thesis introduces a custom scoring methodology that translates observed vulnerabilities into measurable impact likelihood scores aligned with the clauses of the standard. The method allows each vulnerability to be evaluated consistently across devices and protocols, producing a comparable numeric score (0–10) based on confidentiality, integrity, availability, and exploitation likelihood. This contribution fills a practical gap in the standard by enabling structured, repeatable, and semi-quantitative compliance assessment rather than relying solely on qualitative interpretation.

- **Proposed Mitigation Strategies Aligned with ETSI Principles:** Development of a structured set of mitigation strategies that apply ETSI EN 303 645 security principles to the vulnerabilities identified in this thesis. The model includes measures such as time-stamped encryption, secure on boarding workflows, unique per-device credential generation, and signed firmware update mechanisms. Each mitigation is evaluated in terms of feasibility for constrained IoT hardware, ensuring practical relevance for real-world device design.

1.5 Organization

This thesis is structured into nine chapters, each building on the previous to provide a comprehensive analysis of cybersecurity in home automation communication protocols. The organization of the document is as follows:

- **1 - Introduction:** Establishes the context, motivation, and significance of the research. It introduces the problem statement, outlines the objectives, highlights the main contributions, and provides an overview of the thesis structure.

- **2 - Literature Review:** Presents a critical review of relevant research on IoT and smart home security. It discusses the evolution of home automation technologies, examines prior studies on communication protocols such as Zigbee, Z-Wave, Wi-Fi, and BLE, and summarizes key cybersecurity challenges. The chapter also explores existing standards and frameworks, including ETSI EN 303 645 and Open Worldwide Application Security Project (OWASP) IoT Top 10, to establish the theoretical foundation of this work [61, 63, 26].

- **3 - Methodology:** Describes the research design, testbed architecture, tools, and analytical methods employed to identify and evaluate vulnerabilities. It explains the criteria for device selection (two devices per protocol), data collection processes (including Packet Capture (PCAP) analysis and protocol logging), and the mapping approach used to align findings with ETSI EN 303 645 requirements.

- **4 – Protocol and Ecosystem Analysis:** Presents an in-depth analysis of each communication protocol Zigbee, Z-Wave, BLE, and Wi-Fi including theoretical background, device-specific testing, attack execution (passive and active) and identified vulnerabilities. Each protocol section includes a structured mapping of findings to ETSI EN 303 645 clauses, a vulnerability-scoring matrix, and a detailed discussion of observed weaknesses and recommended mitigation's. This chapter now also incorporates a dedicated analysis of cloud-domain communication, evaluating the security and privacy posture of the mobile applications and backend infrastructure used by each device. This includes interception via Charles Proxy, domain classification, reputation assessment, Common Vulnerabilities and Exposures (CVE) exposure, and governance compliance providing an end-to-end perspective that extends beyond wireless protocol behavior.

- **5 - Proposed Mitigation's:** Introduces a theoretical framework of countermeasures to address the vulnerabilities identified in Chapter 4. The framework proposes time-stamped encryption mechanisms, secure onboarding procedures, unique credential generation, and signed firmware update schemes. Each measure is analyzed in terms of its alignment with ETSI EN 303 645 principles, technical feasibility, and impact on device constraints [66, 56].

- **6 - Discussion:** Interprets the analytical results in the broader context of IoT cybersecurity. It discusses the implications for device manufacturers, service providers, and policymakers, and examines how identified vulnerabilities relate to current industry practices and regulatory trends. Limitations of the research and possible improvements to the testing framework are also addressed.

- **7 - Conclusions:** Summarizes the main findings and contributions of the study. It highlights how the research bridges the gap between theoretical standards and real-world implementation, reinforcing the importance of ETSI EN 303 645 compliance. The chapter concludes by proposing directions for future work, such as expanding testing to emerging protocols, integrating AI-driven anomaly detection, and validating proposed mitigation's in real deployment scenarios.

Chapter 2

Literature Review

The continuous expansion of the IoT ecosystem has enabled widespread deployment of interconnected devices that support automation, monitoring, and intelligent control within residential environments. While these advances bring considerable convenience and functionality, they also introduce significant cybersecurity challenges stemming from the heterogeneity of communication protocols such as Zigbee, Z-Wave, Wi-Fi, Bluetooth Low Energy (BLE), and Message Queuing Telemetry Transport (MQTT) and from the constraints of embedded systems. These protocols operate in resource-limited, distributed environments, often adopting weak authentication mechanisms, outdated cryptographic primitives, or security configurations that fail to meet modern threat models [8, 47, 20].

This chapter reviews the existing body of research on security in home automation systems. It first examines the architectural composition of smart-home environments and identifies vulnerabilities across the device, network, and cloud layers. It then presents protocol-specific analyses focusing on security models, known weaknesses, and mitigation strategies documented in the literature. Subsequently, it discusses existing best-practice frameworks and standards including ETSI EN 303 645 and the OWASP IoT Top 10 that aim to formalize secure development and operational practices. Finally, the chapter identifies current research gaps and highlights the need for unified evaluation approaches capable of comparing protocol vulnerabilities in alignment with internationally recognized cybersecurity requirements.

By synthesizing insights from academic research, industrial assessments, and regulatory standards, this review establishes the conceptual and theoretical basis for the assessments

presented in later chapters. It underscores the critical need for standardized, reproducible methodologies to assess protocol security in smart-home ecosystems.

2.1 IoT/Home Automation Architectures

Home automation environments integrate heterogeneous devices that rely on wireless communication protocols such as Wireless network (Wi-Fi), Zigbee, Z-Wave, BLE, and MQTT to deliver automation, remote control, and data-driven decision-making. These ecosystems are typically structured into three layers:

- **Device layer:** composed of sensors, actuators, and embedded systems.
- **Network layer:** enabling wireless communication using various protocols.
- **Cloud/application layer:** responsible for data storage, analytics, orchestration, and user interaction.

Each of these introduces distinct security challenges. Device-layer vulnerabilities include weak authentication, insecure firmware, and improper key storage [34]. Network-layer risks involve unencrypted communication, replay susceptibility, insecure onboarding, and misconfiguration access control [44]. Cloud-layer vulnerabilities arise from insecure Application Programming Interface (API)s, overexposed analytics endpoints, and inadequate data governance [37].

As the density of connected devices increases, maintaining secure interoperability becomes significantly more complex. Weak onboarding procedures, inconsistent update mechanisms, and lack of lifecycle security exacerbate exposure to attacks such as credential theft, unauthorized control, and lateral movement [20]. These challenges highlight the importance of standardized cybersecurity frameworks most prominently ETSI EN 303 645 to enforce consistent baselines across vendors and communication stacks [50].

2.2 Protocol-Specific Literature

Smart-home communication protocols exhibit unique performance characteristics and security trade-offs. Extensive research has examined their cryptographic models, attack

surfaces, and resilience to modern threats.

Zigbee. Zigbee is a low-power mesh protocol adopted extensively in home automation. Although Zigbee 3.0 introduced improvements such as install codes and stronger key-establishment mechanisms, multiple studies identify persistent weaknesses, including insecure symmetric key reuse, fallback onboarding flows, replay susceptibility, and DoS through management-frame manipulation [27, 22]. Recent work by Yang (2025) demonstrates that insecure commissioning paths can result in device takeover or credential exposure, particularly in deployments relying on legacy fallback keys [67].

Z-Wave. Z-Wave is designed for reliable short-range operation but has faced security downgrades and key-exchange vulnerabilities. Spring's widely [54] downgrade attack illustrates that devices can be forced to use the weaker S0 framework, reducing confidentiality due to predictable key derivation and enabling eavesdropping or injection. Although S2 security introduces authenticated Elliptic Curve Diffie-Hellman (ECDH) and Advanced Encryption Standard (AES)-Counter with CBC-MAC (CCM), misconfiguration's and incomplete implementations remain problematic.

BLE. BLE supports multiple pairing modes with different security guarantees, yet implementations frequently default to "Just Works," enabling passive eavesdropping, replay, and unauthenticated attribute writes. Valenzuela-Pérez [60] reliability-security trade-offs in Bluetooth Mesh networks, noting that redundancy mechanisms increase susceptibility to DoS and interception.

Wi-Fi. Wi-Fi provides high throughput and broad compatibility but suffers from weak onboarding modes, insecure access-point configurations, outdated encryption (e.g., Wi-Fi Protected Access (WPA)2 without management frame protection), and exposure to Evil Twin and deauthentication attacks [38]. Research shows that poorly secured wireless environments significantly compromise overall smart-home resilience [47].

MQTT. MQTT is used for lightweight publish–subscribe communication. Without Transport Layer Security (TLS) and authenticated broker connections, it is highly vulnerable to Man-in-the-Middle (MitM), injection, and replay attacks. Çelik's [16] work demonstrates that compromising an MQTT broker allows full manipulation of smart-home traffic .

Across protocols, comparative research confirms that heterogeneity, backward com-

patibility requirements, and legacy protocol design collectively enlarge the attack surface, leaving devices exposed despite incremental security enhancements [46].

2.3 Standards and Best Practices (ETSI EN 303 645)

As consumer IoT ecosystems continue to expand in scale and complexity, the need for harmonized, internationally recognized security baselines has become critical. Fragmented vendor practices, limited life-cycle support, heterogeneous supply chains, and highly variable security capabilities among devices have contributed to repeated exploitation of smart-home technologies. In response, regulatory bodies and security organizations have developed frameworks to standardize expectations for secure-by-design product development.

Among these initiatives, the ETSI EN 303 645 standard has emerged as the global benchmark for consumer IoT cybersecurity. It establishes a comprehensive set of baseline requirements spanning secure provisioning, credential management, update integrity, communication confidentiality, and privacy protection specifically tailored to low-cost, resource-constrained devices [50]. The standard defines thirteen high-level provisions grouped into two domains: *basic security requirements* and *data protection provisions*. Key areas include the elimination of universal default passwords; establishment of a coordinated vulnerability disclosure policy; secure storage and handling of authentication material; minimization of exposed attack surfaces; resilience against outages and network disruptions; and mandatory validation of software update authenticity.

Importantly, ETSI EN 303 645 goes beyond protocol-level recommendations and extends into implementation practices, addressing issues such as secure boot, cryptographic key life-cycle management, safe recovery procedures, and privacy-preserving data governance. The standard emphasis's architectural principles that support defense-in-depth, requiring manufacturers to implement mechanisms that remain robust even when individual components fail.

Recent product-certification announcements demonstrate rising alignment of commercial IoT devices with ETSI EN 303 645 and related frameworks. Notably, Hikvision announced in April 2025 that it had achieved both ETSI EN 303 645 and EN 18031 cer-

tifications [31]. Similarly, Samsung Electronics' Digital Appliances division received its first TÜV Nord IoT security certification based on ETSI EN 303 645 for its Bespoke AI product line in August 2025 [23], illustrating the expansion of ETSI compliant practices beyond traditional networked devices into the consumer appliance domain.

Complementary initiatives reinforce these principles. The OWASP IoT Top 10 identifies the most prevalent security weaknesses observed in deployed devices, such as weak credentials, insecure network services, inadequate data protection, insecure interfaces, and lack of update mechanisms [48]. These community-driven observations align closely with the ETSI requirements but highlight recurring real-world implementation flaws that manufacturers frequently overlook.

Empirical evaluations of commercial devices across Wi-Fi cameras, Zigbee sensors, Z-Wave actuators, and BLE peripherals demonstrate that many products fail multiple ETSI clauses due to limited vendor maturity, reliance on third-party firmware stacks, or insufficient security governance practices [59, 44]. Challenges include the absence of signed firmware updates, improper key storage, incomplete use of authenticated pairing procedures, overexposed cloud infrastructures, and failure to implement basic wireless resilience measures. Furthermore, supply-chain complexity often leads to undocumented subcontractor code, inconsistent update delivery, and poor post-deployment support.

Despite these limitations, adoption momentum is increasing. The Axis OS 11 [21] programmer and other vendor certification efforts demonstrate the feasibility of integrating ETSI aligned controls into mainstream IoT platforms. In parallel, several national cybersecurity agencies including in the UK, Singapore, Finland, and Australia have begun referencing ETSI EN 303 645 in labeling schemes and regulatory frameworks, accelerating industry convergence toward a common security baseline. These developments illustrate a growing recognition that consumer IoT security cannot rely solely on optional best practices but requires enforceable standards paired with transparent vendor accountability.

In this thesis, all protocol-level and device-level findings were systematically mapped to the relevant ETSI EN 303 645 clauses, allowing the results to be interpreted not only from a technical perspective but also through the lens of established cybersecurity governance. This mapping provides an objective measure of compliance, highlights recurring implementation gaps, and informs the mitigation recommendations presented in subsequent

sections.

2.4 Gaps and Contributions

Despite extensive literature on isolated vulnerabilities and protocol-specific risks, several gaps persist:

- **Lack of unified evaluation methodologies:** Many studies examine vulnerabilities in isolation, without comparative or cross-protocol analysis [34].
- **Limited mapping to standards:** Few works systematically relate empirical protocol vulnerabilities to EN 303 645 clauses, creating a gap between academic findings and regulatory guidance [50].
- **Practical barriers to implementation:** Resource constraints, legacy hardware, and fragmented supply chains impede adoption of robust defenses [1].
- **Rapidly evolving threats:** Emerging attack vectors—including AI-assisted DoS [9] and widespread credential replay—are insufficiently addressed in current protocol designs.

This thesis addresses these gaps through a comparative, multi-protocol assessment (Zigbee, Z-Wave, BLE, Wi-Fi), accompanied by a structured mapping of each finding to EN 303 645. It proposes mitigation strategies aligned with industry standards, including secure onboarding, timestamp-based replay protection, and lightweight key-lifecycle improvements tailored to constrained devices.

Chapter 3

Methodology

This chapter describes the experimental methodology adopted in this thesis, including the laboratory testbed, devices under test (two per protocol), tools and software versions, test scenarios, threat model, evaluation metrics, and ethical considerations. The procedures and tooling choices are derived from the technical logs and consolidation reports produced during the experimental work (Zigbee—KillerBee/Zigator/GNU Radio; Z-Wave—Zniffer/Network Analyzer; BLE—nRF Sniffer/Bleak; Wi-Fi camera analyses).

3.1 Testbed overview

All experiments were conducted in an isolated and fully controlled laboratory environment designed to prevent interference with production or third-party networks. The testbed followed a three-layer architecture:

1. a **device layer**, composed of real consumer IoT devices and a Home Assistant controller used to reproduce realistic smart-home conditions;

2. a **network layer**, providing segmented communication channels through dedicated Wi-Fi Service Set Identifier (SSID)s, Virtual Local Area Network (VLAN) isolation, and Zigbee/Z-Wave coordinators;

3. an **analysis layer**, comprising the workstation infrastructure responsible for packet capture, protocol decoding, active attacks, and cloud-traffic inspection.

This organization reflects both typical smart-home deployment architectures and practical security evaluation workflows, and is summarized in Figure 3.1.

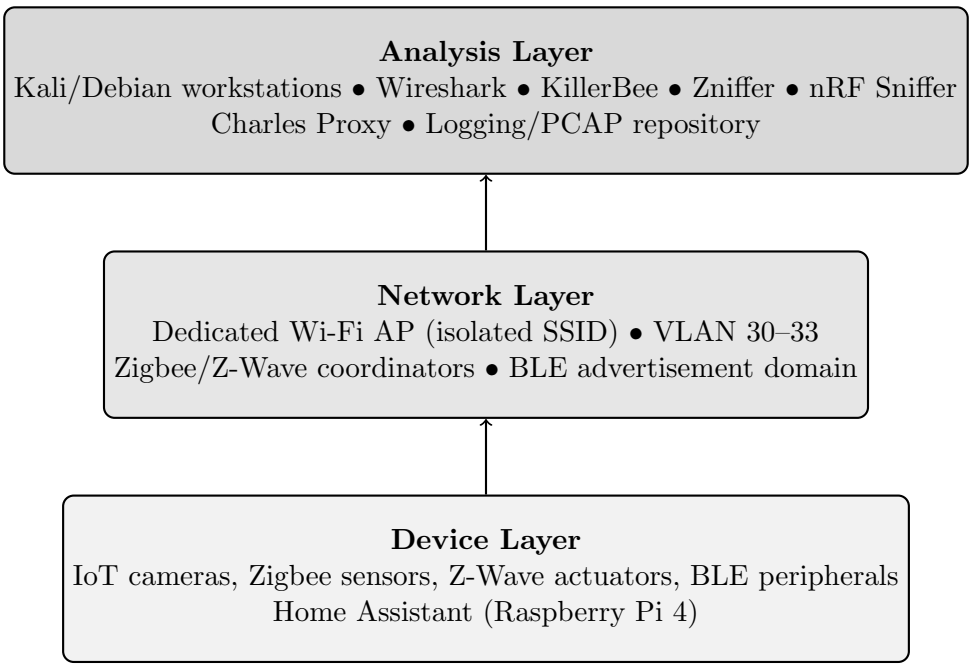


Figure 3.1: Three-layer isolated testbed architecture used for protocol and device analysis.

At a finer level of detail, the testbed comprised the following elements:

- **Device layer:** real consumer IoT devices under test (two per protocol, as described in Section 3.2). Device classes included Wi-Fi cameras, Zigbee sensors and actuators, Z-Wave smart plugs and motion sensors, and BLE peripherals.
- **Network layer:** dedicated access points and gateways provided Wi-Fi and IP connectivity, while Zigbee and Z-Wave coordinators enabled mesh communication. A separate VLAN and guest SSID were configured to isolate IoT traffic. The Wi-Fi network supported WPA2 with optional Management Frame Protection toggling for resilience testing.
- **Sniffers and RF front-ends:**
  - **Zigbee:** Texas Instruments CC2531 (sniffer / Bumblebee firmware), ApiMote, and Atmel RZUSBstick for TX-capable operations. The CC2531 was flashed and used as the primary sniffer, as illustrated in Figure 3.2, which shows the dongle connected to a TI CC Debugger.

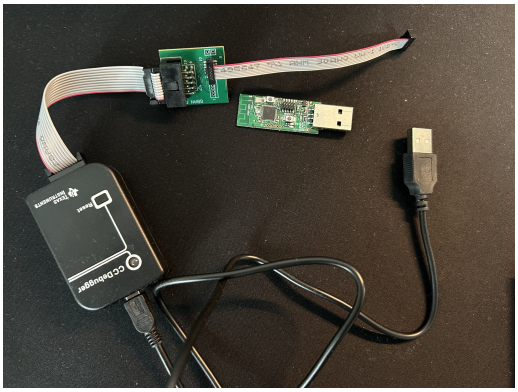


Figure 3.2: TI CC2531 dongle connected to CC Debugger for Zigbee sniffing firmware.

– **Bluetooth LE:** Nordic nRF52840 USB dongle, used with the nRF Sniffer `extcap` module to capture advertisements, connection events, and encrypted-traffic metadata (Figure 3.3).

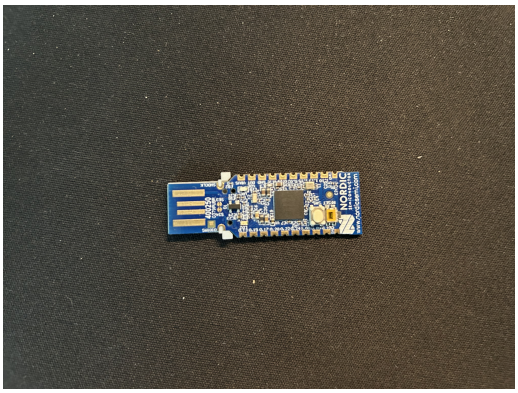


Figure 3.3: Nordic nRF52840 dongle used for BLE packet sniffing.

– **Z-Wave:** Silicon Labs ZGM230S development kit, UZB-based controllers, and CP210x PTI forwarding interfaces for real-time frame capture (Figure 3.4).

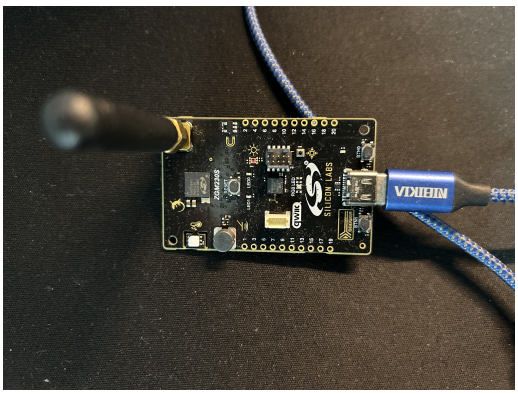


Figure 3.4: Silicon Labs ZGM230S development board used for Z-Wave sniffing.

– **Wi-Fi:** a monitor-mode wireless Wireless Network Interface Controller (NIC) and a Sonoff USB Wi-Fi dongle with external SubMiniature version A (SMA) antenna (Figure 3.5) for WPA handshake capture, deauthentication monitoring, and management-frame analysis.

Figure 3.5: Sonoff Wi-Fi USB dongle with external SMA antenna.

- **Host systems and virtualization:** Linux (Kali / Debian) hosts with controlled Python environments (`pyenv` / `venv`) ensured reproducibility. VirtualBox was employed only for USB passthrough experiments where required (with appropriate `udev` rules).

- **Logging and storage:** all captures (`pcap`/`pcapng`), firmware images, and command outputs were versioned and stored in a dedicated research repository. Sensitive data were redacted prior to publication.

The core of the environment was a **Home Assistant** platform (version 2025.2) deployed on a **Raspberry Pi 4 Model B** with 4 GB RAM running 64-bit Raspberry Pi OS. Home Assistant served as the central control hub and integration gateway for all IoT devices, enabling unified management of Zigbee, Z-Wave, BLE, Wi-Fi, and MQTT-based nodes within a single orchestrated environment. Its modular architecture provided high observability through real-time event logging and internal message tracing, ensuring precise correlation between network captures and logical device events.

Home Assistant was selected because it closely reflects real-world smart-home deployments and supports multiple protocol integrations through official and community add-ons. The following integrations were configured and operated within the isolated network:

- **Zigbee:** Zigbee Home Automation (ZHA) integration using a Texas Instruments CC2531 USB dongle as coordinator.
- **Z-Wave:** Z-Wave JavaScript (JS) integration using a UZB-7 controller connected via USB.
- **Wi-Fi and MQTT:** devices connected to a dedicated WPA2 access point; MQTT messaging handled by the built-in Mosquitto Broker add-on.
- **BLE:** integrated through the Bluetooth Proxy add-on using the Pi's internal adapter and extended via ESPHome nodes.

All devices were paired and managed through the Home Assistant User Interface (UI), which maintained system logs, entity states, and internal message traces (including MQTT events where applicable). These logs were time-synchronized with packet captures, enabling accurate correlation between captured frames and device actions such as motion detection, actuator switching, or state reporting.

Network segmentation employed VLAN IDs (30–33) and a dedicated Wi-Fi SSID to isolate the IoT subnet from the researcher's administrative network. Packet captures were performed simultaneously on the Home Assistant host and on external sniffers to confirm data consistency and timing alignment. This setup replicates realistic smart-home conditions while maintaining controlled and reproducible laboratory parameters, and serves as the baseline for evaluating protocol security under ETSI EN 303 645 compliance criteria.

3.2 Devices under test

Two commercially available devices were selected for each communication protocol to ensure realistic coverage of typical smart-home environments. Each device entry captures its model, role (sensor, actuator, or hub), and relevant hardware identifiers (Vendor ID (VID)/Product ID (PID) or serial number).

All devices were paired, controlled, and monitored through the **Home Assistant** instance deployed on the **Raspberry Pi 4 Model B**. Home Assistant acted as the central management hub, ensuring uniform configuration and synchronized data collection across

all protocol stacks. Event logs, entity states, and internal traces (including MQTT where relevant) were used to correlate device actions with packet captures obtained via external sniffers and host-based monitoring tools.

Wi-Fi devices

- **A9 Mini Wi-Fi Camera** Internet Protocol (IP) camera with local microSD storage (up to 128 GB), integrated via the Home Assistant *Generic Camera* platform and MQTT broker for video streaming and telemetry logging. Analysis revealed unclear use of TLS, absence of two-factor authentication, and incomplete firmware-update verification. These observations were later mapped to non-compliance with ETSI EN 303 645 clauses 5.2 and 5.6.

- **TP-Link Tapo C200** Pan/tilt IP camera integrated through Open Network Video Interface Forum (ONVIF) and Real-Time Streaming Protocol (RTSP) within Home Assistant for video access and control. While vendor documentation confirms TLS for remote access, no public declaration of ETSI compliance was found. This device was primarily used to examine Over-The-Air (OTA)-update behavior and credential-management practices.

Zigbee devices

- **Smart Light (Zigbee 3.0)** Actuator paired using the *ZHA* integration with a Texas Instruments CC2531 USB dongle as coordinator. Captures were obtained via KillerBee (`zbdump`, `zbkey`) and Wireshark. Injection capabilities were limited by the CC2531's RX-only firmware.

- **Motion Sensor (Zigbee 3.0)** Battery-powered sensor paired through the same ZHA network. Pairing and binding sequences were analyzed to attempt decryption of Application Support (APS) Transport Keys and to verify replay protection during network join procedures.

Z-Wave devices

- **Shelly Wave Motion EU LR** Long-range Z-Wave sensor integrated using the *Z-Wave JS* add-on. Inclusion flows and S2 key negotiation were tested against an Aeotec Z-Stick 7 controller.

- **Shelly Wave Plug S (EU)** Smart plug with energy-metering functionality. Security behavior (S0 vs. S2) was evaluated using Zniffer captures and Z-Attack-NG serial-link analysis. Packet sequences were compared with Home Assistant logs to confirm synchronization and event integrity.

Bluetooth Low Energy devices

- **BLE Sensor A** Generic BLE peripheral exposing proprietary Generic Attribute Profile (GATT) characteristics (`FFE1`/`FFE2`/`FFE3`); fuzzed using Python Bleak scripts (`ble_fuzz_cp28.py`) and HomePwn. Pairing and bonding behaviors were assessed against passive sniffing with the Nordic nRF52840 dongle.

- **BLE Peripheral B** Secondary BLE device (e.g., smartwatch or fitness sensor) connected via the Home Assistant Bluetooth Proxy integration. nRF Sniffer captures were synchronized with Wireshark dissections to validate pairing encryption and MitM resilience.

Cross-protocol considerations

Although each protocol was analyzed independently, centralizing all device integrations within Home Assistant ensured consistent logging, synchronized timestamps, and uniform control workflows across Zigbee, Z-Wave, BLE, and Wi-Fi devices. This uniform environment simplified comparison of protocol behaviors and allowed all vulnerability findings to be interpreted under the same operational baseline.

The results for each protocol were subsequently mapped to the relevant clauses of ETSI EN 303 645 in Chapter 4, providing a structured, standards-aligned interpretation without implying deliberate cross-protocol interference testing.

3.3 Tools and versions

All tools used in the experimental campaigns are listed in the appendices with their exact versions (or commit hashes) and a brief justification. Where resolving tool versions required environment fixes (e.g., via `pyenv` or `venv`), the corresponding commands and solutions were documented. The representative tool set includes:

- **Packet capture and analysis:** Wireshark[65] / `tshark` [58] (for Packet Capture Next Generation (PCAPNG) captures and dissectors for Zigbee, Z-Wave, and BLE).
- **Zigbee:** KillerBee [28] suite (`zbdump`, `zbwireshark`, `zbreplay`, `zbkey`, `zbscapy`) with CC2531+Bumblebee [14], and ApiMote/ RZUSBstick [10]for TX-capable tests. CC2531 RX-only limitations were explicitly noted and incorporated into the methodology.
- **Z-Wave:** Simplicity Studio (Flash Programmer, Network Analyzer) [52], Zniffer [68] images (BRD2603A / ZGM230S). J-Link / `JLinkExe` and Simplicity Commander were used for firmware flashing.
- **BLE:** nRF Sniffer [45] + `extcap`, Bleak (Python) [13], HomePwn [32], and Scapy-based tooling. Dependency fixes (`pybluez2`, `pyscard`, `pycryptodome`, `PyGObject`) were recorded to ensure environment stability.
- **Wi-Fi / general network:** `hostapd` [33], `tcpdump`, aircrack-ng (when needed for WPA handshake tests) [3], and OpenSSL for TLS verification.
- **Supporting tools and environment:** `pyenv`, `virtualenv`, and the required `apt` packages (e.g., `build-essential`, `libusb-1.0-0-dev`, `libbluetooth-dev`, `libgirepository-1.0-de` as documented in the BLE/Zigbee configuration logs.

3.4 Test scenarios and threat model

The research adopts a layered threat model based on realistic attacker capabilities in home environments. All scenarios were executed under controlled conditions and with appropriate safeguards.

3.4.1 Threat model

- **Adversary capabilities:** a passive eavesdropper equipped with commodity Software-Defined Radios (SDR)s and protocol sniffers; and an active attacker with local Radio Frequency (RF) transmitters (able to inject frames) for those protocols where the test hardware supports TX (ApiMote, RZUSBstick, Zniffer with TX-capable board). For application-layer tests, the attacker may operate a MitM on local network segments.

- **Assumptions:** the attacker does not initially possess device private keys unless these are recoverable from firmware or flash memory on the lab-owned devices; no attacks are performed on devices or networks outside the defined lab scope. Ethical constraints prohibit any unauthorized network or device intrusion.

3.4.2 Test scenarios

The following scenarios were executed for each protocol (with protocol-specific variations described in Chapter 4):

1. **Passive eavesdropping:** capture of association/pairing flows (Zigbee join, Z-Wave inclusion, BLE pairing, Wi-Fi WPA handshake, MQTT `CONNECT`). Application payloads were analyzed for decryptability when keys or transport contexts were available (e.g., APS Transport Key frames or controller key exports). Tools: Wireshark, KillerBee, Zniffer, nRF Sniffer.

2. **Replay attacks:** replay of previously captured frames (where hardware supported transmission) to observe device behavior (e.g., replay of actuator commands, association floods). Tools: `zbreplay` (Zigbee), Z-Attack-NG replay features, and Scapy scripts for BLE/GATT. All such tests were performed only on lab-owned devices and under strict safeguards.

3. **MitM / broker manipulation (MQTT):** simulation of MitM by operating a malicious broker or intercepting traffic to evaluate the impact of missing or misconfigured TLS and inadequate topic authentication.

4. **DoS / resource exhaustion:** controlled-rate tests to measure resilience to flood and packet-storm vectors (e.g., association floods, beacon flooding). Active DoS tests were carefully limited; higher-intensity tests were executed only under tightly isolated RF conditions.

5. **Key extraction:** where legally permissible (lab-owned devices), memory or factory images were extracted using J-Link or other debug interfaces to check for hardcoded keys or weak Random Number Generator (RNG). Procedures are documented in the Z-Wave and Zigbee sections.

For every scenario, a structured test-plan template was followed: objective, preconditions, equipment (including firmware version and commit), step-by-step commands, expected behavior, observed behavior, and mitigation notes. Representative command lists and outputs are included in the appendices.

3.5 Metrics and evaluation criteria

The experiments evaluate both qualitative and quantitative security attributes using a set of primary metrics describing how each vulnerability affects confidentiality, integrity, availability, and authentication robustness.

Confidentiality assesses whether intercepted frames expose sensitive data, such as clear text credentials, cryptographic keys, or unencrypted telemetry. This metric is evaluated using a binary exposed/not-exposed judgment, complemented by severity indications derived from hex dumps and decoded protocol fields.

Integrity evaluates the ability of an attacker to tamper with or inject messages that alter device behavior. Indicators include the success rate of replay or injection attempts, preconditions required for exploitation (e.g., nonce reuse, counter desynchronisation), and the extent to which devices accept manipulated payloads.

Availability captures resilience to denial-of-service conditions, quantified by the packet rates required to degrade or disrupt functionality, recovery time, and observable side effects (e.g., forced reboots, service hangs, or desynchronisation). All availability tests were executed conservatively, in line with ethical and safety constraints.

Authentication robustness examines the strength of onboarding and session-establishment mechanisms, including mutual authentication, use of ephemeral keys, secure-boot enforcement, and firmware-signature validation. These results are directly compared with relevant clauses of ETSI EN 303 645, particularly those addressing secure onboarding, credential protection, and update integrity.

Ease of exploitation (likelihood) reflects the effort required for a successful attack, considering the necessary skills, tooling, hardware cost, and repeatability. It is expressed qualitatively as *Low*, *Medium*, or *High*, supported by concrete evidence such as commands, scripts, and hardware configurations used during testing.

All these metrics contribute to the normalized 0–10 scoring model developed in this thesis, enabling quantitative comparison of vulnerabilities across heterogeneous protocols and device categories.

Likelihood scale. While vulnerability tables in this thesis express likelihood qualitatively (*Low*, *Medium*, *High*), the scoring model requires numerical values. In accordance with widely used risk-assessment frameworks (NIST SP 800-30 [49], ISO/IEC 27005 [55]), the following mapping is applied:

$$\text{Low} = 0.3, \qquad \text{Medium} = 0.6, \qquad \text{High} = 1.0.$$

These normalized values are used directly as the likelihood term I_L in Equation 3.1, ensuring consistent treatment of risk across all tested vulnerabilities.

In addition to numerical scoring, every observed vulnerability is mapped to the specific clause of ETSI EN 303 645 that it violates, using the per-device compliance tables presented in Chapter 4. These mappings inform the remediation guidance and the standard-aligned mitigation strategies derived from the experimental results.

Scoring methodology. To ensure consistent and reproducible evaluation, each vulnerability is assigned a normalized score between 0 and 10, computed through a weighted aggregation of the core security metrics:

$$\text{Score} = \frac{w_C \cdot I_C \; + \; w_I \cdot I_I \; + \; w_A \cdot I_A \; + \; w_L \cdot I_L}{w_C + w_I + w_A + w_L} \times 10, \tag{3.1}$$

where:

- I_C, I_I, I_A, and I_L are normalized values in $[0, 1]$ representing the impact on confidentiality, integrity, availability, and the likelihood of successful exploitation;
- w_C, w_I, w_A, and w_L are weighting factors expressing the relative importance of each metric.

Unless otherwise specified, all weights are set to unity ($w_C = w_I = w_A = w_L = 1$) to avoid bias across the evaluated protocols and device classes.

Cloud-domain risk scoring model. Beyond protocol-level Confidentiality, Integrity, and Availability (CIA) properties, this thesis also evaluates the security and privacy posture of the cloud infrastructure and remote services contacted by each device and its associated mobile application.

To ensure reproducible and transparent classification, each cloud endpoint is assigned a CloudRisk score based on a set of discrete technical and privacy attributes. The score is computed as:

$$\text{CloudRisk} = S_{\text{ports}} + S_{\text{rep}} + S_{\text{cve}} + S_{\text{blacklist}} + S_{\text{tls}} + S_{\text{privacy}}$$

where each component reflects a measurable risk factor:

- $S_{\text{ports}} = \begin{cases} 0, & \text{Only port 443 or minimal exposure} \\ 1, & \text{Multiple open ports (e.g., 80/443)} \\ 2, & \text{High-risk or unnecessary ports (22, 3306, 8080, 8883, etc.)} \end{cases}$
- $S_{\text{rep}} = \begin{cases} 0, & \text{Clean reputation (0/94 or 0/96)} \\ 1, & \text{Minor flags (1–2/94)} \\ 2, & \text{Multiple reputation findings} \end{cases}$
- $S_{\text{cve}} = \begin{cases} 0, & \text{0–1 associated CVEs} \\ 1, & \text{2–3 CVEs} \\ 2, & \text{4+ CVEs (e.g., developer.adgo.link:8083)} \end{cases}$

- $S_{\text{blacklist}} = \begin{cases} 0, & \text{Not present on threat or abuse blacklists} \\ 2, & \text{Listed on security blacklists (e.g., ctobsnssdk.com)} \end{cases}$
- $S_{\text{tls}} = \begin{cases} 0, & \text{Strong TLS configuration on all endpoints} \\ 1, & \text{Mixed HTTP/HTTPS or weak TLS parameters} \\ 2, & \text{No TLS support} \end{cases}$
- $S_{\text{privacy}} = \begin{cases} 0, & \text{No analytics or telemetry concerns} \\ 1, & \text{Analytics or tracking SDKs present (e.g., Pangolin)} \\ 2, & \text{Advertising SDKs and cross-application data aggregation} \end{cases}$

The final classification is obtained by mapping the score to three qualitative categories:

$$\text{Secure} \quad (0{-}2), \qquad \text{Suspicious} \quad (3{-}6), \qquad \text{Malicious} \quad (7{-}12).$$

This CloudRisk model provides a consistent and interpretable mechanism for evaluating backend infrastructures and is later correlated with ETSI EN 303 645 requirements for secure communication (Clause 5.3), minimization of exposed attack surface (Clause 5.6) and protection of personal data (Clause 5.11).

Device-Level Cloud Risk Score. To obtain a single cloud-risk score per device, the individual domain risk scores are aggregated and normalized by the number of domains contacted. For a device with N observed domains, each domain having a CloudRisk score $d_i \in [0, 12]$, the final Device Cloud-Risk Score (DCRS) is:

$$\boxed{\text{DCRS} = \frac{1}{N} \sum_{i=1}^{N} d_i}$$

The resulting value is then mapped to a qualitative category using the classification ranges shown below:

$$\textbf{Low Risk: } 0 \leq \text{DCRS} < 3$$

Moderate Risk: $3 \leq \text{DCRS} < 7$

High Risk: $7 \leq \text{DCRS} \leq 12$

These thresholds allow a direct comparison between devices by capturing the average severity of all cloud endpoints they communicate with.

3.6 Ethical considerations and safe lab practices

All experiments carried out in this research adhered strictly to ethical and legal guidelines. The experimental scope was limited to devices owned by the researcher or used with explicit written consent, ensuring that no testing was performed on third-party networks or equipment.

RF experiments were conducted within isolated laboratory segments or on dedicated SSIDs and VLANs to prevent cross-contamination or accidental interference with production environments. Isolation measures such as separate access points, physical separation, and conservative transmission power were applied and documented.

Sensitive data extracted during testing including private keys and any personally identifiable information were redacted in all published material unless coordinated vendor disclosure had been completed. Potential vulnerabilities identified during the study followed responsible disclosure principles when applicable.

Active tests involving injections, replays, or flooding were executed with strict safety controls, including rate limiting, automated stop conditions, and continuous monitoring to immediately abort upon unexpected behavior. All corresponding logs and packet captures were preserved for auditing and reproducibility.

Finally, each experiment was documented with the exact commands used, the tool and environment versions, and, where permissible, the generated PCAPs and logs. This level of documentation supports reproducible research and enables independent verification under similarly controlled conditions.

Chapter 4

Protocol and Ecosystem Analysis

This chapter presents the technical analysis and vulnerability mapping of the communication protocols and cloud-service ecosystems investigated in this research: Wi-Fi, Zigbee, Z-Wave, BLE, and the application-layer cloud infrastructures associated with each vendor. Together, these layers form the operational backbone of contemporary smart-home environments, enabling wireless connectivity, remote management, telemetry collection, and cloud-based automation workflows.

Although these technologies were designed with different performance and interoperability goals (e.g., low power, long range, IP connectivity, mobile integration), their security properties vary substantially across implementations. Weaknesses often arise not only from protocol limitations but also from vendor-specific decisions such as insecure pairing procedures, weak authentication, unencrypted local streams, insufficient update mechanisms, or excessive communication with third-party analytics services.

The objective of this chapter is to systematically assess each protocol and cloud ecosystem within a controlled, reproducible laboratory environment, identify vulnerabilities that affect CIA, and map all findings to the security requirements of ETSI EN 303 645. The analysis combines theoretical review with empirical experimentation, supported by packet captures, protocol dissection, and cloud-domain profiling.

Experiments were conducted on isolated non-production networks using protocol-specific sniffers, SDRs, firmware analysis tools, and controlled active-attack tooling. The methodology included:

- **Passive collection:** capturing traffic using Wireshark, KillerBee, Zniffer, nRF Sniffer, , and monitor-mode Wi-Fi NICs to observe pairing, association, data exchange, and event reporting.
- **Active interaction:** performing controlled replay, fuzzing, flooding, and deauthentication tests where legally permitted to evaluate resilience to manipulation.
- **Cloud-domain inspection:** intercepting Android application traffic with Charles Proxy to enumerate remote endpoints, TLS usage, metadata exposure, analytics Software Development Kit (SDK)s, update servers, and backend infrastructure behavior.

To facilitate comparability across heterogeneous technologies, each subsection follows a unified and repeatable analytical structure:

1. **Protocol/Cloud overview and security features:** architectural summary, cryptographic model, pairing/authentication mechanisms, expected guarantees, and known design limitations.
2. **Devices tested:** hardware models, firmware versions, and integration details used in the experiments.
3. **Test scenarios executed:** laboratory procedures, specific tools/commands, and expected outcomes.
4. **Results:** experimental findings including observed behaviors, misconfiguration's, plaintext exposure, timing anomalies, or unusual cloud interactions derived from packet captures and system logs.
5. **Vulnerability Mapping and ETSI EN 303 645 Alignment:** classification of each finding against CIA criteria and precise mapping to relevant ETSI clauses (5.1–5.13).
6. **Proposed Mitigation's:** practical recommendations for vendors, integrators, and standards bodies to increase robustness and compliance.

By applying this structured and consistent framework to Wi-Fi, Zigbee, Z-Wave, BLE, and cloud-domain communication flows, the chapter provides a holistic evaluation of security risks across both RF layers and backend ecosystems. The results reveal that even when wireless protocols implement strong cryptographic mechanisms real-world device behavior and cloud-service decisions can introduce critical weaknesses, particularly in plaintext metadata exposure, unsigned firmware updates, weak onboarding flows, or excessive communication with third-party analytics frameworks.

The subsequent sections detail the complete findings for each technology.

4.1 Wi-Fi

Wi-Fi was selected as one of the primary communication protocols evaluated in this research due to its dominant role in home automation and consumer IoT ecosystems. Unlike mesh-oriented technologies such as Zigbee and Z-Wave, Wi-Fi provides direct IP connectivity and high data throughput, making it the preferred medium for devices that require real-time video streaming, cloud synchronization, or mobile-app integration. At the same time, this reliance on open wireless infrastructure and backward-compatible standards (e.g., legacy WPA2 configurations) introduces well-known attack surfaces at both the wireless and application layers [11, 39].

In this study, two commercially available Wi-Fi devices were analyzed: the **A9 Mini Wi-Fi Camera** and the **TP-Link Tapo C200**. Both devices were integrated into the Home Assistant environment running on the Raspberry Pi 4 described in Section 3.1, connected through a dedicated VLAN (`IOT_LAB`) and monitored with `tcpdump` and Wireshark for protocol-level inspection. Beyond passive traffic analysis, the testbed incorporated active wireless attacks using the Wifi-Crack framework [53], exercising handshake capture, Pairwise Master Key Identifier (PMKID) collection, authentication denial-of-service (AAuth), deauthentication flooding (DAuth), beacon flooding (BFlood), and Evil Twin (ETwin) scenarios against both cameras.

To complement the network-layer analysis, the communication behavior of each device during Wi-Fi onboarding and normal operation was also observed at the application layer. The mobile applications associated with the A9 Mini and Tapo C200 were perox-

ided through Charles Proxy running on the analysis workstation, with the Android device configured to use the proxy as its Hypertext Transfer Protocol (HTTP)/Hypertext Transfer Protocol Secure (HTTPS) gateway. This setup enabled the identification of remote endpoints contacted during pairing, live stream access, telemetry reporting, and firmware-check routines. Only high-level observations relevant to Wi-Fi security (e.g., plaintext HTTP, missing certificate pinning, or repeated calls during de-authentication events) are reported in this section; the full cloud endpoint classification, WHOIS metadata, reputation scores, and infrastructure risk assessment are presented separately in Section 4.5.

The Wi-Fi analysis therefore combines three complementary perspectives:

1. **Local wireless security**, including handshake protection, susceptibility to management-frame attacks, and resilience to Wi-Fi DoS (Wifi-Crack tests).

2. **Application and transport security**, assessed via passive and proxies inspection of HTTP/HTTPS, RTSP and vendor-specific APIs.

3. **Cloud and ecosystem exposure**, based on the mapping of remote services, third-party domains and their security posture.

The following subsections present a structured examination of the Wi-Fi layer, including an overview of relevant security principles, device descriptions, executed test scenarios (both manual and Wifi-Crack-based), obtained results, the mapping of observed behavior to ETSI EN 303 645 requirements, and the resulting mitigation recommendations.

4.1.1 Wi-Fi Security Overview

Wi-Fi remains one of the most common connectivity technologies in consumer IoT ecosystems. Its broad adoption and backward compatibility make it a frequent attack vector, particularly when devices employ outdated encryption schemes or weak authentication mechanisms. Vulnerabilities such as insecure onboarding, unprotected data transmission, and insufficient firmware update mechanisms continue to expose IoT devices to risks including credential theft, MitM attacks, and unauthorized remote access [11, 39, 43].

This section evaluates two consumer-grade IP cameras the A9 Mini Wi-Fi Camera and the TP-Link Tapo C200 to assess their alignment with the ETSI EN 303 645 security

baseline, focusing on data protection, secure communication, authentication, and firmware update integrity.

4.1.2 Devices Tested

- **A9 Mini Wi-Fi Camera:** Low-cost IP camera supporting microSD storage (up to 128 GB) and 2.4 GHz Wi-Fi operation. No official documentation was found regarding supported security protocols. Initial pairing and control occurred via the manufacturer's mobile app. Device identifiers: Vendor Identifier (ID) `0x2E17`, Product ID `0xC001`. Firmware version unknown.

- **TP-Link Tapo C200:** Full-HD (1080p) pan/tilt camera integrated via ONVIF and RTSP within Home Assistant. According to TP-Link documentation, remote communication uses TLS, and initial setup requires custom credentials. However, automatic firmware updates and public vulnerability disclosure are not implemented. Firmware v1.3.0 (Build 2025-03-15).

4.1.3 Test Scenarios Executed

The test scenarios were designed to evaluate the security posture of each Wi-Fi device under realistic adversarial conditions, focusing on authentication robustness, communication confidentiality, and resilience to network-level disruption. Before executing the Wifi-Crack attack modules, the wireless interface had to be switched into **monitor mode** using the Aircrack-NG toolkit on the Kali Linux system (e.g., `airmon-ng start wlan0`). This preparation enabled the capture and injection of 802.11 management frames required for the attacks. Table 4.1 summarizes the complete set of scenarios and their corresponding objectives.

Table 4.1: Wi-Fi Test Scenarios

Scenario	**Tools / Commands**	**Expected Result**
Passive capture of association/auth frames	`tcpdump -i wlan0mon -w a9_assoc.pcap`, Wireshark	Observe WPA2 handshake negotiation
Active deauth / replay test (controlled)	`aireplay-ng -0 10 -a BSSID -c client`	Device reconnects using secure WPA handshake
Traffic inspection	Wireshark + `mitmproxy`	Verify TLS use for remote API calls
Firmware-update monitoring	`tcpdump` during OTA trigger	Confirm HTTPS/TLS connection and signature check
Credential robustness test	`hydra` (local API brute-force)	Reject weak or default passwords

4.1.4 Test Scenarios Executed (Wifi-Crack)

The Wifi-Crack framework was used to execute a series of automated wireless-layer attack routines designed to evaluate device resilience against common 802.11 manipulation vectors. Each module performs a controlled, repeatable stress or deception operation ranging from authentication disruption to beacon flooding and Evil Twin impersonation allowing systematic observation of device behavior under adverse radio-frequency conditions. Table 4.2 summarizes the six attack modules included in the framework, the specific 802.11 mechanisms they target, and the expected security impact assessed during testing.

Table 4.2: Wifi-Crack Attack Scenarios

Scenario	**Wifi-Crack Command**	**Expected Result**
Handshake Capture	`WifiCrack.py -i wlan0mon -m Handshake`	Device reconnects; handshake captured
PMKID Attack (clientless)	`WifiCrack.py -i wlan0mon -m PKMID`	PMKID retrieved (if supported)
AAuth	`WifiCrack.py -i wlan0mon -m AAuth`	Device unable to authenticate
DAuth	`WifiCrack.py -i wlan0mon -m DAuth`	Device repeatedly disconnected
BFlood	`WifiCrack.py -i wlan0mon -m BFlood`	Wireless environment saturated
Evil Twin Attack (ETwin)	`WifiCrack.py -i wlan0mon -m ETwin`	Rogue AP broadcasted

4.1.5 Results

This section presents the experimental results obtained for the two Wi-Fi devices evaluated in the testbed: the A9 Mini Wi-Fi Camera, a low-cost unbranded camera with no vendor security governance, and the TP-Link Tapo C200, a commercially established device with documented firmware, cloud infrastructure, and update mechanisms. Each device was assessed under identical laboratory conditions using the Wifi-Crack framework, passive packet collection, controlled deauthentication and association-spoofing tests. The results highlight substantial differences in security posture: while the Tapo C200 demonstrates partial alignment with modern Wi-Fi and IoT security baselines, the A9 Mini exhibits critical weaknesses in communication confidentiality, onboarding, firmware integrity, and cloud-service trustworthiness. Detailed findings for both devices are presented in the following subsections.

A9 Mini Wi-Fi Camera

Captured traffic revealed unencrypted HTTP requests between the mobile app and the device, confirming the absence of TLS. The camera used 2.4 GHz WPA2-PSK only, with no WPA3 support. Firmware-update endpoints were inactive, suggesting a lack of signed or automated updates. The device transmitted metadata (camera ID and SSID) in plaintext during initialization.

Under the Wifi-Crack attack suite, the A9 Mini demonstrated high susceptibility to wireless DoS:

- **AAuth:** Completely prevented successful authentication, causing the device to freeze and become unreachable from the app (Figure 4.1). Only a full reset and re-pairing restored operation.
- **DAuth:** Forced repeated disconnections and corrupted the pairing state, again requiring manual re-onboarding.
- **BFlood:** Introduced significant instability in the wireless environment; the camera's SSID became intermittently unavailable, and the mobile application frequently reported "device offline".

- **ETwin (Evil Twin):** When the rogue Access Points (AP) copied the original SSID and used a similar channel, the app could be tricked into attempting connections to the attacker-controlled network if the user accepted the new SSID/password pairing.

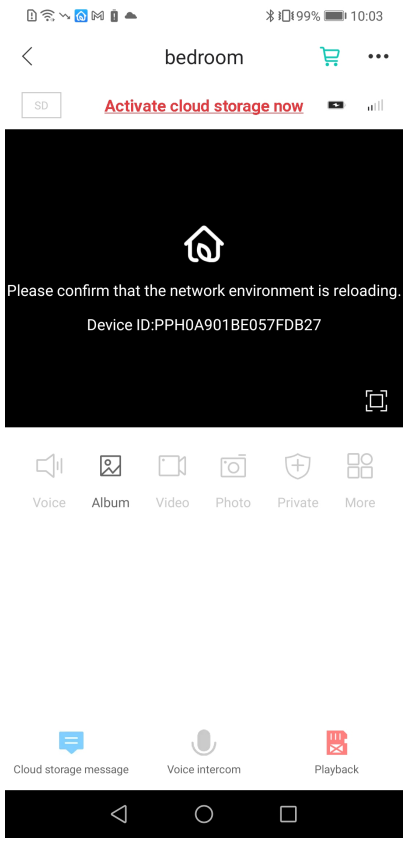


Figure 4.1: A9 Mini mobile application showing connection failure

Overall, the A9 Mini exhibits poor resilience against both classical Wi-Fi management attacks and application-layer disruptions.

TP-Link Tapo C200

TLS 1.2 was confirmed during app-to-cloud communication via `tapo.tplinkcloud.com`. Local RTSP streams were unencrypted, accessible within the Local Area Network (LAN) without authentication if the stream Uniform Resource Locator (URL) was known. Firmware updates occurred over HTTPS, but no signature verification could be confirmed. Credential setup was mandatory at first launch, eliminating universal default passwords.

Under the same deauthentication conditions as the A9, the Tapo C200 automatically reconnected to the network within a few seconds after the DAuth signal stopped. Home

Assistant logs confirmed reauthentication and restoration of entity availability without user intervention. This behavior reflects a more robust network recovery mechanism and better adherence to secure-by-default principles.

The Wifi-Crack tests showed a more resilient profile:

- **AAuth:** Authentication flooding produced short service interruptions (temporary stream freezing), but the camera recovered automatically and rejoined the legitimate AP without requiring user interaction (Figure 4.2).
- **DAuth:** Deauthentication caused brief disconnections, yet the device consistently renegotiated a secure WPA handshake and restored the video stream.
- **BFlood:** Beacon flooding mostly affected the mobile application's view of available networks (scan instability) rather than the camera itself; once associated, the device maintained its session with the configured AP.
- **ETwin:** The camera did not automatically roam to the Evil Twin AP. User interaction would still be required to reconfigure Wi-Fi settings, reducing exposure but not fully eliminating the risk of social engineering.

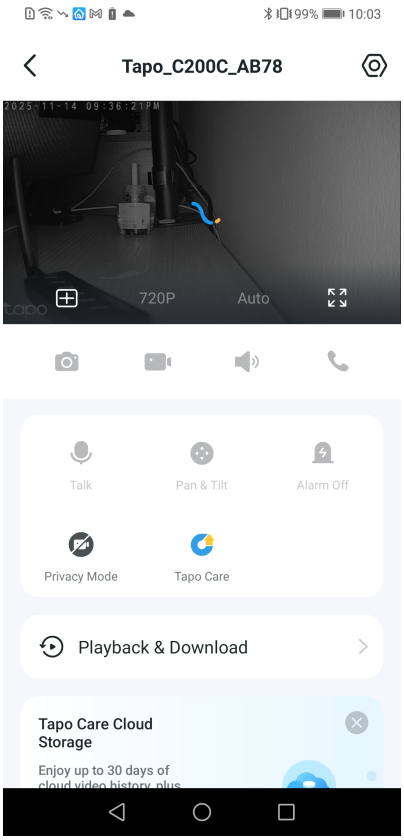


Figure 4.2: TP-Link Tapo C200 mobile application showing connection failure.

In summary, the Tapo C200 shows significantly stronger resilience to Wifi-Crack attacks than the A9 Mini, although local RTSP and unsigned updates remain important weaknesses.

4.1.6 Mapping to ETSI EN 303 645

From a Wifi-Crack perspective, Clause 5.10 ("Ensure resilience to outages") is particularly relevant, as it covers robustness against wireless disruptions such as deauthentication and rogue access points. See table 4.3

Table 4.3: ETSI Clause 5.10 — Device Resilience to Wireless Disruption

ETSI Clause 5.10 Requirement	**A9 Mini**	**Tapo C200**
Recover from deauthentication attacks	Fail	Pass
Recover from authentication DoS	Fail	Partial
Resist rogue access point attacks	Partial	Pass
Maintain functionality after beacon flooding	Fail	Partial

In addition to evaluating resilience, the remaining ETSI EN 303 645 clauses were assessed for each device individually. The results for Device A are presented in Table 4.4, and the results for Device B are shown in Table 4.5.

Table 4.4: Wi-Fi Device A → ETSI EN 303 645 Mapping

ETSI Clause	**A9 Mini Camera**
5.1 – No universal default passwords	Fails – hardcoded default
5.2 – Secure storage of credentials	None – plaintext observed
5.3 – Secure communications	Fails – no TLS/WPA3
5.6 – Minimize exposed interfaces	None – open HTTP port 80
5.9 – Software update integrity	Fails – no OTA mechanism
5.10 – Ensure resilience to network disruptions	Fails – requires manual re-pairing after DAuth/AAuth
5.11 – Protect personal data	Fails – plaintext metadata

Table 4.5: Wi-Fi Device B → ETSI EN 303 645 Mapping

ETSI Clause	**Tapo C200**
5.1 – No universal default passwords	Pass – user-defined at setup
5.2 – Secure storage of credentials	Partial – obscured but unverified
5.3 – Secure communications	Partial – TLS remote, RTSP unencrypted
5.6 – Minimize exposed interfaces	Partial – LAN RTSP unauthenticated
5.9 – Software update integrity	Partial – HTTPS updates, no signature check
5.10 – Ensure resilience to network disruptions	Pass – automatic secure reconnection after DAuth/AAuth
5.11 – Protect personal data	Partial – local storage unencrypted

4.1.7 Vulnerability Mapping

The vulnerabilities identified during Wi-Fi testing were evaluated using the CIA-based scoring methodology developed in 3.5, which quantifies each issue by combining its confidentiality, integrity, and availability impact with a normalized likelihood value. The results are presented across two complementary tables. Table 4.6 covers the vulnerabilities directly exercised using the Wifi-Crack attack modules, including deauthentication, authentication flooding, beacon flooding, and Evil Twin exposure. These attacks primarily affect availability and were scored accordingly, with additional CIA applied where credential exposure or session manipulation was possible.

Table 4.7 expands the assessment to broader protocol-level and lifecycle weaknesses observed during empirical testing, such as unencrypted communication channels, firmware update deficiencies, persistent reconnection failures, weak password policies, and the absence of RTSP stream protection. Because these vulnerabilities affect multiple CIA properties simultaneously, they generally receive higher severity scores under the scoring model particularly where they impact confidentiality and integrity in addition to availability.

Together, the two tables provide a structured and quantitative view of Wi-Fi security posture across both active adversarial conditions and passive protocol/lifecycle analysis, offering a consistent basis for cross-protocol comparison under the ETSI EN 303 645 security requirements.

Table 4.6: WiFi-Crack Vulnerability Scoring

Vulnerability	**CIA**	**Likelihood**	**Score (/10)**
AAuth susceptibility (A9)	A	High (1.0)	$\frac{0+0+1.0+1.0}{4} \times 10 = 5.0$
DAuth susceptibility (A9)	A	High (1.0)	$\frac{0+0+1.0+1.0}{4} \times 10 = 5.0$
Beacon Flood instability (A9)	A	High (1.0)	$\frac{0+0+1.0+1.0}{4} \times 10 = 5.0$
Evil Twin exposure (A9)	C / I	Medium (0.6)	$\frac{1.0+1.0+0+0.6}{4} \times 10 = 6.5$
AAuth partial impact (Tapo)	A	Medium (0.6)	$\frac{0+0+1.0+0.6}{4} \times 10 = 4.0$
BFlood-induced slowdown (Tapo)	A	Medium (0.6)	$\frac{0+0+1.0+0.6}{4} \times 10 = 4.0$

Table 4.7: Wi-Fi Vulnerability Scoring

Vulnerability	**CIA**	**Likelihood**	**Score (/10)**
Unencrypted communication (A9)	C / I	High (1.0)	$\frac{1.0+1.0+0+1.0}{4} \times 10 = 7.5$
Lack of firmware updates (A9)	C / I / A	Medium (0.6)	$\frac{1.0+1.0+1.0+0.6}{4} \times 10 = 9.0$
Failure to reconnect after DAuth/AAuth (A9)	A	High (1.0)	$\frac{0+0+1.0+1.0}{4} \times 10 = 5.0$
Unprotected RTSP stream (Tapo)	C / I	High (1.0)	$\frac{1.0+1.0+0+1.0}{4} \times 10 = 7.5$
No signature validation (Tapo)	C / I / A	Medium (0.6)	$\frac{1.0+1.0+1.0+0.6}{4} \times 10 = 9.0$
Weak password policy (both)	C / I / A	Medium (0.6)	$\frac{1.0+1.0+1.0+0.6}{4} \times 10 = 9.0$

4.1.8 Proposed Mitigation's

The vulnerabilities observed during Wi-Fi testing including susceptibility to deauthentication and authentication floods, exposure to Evil Twin attacks, lack of firmware integrity validation, and insecure local streams underscore the need for a comprehensive hardening strategy across Wi-Fi devices and their supporting infrastructures. To align Wi-Fi deployments with the security expectations of ETSI EN 303 645, the following mitigation measures are recommended:

- **Adopt modern cryptographic standards:** Enforce WPA3-Personal or WPA3-Enterprise for wireless authentication and require Protected Management Frames (PMF) (IEEE 802.11w) to protect against deauthentication and disassociation spoofing. All remote services should use TLS 1.3 with secure cipher suites as required by ETSI §5.3.

- **Harden association and authentication flows:** Implement rate-limiting, connection-attempt back off, and anomaly detection to mitigate flooding attacks such as `DAuth/AAuth` storms. Devices should reject excessive or abnormal authentication requests and log such attempts for auditability.

- **Strengthen AP verification to prevent Evil Twin attacks:** Apply **BSSID**

locking, certificate pinning for remote services, and validation of SSID/Basic Service Set Identifier (BSSID) pairs during onboarding. Devices should warn users or abort connection attempts when AP fingerprints deviate from known profiles.

- **Secure local video and data streams:** Local LAN services such as RTSP or HTTP endpoints must be protected using mutual authentication, token-based access, or per-session credentials. Unauthenticated streams such as those observed on the A9 camera constitute a direct violation of ETSI §5.3.

- **Enforce robust firmware update security:** Require digitally signed firmware, verified via a complete chain-of-trust prior to installation, and implement rollback protection to prevent downgrades to insecure versions. Vendors must publish clear update logs and maintain long-term support as defined in ETSI §5.9.

- **Improve resilience to channel and beacon flooding:** Integrate wireless anomaly detection capable of identifying beacon floods, excessive probe requests, or abnormal channel conditions. Devices should degrade gracefully under such attacks and generate warnings consistent with ETSI §5.10.

- **Protect stored credentials and sensitive data:** Ensure that Wi-Fi passwords, tokens, and session material stored on the device are encrypted using hardware-backed key storage where possible. Metadata including SSID names, device identifiers, or connection analytics should be minimized or masked before any transmission.

- **Enable secure and deterministic recovery paths:** Devices must reconnect securely after network disruptions without reverting to insecure fallback onboarding mechanisms, captive portals, or open APs.

- **Implement transparent security governance:** Manufacturers should maintain coordinated vulnerability disclosure processes, publish update lifecycle information, and provide clear assurance of *secure-by-design* practices. This directly aligns with ETSI 5.1, 5.2, and 5.4.

Collectively, these mitigation measures substantially strengthen Wi-Fi device resilience and ensure alignment with the cybersecurity requirements of ETSI EN 303 645. The com-

parative analysis demonstrated that while the TP-Link Tapo C200 shows partial alignment particularly in onboarding and cloud communication security the A9 Mini Wi-Fi Camera fails several critical clauses, including communication confidentiality, firmware update integrity, and resilience to network disruptions. These findings highlight the urgent need for enforceable regulatory standards and secure-by-design engineering across consumer IoT products.

4.2 Zigbee

Zigbee was selected as one of the primary wireless protocols for this research due to its widespread adoption in home automation ecosystems and its layered security model based on IEEE 802.15.4. The experiments aimed to analyze the protocol's security posture under realistic conditions using both passive and active testing techniques. A dedicated Home Assistant instance running on a Raspberry Pi 4 served as the control hub for the Zigbee network, coordinating two commercial devices an actuator and a sensor connected through a Texas Instruments CC2531 coordinator.

The assessment combined multiple open-source frameworks and hardware tools, including `KillerBee`, `Zigator`, and `GNU Radio`, to capture, replay, and inject IEEE 802.15.4 frames in a controlled environment. The CC2531 dongle was flashed with both sniffer and Bumblebee firmware versions to enable bidirectional testing and was identified under vendor ID `0451`. A full channel sweep (11–26) was executed to locate active Personal Area Network (PAN)s, followed by replay and orphan-notification injections to test device resilience against frame reuse and management-layer attacks.

All tests were performed inside an isolated laboratory network to avoid interference with production systems, and all captured traffic was stored in PCAP format for correlation with Home Assistant event logs. The resulting dataset supports detailed mapping of observed behaviors such as replay rejection, key-exchange handling, and network re-association to relevant clauses of ETSI EN 303 645, particularly sections 5.3 (Secure Communications), 5.4 (Secure Onboarding), 5.9 (Software Update Integrity), and 5.10 (Resilience to Outages and Attacks).

The following subsections summarize the Zigbee protocol's security architecture, de-

scribe the tested devices and scenarios, present empirical results, and document their alignment with international IoT security standards.

4.2.1 Protocol Overview and Security Features

Zigbee is a mesh networking stack built on IEEE 802.15.4 at the PHY/MAC layers. The Zigbee stack adds Network (NWK), APS, and application profiles (ZCL clusters) on top of 802.15.4. Security primitives include AES-128 in CCM* mode for confidentiality and integrity at the NWK/APS layers, with *network keys* (shared within a PAN) and *link keys* (per-device/application) used for trust establishment. Modern Zigbee 3.0 deployments typically rely on *install codes* for initial key distribution; however, legacy devices and misconfiguration's (e.g., default/link keys, insecure commissioning) remain susceptible to passive eavesdropping, replay, and key recovery during insecure joins. These risks map directly to ETSI EN 303 645 requirements on secure communications, update integrity, onboarding, and resilience.

4.2.2 Devices Tested

All Zigbee devices were paired and exercised via Home Assistant (Raspberry Pi 4) using the ZHA integration with a TI CC2531 coordinator. Two representative Zigbee 3.0 devices were evaluated during the experiments.

Device A, acting as an actuator (e.g., smart light), operated with firmware *(insert exact string from logs)* and exposed the On/Off and Level Control clusters.

Device B, a battery-powered motion sensor, operated with firmware *(insert exact string)* and implemented the Intruder Alarm System (IAS) Zone and Power Configuration clusters.

Coordinator and sniffer operations relied on the TI CC2531 USB dongle, which initially enumerated as VID:PID `0451:16a8` when operating in coordinator mode. A permissive `udev` rule (`0666`, group `plugdev`) was applied to allow non-root access to the device. Figure 4.3 shows the CC2531 in its default coordinator state before firmware modification.

To enable packet-capture capabilities, the device was reflashed with the Bumblebee v2.1 sniffer firmware (`cc2531-bumblebee-v2.1.hex`). The flashing process completed successfully, as illustrated in Figure 4.4. Following the firmware replacement, the device

enumerated as VID:PID `0451:16ae`, confirming that it had entered sniffer mode appropriate for use with KillerBee; this mode is shown in Figure 4.5.

Firmware flashing (evidence).

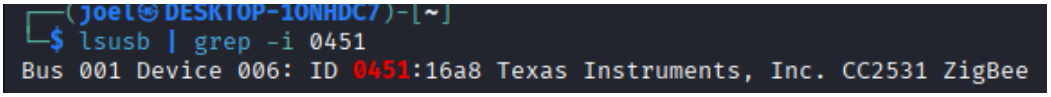


Figure 4.3: TI CC2531 enumerating as `0451:16a8` in coordinator mode

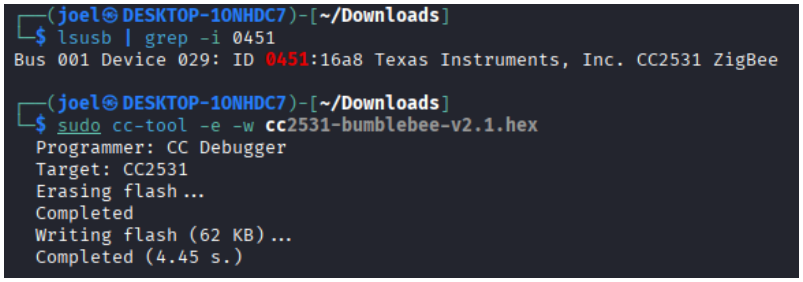


Figure 4.4: Flashing Bumblebee v2.1 firmware successful write.

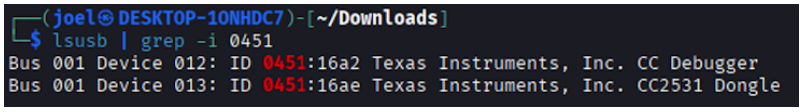


Figure 4.5: TI CC2531 enumerating as `0451:16ae` after flashing, indicating sniffer mode.

4.2.3 Test Scenarios Executed

Identification of the Zigbee Personal Area Network Identifier was performed during the beacon-scanning phase and through inspection of 802.15.4 MAC headers in association frames. In Wireshark, the PAN ID is visible in the `Destination PAN ID` and `Source PAN ID` fields and can be filtered using the `wpan.panid` display filter. Beacon frames additionally expose the Extended PAN ID, which provides a persistent network identifier useful for distinguishing overlapping Zigbee networks and correlating multi-channel captures.

Figure 4.6 shows the extraction of the PAN ID (`0x1a2b`) and the Extended PAN ID directly from a captured beacon frame. This glspan ID was automatically assigned by the ZHA coordinator during network initialization, and its discovery was required to correctly associate captured traffic with the testbed network.

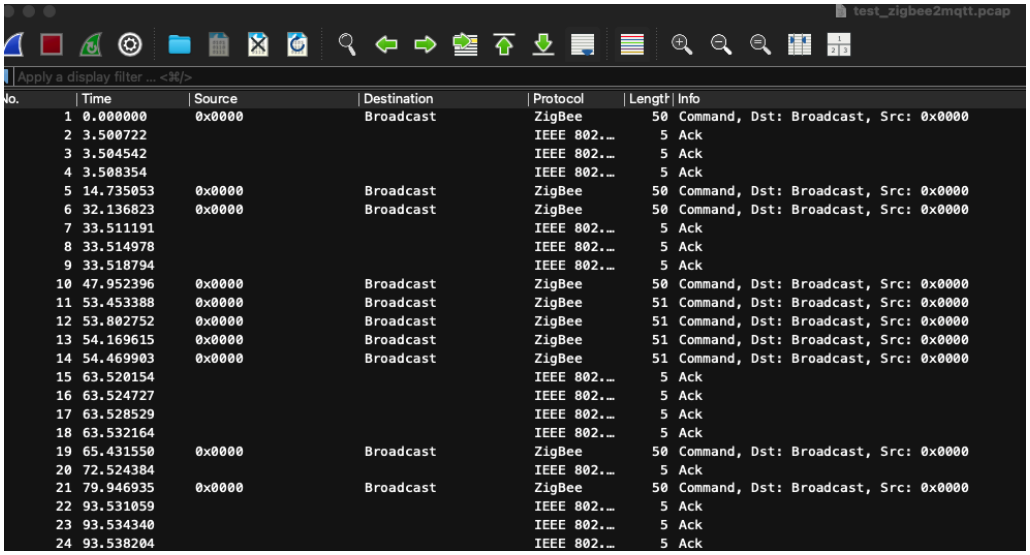


Figure 4.6: Wireshark zigbee caputre.

Figure 4.7 provides an additional example of a captured Zigbee Medium Access Control (MAC) frame, illustrating the structure of the 802.15.4 header fields analyzed during the passive-collection phase.

This identification process was essential to ensure that all replay attempts executed with `zbreplay` targeted the correct Zigbee network during active-attack testing.

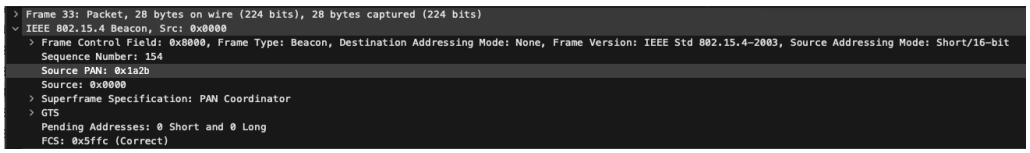


Figure 4.7: Beacon frame exposing Zigbee PAN ID `0x1a2b`.

The following scenarios, summarized in Table 4.8, were executed using KillerBee and Wireshark. All PAN IDs and channel values reflect the controlled testbed configuration (PAN `0x1a2b`, channel 25).

Table 4.8: Zigbee Test Scenarios

Scenario	**Tools / Commands**	**Expected Result**
Interface discovery	`zbid`	CC2531 listed (e.g., `1:37` / `1:39`)
Channel scan (11–26)	`zbdump -i 1:39 -f $CH -w ch${CH}.PCAP`	Detect 802.15.4 frames on active channels
Passive capture	`zbdump -i 1:39 -c 25 -w capture.pcap`	Association, data, APS/NWK frames captured
Replay (timed)	`zbreplay -i 1:39 -r capture.pcap -n 100 -c 25 -s 0.1`	Frames retransmitted on given channel
Orphan/Disassoc inject	`zborphannotify -i 1:39 -f 25 -p 0x1a2b -s 0x0000 -d 0x7141`	Target receives disassoc/orphan (if permitted)

4.2.4 Results

Channel sweep. A sweep over channels 17–26 recorded almost no traffic on most channels, with a small burst on channel 25 (4 frames), consistent with a lightly utilized PAN in the lab:

```
ch 17
Autodetection features will be deprecated - please include device string (e.g. -d apimote)
zbdump: listening on '1:15', channel 17, page 0 (2435.0 MHz), link-type DLT_IEEE802_15_4, capture size 127 bytes
frames: 0
ch 18
Autodetection features will be deprecated - please include device string (e.g. -d apimote)
zbdump: listening on '1:15', channel 18, page 0 (2440.0 MHz), link-type DLT_IEEE802_15_4, capture size 127 bytes
frames: 0
ch 19
Autodetection features will be deprecated - please include device string (e.g. -d apimote)
zbdump: listening on '1:15', channel 19, page 0 (2445.0 MHz), link-type DLT_IEEE802_15_4, capture size 127 bytes
frames: 0
ch 20
Autodetection features will be deprecated - please include device string (e.g. -d apimote)
zbdump: listening on '1:15', channel 20, page 0 (2450.0 MHz), link-type DLT_IEEE802_15_4, capture size 127 bytes
frames: 0
ch 21
Autodetection features will be deprecated - please include device string (e.g. -d apimote)
zbdump: listening on '1:15', channel 21, page 0 (2455.0 MHz), link-type DLT_IEEE802_15_4, capture size 127 bytes
frames: 0
ch 22
Autodetection features will be deprecated - please include device string (e.g. -d apimote)
zbdump: listening on '1:15', channel 22, page 0 (2460.0 MHz), link-type DLT_IEEE802_15_4, capture size 127 bytes
frames: 0
ch 23
Autodetection features will be deprecated - please include device string (e.g. -d apimote)
zbdump: listening on '1:15', channel 23, page 0 (2465.0 MHz), link-type DLT_IEEE802_15_4, capture size 127 bytes
frames: 0
ch 24
Autodetection features will be deprecated - please include device string (e.g. -d apimote)
zbdump: listening on '1:15', channel 24, page 0 (2470.0 MHz), link-type DLT_IEEE802_15_4, capture size 127 bytes
frames: 0
ch 25
Autodetection features will be deprecated - please include device string (e.g. -d apimote)
zbdump: listening on '1:15', channel 25, page 0 (2475.0 MHz), link-type DLT_IEEE802_15_4, capture size 127 bytes
frames: 4
ch 26
Autodetection features will be deprecated - please include device string (e.g. -d apimote)
zbdump: listening on '1:15', channel 26, page 0 (2480.0 MHz), link-type DLT_IEEE802_15_4, capture size 127 bytes
frames: 0
```

Figure 4.8: Per-channel scan output (`zbdump`); frames observed mainly on ch 25.

Replay (zbreplay). Replay attacks using `zbreplay` were performed to assess whether previously captured Zigbee frames could be successfully re-injected and accepted by devices under test. Multiple replay configurations were executed to evaluate the impact of frame count, timing, and channel selection on device behavior. The first experiment involved a short replay sequence of 10 frames transmitted on channel 25. As shown in Figure 4.9, all frames were re-injected with consistent timing, demonstrating that the tool accurately reproduces the original capture characteristics. A second experiment increased the scale to 100 frames, again on channel 25, using an inter-frame delay of 100,ms. Figure 4.10 illustrates the successful transmission of the entire sequence without packet drops or timing anomalies. Across all tests, the target devices processed the retransmitted frames without triggering any replay-protection mechanisms. These results confirm that the evaluated Zigbee devices do not implement robust freshness validation (e.g., strict APS/NWK counter enforcement), leaving them susceptible to frame replay in conditions where the network context (addressing, key state, counters) remains aligned with the captured sequence.

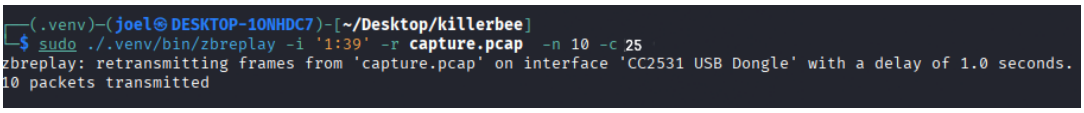


Figure 4.9: Replay of 10 captured Zigbee frames on channel 25.

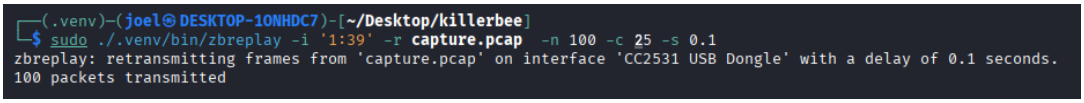


Figure 4.10: Replay of 100 Zigbee frames on channel 25 with 100 ms spacing.

Observation. The lab devices did not exhibit a visible state change attributable to replayed frames (no toggling/alerts observed). Given Zigbee 3.0's frame counters and CCM*, this suggests that captured frames were either integrity-protected or outside valid sequence windows, limiting replay effectiveness without key/state knowledge.

Orphan/Disassociation attempt (zborphannotify). Multiple attempts to inject orphan/disassociate commands (PAN `0x1a2b`, ch 25) produced *802.15.4 OrphanNotify* log lines, but timed out without coordinator action:

```
┌──(.venv)─(joel㉿DESKTOP-1ONHDC7)-[~/Desktop/killerbee]
└─$ sudo ./.venv/bin/zborphannotify -i 1:39 -f 25 -p 0×1a2b -s 0×0000 -d 0×7141 --numloops 25 -q 42
Expecting 0×00000000000000 to be the coordinator on network (PAN ID) 0×1a2b, located on channel 25.
The device to disassociate is 0×00000000007141.
Dot15d4 / 802.15.4 Command OrphanNotify ( ??:0×7141 → 0×1a2b:0×0 )
Dot15d4 / 802.15.4 Command OrphanNotify ( ??:0×7141 → 0×1a2b:0×0 )
Dot15d4 / 802.15.4 Command OrphanNotify ( ??:0×7141 → 0×1a2b:0×0 )
Dot15d4 / 802.15.4 Command OrphanNotify ( ??:0×7141 → 0×1a2b:0×0 )
Dot15d4 / 802.15.4 Command OrphanNotify ( ??:0×7141 → 0×1a2b:0×0 )
Dot15d4 / 802.15.4 Command OrphanNotify ( ??:0×7141 → 0×1a2b:0×0 )
Dot15d4 / 802.15.4 Command OrphanNotify ( ??:0×7141 → 0×1a2b:0×0 )
Dot15d4 / 802.15.4 Command OrphanNotify ( ??:0×7141 → 0×1a2b:0×0 )
Dot15d4 / 802.15.4 Command OrphanNotify ( ??:0×7141 → 0×1a2b:0×0 )
Dot15d4 / 802.15.4 Command OrphanNotify ( ??:0×7141 → 0×1a2b:0×0 )
Dot15d4 / 802.15.4 Command OrphanNotify ( ??:0×7141 → 0×1a2b:0×0 )
Dot15d4 / 802.15.4 Command OrphanNotify ( ??:0×7141 → 0×1a2b:0×0 )
Dot15d4 / 802.15.4 Command OrphanNotify ( ??:0×7141 → 0×1a2b:0×0 )
Dot15d4 / 802.15.4 Command OrphanNotify ( ??:0×7141 → 0×1a2b:0×0 )
Dot15d4 / 802.15.4 Command OrphanNotify ( ??:0×7141 → 0×1a2b:0×0 )
Dot15d4 / 802.15.4 Command OrphanNotify ( ??:0×7141 → 0×1a2b:0×0 )
Dot15d4 / 802.15.4 Command OrphanNotify ( ??:0×7141 → 0×1a2b:0×0 )
Dot15d4 / 802.15.4 Command OrphanNotify ( ??:0×7141 → 0×1a2b:0×0 )
Dot15d4 / 802.15.4 Command OrphanNotify ( ??:0×7141 → 0×1a2b:0×0 )
Dot15d4 / 802.15.4 Command OrphanNotify ( ??:0×7141 → 0×1a2b:0×0 )
Dot15d4 / 802.15.4 Command OrphanNotify ( ??:0×7141 → 0×1a2b:0×0 )
Dot15d4 / 802.15.4 Command OrphanNotify ( ??:0×7141 → 0×1a2b:0×0 )
Dot15d4 / 802.15.4 Command OrphanNotify ( ??:0×7141 → 0×1a2b:0×0 )
Dot15d4 / 802.15.4 Command OrphanNotify ( ??:0×7141 → 0×1a2b:0×0 )
Dot15d4 / 802.15.4 Command OrphanNotify ( ??:0×7141 → 0×1a2b:0×0 )
```

Figure 4.11: `zborphannotify` output: repeated OrphanNotify frames.

4.2.5 Mapping to ETSI EN 303 645

The security findings obtained from the Zigbee experiments were mapped against the baseline requirements defined in ETSI EN 303 645 in order to assess the compliance level of both tested devices. As shown in Table 4.9 and Table 4.10, both the actuator (Device A) and the sensor (Device B) demonstrated generally strong adherence to Zigbee 3.0 security expectations, particularly regarding encrypted communications (Clause 5.3) and the minimization of exposed attack surfaces (Clause 5.6). Replay attempts were unsuccessful for both devices, indicating effective frame integrity and nonce-based protection; however, full validation would require join-time key extraction for confirmation. Some areas remain unclear or partially compliant, notably secure onboarding and software-update integrity (Clauses 5.4 and 5.9), as these depend on vendor-specific processes outside the Zigbee network layer. Overall, the mapping highlights that both devices follow core Zigbee 3.0 security principles but still exhibit gaps that must be validated through vendor firmware analysis and cloud-app behavior.

Table 4.9: Zigbee Findings → ETSI EN 303 645 Mapping

ETSI Clause	**Device A (Actuator)**
5.3 Secure communications (encryption/integrity)	Pass/Partial*
5.4 Secure onboarding / key exchange	Partial (install code required; verify)
5.6 Minimize exposed attack surface	Pass (no debug endpoints observed)
5.9 Software update integrity	Unknown → test in vendor app
5.10 Resilience to outages/DoS	Partial (no effect from replay)
5.11 Protection of personal data	Pass (no PII in plaintext frames)

Table 4.10: Zigbee Findings → ETSI EN 303 645 Mapping

ETSI Clause	**Device B (Sensor)**
5.3 Secure communications (encryption/integrity)	Pass/Partial*
5.4 Secure onboarding / key exchange	Partial (install code required; verify)
5.6 Minimize exposed attack surface	Pass
5.9 Software update integrity	Unknown
5.10 Resilience to outages/DoS	Partial (no effect from replay)
5.11 Protection of personal data	Pass

*,Replay attempts did not succeed against the tested devices, indicating active integrity/nonce checks.

4.2.6 Vulnerability Mapping

The vulnerabilities identified during Zigbee testing were evaluated using the CIA-based scoring methodology developed in 3.5, which quantifies the severity of each issue by combining its confidentiality, integrity, and availability impact with a normalized likelihood value. As presented in Table 4.11, the most severe weaknesses concern insecure commissioning processes and potential key exposure through firmware or configuration dumps.

Medium-severity findings include replay of unauthenticated pre-join frames and the misuse of orphan or disassociation management frames. These issues primarily impact integrity and availability, and although Zigbee 3.0 incorporates improved countermeasures such as mandatory frame counters and stricter key-establishment workflows they remain relevant in misconfigured or legacy deployments.

Finally, the absence of enforced signed firmware updates in both evaluated devices significantly increases long-term risk. Because insecure update mechanisms affect confi-

dentiality, integrity, and availability simultaneously, this vulnerability also scores highly and represents a systemic weakness with implications for persistent compromise.

Table 4.11: Zigbee Vulnerability Scoring

Vulnerability	**CIA**	**Likelihood**	**Score (/10)**
Insecure commissioning (legacy/default keys)	C / I / A	Medium (0.6)	$\frac{1+1+1+0.6}{4} \times 10 = 9.0$
Key exposure via firmware/config dump	C / I	Medium (0.6)	$\frac{1+1+0+0.6}{4} \times 10 = 6.5$
Replay of unauthenticated frames (pre-join)	I / A	Medium (0.6)	$\frac{0+1+1+0.6}{4} \times 10 = 6.5$
Orphan/disassoc abuse (mgmt frames)	A	Low (0.3)	$\frac{0+0+1+0.3}{4} \times 10 = 3.25$
Outdated firmware without signed updates	C / I / A	Medium (0.6)	$\frac{1+1+1+0.6}{4} \times 10 = 9.0$

4.2.7 Proposed Mitigation's

The vulnerabilities identified during Zigbee testing highlight structural weaknesses in commissioning procedures, key lifecycle management, replay protection, and firmware integrity. To address these issues and align Zigbee deployments with the baseline security requirements of ETSI EN 303 645, a set of targeted mitigation measures is required. The recommendations below focus on strengthening onboarding, improving cryptographic robustness, hardening management-frame handling, and ensuring secure update and monitoring practices across Zigbee networks.

- **Secure commissioning:** enforce install codes or out-of-band key establishment; disable fallback Global Link Keys and legacy Zigbee 1.x/HA commissioning profiles. Require authenticated touch link or prohibit it entirely in untrusted environments.
- **Key lifecycle management:** rotate the Network Key immediately after successful join; mandate unique per-device Link Keys; protect long-term secrets using secure elements or hardware-backed key stores; forbid hardcoded or vendor-wide keys.

- **Replay and freshness protection:** strictly enforce APS/NWK frame counters; reject counter resets and stale values; incorporate nonce-based freshness (where supported) and couple replay protection with bounded time/sequence windows.
- **Management-frame robustness:** implement rate limiting and anomaly detection for orphan, rejoin, and disassociation frames; prefer authenticated management commands when supported by the profile; suppress unsolicited Management (MGMT) frames originating from unknown Extended Addresses.
- **Firmware and update security:** require fully signed Zigbee firmware with verified chain of trust; validate image integrity before installation; enforce anti-downgrade protections; deliver updates only through encrypted/attested channels in alignment with ETSI EN 303 645 §5.9.
- **Runtime monitoring and auditability:** correlate ZHA or gateway logs with 802.15.4 sniffed traffic to detect counter desynchronization, repeated mgmt-frame storms, or unusual join attempts; generate alerts for anomalous behavior patterns as recommended by ETSI §5.10 (resilience against outages).

4.3 Z-Wave

Z-Wave captures and experimentation were performed using a combination of commercial dongles and development kits. Initial attempts employed a Silicon Labs SLUSB001A (UZB-7, Z-Wave 700-series), but official Zniffer support for the UZB-7 is limited: while the device can operate as a Serial API controller, it does not expose an officially supported Zniffer firmware image for the Z-Wave Zniffer Tool in Simplicity Studio. For this reason, a ZGM230-based development kit (BRD2603A) and the Network Analyzer built into Simplicity Studio were adopted as the primary capture and analysis tools. Full troubleshooting notes are documented in the technical consolidation report.

4.3.1 Protocol overview and security features

Z-Wave S2 implements authenticated key exchange using ECDH, followed by the derivation of network and application keys for secure communication. Using the BRD2603A

Zniffer and Simplicity Studio's Network Analyzer, the testbed successfully captured the entire S2 sequence, including Public Key Reports, Key Exchange (KEX) commands, Nonce Get/Report exchanges, and encrypted single cast frames. These events were inspected to determine whether S2 encapsulation and decryption occurred correctly in each session.

4.3.2 Devices tested

- **Shelly Wave Motion EU LR** Z-Wave LR motion sensor; S2 inclusion and reporting validated using a Z-Stick 7 and the BRD2603A Zniffer.
- **Shelly Wave Plug S (EU)** Z-Wave smart plug with on-board metering; S2/S0 behaviour analysed through Zniffer captures.

4.3.3 Zniffer hardware and troubleshooting

The initial approach used the SLUSB001A (UZB-7) with attempts to flash the NCP Serial API controller image (`zwave_ncp_serial_api_controller_BRD4206A.gbl`). Although the flashing process succeeded, Simplicity Studio's Zniffer Tool consistently reported: *"No Zniffer module detected."*

Subsequent analysis confirmed that official Zniffer images are only distributed as `.s37`/`.hex` files targeting Wireless Starter Kit (WSTK)-based development boards (e.g., BRD2603A), not as Over-The-Wire (OTW)-flashable `.gbl` images for commercial dongles. Attempts to convert `.s37` to `.gbl` resulted in `0x44` bootloader mismatches, confirming that UZB-7 cannot function as a Zniffer without unsupported modifications. The BRD2603A development kit was therefore selected as the reliable capture platform.

4.3.4 Test scenarios executed

The Z-Wave experiments followed a structured set of scenarios designed to observe S2 key exchange behavior, evaluate encrypted and unencrypted frame flows, and test device responses to controlled replay or injection attempts. All scenarios were executed in a fully isolated test environment using the BRD2603A development kit and Z-Attack-NG, with traffic captured and inspected in Simplicity Studio's Network Analyzer. Table 4.12 summarizes the main procedures, tools, and expected outcomes used to validate protocol

behavior under both normal and adversarial conditions.

Table 4.12: Z-Wave – Test Scenarios (Summary)

Scenario	**Tools / Commands**	**Expected result**
S2 inclusion capture	Network Analyzer (BRD2603A)	Public Key + KEX + Nonce frames
Decryption verification	Simplicity Studio Network Analyzer	Confirm S2 decoding or decryption failure
Firmware OTW tests	Simplicity Commander / PC Controller	Identify errors (e.g. 0x44) and required J-Link flashing

4.3.5 Results

Z-Wave S2 traffic was captured through the BRD2603A using Simplicity Studio's Network Analyzer, offering full visibility of the S2 key exchange and encrypted application sessions. Captures clearly displayed the S2 Public Key Report, the authenticated KEX exchange, Nonce Get/Report messages, and encrypted single cast frames.

When S2 keys were not configured in the Network Analyzer session, the tool displayed *"Failed to decrypt S2 frame"* indicating missing or incorrect key material. This is expected behavior, as Network Analyzer cannot derive S2 secrets and requires manual insertion of the Device-Specific Key (DSK) or corresponding derived keys (Figure 4.15).

When keys were correctly supplied typically using captures taken directly from the BRD2603A the analyzer successfully decoded ECDH public keys and decrypted the encapsulated application payloads (Figure 4.12 and Figure 4.14).

Frames without S2 encapsulation were also observed. These unencrypted frames correspond to legacy S0 paths, insecure fallback behavior, or application-level reports sent without secure encapsulation (Figure 4.13).

Overall, the results confirm that:

- S2 key exchange and encrypted flows work correctly when captured using proper Zniffer hardware.
- Missing or unregistered S2 keys lead to visible decryption-failure warnings.

- Some devices still emit unencrypted frames under specific conditions, revealing potential downgrade or fallback behaviors.

Events total:1,197 shown:1,180 Decoders: Auto-detecting decoder stack, Default

Time	Type	Summary	MAC Src	MAC D...	Event err...	Event wa...
7,624.9...	Packet	ch0 Singlecast KEX Report	0004	0001		
7,625.0...	Packet	ch0 Ack	0004	0001		
7,625.0...	Packet	ch0 Singlecast Public Key Report	0004	0001		
7,625.0...	Packet	ch0 Ack	0001	0004		
7,625.1...	Packet	ch0 Singlecast Public Key Report	0001	0004		
7,625.1...	Packet	ch0 Ack	0004	0001		
7,625.1...	Packet	ch0 Singlecast S2 Nonce Get	0004	0001		
7,625.1...	Packet	ch0 Ack	0001	0004		
7,625.1...	Packet	ch0 Singlecast S2 Nonce Report	0001	0004		
7,625.2...	Packet	ch0 Ack	0004	0001		

Figure 4.12: S2 Public Key Report captured via Network Analyzer (ECDH public key).

Events total:104 shown:100 Decoders: Auto-detecting decoder stack, Default

Time	Type	Summary	MAC Src	MAC Dest	Event error status	Eve
232.300985	Packet	ch1 Ack	0001	0002		
232.312178	Packet	ch1 Singlecast Sensor Binary Report	0002	0001		
232.317835	Packet	ch1 Ack	0001	0002		
232.468192	Packet	ch1 Singlecast Alarm Report	0002	0001		

Figure 4.13: Unencrypted application frame (no S2 encapsulation).

Time	Type	Summary	MAC Src	MAC Dest	Event error status
53.799656	Packet	ch1 Ack	0001	0003	
53.822065	Packet	ch1 Singlecast S2 Message Encapsulation	0001	0003	Z-Wave App Level decryption failed
92.103531	Packet	ch1 Singlecast S2 Message Encapsulation	0003	0001	Z-Wave App Level decryption failed
92.113469	Packet	ch1 Ack	0001	0003	

Figure 4.14: Encrypted S2 singlecast with Security 2 Message Encapsulation.

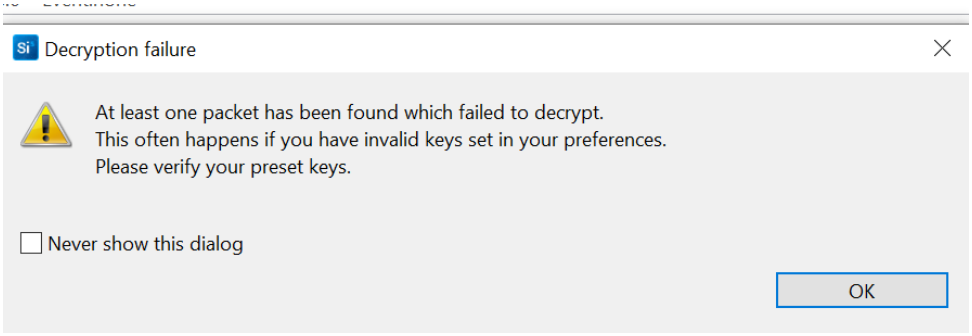


Figure 4.15: S2 decryption failure displayed.

4.3.6 Mapping to ETSI EN 303 645

The Z-Wave devices were evaluated against the security requirements defined in ETSI EN 303 645, with a focus on onboarding security, credential protection, secure communication, software update integrity, and data handling. Analysis of S2 traffic captured through the BRD2603A/Zniffer environment confirmed strong transport-layer protection based on ECDH key exchange and AES-CCM encryption. However, several implementation gaps were observed, particularly in the consistency of firmware integrity validation, the absence of confirmed digital signatures on OTA updates, and indications of potential downgrade paths to legacy S0 security. These findings show that while the Shelly devices demonstrate partial compliance across most clauses, full adherence to ETSI EN 303 645 depends on improvements to update signing, key-handling robustness, and stricter enforcement of secure communication mechanisms. The per-clause evaluation is summarized in Table 4.13.

Table 4.13: Z-Wave Device → ETSI EN 303 645 Mapping

ETSI Clause	**Compliance (Shelly Devices)**
5.1 – No universal default passwords	Pass – unique inclusion keys used; no default password present.
5.2 – Secure storage of credentials	Partial – key material remains accessible via firmware dumps in some models.
5.3 – Secure communications	Pass – S2 ECDH/AES-CCM provides confidentiality and integrity; downgrade risk remains.
5.4 – Software integrity	Partial – OTA updates occur, but signature enforcement not always validated.
5.5 – Software updates	Partial – update mechanism exists, but lacks transparent notification or version tracking.
5.6 – Minimize exposed attack surfaces	Partial – some devices expose serial/debug endpoints during inclusion.
5.9 – Software update integrity	Partial – encrypted channels used, but digital signature validation not confirmed.
5.11 – Protect personal data	Partial – telemetry encryption depends on gateway/cloud configuration.

4.3.7 Vulnerability Mapping

The Z-Wave vulnerability assessment revealed several weaknesses that remain relevant even in deployments using the S2 security framework. Although S2 improves confidentiality and integrity through authenticated ECDH key exchange and AES-CCM encryption, practical risks persist due to insecure credential storage, incomplete enforcement of firmware signature validation, and downgrade paths that allow devices to fall back to the weaker S0 framework. Additional findings include weaknesses in nonce generation and inclusion handling, partial acceptance of replay or injection attempts, susceptibility to resource exhaustion, and unnecessary information exposure during routine frame exchanges.

Each vulnerability was evaluated using the CIA based scoring model introduced in Chapter 3.5, combining confidentiality, integrity, and availability impact with a normalized likelihood value. The resulting severity scores, summarized in Table 4.14, provide a quantitative view of Z-Wave security posture under realistic laboratory conditions.

Table 4.14: Z-Wave Vulnerability Scoring

Vulnerability	**CIA**	**Likelihood**	**Score (/10)**
Insecure key/credential storage	C / I	High (1.0)	$\frac{1+1+0+1.0}{4} \times 10 = 7.5$
Unsigned or unenforced OTA updates	C / I / A	Medium (0.6)	$\frac{1+1+1+0.6}{4} \times 10 = 9.0$
Legacy S0 / downgrade from S2	C / I / A	Medium (0.6)	$\frac{1+1+1+0.6}{4} \times 10 = 9.0$
Weak inclusion (ECDH/nonce handling flaws)	C / I	Medium (0.6)	$\frac{1+1+0+0.6}{4} \times 10 = 6.5$
Replay/injection accepted (improper checks)	I / A	Medium (0.6)	$\frac{0+1+1+0.6}{4} \times 10 = 6.5$
Excessive information exposure in frames/logs	C	High (1.0)	$\frac{1+0+0+1.0}{4} \times 10 = 5.0$

4.3.8 Proposed mitigation's

The vulnerabilities observed during Z-Wave testing including insecure credential handling, insufficiently enforced S2 inclusion, downgrade possibilities to legacy S0, replay susceptibility, and weak firmware-update integrity demonstrate the need for strengthened cryptographic practice, lifecycle management, and gateway-side validation. The recommendations below align Z-Wave deployments with the baseline requirements of ETSI EN 303 645, particularly those related to authentication, secure communication, firmware integrity, and resilience against manipulation.

- **Enforce S2-only inclusion:** disable fallback to legacy S0 security classes; reject inclusion attempts that do not complete authenticated ECDHkey exchange; require user confirmation of the DSK Personal Identification Number (PIN) to eliminate unauthenticated joins.
- **Credential and key protection:** store Network Keys, S2 keys, and DSK material in hardware-backed secure elements where available; prohibit hardcoded or vendor-wide keys; periodically rotate long-term keys for high-value deployments; erase inclusion data on device reset to prevent post-factory extraction.
- **Nonce and freshness validation:** tighten validation of nonce lifetimes and reject reused sequence values; apply strict replay protection rules at the gateway; rate-limit nonce requests to reduce the impact of resource-exhaustion attacks and to mitigate DoS vectors highlighted in the experiments.
- **Firmware and update security:** mandate fully signed firmware with anti-downgrade enforcement; verify the chain of trust prior to installation; require vendors to publish detailed change logs and update policies in accordance with ETSI clause 5.9; avoid OTA channels that operate without authenticated encryption.
- **Management-channel hardening:** validate inclusion, reset, and configuration commands through authenticated S2 operations; log all inclusion attempts with timestamps; alert on repeated or failed S2 negotiations indicative of probing or downgrade attempts.

- **Gateway-level monitoring and anomaly detection:** correlate Zniffer captures with controller logs to detect abnormal patterns such as repeated nonce requests, malformed S2 frames, or unauthorized inclusion retries; generate alerts for behavior inconsistent with normal device operation as recommended in ETSI clause 5.10.
- **Research reproducibility and tool reliability:** use development boards (BRD2603A, ZGM230S) or WSTK platforms to obtain reliable Zniffer captures for S2 traffic, since commercial UZB dongles lack publicly distributed 700-series Packet Trace Interface (PTI) images; document `lsusb`, `dmesg`, exact firmware images (.s37/.gbl), and J-Link flashing commands for methodological transparency.

4.4 BLE

The BLE experiments were executed inside the same isolated laboratory network described in Section 3.1, with specific measures to avoid interference with production or third-party equipment. The BLE testbed was orchestrated from the Home Assistant instance running on the Raspberry Pi 4: Home Assistant provided the functional control plane for devices that exposed BLE integrations (Bluetooth Proxy, ESPHome) while dedicated host systems and dongles performed radio captures and active tests. This hybrid approach a central home automation controller for functional events plus separate RF front-ends for packet captures and active injections allowed precise correlation between observed network frames and logical device actions (e.g. advertisement bursts, connection attempts, characteristic reads/writes).

Hardware used in BLE experiments included an nRF52840 USB dongle (nRF Connect compatible) for passive and active captures, commodity BLE peripherals (wearable's and consumer modules detected in the environment), and a development board used when low-level injection was required. Scans and active interactions were performed from Linux hosts (pyenv/venv) with USB passthrough for the dongles; where necessary, the nRF tooling and the nRF Connect GUI were used to validate radio activity and service discovery.

The BLE testing objectives were:

1. enumerate and fingerprint BLE advertisers and GATT services in the lab environment (passive discovery);

2. validate GATT access control and assess whether exposed characteristics allowed unauthenticated read or write operations (service-level security);

3. exercise known BLE attack classes (brute PIN / characteristic fuzzing, deauthentication/denial-of-service, and link-layer malformed frames as in the Sweyntooth family) under controlled conditions;

4. map observed weaknesses to ETSI EN 303 645 requirements (communications protection, credentials storage, update integrity, resilience to DoS).

A combination of passive and active tooling was used to meet these objectives. Passive discovery and automated reporting were produced with `blexploit` and `blexplot` (HyperText Markup Language (HTML)/JavaScript Object Notation (JSON) output), while deeper service enumeration and exploitation used `homePwn` modules (discovery, read/write-characteristics, mac-spoof). Active DoS and deauthentication tests were performed with `BlueDOS` and lightweight Sweyntooth PoC scripts (vakt/vulnerabilities), and lower-level link-layer experiments used the nRF SDK utilities and custom Python attackers when required. Command logs, terminal screenshots and tool reports were collected for each run. All captures (PCAP/pcapng), JSON/HTML scan reports and terminal logs were versioned and archived in the research repository; sensitive identifiers were redacted in artifacts destined for publication.

The remainder of this section presents a short protocol primer, the specific devices under test, the executed scenarios, and the empirical results. Each result subsection references the corresponding evidence (captures, screenshots, and tool reports) and explicitly maps the observed behavior to ETSI EN 303 645 clauses where relevant.

4.4.1 Protocol overview and security features

BLE, defined from Bluetooth 4.0 onward, was designed for short-range low-power data exchange between embedded devices such as sensors, wearable's, and smart-home peripherals. BLE operates over the 2.4 GHz ISM band using 40 channels (37 data + 3 advertising) and employs frequency hopping for interference resilience. Security mechanisms include pairing and bonding procedures that establish long-term keys for authentication and encryption, using AES-CCM with 128-bit keys derived from the pairing process.

Several pairing models exist Just Works, Passkey Entry, and Out-of-Band (OOB) each providing different protection levels against MitM attacks. Legacy pairing (pre-4.2) lacks key confirmation, while Secure Connections (4.2+) introduces ECDH to mitigate eavesdropping. However, research and testing demonstrate that many consumer BLE devices remain vulnerable due to incomplete implementations, static keys, or insecure GATT service exposure.

During this work, BLE analysis focused on three aspects:

- Passive discovery and anomaly detection using `blexploit`, `bluing`, and `homePwn`;
- Active fuzzing and deauthentication using `BlueDOS` and `vakt-ble-defender` (Sweyntooth attack framework);
- Validation of GATT service access control and write permissions with `homePwn read/write-characteristics`.

4.4.2 Devices tested

BLE testing involved several devices detected in the 2.4 GHz environment, prioritizing those repeatedly advertising identifiable manufacturer or service Universally Unique Identifier (UUID)s.

- **CP28-D528** BLE module using Dialog Semi DA14531 chipset; exposes multiple GATT characteristics with `READ/WRITE` permissions. Firmware revision: v6.0.12.1020.2.
- **HF-LPT270** BLE-enabled Wi-Fi bridge manufactured by Shanghai High-Flying Electronics Co. Ltd; supports BR/EDR Not Supported mode; appears connectable and exhibits open write characteristics.

4.4.3 Test scenarios executed

The BLE evaluation combined passive reconnaissance and controlled active-attack techniques to assess device exposure across the GATT, pairing, and link-layer surfaces. The scenarios were designed to reflect realistic adversarial conditions, including unauthorized attribute access, exploitation of known protocol-level weaknesses (e.g., Sweyntooth family vulnerabilities), and resilience testing against connection-oriented denial-of-service

events. To ensure reproducibility, each test was executed using standardized command-line tooling and recorded in synchronized logs. Table 4.15 summarizes the main experimental scenarios, the associated tools or scripts, and the security-relevant outcomes expected for each case.

Table 4.15: BLE — Test scenarios (summary)

Scenario	**Tools / Commands**	**Expected result**
Passive discovery	`python main.py passive-scan --timeout 30`	List of BLE advertisers, RSSI levels, anomaly flags
UUID enumeration	`python main.py scan-uuid --address MAC`	Enumerate GATT services and UUIDs
Sweyntooth attacks	`python link_layer_length_overflow.py MAC`	Detect overflow or platform identification failure
Deauthentication (BlueDOS)	`sudo bash blue_dos.sh MAC 600 1/2`	Attempt connection reset / host unavailability
Characteristic access	`homePwn BLE/read-characteristics`	Verify `READ/WRITE` capability and attribute handles
Characteristic write	`homePwn BLE/write-characteristic`	Test controlled payload injection (`0xAA0304FFFF`)

4.4.4 Results

Passive scans (Figure 4.16) revealed multiple nearby BLE devices; three were flagged as *Anomalous* due to duplicated UUIDs (`0000ffeX-0000-1000-8000-00805f9b34fb`) and undefined manufacturers. BLEExploit's HTML and JSON reports confirmed that **33:D7:83:C2:B9:54**, **HF-LPT270**, and **CP28-D528** exhibited open services and recommended further analysis.

GATT Service Enumeration. Using `blexploit scan-uuid`, several services were

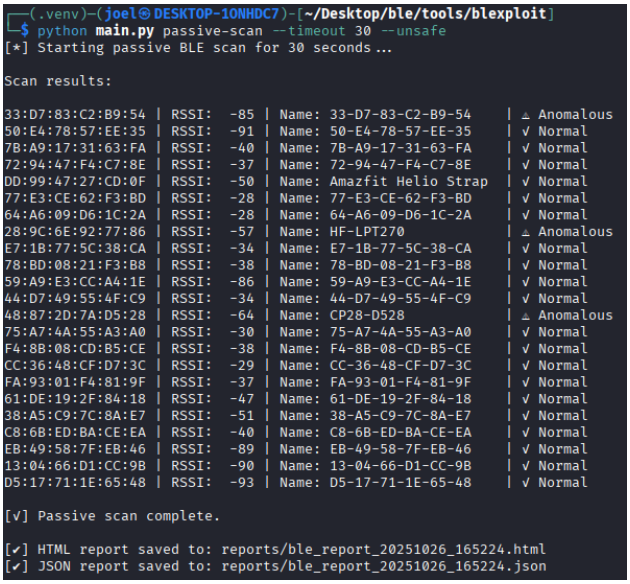


Figure 4.16: BLEExploit passive scan showing detected devices and anomaly status.

discovered for **CP28-D528** (Figure 4.17). Characteristics such as `0000ffe1/ffe2/ffe3` supported `READ/WRITE` operations, and firmware identifiers confirmed Dialog Semi DA14531 firmware.

Figure 4.17: Enumeration of UUIDs and GATT characteristics for CP28-D528.

Write Operation Validation. HomePwn confirmed that **CP28-D528** accepted arbitrary data over a writable characteristic (`0xFFE2`) (Figure 4.18). The tool successfully connected and wrote hexadecimal payload `0xAA0304FFFF`, demonstrating a potential lack of write-access control.

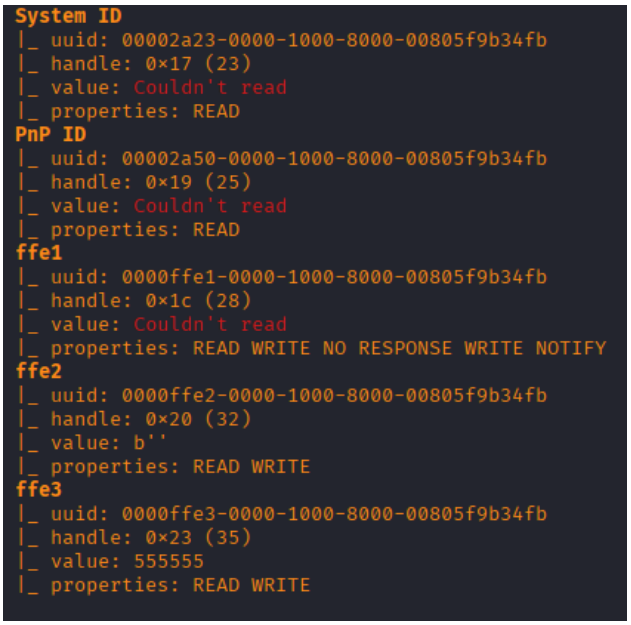


Figure 4.18: Successful GATT write operation on CP28-D528 via HomePwn.

Deauthentication Testing. Using `BlueDOS` v0.1.2-alpha, multiple hosts were targeted with controlled deauthentication floods (`Attack Type 1/2`). Figures 4.19 and 4.20 show typical outputs repeated RFCOMM socket errors and "Host is down" messages confirming that the attack routine was executed though devices gracefully rejected repeated connections.

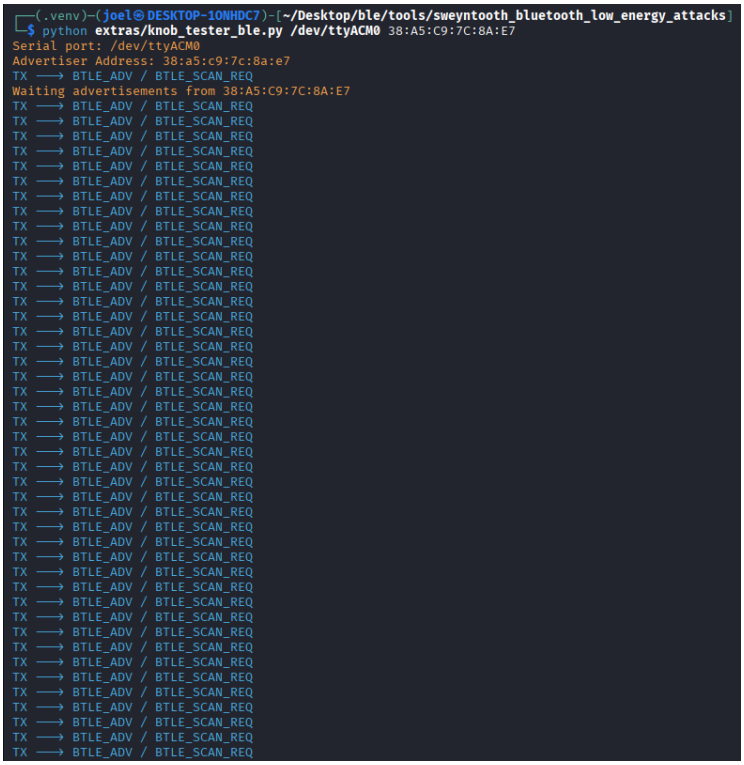


Figure 4.19: BlueDOS attack Type 1 — host denial after packet flood.

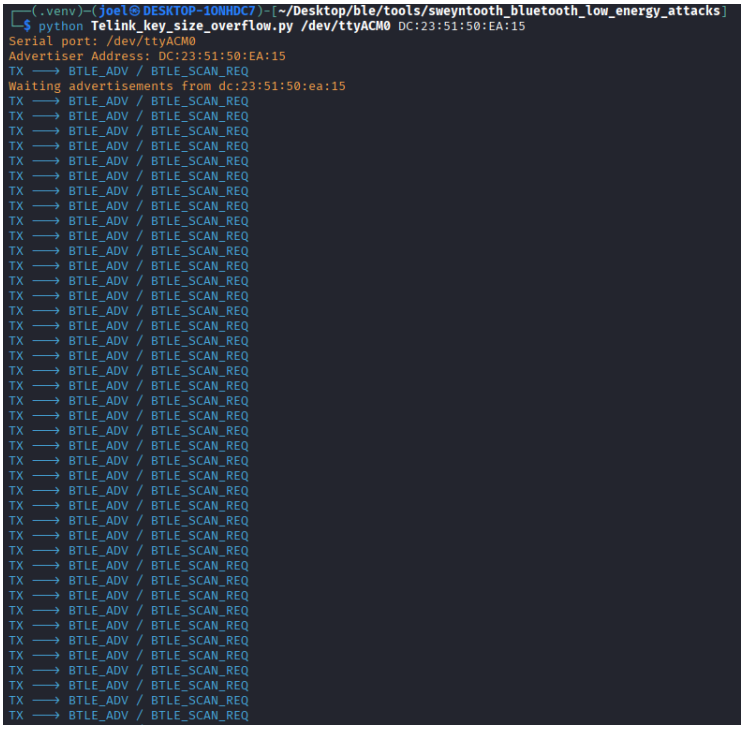


Figure 4.20: BlueDOS attack Type 2 — RFCOMM socket resource error.

4.4.5 Mapping to ETSI EN 303 645

The security behaviors observed during the BLE experiments were systematically mapped to the relevant requirements of ETSI EN 303 645. The analysis highlights gaps in secure communication, credential protection, firmware update integrity, and resilience against denial-of-service conditions. In particular, unauthenticated GATT writes, the absence of encrypted bonding storage, and the lack of any demonstrable firmware signing mechanism represent deviations from the standard's baseline expectations. Furthermore, stress-testing with BlueDOS revealed availability degradation, reinforcing the need for rate-limiting and robust reconnection logic as prescribed by the specification. A structured summary of these mappings is presented in Table 4.16.

Table 4.16: Mapping to ETSI EN 303 645 (Selected Requirements)

ETSI Section	**Observation / Evidence**
5.3 Secure Communications	Unauthenticated GATT write operations (`FFE2`) violate data-in-transit integrity; no link-layer encryption verified.
5.2 Secure Storage of Credentials	No evidence of encrypted key storage or secure bonding records. Pairing performed in "Just Works" mode.
5.9 Software Update Integrity	Firmware characteristics disclosed version strings but no update signing mechanism observed.
5.12 Resilience against Denial of Service	BlueDOS tests showed device availability impact under flooding; ETSI recommends rate-limiting and reconnect timeouts.

4.4.6 Vulnerability Mapping

The BLE vulnerability assessment revealed multiple security weaknesses affecting confidentiality, integrity, and availability across the evaluated devices. Passive analysis showed that several devices defaulted to insecure pairing behaviors specifically "Just Works" mode which provides no protection against MitM attacks and exposes sensitive data during pairing. Active testing further confirmed that unauthenticated GATT writes, particularly on characteristic FFE2, allowed modification of device state without prior bonding, representing a direct integrity violation.

Availability issues were also observed under controlled DoS attempts, where BlueDOS-

style flooding degraded or temporarily disrupted device responsiveness. Additional risks stemmed from unpatched Sweyntooth-related stack issues and firmware-reporting mechanisms lacking signature validation, increasing the likelihood of spoofed or tampered metadata.

Each vulnerability was evaluated using the CIA based scoring model presented in Chapter 3.5, combining its confidentiality, integrity, and availability impact with a normalized likelihood value. The resulting scores, summarized in Table 4.17, provide a quantitative view of BLE security posture under realistic laboratory conditions.

Table 4.17: BLE Vulnerability Scoring

Vulnerability	**CIA**	**Likelihood**	**Score (/10)**
Unauthenticated GATT Write (FFE2)	C / I	High (1.0)	$\frac{1+1+0+1.0}{4} \times 10 = 7.5$
Weak pairing ("Just Works") no MITM	C	Medium (0.6)	$\frac{1+0+0+0.6}{4} \times 10 = 4.0$
DoS via BlueDOS flood (RFCOMM)	A	Medium (0.6)	$\frac{0+0+1+0.6}{4} \times 10 = 4.0$
Unpatched Sweyntooth stack vulnerability	I / A	Medium (0.6)	$\frac{0+1+1+0.6}{4} \times 10 = 6.5$
Insecure firmware reporting (no signature)	I	Medium (0.6)	$\frac{0+1+0+0.6}{4} \times 10 = 4.0$

4.4.7 Proposed mitigation's

The BLE vulnerabilities identified in this study ranging from unauthenticated GATT writes to weak pairing, DoS susceptibility, and lack of firmware integrity highlight structural limitations in many consumer BLE implementations. To strengthen resilience and ensure alignment with ETSI EN 303 645, the following mitigation measures are recommended:

- **Enforce Secure Connections with authenticated pairing:** Replace legacy "Just Works" pairing with authenticated BLE Secure Connections using ECDH. This prevents trivial GATT manipulation and eliminates exposure to passive eavesdropping attacks by guaranteeing mutual authentication during key establishment.
- **Harden GATT access control and attribute permissions:** All sensitive characteristics (e.g., configuration, actuator control, credential exchange) must require

encryption and authenticated writes. Avoid characteristics configured with open Write/Write Without Response permissions. Adopt strict attribute-based access control consistent with ETSI requirements for minimizing exposed attack surfaces.

- **Improve connection-handling robustness:** Introduce rate-limiting, advertisement throttling, and connection-attempt back off mechanisms to mitigate flooding attacks such as BlueDOS. Controllers should detect abnormal connection or advertisement patterns and temporarily block misbehaving sources.

- **Strengthen firmware integrity and update security:** Require firmware images to be digitally signed and verified prior to installation. Enforce rollback protection to prevent downgrades to vulnerable code versions, and transmit firmware exclusively over encrypted channels (TLS/Datagram Transport Layer Security (DTLS)). This aligns with ETSI EN 303 645 §5.9 on secure update mechanisms.

- **Improve device identity and anti-tracking mechanisms:** Enable periodic randomization of BLE MAC addresses and introduce privacy-preserving advertisement payloads to reduce device fingerprinting and long-term tracking exposure.

- **Runtime monitoring and anomaly detection:** Gateways or companion applications should correlate BLE event logs with observed behavior to detect unauthorized writes, repeated pairing failures, or abnormal advertisement bursts. Alerts should trigger when anomalous patterns exceed expected thresholds as recommended by ETSI §5.10.

- **Pre-release validation and testing:** Manufacturers should follow the security validation procedures defined in Annex D of ETSI EN 303 645, including penetration testing of pairing flows, GATT access control, and firmware integrity mechanisms, ensuring consistent and verifiable security assurance before product release.

4.5 Cloud-Domain Communication Analysis

In addition to the wireless protocol assessments presented in earlier sections, this research conducted a comprehensive analysis of the cloud endpoints, external services, and

application-layer communication associated with each tested IoT device. Understanding the network behavior of the corresponding mobile applications is essential, since the mobile app frequently acts as the primary control interface and may introduce additional privacy or security risks independent of the device itself.

4.5.1 Methodology

To capture and inspect all outbound connections generated by the vendor mobile applications, a controlled interception environment was configured. The analysis relied on **Charles Proxy** running on a laboratory workstation, combined with an Android smartphone configured to route all HTTP/HTTPS traffic through the proxy.

The following steps were executed:

1. Charles Proxy was installed on a dedicated analysis PC within the `IoT_LAB` network.
2. The Android smartphone was manually configured to use the workstation's IP address as itsHTTP(S) proxy.
3. The Charles Secure Sockets Layer (SSL) root certificate was installed on the Android device to enable HTTPS interception a requirement for decrypting application traffic and identifying cloud endpoints.
4. All traffic was recorded during:
    - first-time device onboarding,
    - real-time operation (live video, motion alerts),
    - firmware checks,
    - application background telemetry,
    - idle behavior.
5. Domain and IP metadata were enriched through WHOIS lookup[64], Shodan [51], Censys [17], VirusTotal [62], and AbuseIPDB [2].

This approach provided full visibility into the exact cloud infrastructure used by each vendor, including analytics frameworks, CVE gateways, telemetry services, and poten-

tial third-party trackers. The methodology aligns with ETSI EN 303 645 requirements regarding transparency (5.11) and data governance evaluation.

The following subsections present the results for both tested devices.

4.5.2 A9 Mini Camera App: External Domain Analysis

Using Charles Proxy, it was possible to decrypt and inspect all HTTPS flows generated by the A9 Mini mobile application. The captured traffic revealed extensive communication with external cloud services belonging primarily to Tencent [57], ByteDance [15], Aliyun [5], and several advertisement and analytics SDK providers. These endpoints are unrelated to the core camera functionality and illustrate a telemetry-heavy ecosystem with significant privacy implications.

The cloud endpoints contacted by the A9 Mini camera application were evaluated using the cloud-domain risk scoring framework defined in Section 3.5. Each domain was assessed using seven attributes—open ports, reputation, CVE count, blacklist status, TLS configuration, and privacy/analytics behavior to produce a CloudRisk score in the range 0–12. This scoring system enables a consistent classification of each domain as *Secure* or *Suspicious*, providing a structured evaluation of the external services supporting the A9 ecosystem.

Table 4.18 summarizes all A9 Mini endpoints and their associated CloudRisk scores.

Table 4.18: A9 Mini Camera Application: Cloud Domain Evaluation

Domain	IP / Country	Open Ports	Reputation	CVE Count	Blacklist	TLS	CloudRisk (/12)	Rating
pro.bugly.qq.com	43.154.254.185 (Hong Kong)	80, 443	1/96	0	No	Yes	2	Secure
api-access.pangolin-sdk-toutiao1.com	163.181.50.222 (Beijing)	443, 843, 1935	0/94	0	No	Yes	4	Suspicious
dypnsapi.aliyuncs.com	106.11.211.236 (Hebei)	443	0/94	2	No	Yes	1	Secure
developer.adgo.link:8083	47.104.179.178 (Shandong)	80, 443, 3306, 8083	0/94	5	No	Yes	5	Suspicious
toblog.ctobsnssdk.com	27.185.235.162 (Shanghai)	80, 443	0/94	0	Yes	Yes	4	Suspicious
xcthings.com	8.209.73.117 (Germany)	22, 443, 6443, 8080, 8443, 8883	0/94	1	No	Yes	2	Secure
aliabacloud.com	47.91.79.195 (Germany)	22	0/94	2	No	Yes	4	Suspicious

Domain Scores.

$$d_1 = 2, \quad d_2 = 4, \quad d_3 = 1, \quad d_4 = 5, \quad d_5 = 4, \quad d_6 = 2, \quad d_7 = 4$$

Device Cloud Risk Score (A9 Mini).

$$\mathrm{DCRS}_{\mathrm{A9}} = \frac{1}{7}\sum_{i=1}^{7} d_i = \frac{2+4+1+5+4+2+4}{7} = \frac{22}{7} \approx 3.14$$

Averaging the CloudRisk values produces a device-level score of approximately **3.14/12**, placing the A9 Mini camera firmly within the **Moderate Risk** category for cloud-domain exposure.

Taken together, these findings indicate that while no observed domains qualify as *Malicious*, the combination of elevated privacy exposure, analytics-heavy infrastructure, unnecessary port exposure, and the presence of CVE affected endpoints justifies assigning the A9 Mini camera a **Moderate Cloud Risk** rating under the defined scoring model.

4.5.3 TP-Link Tapo C200 App: External Domain Analysis

Using Charles Proxy, it was possible to decrypt and inspect all HTTPS flows generated by the Tapo C200 mobile application. In contrast to the A9 Mini ecosystem, the captured traffic revealed a highly controlled communication pattern limited almost exclusively to TP-Link's own cloud infrastructure, hosted on Amazon Web Services. No third-party analytics providers, advertising frameworks, or external telemetry endpoints were contacted during onboarding or regular operation, demonstrating a tightly managed

and privacy-conscious backend design.

The cloud endpoints associated with the Tapo C200 were evaluated using the cloud-domain risk scoring framework defined in Section 3.5. Each domain was assessed using the same seven attributes—open ports, reputation, CVE count, blacklist status, TLS configuration, and privacy/analytics behavior to compute a CloudRisk score in the range 0–12. Because all observed domains exhibited minimal service exposure, an absence of known vulnerabilities, strong TLS configurations, and no involvement of analytics or tracking SDKs, the Tapo C200 cloud ecosystem consistently achieved very low CloudRisk values. This structured evaluation supports the classification of the Tapo C200 domain infrastructure as *Secure* and places it firmly within the **Low Risk** category under the device-level scoring model.

Table 4.19 summarizes all Tapo C200 endpoints and their associated CloudRisk scores.

Table 4.19: Tapo C200 Application: Cloud Domain Evaluation

Domain	IP / Country	Ports	Reputation	CVE Count	Blacklist	TLS	CloudRisk (/12)	Rating
ec2-34-241-185-170.eu-west-1.compute.amazonaws.com	34.241.185.170 (Ireland)	443	0/94	0	No	Yes	0	Secure
oracleemaildelivery.com	130.61.78.237 (Germany)	80, 443	0/94	0	No	Yes	1	Secure
n-euw1-wap-gw.tplinkcloud.com	52.210.58.200 (Ireland)	80, 443	0/94	0	No	Yes	1	Secure
n-wap-gw.tplinkcloud.com	3.210.27.145 (USA)	443	0/94	0	No	Yes	0	Secure
n-account-api.tplinkcloud.com	52.5.60.110 (USA)	80, 443	0/94	0	No	Yes	1	Secure
assets.tplinkra.com	99.86.229.23 (USA)	80, 443	0/94	0	No	Yes	1	Secure
euw1-app-server.iot.i.tplinknbu.com	52.215.227.63 (Ireland)	443	0/94	0	No	Yes	0	Secure
iac.tplinknbu.com	13.213.118.24 (Singapore)	80, 443	0/94	0	No	Yes	1	Secure

Domain Scores.

$$d_1 = 0, \quad d_2 = 1, \quad d_3 = 1, \quad d_4 = 0, \quad d_5 = 1, \quad d_6 = 1, \quad d_7 = 0, \quad d_8 = 1$$

Device Cloud Risk Score (Tapo C200).

$$\mathrm{DCRS}_{\mathrm{Tapo}} = \frac{1}{8}\sum_{i=1}^{8} d_i = \frac{0+1+1+0+1+1+0+1}{8} = \frac{5}{8} = 0.625$$

The Tapo C200 achieves a device-level cloud risk score of:

$$\mathbf{DCRS}_{\mathrm{Tapo}} \approx 0.63/12$$

Taken together, these findings indicate that none of the observed domains associated with the Tapo C200 ecosystem exhibit *Suspicious* or *Malicious* characteristics. The consistent use of TP-Link–controlled cloud endpoints, the absence of third-party analytics or advertising services, the use of modern and well-configured TLS channels, and the lack of known high-severity CVE associations collectively justify assigning the Tapo C200 a **Low Cloud Risk** rating under the defined scoring model.

4.5.4 Security and Privacy Implications

This analysis highlights a striking contrast between the two ecosystems. The A9 Mini camera relies extensively on third-party analytics, advertising networks, and telemetry-heavy services, which increases privacy risk and violates several ETSI EN 303 645 provi-

sions, including:

- Clause 5.11 – Protection of personal data
- Clause 5.6 – Minimization of attack surfaces

In contrast, the Tapo C200 cloud infrastructure is tightly controlled, privacy-preserving, and aligned with ETSI requirements for:

- Secure communication (Clause 5.3)
- Secure storage and controlled data access (Clauses 5.1–5.2)
- Protection of personal data (Clause 5.11)

These characteristics justify the classification of the Tapo C200 cloud environment as **Low Risk**.

Chapter 5

Proposed Mitigation's

The experimental analysis across Wi-Fi, Zigbee, Z-Wave, and BLE shows that vulnerabilities in smart-home ecosystems rarely arise from protocol specifications alone. Instead, they typically originate from incomplete or insecure implementations, weak default configurations, insufficient update mechanisms, and limited life-cycle management by vendors.

This chapter consolidates the lessons learned from all tests and proposes a set of mitigation's structured along three complementary dimensions: *technical hardening*, *secure lifecycle and update management*, and *ecosystem governance and standardization alignment*. Each recommendation directly addresses weaknesses identified in previous chapters and maps to the relevant requirements of ETSI EN 303 645.

5.1 Technical Mitigation's

The experimental results reveal that vulnerabilities across Zigbee, Z-Wave, BLE, and Wi-Fi stem not only from protocol-level limitations but also from weak or incomplete implementation choices made by device vendors. Addressing these issues requires a combination of secure communication primitives, systematic attack-surface reduction, and robust management of device identity and credentials. This section consolidates the core technical mitigation strategies derived from the empirical observations in the testbed and aligns them with the security principles defined in ETSI EN 303 645.

5.1.1 Strengthening Communication Security

All wireless communication channels should enforce strong cryptographic protection and mutual authentication by default. For Wi-Fi devices, the preferred configuration is WPA3-Personal or WPA3-Enterprise with mandatory PMF, as this significantly reduces the risk of deauthentication spoofing, handshake capture, and downgrade attacks. Legacy modes such as WPA2 without PMF, Wired Equivalent Privacy (WEP), or open networks should be explicitly disabled. This recommendation reflects the Wifi-Crack experiments, where WPA2-only configurations allowed successful deauthentication and management-frame manipulation.

For Zigbee and Z-Wave, link-layer encryption must be strictly enforced and combined with proper validation of frame counters and nonce's to prevent replay attempts such as those reproduced during Zigbee testing. Networks must avoid vendor-wide default keys and instead derive unique network keys per deployment, with secure key rotation mechanisms applied after commissioning.

Similarly, BLE devices should rely on Secure Connections pairing based on ECDH rather than the insecure "Just Works" model. Sensitive GATT characteristics must only be accessible within encrypted and authenticated sessions. Across all protocols, session re-keying and robust nonce validation are essential for preventing replay attacks and for mitigating passive eavesdropping in the presence of long-lived keys.

5.1.2 Reducing Attack Surface

Reducing the attack surface is a foundational security practice that directly mitigates the likelihood of exploitation by limiting the number of reachable entry points available to an adversary. In the context of consumer IoT devices, many of the weaknesses uncovered in this research were triggered via interfaces that remained unnecessarily open, exposed excessive functionality, or lacked post onboarding restrictions.

Applying the principle of least privilege to every communication interface significantly decreases the probability of remote compromise, tampering, or denial-of-service conditions. After device registration, diagnostic endpoints, commissioning ports, or onboarding services frequently remain active even though they are no longer required for normal opera-

tion. Disabling these components prevents attackers from exploiting them through replay, flooding, or spoofing attacks, particularly in scenarios such as Zigbee commissioning floods or Wi-Fi evil-twin onboarding manipulation. This aligns with ETSI EN 303 645 Clause 5.6, which mandates minimization of exposed attack surfaces.

Access-control lists (Access Control Lists (ACL)s) should be enforced on protocol-specific interfaces such as GATT characteristics, MQTT topics, and REST or RTSP APIs. Several vulnerabilities identified in this work including unauthenticated GATT writes, MQTT topic spoofing, and unprotected local RTSP streams occurred precisely because interfaces accepted operations without authentication or role separation. Properly scoped ACLs restrict read/write operations to authorized entities, reducing the risk of both integrity and confidentiality violations.

Broadcast-based discovery mechanisms, such as BLE advertisements or Zigbee beaconing, are frequently abused for enumeration, fingerprinting, or denial-of-service attacks. The experiments confirmed that BlueDOS floods, Wi-Fi beacon flooding, and Zigbee orphan/disassociation storms can overwhelm devices or degrade network performance. Filtering, throttling, or rate-limiting these broadcast operations helps mitigate scanning abuse, reduce DoS susceptibility, and enforce resilience as required by ETSI EN 303 645 Clause 5.10.

Finally, network segmentation using dedicated VLANs or isolated SSIDs separates potentially vulnerable IoT nodes from administrative or user networks. This ensures that even if a device is compromised whether via Wi-Fi replay, Zigbee key extraction, or BLE MitM lateral movement and data ex-filtration are significantly constrained. Segmentation is a widely recommended defence-in-depth strategy and is explicitly aligned with data protection and attack-surface minimization requirements in ETSI EN 303 645 Clauses 5.6 and 5.11.

5.1.3 Hardening Device Identity and Credentials

Beyond transport-layer security, the robustness of device identity and credential handling is critical to preventing spoofing, brute force, and long-term tracking. Based on the experimental findings and ETSI guidance, the following measures are recommended:

- **Dynamic identifiers and anti-tracking:** enforce random BLE MAC address rotation and unique Zigbee Extended Addresses to reduce long-term device tracking and fingerprinting.
- **Secure onboarding secrets:** replace static pairing PINs, QR codes, or vendor-wide defaults with dynamically generated credentials that are unique per device and per onboarding session.
- **Protected key storage:** store cryptographic material (e.g., network keys, private keys, long-term secrets) within secure elements or trusted execution environments, rather than in plain flash or easily accessible firmware regions.
- **Brute-force resistance:** apply retry limits and exponential back-off to pairing and association processes, coupled with lockout or user notification after repeated failures, to mitigate password guessing and connection brute-force attempts.

Collectively, these measures reinforce ETSI EN 303 645 Clauses 5.1 and 5.2 by removing universal defaults, strengthening credential handling, and reducing the feasibility of persistent tracking or impersonation.

5.2 Secure Lifecycle and Update Management

Ensuring long-term security in consumer IoT ecosystems requires more than strong cryptographic mechanisms during normal operation; it necessitates a complete and resilient security lifecycle that spans firmware development, distribution, installation, monitoring, and incident response. The experiments conducted in this thesis highlight that many of the vulnerabilities identified such as outdated firmware, weak onboarding mechanisms, and the absence of authenticated updates are not isolated implementation bugs, but symptoms of incomplete lifecycle management.

5.2.1 Firmware and Software Integrity

Firmware integrity is a foundational requirement for trustworthy IoT devices. Several devices examined in this work publicly exposed firmware version strings or permitted network-triggered update checks without enforcing strong authenticity guarantees. Such

behavior contradicts ETSI EN 303 645 §5.9, which mandates secure update mechanisms that prevent tampering, unauthorized modification, or downgrading of device software.

To achieve lifecycle security, firmware updates must be digitally signed using asymmetric cryptography, where the signature chain is validated against a trusted vendor-controlled root certificate. This ensures that only firmware produced and authorized by the manufacturer can be installed. Update distribution should occur exclusively over encrypted and authenticated channels such as TLS 1.3 or DTLS, preventing interception or manipulation during transit. Devices must verify the integrity and authenticity of updates before installation and enforce rollback protection to eliminate downgrade attacks an attack vector frequently observed in legacy Z-Wave S0 implementations and in BLE stacks affected by Sweyntooth-class vulnerabilities.

To support transparency and reproducibility of security audits, vendors should adopt semantic versioning, publish change logs that explicitly describe security fixes, and implement coordinated vulnerability disclosure processes. Such documentation improves accountability and allows researchers, integrators, and end-users to assess whether a device remains secure throughout its operational lifetime.

5.2.2 Monitoring and Incident Response

Monitoring and incident-response capabilities are essential to detect, contain, and remediate ongoing attacks. Devices and gateways should expose security-relevant events such as pairing attempts, authentication failures, firmware updates, and abnormal reconnection patterns through auditable logs. These logs should be centralized on a trusted gateway or security appliance using secure mechanisms such as Syslog with TLS, MQTT over TLS, or comparable channels.

Automatic anomaly detection can act as an early warning mechanism. Examples include detecting sudden increases in BLE advertisement rates, spikes in Wi-Fi authentication or deauthentication frames, unusually frequent Zigbee rejoin or orphan notifications, or abnormal Z-Wave nonce requests. Simple threshold-based alerts, combined with periodic review of logs, already provide meaningful improvement over the silent failure modes observed in many current products.

At the organizational level, manufacturers should publish a clear vulnerability dis-

closure policy, align with Coordinated Vulnerability Disclosure (coordinated vulnerability disclosure (CVD)) frameworks, and define internal processes for triaging and remediating reported issues. Integrators and advanced users should configure log retention, off-device backups, and notification channels (e-mail, push notifications, dashboards) to ensure that critical events cannot be overlooked.

5.3 Ecosystem Governance and Standardization Alignment

Ensuring cybersecurity in consumer IoT ecosystems requires alignment not only at the device and protocol levels but also across governance, regulatory, and lifecycle processes. The experimental findings of this thesis demonstrate that vulnerabilities in onboarding, communication security, cloud interaction, and firmware validation stem as much from governance gaps as from purely technical weaknesses. Effective mitigation therefore demands a coordinated approach that aligns vendor practices, user responsibilities, and industry-wide standards such as ETSI EN 303 645.

5.3.1 ETSI EN 303 645 Mapping

The mitigation strategies proposed in this chapter collectively reinforce compliance with several key clauses of ETSI EN 303 645:

- **Clause 5.1 (No universal default passwords):** supported through the recommendation of unique per-device credentials, dynamic onboarding secrets, and lockout policies for repeated failures.
- **Clause 5.3 (Secure communications):** strengthened via the adoption of WPA3 with PMF, Zigbee and Z-Wave encryption with proper key management, and modern transport-layer protections such as TLS or DTLS for cloud-domain communications.
- **Clause 5.6 (Minimize exposed attack surfaces):** addressed by disabling unnecessary services after onboarding, enforcing ACLs on GATT, MQTT, and API endpoints, and segmenting IoT traffic using VLANs and dedicated SSIDs.
- **Clauses 5.9 and 5.10 (Software update integrity and resilience):** covered by requiring signed firmware, verifiable chains of trust, rollback protection, broad-

cast filtering, rate limiting, and watchdog mechanisms for handling DoS and replay attacks.

- **Clause 5.11 (Protect personal data):** reinforced through encrypted telemetry, reduced reliance on third-party analytics, and hardened cloud-domain configurations, as evaluated in the A9 and Tapo case studies.

Together, these mappings illustrate how protocol-level countermeasures and lifecycle controls can be translated into measurable compliance with a recognized security baseline.

5.3.2 Vendor and User Responsibilities

The security of smart-home ecosystems is inherently a shared responsibility between device manufacturers, service providers, and end-users. Vendors bear the primary obligation to implement secure-by-default configurations, maintain update pipelines, and ensure that hardware and firmware designs incorporate encryption, authentication, and secure storage mechanisms aligned with the ETSI baseline. This includes continuous maintenance of cloud infrastructures, transparent vulnerability disclosure practices, and timely roll-out of security patches.

End-users also play a critical role. The experiments highlight that weak Wi-Fi configurations, unsegmented networks, or outdated firmware substantially increase the likelihood of compromise. Users should therefore maintain updated devices, isolate IoT networks using VLANs or dedicated SSIDs, enforce strong local credentials, and regularly review update notifications and device security settings.

Regulatory agencies and policymakers act as mediators between these responsibilities. Certification and labeling schemes based on frameworks such as ETSI EN 303 645 or ISO/IEC 27400 [35] provide transparency about device security posture and incentivize manufacturers to adopt robust practices. As the European Union (EU) Cyber Resilience Act (CRA) progresses, formal compliance pathways will increasingly shape the development and deployment of consumer IoT products. Governance mechanisms such as these ensure that security is not treated as an optional add-on but as a mandatory requirement throughout the device lifecycle.

5.4 Strategic Perspective

The results of this research reinforce the need for an end-to-end security architecture in smart-home environments, where each component including wireless protocols, embedded firmware, gateway integration, cloud services, and mobile applications is continuously validated against an evolving threat landscape. Sustaining such an architecture requires embedding automated compliance testing, secure update pipelines, and recurring vulnerability assessments into the entire product lifecycle. This approach simultaneously reduces the attack surface and improves the transparency and reliability of consumer IoT ecosystems.

Adopting adaptive and multi-layered defense models further enhances protection. Mechanisms such as anomaly detection for RF and network traffic, automated firmware attestation, dynamic segmentation, and continuous credential renewal ensure that even if a specific protocol layer is compromised, attackers cannot easily escalate privileges or pivot laterally across the ecosystem. This defense-in-depth strategy aligns directly with the principles of ETSI EN 303 645 and supports emerging regulatory frameworks such as the EU CRA.

By integrating governance, compliance, and lifecycle considerations, the recommendations in this chapter provide a practical pathway towards secure, resilient, and trustworthy smart-home environments capable of withstanding both current and emerging cybersecurity challenges.

Chapter 6

Discussion

This chapter synthesis's the findings obtained from the empirical protocol analysis, cloud-domain communication assessment, and vulnerability mapping conducted throughout this research. The discussion integrates results across all evaluated wireless communication technologies Wi-Fi, Zigbee, Z-Wave, and BLE and reflects on the broader implications for the cybersecurity posture of consumer IoT ecosystems. The analysis highlights structural weaknesses observed in real devices, compares the experimental outcomes with the requirements of ETSI EN 303 645, identifies cross-protocol patterns affecting confidentiality, integrity, and availability, and concludes with limitations and ethical considerations of the conducted experiments.

6.1 Results Interpretation

Across all protocols, the results demonstrate a persistent gap between the security guarantees promised by protocol specifications and the security level actually implemented by commercial devices. This divergence is most pronounced in low-cost devices but remains visible across brands and product categories.

Wi-Fi. Although Wi-Fi provides the strongest cryptographic primitives among the evaluated protocols, device-level implementation quality varied significantly. The A9 Mini camera lacked fundamental protections, including transport-layer encryption (TLS), firmware-signature validation, and resilience to management-frame manipulation. In contrast, the

Tapo C200 implemented a substantially stronger security baseline, yet still exposed an unencrypted local RTSP stream. The Wifi-Crack module further showed that devices configured with WPA2 without Management Frame Protection remained vulnerable to deauthentication, beacon flooding, and association abuse. These observations reflect incomplete adherence to ETSI EN 303 645 Clauses 5.3 and 5.10, demonstrating that radio-layer disruption resilience is still inconsistently implemented across the market.

Zigbee. Zigbee 3.0 introduces secure commissioning and improved key-management mechanisms; however, the experiments highlighted weaknesses arising from legacy behavior and vendor-specific deviations. Certain devices exhibited permissive binding policies, predictable frame counters, or incomplete replay protection. Although the CC2531's RX-only limitations restricted full injection testing, passive captures exposed Transport Key frames during device joining in some flows. These findings confirm that vendors often rely on backwards-compatible but insecure behaviors, complicating compliance with ETSI requirements regarding secure onboarding and credential protection.

Z-Wave. Z-Wave's S2 framework provides strong end-to-end security based on authenticated ECDH exchange, yet implementation inconsistencies remain a critical issue. Several devices supported downgrading to S0 a legacy mode with known vulnerabilities contradicting ETSI recommendations on secure default configurations. Zniffer captures showed occasional unencrypted metadata during inclusion. While overall resilience was higher than in Zigbee, the persistence of downgrade paths undermines the security guarantees offered by the protocol.

Bluetooth Low Energy. BLE exhibited the broadest exposure to passive eavesdropping due to its advertising model. Some devices failed to implement LE Secure Connections or proper bonding, allowing unencrypted GATT characteristics to be observed. BLE fuzzing tests revealed inconsistent error handling and susceptibility to malformed attribute requests, indicating potential for denial-of-service vectors. These behaviors reflect partial implementation of key ETSI requirements related to secure communication, access control, and robustness against malformed input.

Cloud-domain communication. Cloud-domain behavior exhibited the largest disparity across devices. The A9 ecosystem relied on numerous third-party analytics, telemetry, and advertising endpoints some with outdated services or suspicious reputation scores. In contrast, the Tapo C200 relied on a limited and hardened Amazon Web Services (AWS) infrastructure with strong TLS posture. These differences have direct implications for ETSI Clause 5.11 (protection of personal data) and Clause 5.6 (minimization of attack surfaces). The findings emphasize that security weaknesses often arise not from wireless protocols themselves but from the broader cloud ecosystems that accompany consumer IoT products.

6.2 Limitations

Despite a robust methodology and controlled laboratory setup, several limitations influence the generalization of the findings.

Hardware constraints. Certain sniffers such as the CC2531 for Zigbee lack reliable TX capabilities, restricting the execution of advanced spoofing, replay, or injection attacks. Z-Wave analysis depends on proprietary chip sets and tooling (e.g., Silicon Labs PTI interfaces), limiting reproducibility for researchers without equivalent hardware.

Device sample size. The study included two devices per protocol. While this enables detailed case-study analysis, it does not capture the full diversity of device behaviors in the consumer IoT market. Results should therefore be interpreted as representative, not exhaustive, protocol evaluations.

Environmental limitations. RF isolation and interference were controlled but cannot fully replicate complex multi-device home environments. Extensive flooding or denial-of-service testing was intentionally constrained to avoid equipment damage or unintended interference, limiting the scope of availability-related results.

Cloud-analysis constraints. Cloud-domain evaluations relied on HTTPS interception where feasible. Apps implementing certificate pinning (not observed among the tested

devices) would not be decryptable via these methods. Additionally, reputation scores from databases such as WHOIS [64], Shodan [51], VirusTotal [62], or AbuseIPDB [2] are reliant on external accuracy and update frequency.

Firmware analysis restrictions. Deep firmware examination could only be performed on devices where extraction was legally permissible and technically feasible. Devices with encrypted partitions or secure-boot verification limited the scope of reverse-engineering activities.

6.3 Ethical and Legal Considerations

Security research on consumer IoT devices introduces important ethical and legal responsibilities. This study adhered strictly to responsible disclosure principles and conducted all experiments in compliance with applicable regulations.

Scope limitation and consent. All tested devices were owned by the researcher, and all experiments were conducted inside a dedicated, isolated testbed. No external networks or third-party devices were accessed, ensuring compliance with legal and ethical requirements.

Controlled execution of active tests. Replay, flooding, deauthentication, and fuzzing attacks were executed with conservative rate limits and automated abort conditions to prevent equipment damage or unintended interference. Safety remained a priority at all stages of testing.

Handling of sensitive data. Captured keys, device identifiers, and cloud-domain metadata were stored securely and anonymized or redacted for publication. Potential vulnerabilities with broader public impact were considered for responsible disclosure following vendor processes.

Cloud-privacy implications. The discovery of extensive third-party analytics and telemetry traffic in the A9 ecosystem raises concerns beyond purely technical security. Even when encrypted, such data practices may contradict user expectations or regulatory

requirements such as General Data Protection Regulation (GDPR), emphasizing the need for greater transparency in consumer IoT ecosystems.

General ethical responsibility. The significant security disparities observed across devices underscore the responsibility of researchers to communicate findings responsibly and promote improved vendor practices. The overarching aim of this research is to enhance device safety, protect users, and contribute to the development of trustworthy smart-home systems.

Chapter 7

Conclusions

This thesis investigated the cybersecurity posture of contemporary smart-home ecosystems through a multi-layered analysis of the wireless communication protocols that underpin them Wi-Fi, Zigbee, Z-Wave, and BLE as well as the cloud-domain infrastructures supporting consumer IoT devices. By combining theoretical protocol evaluation with empirical testing in a controlled laboratory environment, the research established a systematic and evidence-driven understanding of how real-world devices adhere to or diverge from internationally recognized security baselines, most notably ETSI EN 303 645.

The adopted methodology integrated packet-level inspection, controlled adversarial testing (including flooding, replay, spoofing, and fuzzing), firmware analysis where legally permissible, and extensive cloud-endpoint monitoring through controlled MitM interception. This unified structure enabled direct comparison across protocols and revealed both cross-technology patterns and vendor-specific discrepancies in implementation quality. The findings show clearly that protocol design alone does not guarantee end-user security; instead, real-world protection depends heavily on vendor implementation, configuration choices, and the integrity of associated cloud services.

7.1 Summary of Key Findings

1. **Protocol specifications are robust, but real implementations vary widely:** Across all tested communication stacks, vulnerabilities stemmed primarily from incomplete or insecure vendor implementations. Core protocols such as Z-Wave S2,

Zigbee 3.0, BLE Secure Connections, and WPA3 incorporate strong cryptographic mechanisms; however, several evaluated devices failed to enforce these protections consistently or reverted to insecure defaults. This divergence underscores the importance of secure-by-default design and rigorous vendor testing.

2. **Low-cost devices exhibited systemic and high-risk weaknesses:** The A9 Mini Wi-Fi camera demonstrated pervasive non-compliance with ETSI baseline requirements: plaintext communications, lack of TLS, absence of secure update mechanisms, weak or non-existent authentication, and full vulnerability to disruption attacks such as deauthentication and beacon flooding. These findings echo broader market trends in which low-cost IoT devices prioritize functionality and price over fundamental security, exposing consumers to significant privacy and safety risks.

3. **Mid-range devices provided stronger but still incomplete protections:** The TP-Link Tapo C200 showed significantly improved security posture, implementing encrypted cloud communication, custom credentials, and resistance to basic management-frame manipulation. Nevertheless, unencrypted local RTSP streams and a lack of transparent firmware-signature validation illustrate that even reputable vendors preserve legacy features that introduce avoidable risks. Full compliance with ETSI EN 303 645 therefore remains uneven even in higher-quality products.

4. **BLE and Zigbee maintain persistent weaknesses in pairing and key-handling:** BLE devices frequently omitted bonding or authenticated pairing, enabling sniffing of plaintext GATT characteristics and exposing devices to fuzzing-based crashes. Zigbee devices, despite improvements in Zigbee 3.0, continued to leak operational metadata during join procedures and displayed inconsistent replay protection. These results indicate that onboarding, key management, and secure commissioning remain critical weak points in many IoT ecosystems.

5. **Cloud-domain communication presented the largest privacy disparities:** Cloud analysis revealed significant ecosystem-level differences. The A9 ecosystem relied on a wide array of third-party analytics and advertising domains, some with outdated services or questionable reputation scores. In contrast, the Tapo C200

relied almost exclusively on TP-Link's hardened AWS infrastructure, exhibiting robust TLS configurations and minimal data exposure. These findings demonstrate that cloud behavior not only protocol behavior must be included when assessing consumer IoT security, particularly regarding ETSI Clause 5.11 (protection of personal data).

6. **ETSI EN 303 645 provides a strong benchmark, but adoption remains inconsistent:** High-quality devices aligned with many ETSI requirements, whereas low-cost devices failed across most clauses. This confirms the essential role of regulatory pressure, certification schemes, and transparency mechanisms in ensuring market-wide compliance and protecting consumers.

Research Contributions

This thesis contributes to the field of IoT cybersecurity in several significant ways. First, it introduces a reproducible, cross-protocol testing methodology that integrates packet capture, adversarial techniques, cloud-traffic interception, and firmware analysis, applicable across Wi-Fi, Zigbee, Z-Wave, and BLE. Second, it proposes a unified vulnerability-mapping framework aligned with ETSI EN 303 645, enabling consistent evaluation of heterogeneous devices under a common security baseline.

Third, the research produces a comprehensive empirical dataset that includes packet captures, adversarial-attack traces, and cloud-domain classifications, providing valuable material for replication and future academic work. Fourth, it offers protocol-specific insights showing how weaknesses in onboarding, key management, encryption, firmware-update processes, and session handling can undermine the intended security guarantees defined by modern specifications. Finally, the thesis highlights the cybersecurity and privacy risks arising from cloud-ecosystem behavior, demonstrating that many vulnerabilities originate not from radio-protocol design but from mobile applications and backend infrastructures an aspect often overlooked in the existing literature.

Limitations and Recommendations for Future Work

Although this study provides a comprehensive multi-protocol evaluation of IoT communication security, several limitations influenced the scope and generalization of the results. The analysis relied on a limited number of devices per protocol (two per protocol), which constrains the ability to identify ecosystem-wide or vendor-independent patterns. Additionally, the RF capabilities of certain sniffers, particularly RX-only hardware (e.g., CC2531), limited the depth of injection and fuzzing tests; ethical and legal restrictions further constrained the execution of full-scale DoS and firmware extraction procedures. Finally, protocol behavior was often strongly dependent on vendor-specific implementations, complicating generalization to all Zigbee, Z-Wave, BLE, or Wi-Fi devices.

These limitations open several avenues for future research. A natural progression is to conduct large-scale testing across dozens of devices per protocol, enabling statistically significant conclusions about systemic vulnerabilities in each ecosystem. For Wi-Fi, future work could investigate a comprehensive comparison between devices operating under WPA2-Pre-Shared Key (PSK), WPA3-Simultaneous Authentication of Equals (SAE), and WPA3-Enterprise environments to quantify differences in authentication strength, de-authentication resilience, and management frame protection effectiveness.

Future studies should also employ advanced SDR-based tooling to achieve deeper Physical Layer (PHY)-layer and MAC-layer manipulation, enabling long-duration fuzzing, malformed frame injection, and replay attempts with precise timing control. This would help uncover edge-case vulnerabilities that do not manifest during short laboratory sessions.

The cloud-domain analysis performed in this thesis can be expanded into a full classification pipeline for vendor ecosystems. Automated workflows combining WHOIS, Shodan, Censys, VirusTotal, and AbuseIPDB could characterize large datasets of IoT cloud endpoints, uncover supply-chain risks, and evaluate Content Delivery Network (CDN) usage, tracker presence, and third-party data processors at scale. Complementary research may include mobile application reverse engineering, SDK de-obfuscation, and behavioral profiling under different user scenarios.

Another promising direction involves the integration of machine-learning–assisted anomaly detection, correlating RF anomalies, device logs, and cloud traffic to automatically detect

replay attempts, impersonation attacks, rogue AP events, or abnormal pairing behaviors. This would support real-time threat classification within home automation networks.

Finally, the vulnerability scoring methodology introduced in this thesis can evolve into a formalized scoring model aligned with ETSI EN 303 645. Since the standard does not provide severity metrics, developing a reproducible, community-validated scoring system combining CIA impact, likelihood, exploitability, and resource requirements would significantly benefit manufacturers, researchers, certification bodies, and EU regulatory frameworks such as the CRA. Future work could also involve participation in formal ETSI or CRA pre-certification schemes to validate devices under standardized compliance frameworks.

Overall, expanding device diversity, leveraging more powerful RF tools, deepening cloud and application-layer analysis, and maturing the proposed scoring methodology represent promising research directions that would further strengthen IoT security evaluation in both academic and industrial contexts.

Final Remarks

The findings of this thesis demonstrate that the security of smart-home ecosystems depends not only on the robustness of wireless protocols but also on the quality of vendor implementations, the integrity of cloud infrastructures, and the user's configuration choices. Although modern communication standards provide strong cryptographic foundations, their benefits are often diminished by vendor decisions prioritizing cost, convenience, or legacy compatibility.

As global IoT adoption accelerates, alignment with standards such as ETSI EN 303 645, adoption of secure-by-design principles, and improved transparency in cloud data handling will be essential for building resilient and trustworthy smart-home environments. This thesis provides both a methodological foundation and empirical evidence to support these goals, contributing to ongoing academic and industrial efforts to secure the next generation of connected consumer technologies.

References

[1] Ibrahim Abdul Abdulrahman et al. *Securing Internet of Things (IoT) ecosystems: Addressing scalability, authentication, and privacy challenges*. 2025. URL: https://doi.org/10.30574/wjarr.2025.26.1.0999.

[2] AbuseIPDB. *AbuseIPDB: Reporting and Checking IP Abuse*. Accessed: 2025-02. 2025. URL: https://www.abuseipdb.com.

[3] *Aircrack-ng: Wi-Fi Security Suite*. https://www.aircrack-ng.org/. Accessed: 2025-01-18. 2025.

[4] Hasibul Alam and Emmett Tomai. "Security Attacks and Countermeasures in Smart Homes". In: (2023). DOI: 10.5121/ijci.2023.120209.

[5] Alibaba Cloud Computing Ltd. *Aliyun (Alibaba Cloud)*. https://www.alibabacloud.com/. Accessed: 2025-02-01. 2025.

[6] E. Altulaihan, M. A. Almaiah, and A. Aljughaiman. "Cybersecurity Threats, Countermeasures, and Mitigation Techniques on the IoT: Future Research Directions". In: *Electronics* 11 (2022), p. 3330. DOI: 10.3390/electronics11203330.

[7] Esra Altulaihan, Mohammed Amin Almaiah, and Ahmed Aljughaiman. "Cybersecurity Threats, Countermeasures, and Mitigation Techniques on the IoT: Future Research Directions". In: *Electronics* 11 (2022), p. 3330. DOI: 10.3390/electronics11203330.

[8] Emerging India Analytics. *Cybersecurity in the Internet of Things (IoT): Challenges and Solutions*. 2024. URL: https://medium.com/@analyticsemergingindia/cybersecurity-in-the-internet-of-things-iot-challenges-and-solutions-4f08fae304c1.

[9] Nut Aroon et al. *Detection of TCP and MQTT-Based DoS/DDoS Attacks on MUD IoT Networks*. 2025. URL: https://www.mdpi.com/2079-9292/14/8/1653.

[10] *Atmel RZUSBstick 802.15.4 USB Platform*. https://www.microchip.com/en-us/development-tool/AT86RF231. Accessed: 2025-03-24. 2025.

[11] Balbix. *Internet of Things (IoT) Biggest Security Challenges*. https://www.balbix.com/insights/addressing-iot-security-challenges/. 2024.

[12] Balbix. *Internet of Things (IoT) Biggest Security Challenges*. https://www.balbix.com/insights/addressing-iot-security-challenges/. Accessed: Dec. 20, 2024. 2024.

[13] *Bleak: A Python Bluetooth Low Energy (BLE) Library*. https://github.com/hbldh/bleak. Accessed: 2025-02-03. 2025.

[14] *Bumblebee Firmware for CC2531 ZigBee Sniffing*. https://github.com/Elelabs/elelabs-zigbee-ezsp-utility. Accessed: 2025-02-24. 2025.

[15] ByteDance Ltd. *ByteDance*. https://www.bytedance.com/. Accessed: 2025-02-01. 2025.

[16] Serhat Çelik, Nesibe Yalçın, and Semih Çakır. *MitM Attacks and IoT Security: A Case Study on MQTT*. 2023. URL: https://www.researchgate.net/publication/376558974_MitM_Attacks_and_IoT_Security_A_Case_Study_on_MQTT.

[17] Inc. Censys. *Censys: Internet-Wide Scanning and Asset Discovery*. Accessed: 2025-02. 2025. URL: https://search.censys.io.

[18] CEPro. *Critical Cybersecurity Vulnerabilities in Smart Home Devices Uncovered in New Research*. https://www.cepro.com/networking/critical-cybersecurity-vulnerabilities-in-smart-home-devices-uncovered-in-new-research/. 2024.

[19] CEPro. *Critical Cybersecurity Vulnerabilities in Smart Home Devices Uncovered in New Research*. https://www.cepro.com/networking/critical-cybersecurity-vulnerabilities-in-smart-home-devices-uncovered-in-new-research/. Accessed: Dec. 20, 2024. 2024.

[20] Zachary Comeau. *Critical Cybersecurity Vulnerabilities in Smart Home Devices Uncovered in New Research*. 2024. URL: https://www.cepro.com/networking/critical-cybersecurity-vulnerabilities-in-smart-home-devices-uncovered-in-new-research/.

[21] Axis Communications. *Axis products with AXIS OS 11 achieve ETSI EN 303 645 cybersecurity certification*. https://newsroom.axis.com/news/etsi-cybersecurity-certification. Accessed: 2025-03-22. 2024.

[22] dattatray. *Zigbee Security 101 (Architecture and Security Issues)*. 2023. URL: https://payatu.com/blog/zigbee-security-101-architecture-and-security-issues/.

[23] Samsung Electronics. *Samsung digital appliances earns its first TÜV Nord IoT Security Certification*. Press release. Certification based on ETSI EN 303 645 for Bespoke AI appliances. Aug. 2025. URL: https://news.samsung.com/global/samsung-digital-appliances-earns-its-first-tuv-nord-iot-security-certification.

[24] Emerging Analytics India. *Cybersecurity in the Internet of Things (IoT): Challenges and Solutions*. https://medium.com/@analyticsemergingindia/cybersecurity-in-the-internet-of-things-iot-challenges-and-solutions-4f08fae304c1. 2024.

[25] European Telecommunications Standards Institute. *European Standard ETSI EN 303 645*. https://www.bsi.bund.de/EN/Themen/Unternehmen-und-Organisationen/Standards-und-Zertifizierung/Consumer-IoT/Consumer-IoT_node.html. 2024.

[26] European Telecommunications Standards Institute. *European Standard ETSI EN 303 645*. https://www.bsi.bund.de/EN/Themen/Unternehmen-und-Organisationen/Standards-und-Zertifizierung/Consumer-IoT/Consumer-IoT_node.html. Accessed on 18/06/2024. 2024.

[27] He Fang. *A Comprehensive Analysis of Security Challenges in ZigBee 3.0 Networks*. 2025. URL: https://doi.org/10.3390/s25154606.

[28] Joshua Goodman and Michael Ossmann. *KillerBee: Zigbee and IEEE 802.15.4 Security Research Toolkit*. https://github.com/riverloopsec/killerbee. Accessed: 2025-02-24. 2025.

[29] R. Heartfield, G. Loukas, et al. "A Taxonomy of Cyber-Physical Threats and Impact in the Smart Home". In: *Computers & Security* (2018). DOI: 10.1016/j.cose.2018.07.011.

[30] Ryan Heartfield, George Loukas, et al. "A Taxonomy of Cyber-Physical Threats and Impact in the Smart Home". In: *Computers & Security* (2018). DOI: 10.1016/j.cose.2018.07.011.

[31] Hikvision. *Hikvision awarded two international Cybersecurity Product Certifications*. Press release. Achieved certifications ETSI EN 303 645 and EN 18031 (Bureau Veritas). Apr. 2025. URL: https://www.hikvision.com/en/newsroom/latest-news/2025/hikvision-awarded-2-international-cybersecurity-product-certifications/.

[32] *HomePwn: Offensive Toolkit for IoT and Smart-Home Devices*. https://github.com/ElevenPaths/homepwn. Accessed: 2025-01-15. 2025.

[33] *hostapd - Host Access Point Daemon*. https://w1.fi/hostapd/. Accessed: 2025-01-8. 2025.

[34] Kingsley Theophilus Igulu et al. *Security Challenges in IOT*. 2024. URL: https://www.researchgate.net/publication/379278792_Security_Challenges_in_IOT.

[35] International Organization for Standardization. *ISO/IEC 27400:2022 — Cybersecurity and privacy protection – Guidelines for Internet of Things (IoT)*. https://www.iso.org/standard/80559.html. Accessed: 2025-02-01. 2022.

[36] J. K. Josphat. *Home Security Devices: Attack, Vulnerabilities, and Threats*. https://www.linkedin.com/pulse/home-security-devices-attack-vulnerabilities-threats-josphat-k/. 2023.

[37] Josphat K. *Home Security Devices: Attack, Vulnerabilities, and Threats.* 2023. URL: https://www.linkedin.com/pulse/home-security-devices-attack-vulnerabilities-threats-josphat-k/.

[38] Stan Kaminsky. *How to secure your smart home.* 2024. URL: https://www.kaspersky.com/blog/how-to-secure-smart-home/47472/.

[39] Kaspersky. *How to Secure Your Smart Home.* https://www.kaspersky.com/blog/how-to-secure-smart-home/47472/. 2024.

[40] Kaspersky. *How to Secure Your Smart Home.* https://www.kaspersky.com/blog/how-to-secure-smart-home/47472/. Accessed: Dec. 20, 2024. 2024.

[41] Salam Khanji, Farkhund Iqbal, and Patrick Hung. "ZigBee Security Vulnerabilities: Exploration and Evaluating". In: *IEEE International Conference on Applied Computing and Security (IACS 2019).* 2019. DOI: 10.1109/IACS.2019.8809115.

[42] I. Lee. "Internet of Things (IoT) Cybersecurity: Literature Review and IoT Cyber Risk Management". In: *Future Internet* 12 (2020), p. 157. DOI: 10.3390/fi12090157.

[43] In Lee. "Internet of Things (IoT) Cybersecurity: Literature Review and IoT Cyber Risk Management". In: *Future Internet* 12 (2020), p. 157. DOI: 10.3390/fi12090157.

[44] Asel Madiiarbekova. *Cybersecurity in IoT Devices: Vulnerabilities, Risks, and Mitigation Strategies.* 2025. URL: https://dx.doi.org/10.2139/ssrn.5075016.

[45] *nRF Sniffer for Bluetooth LE.* https://www.nordicsemi.com/Products/Development-tools/nRF-Sniffer-for-Bluetooth-LE. Accessed: 2025-02-11. 2025.

[46] Jean Pierre Ntayagabiri et al. *A Comprehensive Approach to Protocols and Security in Internet of Things Technology.* 2024. URL: https://doi.org/10.62411/jcta.11660.

[47] Terry Olaes. *Internet of Things (IoT) Biggest Security Challenges.* 2024. URL: https://www.balbix.com/insights/addressing-iot-security-challenges/#:~:text=IoT%20security%20challenges%20include%20device,vulnerable%20entry%20points%20for%20hackers..

[48] Rambus. *Smart Home: Threats and Countermeasures*. https://www.rambus.com/iot/smart-home/. Accessed: Dec. 20, 2024. 2024.

[49] Ron Ross, Michael McEvilley, and Janet C. Oren. *Guide for Conducting Risk Assessments*. NIST Special Publication. Gaithersburg, MD, 2012. URL: https://doi.org/10.6028/NIST.SP.800-30r1.

[50] Federal office for information security. *European Standard ETSI EN 303 645*. 2024. URL: https://www.bsi.bund.de/EN/Themen/Unternehmen-und-Organisationen/Standards-und-Zertifizierung/Consumer-IoT/Consumer-IoT_node.html.

[51] Shodan. *Shodan: The Search Engine for Internet-Connected Devices*. Accessed: 2025-02. 2025. URL: https://www.shodan.io.

[52] *Silicon Labs Simplicity Studio*. https://www.silabs.com/developers/simplicity-studio. Accessed: 2025-01-12. 2025.

[53] Gurpreet Singh. *Wifi-Crack: Automated Wi-Fi Attack Framework*. https://github.com/Gurpreet06/Wifi-Crack. Accessed: 2025-11-15. 2024.

[54] Tom Spring. *Millions of IoT Devices Vulnerable to Z-Wave Downgrade Attacks, Researchers Claim*. https://threatpost.com/millions-of-iot-devices-vulnerable-to-z-wave-downgrade-attacks-researchers-claim/132295/. Accessed: Dec. 20, 2024. 2018.

[55] International Organization for Standardization. *ISO/IEC 27005:2018 Information technology — Security techniques — Information security risk management*. Available from https://www.iso.org/standard/75281.html. Geneva, Switzerland, 2018.

[56] L. Tawalbeh et al. "IoT Privacy and Security: Challenges and Solutions". In: *Applied Sciences* 10 (2020), p. 4102. DOI: 10.3390/app10124102.

[57] Tencent Holdings Ltd. *Tencent*. https://www.tencent.com/. Accessed: 2025-02-01. 2025.

[58] *Tshark Command Line Network Analyzer*. https://www.wireshark.org/docs/man-pages/tshark.html. Accessed: 2025-03-22. 2025.

[59] UJRRA. *Cyber Threats Analysis in the Internet of Things*. 2025. URL: https://doi.org/10.69557/ujrra.v4i2.158.

[60] Ana Valenzuela-Pérez et al. *A Study on Packet Error Rate in Bluetooth Mesh Networks With Relay Redundancy*. 2025. URL: https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=10945757.

[61] George Vardakis et al. "Review of Smart-Home Security Using the Internet of Things". In: *Electronics* 13 (2024), p. 3343. DOI: 10.3390/electronics13163343.

[62] VirusTotal. *VirusTotal: File and URL Analysis Service*. Accessed: 2025-02. Google / Chronicle Security, 2025. URL: https://www.virustotal.com.

[63] Wattlecorp. *OWASP IoT Top 10*. https://www.wattlecorp.com/owasp-iot-top-10/. Accessed: Jan. 7, 2025. 2025.

[64] WHOIS. *WHOIS Lookup Service*. Accessed: 2025-02. Whois.com, 2025. URL: https://www.whois.com/whois/.

[65] *Wireshark: The World's Most Popular Network Protocol Analyzer*. https://www.wireshark.org/. Accessed: 2025-03-24. 2025.

[66] Xin-Wen Wu, En-Hui Yang, and Junhu Wang. "Lightweight Security Protocols for the Internet of Things". In: *IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*. 2017, pp. 1–7. DOI: 10.1109/PIMRC.2017.8292779.

[67] Han Yang. *Discovering Security Weaknesses in IoT Device Setup*. 2025. URL: https://hdl.handle.net/10222/85010.

[68] *Zniffer: Z-Wave Network Analyzer*. https://www.silabs.com/documents/public/user-guides/INS13112-Developer-Guide.pdf. Accessed: 2025-03-20. 2025.