Vulnerability-oriented risk identification framework for IoT risk assessment

Mohammad Beyrouti, Ahmed Lounis, Benjamin Lussier, Abdelmadjid Bouabdallah, Abed Ellatif Samhat

Vulnerability-oriented Risk Identification Framework for IoT Risk Assessment

Mohammad Beyrouti[a], Ahmed Lounis[a], Benjamin Lussier[a], Abdelmadjid Bouabdallah[a], Abed Ellatif Samhat[b]

[a] *Université de Technologie de Compiègne, CNRS, Alliance Sorbonne Université, Heudiasyc, Compiegne, 60203, Compiegne Cedex, France*
[b] *Faculty of Engineering-CRSI, Lebanese University, Beirut, BP2, Hadath, Lebanon*

Abstract

The proliferation of Internet of Things (IoT) systems across diverse applications has led to a notable increase in connected smart devices. Nevertheless, this surge in connectivity has induced a broad spectrum of vulnerabilities and threats, jeopardizing the security and safety of IoT applications. Security risk assessment methods are commonly employed to analyze risks. However, traditional IT and existing IoT-tailored security assessment methods often fail to fully address key IoT aspects: complex assets intercommunication, dynamic system changes, assets' potential as attack platforms, safety impacts of security breaches, and assets resource constraints. Such oversights lead to significant risks being overlooked in the IoT ecosystem. In this paper, we propose a novel vulnerability-oriented risk identification framework comprising a four-step process as a core element of IoT security risk assessment, applicable to any IoT system. Our process enhances both traditional and IoT-specific security risk assessment methods by providing tailored approaches that address their crucial oversights for comprehensive IoT risk assessment. We validate our process with a case study of an IoT smart healthcare system using a proposed expert-driven approach. The results confirm that our process effectively identifies critical attack scenarios originating from the lack of proper security measures, mobility, and intercommunication processes of IoT devices in the healthcare system. Furthermore, our analysis reveals potential attacks that exploit the IoT devices as platforms to target the backend and user domains. We demonstrate the feasibility of our process for identifying realistic risks by conducting simulations of two derived attack scenarios using the Contiki Cooja network simulator.

Keywords: Dynamic IoT, Risk Identification, IoT Paradigm, Attack Scenario Design, Attack Simulation, Safety Impact.

1. Introduction

The Internet of Things (IoT) represents a revolutionary paradigm in which everyday objects, from simple sensors and actuators to complex industrial machinery and control systems, become interconnected [1]. This interconnectivity enables continuous data acquisition, exchange, and processing, enhancing the service and information access via the internet. IoT networks collect extensive data, offering accurate situational awareness for informed decision-making in various applications like smart healthcare, smart grids, and smart homes [2]. The large-scale deployment of IoT systems, predominantly in open environments, presents significant security challenges, including numerous vulnerabilities and potential cyber-attacks. Compromising the security of these systems could lead to environmental damage and threaten human lives, especially in critical domains such as nuclear plants and healthcare [3].

IoT devices often employ lightweight protocols, lack seamless over-the-air (OTA) updates, and possess weak security mechanisms due to the resource constraints of the commercial off-the-shelf hardware and software components aimed at reducing costs and facilitating network integration [4]. Unfortunately, many IoT manufacturers, prioritizing profit and often lacking cybersecurity awareness, release products without thorough security checks [5]. Additionally, some manufacturers neglect to establish policies for patching known vulnerabilities. For instance, a report by the IoT Security Foundation [6] found that only 10% of over 300 IoT companies had policies for disclosing vulnerabilities and providing patches.

Traditional security preventive measures designed for computing systems often demand significant resources, such as complex encryption algorithms, multi-factor authentication, and heavyweight antivirus programs. These measures are generally unsuitable for the heterogeneous nature of IoT devices, which vary widely in hardware capabilities, including limited memory, energy, and processing power. They also feature diverse customizations of operating systems, bare-metal firmware, and communication protocols [7]. Such diversity complicates the implementation of standardized and homogeneous security measures [8]. For example, blockchain has been identified as a promising technology for securing and validating transactions on data. It ensures high security, immutability, and traceability through decentralized data processing [9]. However, the integration of blockchain in IoT is still an emerging field lacking standards, thereby limiting its integration to specific IoT use cases [10]. Also, traditional IP-based Intrusion Detection Systems (IDS) and firewall systems are designed to monitor IP-based traffic but fail to adequately cover or interpret the specifics of short-range wireless communication protocols frequently used by IoT devices at the sensing and controlling

Email addresses: mohammad.beyrouti@hds.utc.fr (Mohammad Beyrouti), ahmed.Lounis@hds.utc.fr (Ahmed Lounis), benjamin.lussier@hds.utc.fr (Benjamin Lussier), madjid.bouabdallah@hds.utc.fr (Abdelmadjid Bouabdallah), samhat@ul.edu.lb (Abed Ellatif Samhat)

layer, such as Bluetooth Low Energy (BLE), Zigbee, Z-wave, RFID, and NFC [11]. For example, an attacker could spoof a genuine BLE sensor to inject false readings into a hybrid network, which traditional systems might consider legitimate traffic as it conforms to standard IP packet structures between the device and the rest of the network. As a result, numerous IoT devices remain vulnerable to both known and unknown risks, lacking adequate security controls. This poses serious impacts to security, business assets, and domain safety if exploited by adversaries [12]. A study [13] found that 385,060 out of 1,362,906 IoT devices (28.25%) had at least one N-day vulnerability.

A potential approach to analyzing these vulnerabilities, threats, and their potential implications requires conducting a risk assessment process. Indeed, the direct application of traditional risk assessment methods to IoT systems often results in significant oversight of key considerations [14]. Most traditional IT methodologies, established before the proliferation of IoT, focus on risks associated with well-known organizational assets using secured standard protocols with robust security features. However, they often overlook the intercommunication processes through which assets are coupled and operate [15]. This oversight is especially problematic for IoT assets that typically use resource-optimized lightweight protocols with dynamic intercommunication processes such as device advertising, pairing, device connections and reconnections, and connection parameter updates. In addition, traditional methods view system assets as valuable properties when assessing their potential risks to provide the necessary security protection, while neglecting their potential to serve as attack platforms. In contrast, IoT assets can more easily be compromised and used as platforms for launching dynamic attacks. Furthermore, these methods assess risks of static system assets through periodic risk assessments that require extensive knowledge and study of the target system. Such practices are often unsuitable for IoT systems, where the dynamic nature and shifting system boundaries cause the system to change frequently, rendering previous traditional risk assessments quickly obsolete. Traditional methods also presuppose that system assets can implement heavyweight security measures and receive updates regularly. This assumption is often impractical for IoT systems, as devices may be resource-constrained and lack seamless OTA update mechanisms [16]. Moreover, traditional risk assessments often neglect factors related to the safety impact on the physical entity domain of a security breach [17]. This is crucial for IoT systems, where a security breach could have direct impacts on human lives and cause environmental damage. For instance, while a security breach in a traditional IT system may compromise data integrity, a compromised IoT device, such as an insulin pump, poses direct threats to human lives. Therefore, IoT-focused risk identification approaches are needed to overcome these challenges. To the best of our knowledge, existing risk assessment methods designed for IoT fall short of fully addressing the aforementioned key aspects in risk identification for IoT, leading to significant risks being overlooked.

The objective of this paper is to develop a risk identification framework as a core element of security risk assessment for IoT, addressing the challenges mentioned above. More precisely, we propose a *vulnerability-oriented* risk identification framework designed as an interconnected process specific to IoT, following the taxonomy presented by NIST in [18]. Our process begins by identifying common weaknesses and vulnerabilities, then contextualizes threats through attack scenarios in the IoT environments using comprehensive approaches. It finally uses qualitative metrics to profile impact and likelihood characteristics factors associated with the identified attack scenarios. Our proposed process consists of four interrelated steps presented in section 4, all developed in accordance with the risk identification factors provided by NIST. Figure 1 represents the risk assessment subprocesses and the projection of our proposed process steps onto the risk identification subprocess. The main contributions of this paper are as follows:

1. We propose a new process to identify common high-risk security weaknesses and vulnerabilities in IoT components through an expert-driven mapping approach between MITRE's Common Weakness Enumeration (CWE[1]) database and Open Web Application Security Project (OWASP[2]) top ten IoT security weakness categories. Our process includes a comprehensive spreadsheet developed by reviewing and analyzing over 600 Common Vulnerabilities and Exposures (CVE) entries from the National Vulnerability Database (NVD[3]) relevant to the IoT domain (Steps 1 and 2).
2. We propose an approach for identifying common attack patterns using a spreadsheet derived from the Common Attack Pattern Enumeration and Classification (CAPEC[4]) database. This spreadsheet helps define the execution of attack actions, which correlates with previously identified CWE-IoT and CVE-IoT entries. This information is utilized to establish novel attack scenarios (Step 3 and Step 4).
3. We propose metrics to profile the impact and likelihood of attack scenarios, evaluating their severity on security attributes, business assets, IoT application safety, and the likelihood of vulnerability exploitation (Step 4).
4. Finally, we present an in-depth validation of our process through a use case study, including simulations of two derived attack scenarios.

This process is well-suited for integration into existing traditional security risk assessment frameworks. It incorporates into its core methodology the potential security risks arising from the dynamic nature of IoT, complex interprocess communications, the use of assets as dynamic attack platforms, and the typical resource constraints of IoT devices. Additionally, it considers crucial safety factors essential for risk estimation and prioritization. Moreover, we believe this process will be particularly relevant to researchers focused on developing new

[1] https://cwe.mitre.org/
[2] https://owasp.org/
[3] https://nvd.nist.gov/
[4] https://capec.mitre.org/

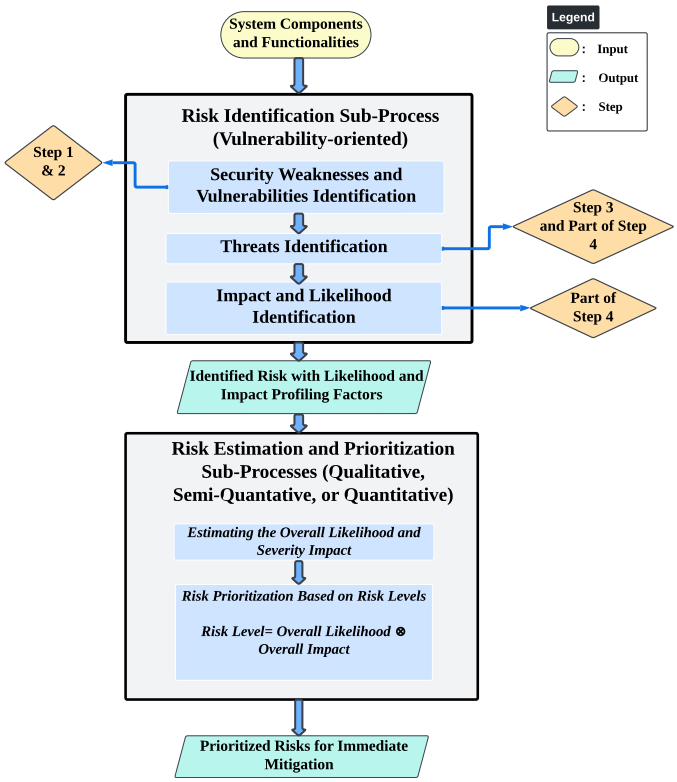

Figure 1: The Proposed Process Steps on the Risk Identification Sub-process of Risk Assessment

defensive mechanisms for IoT networks and using realistic attack scenarios for validation, aligning with one of our primary goals in its development. We have previously outlined the first three steps of our process in [19]. This article focuses on the refined third and fourth steps and demonstrates the capability of the entire process to address key IoT aspects for comprehensive risk assessment.

The structure of the remainder of this paper is outlined as follows: Section 2 delineates the principal security databases, ontologies, and taxonomies employed in our proposed methodology. Section 3 examines both traditional and IoT-specific security risk assessment methods, highlighting their limitations in addressing key aspects of IoT risk identification. Section 4 introduces our novel vulnerability-oriented risk identification framework, designed as a four-step process for IoT. This process systematically identifies common weaknesses, vulnerabilities, and threat events and integrates these elements through attack scenarios complemented by impact and likelihood profiles. Section 5 evaluates how our process effectively addresses the key aspects of IoT risk identification and overcomes the limitations of traditional methods. Section 6 provides a detailed use-case analysis validating our process within a smart healthcare IoT system using a proposed expert-driven approach. It also explores the framework's generalizability and discusses the application of derived attack scenarios across different IoT systems. The paper concludes in Section 7 with simulations of two specific attack scenarios, and Section 8 presents conclusions and future research directions for IoT risk mitigation.

2. Concepts and Definitions

In this section, we provide background on IoT concepts and discuss databases and standards for identifying and assessing security vulnerabilities and threats.

2.1. IoT Architecture

Academic literature divides IoT network architecture into three layers: perception, network, and application, with several studies detailing this structure [20]. The ISO/IEC 30141 standard [21] proposes a model with six domains: physical entity, sensing and controlling, operational and management, resource and interchange, application services, and user domain. However, IoT applications still require a universally endorsed standard by industrial and academic organizations.

2.2. Security Terms

In this subsection, we present different basic terms used in our proposed security assessment process, as cited from [22], [23], [24], OWASP, and NIST [18]:

- **Security Risk Assessment:** The process of identifying, estimating, and prioritizing risks to assets and operations.
- **Security Risk Identification:** Includes identifying common threat events and sources, weaknesses, and vulnerabilities. Additionally, it involves profiling the potential impact and likelihood factors of these threats. When synthesized effectively, these elements facilitate the identification of security risks that may affect the system, setting the stage for the subsequent estimation and prioritization of these risks.
- **Security Risk Estimation and Prioritization:** The process of evaluating the overall severity impact and likelihood of security risks. These assessments are integrated to determine the levels of risk, which inform the prioritization of risk mitigation strategies.
- **Vulnerability:** An exploitable security weakness in a system, identified from previous incidents, which adversaries can leverage to compromise system security. For example, a vulnerability might result from a device's failure to properly validate the length of incoming packets before processing, allowing an attacker to send oversized packets that cause the device to crash.
- **Security Weakness:** A flaw in a system's software or hardware that could lead to one or more vulnerabilities. For example, an improper input validation weakness in a device's protocol stack's communication layers could lead to vulnerabilities if exploited by an individual aiming to breach the system's security.
- **Threat Source:** The attributes of adversaries characterized by their types, skills, and the means they utilize to exploit vulnerabilities.

- **Threat Event:** An event or situation, generally described (e.g., DoS, Spoofing) or detailed using tactics (e.g., CAPEC), potentially impacts system assets adversely when caused by a threat source.
- **Attack Scenario Models:** Scenarios describe the successive methods that an attacker could use to achieve a threat event, exploiting generic potential security weaknesses that affect a component regardless of its specific vendor or version.
- **Attack Scenario Model Implementations:** Specific implementations or instances of attack scenario models tailored to exploit particular vulnerabilities identified in the system.

2.3. Open Web Application Security Project (OWASP) IoT

OWASP documents online security threats and weaknesses in computing systems, including a *Top Ten* list that categorizes common IoT weaknesses based on cyber-attacks, exploitability, detectability, and human safety risks. Table 3 presents detailed descriptions of OWASP's *Top 10 IoT Weakness Categories* from the latest 2018 release, arranged in descending order of criticality.

2.4. Common Weakness Enumeration (CWE) by MITRE

The MITRE Corporation maintains the CWE, a catalog of over 1,000 software and hardware common weaknesses that target IT, OT, and IoT domains, each with a unique code like CWE-20 and detailed description. Relationships depicted within the CWE illustrate various abstraction levels, similarities, and the potential for one weakness to lead to another. The NVD aligns CVE descriptions with relevant CWE through ontologies, categorizing vulnerabilities by type and cause.

2.5. Common Vulnerabilities and Exposures (CVE) by NIST

NVD, maintained by NIST, reports an exhaustive list of vulnerabilities targeting IT, OT, and IoT domains, known as CVE database. Each vulnerability is uniquely identified, acquiring a timestamp for its publication date and a unique sequential number separated by hyphens, such as CVE-2024-26001. The Common Vulnerability Scoring System (CVSS) assigns severity ratings to vulnerabilities on a scale from 1 to 10, evaluating their impact on primary security attributes. However, this scoring system, developed for traditional systems, may not be suitable for IoT vulnerability assessment [25]. The security community trusts NVD, and several vulnerability databases rely on its information. For detailed information on each CVE vulnerability, please refer to the NVD database.

2.6. Common Attack Pattern Enumeration and Classification (CAPEC) by MITRE

The MITRE Corporation's CAPEC catalog documents 559 common cyber-attack patterns. Each pattern is assigned a unique number, such as CAPEC-34, and provides detailed descriptions of preparatory and execution actions illustrating how adversaries exploit common security weaknesses across various domains, including IT, OT, and IoT. To reduce the effectiveness of these attacks, CAPEC recommends specific mitigation techniques. Furthermore, CAPEC's descriptions aid in threat assessment by identifying scenarios in which vulnerabilities are exploited. MITRE aligns CAPEC descriptions with relevant CWE through ontologies in the IT domain.

3. Related works

This section introduces databases akin to the NVD that identify security vulnerabilities, followed by a literature review on security assessment processes for IoT risk analysis.

3.1. Vulnerability Databases

Several databases identify and assess vulnerabilities in the literature. The China National Vulnerability Database (CNVD[5]), akin to NVD; it utilizes the CVSS score to rate vulnerabilities and is managed by the Chinese National CERT. The US-CERT[6] operates a vulnerability database that includes CVE entries. Additionally, US-CERT issues advisories on vulnerabilities in IoT industrial control systems and medical devices categorized by ICSA and ICSMA IDs, through ICS-CERT[7]. Other vulnerability databases mentioned in the literature include Japan's Ipedia Vulnerability Database (JVNDB[8]), China's National Vulnerability Database of Information Security (CNNVD[9]), and the Chinese Industrial Internet Security Emergency Response Center (CN-ICS-CERT[10]).

While most of these databases draw entries from NVD and employ CVSS-based scoring systems, only a few, such as CN-ICS-CERT, US-CERT, and CNVD, specifically target IoT vulnerabilities to some extent. Other repositories, including the NVD, catalogue a multitude of reported vulnerabilities related to software, hardware, and network protocols across diverse systems and technological domains. However, they fall short in offering methodologies for discerning IoT-specific vulnerabilities—characterized by their distinctive characteristics and constraints—from those associated with other technological facets in conventional networks.

3.2. Research Work on Security Assessment in IoT

This subsection presents related works that conduct and propose risk assessment methods for IoT, as documented in the literature, along with their limitations in addressing key aspects of IoT.

[5] http://www.cnvd.org.cn/
[6] https://www.kb.cert.org/vuls/
[7] https://www.cisa.gov/uscert/ics/advisories-by-vendor
[8] https://jvndb.jvn.jp/en/
[9] http://www.cnnvd.org.cn/
[10] https://www.ics-cert.org.cn/

3.2.1. Application of Traditional Security Assessment Methods to IoT Systems

Various traditional security assessment methodologies and tools are employed in the literature to identify and assess threats, vulnerabilities, and risks for network security [4]. Zahra and Abdelhamid [26] conducted a security risk analysis for IoT using the qualitative traditional risk analysis method EBIOS, aiming to identify security risks across the three layers of an IoT architecture and to provide severity impact and likelihood scoring factors for risk identification and evaluation. Ali and Awad [27] undertook a risk assessment process for IoT smart homes using the OCTAVE traditional methodology, pinpointing potential threats, risks, and suggesting mitigation strategies. Bhuyan et al. [28] assessed the communication channel between smartphones and IoT devices, employing the traditional CORAS risk assessment method to identify threats and vulnerabilities, while also examining several cryptographic methods to determine the most suitable solutions for resource-constrained IoT devices. Finally, Hankin et al. [29] developed a framework, similar to our approach, for automating attack generation, employing visual scenario representations based on the CAPEC, CWE, and CVE databases for risk identification. Nevertheless, this tool is specifically tailored for IT organizational systems and utilizes MITRE's ontology mappings of CAPEC, CWE, and CVE in traditional system contexts for attack generation. Consequently, to effectively apply this framework to IoT systems, substantial modifications and configurations in the methodology's scope and dedicated efforts are required to realign the database ontologies within the IoT domain. For instance, this framework could be integrated into our proposed framework for automated attack generation, thereby addressing the distinct key aspects of risk identification inherent in the IoT landscape.

3.2.2. Limitations of Traditional Assessment Methods in IoT Systems

While these methodologies provide a foundation for risk assessment, they fall short when applied to complex and dynamic IoT ecosystems [30]. Firstly, these methods primarily address and analyze risks targeting known organizational assets, which include devices, communication platforms, information, and interfaces. These assets are considered valuable to the organization and require security protection [31]. However, these methods overlook the processes through which devices establish dynamic connections—the coupling processes that enable IoT devices to communicate using protocols such as BLE, Zigbee, Z-wave, and others ($L_{coupling}$). For example, establishing BLE communications in IoT involves several phases: *Advertising*, *Connection*, and possibly *Pairing*. Each phase presents unique risk areas not addressed by conventional methods, particularly during updates to connection parameters like the channel map, latency or connection intervals, and security modes.

Secondly, these methods do not consider that these assets could also serve as platforms for dynamic attacks within their scope analysis (L_{attack_vector}) [32]. This oversight is critical for IoT, where assets can be easily compromised and facilitate significant dynamic attacks. For instance, in a hybrid IoT network, an attacker could spoof a sensor using a vulnerability in a short-range wireless protocol, then inject a malicious SQL payload to be relayed through the gateway, exploiting a server vulnerability to compromise the database.

Thirdly, these methods assess risks periodically, typically in response to significant changes in the business processes of organizations. They require comprehensive knowledge of the system's static assets, as well as potential threats and vulnerabilities. However, given the challenges in developing such in-depth knowledge even within IT organizations [33], the dynamic nature and frequent system changes characteristic of IoT environments cause periodic traditional methods to fail in recognizing or anticipating significant risks ($L_{dynamics}$). Therefore, to provide an early warning of emerging risks, the IoT domain requires predictive consideration of system dynamics and changes [33]. For instance, IoT devices may join networks at any time, even temporarily, to execute specific tasks, and then disconnect upon task completion. Some devices, characterized by dynamic mobility, may constantly connect and disconnect. These scenarios are often overlooked by traditional methods.

Fourthly, the traditional methods assume that security measures and software patches can be implemented by the system's assets without resource constraints. However, IoT devices which often face resource limitations, present challenges in implementing heavyweight security measures and seamless updates recommended by these methods ($L_{resource\_constraints}$). This results in a proliferation of emerging vulnerabilities and threats prior to subsequent assessments, presenting an extensive attack landscape that traditional methods have not adequately addressed [12].

Finally, as IoT systems become increasingly integral to critical applications, such as in healthcare and railway systems, traditional assessment frameworks may not adequately emphasize and prioritize the safety impact factor associated with threats (L_{safety}). For instance, a DoS attack on medical devices, or physical tampering with railway signaling equipment, can have a direct impact on human lives [34].

These limitations highlight the need for IoT-focused approaches to risk identification. These approaches should provide an abstraction of system details while simultaneously considering the aforementioned IoT aspects for comprehensive risk assessment. This remains an open research concern [35]. Table 1 provides a summary of the limitations associated with applying traditional risk assessment methods to IoT systems.

3.2.3. Security Risk Identification Frameworks for IoT

In the literature, various research works have proposed security risk identification and assessment methodologies for IoT. For instance, Wang et al. [36] introduced a vulnerability assessment method for risk identification in industrial IoT (IIoT) that uses attack graphs and maximum flow analysis to assess vulnerabilities and risk paths. This method evaluates network system vulnerabilities based on the CVSS scoring system. Although the method has its strengths in generating dynamic attacks, it primarily focuses on IIoT systems and regards system components as static nodes within the network topology

Table 1: Limitations of Traditional Risk Assessment for IoT Risk Identification

Limitation	Description	Key IoT Aspect
Shortcomings of grasping the advertising and coupling of IoT's internal communications and their potential risks ($L_{coupling}$)	Traditional risk assessment methods primarily focus on well-known assets such as information, devices, communication platforms, and interfaces.	IoT devices use a wide range of lightweight protocols that raise vulnerabilities during the processes of dynamic coupling and bonding, which traditional methods often overlook.
Failure to consider the system's assets as attack platforms (L_{attack_vector})	Traditional risk assessment methods view the studied assets as valuable entities needing protection within organizations. Indeed, the potential for dynamic attacks launched by adversaries using compromised nodes is not adequately addressed by traditional methods.	In the context of IoT, where nodes are more vulnerable due to resource constraints, device mobility, lightweight security mechanisms, and physical accessibility, they can be easily compromised and used as platforms for dynamic attacks.
Periodic risk assessment with limited systems knowledge ($L_{dynamics}$)	Traditional risk assessment methods assume that systems remain largely unchanged over a short period and focus on specific static well-known assets, with all risks linked to this specific assessment.	IoT architecture is inherently dynamic, and frequent asset variability is common (mobility, standby, interoperability requirement, heterogeneity, etc.). Thus, critical risks could be overlooked.
Shortcomings in accounting for the IoT resource constraints ($L_{resource\_constraints}$)	Traditional risk assessment methods assume that the studied assets can readily implement security measures with sufficient resources.	IoT devices are often face resource limitations in implementing recommended traditional security measures and lack seamless over-the-air update capabilities. This situation leads to the significant emergence of both known and unknown vulnerabilities, thereby increasing the attack surfaces.
Failure to consider safety impact metrics for risk estimation and prioritization (L_{safety})	Traditional risk assessment methods often base risk severity impact estimates solely on primary security attributes—confidentiality, integrity, and availability—and business assets.	IoT devices are frequently deployed in critical environments, making safety a crucial factor in IoT risk assessment.

for the generation of attack graphs, without considering coupling and dynamic aspects. The method also does not offer approaches for identifying potential weaknesses and vulnerabilities in IIoT, which poses challenges in the comprehensiveness of the methodology and in identifying risks while abstracting from system details.

Casola et al. [37] developed a methodology for automated threat modeling and risk assessment in IoT systems, as demonstrated by a home automation case study. Their approach encompasses system modeling, threat assessment, risk analysis, and the identification of security controls. However, they treat the system components as static assets requiring protection, without considering the risks that could arise from the system's dynamics. Moreover, by analyzing risks individually for each component, this approach neglects risks associated with internal communication and the use of components as attack platforms. Additionally, the authors' reliance on identifying security controls based on traditional methods is not suitable for resource-constrained devices, and the safety risk factor is not considered as well.

Shivraj et al. [38] presented a generic risk assessment framework for the IoT paradigm that integrates traditional risk assessment techniques with threat models like STRIDE and LINDDUN, alongside novel approaches to defining risk for IoT. Although the framework analyzes risks using a dynamic database of existing threats for IoT and considers the propagation of attack vectors, it lacks approaches to identify possible weaknesses and vulnerabilities, essential components in risk identification. This omission leads to inadequately addressing the risks associated with dynamic IoT components and resource constraints. Additionally, the framework fails to take into account the safety factor.

Kang et al. [39] proposed a multidimensional security risk assessment for IoT systems, employs assets, threats, and vulnerabilities as primary elements in identifying and evaluating security risks. However, their method regards IoT components as static assets, similar to traditional risk identification practices, without considering the dynamics of these assets, their resource constraints, and their potential as attack platforms. Moreover, the method does not take into account the complexity of interprocess communications in the risk assessment, and it also does not consider a safety impact metric within risk estimation and evaluation.

Sicari et al. [40] proposed a general IoT risk assessment framework focusing on evaluating the trustworthiness of IoT middleware, particularly within the distributed processing and storage layers managing data from IoT networks. The authors derived lists of vulnerabilities and threats from standards such as ISO27001/ISO22301 to describe and analyze their solution. However, while these lists are standard in traditional organizational systems and can partly assess middleware security, they do not address the vulnerabilities and threats specific to the lightweight protocols used by resource-constrained IoT nodes in the sensing and control domain. This oversight leaves the middleware vulnerable to attacks from devices within local networks, which attackers could exploit. Furthermore, the authors overlook safety impact factors in their analysis.

Stellios et al. [41] proposed a risk assessment methodology for cyber-physical IoT systems, utilizing modules to model possible cyber-physical interactions and assess risks based on CVE

vulnerabilities, the CVSS scoring system, and threat models by constructing attack paths to determine risk levels. While the authors offer an effective asset-based approach for IoT risk assessment, it necessitates considerable knowledge of the system's assets and their potential interactions. Moreover, the authors do not provide a method for identifying CVE IoT vulnerabilities from the extensive and uncategorized list of vulnerabilities in the NVD database, which is essential for addressing the dynamic nature of IoT. This represents a limitation in identifying a comprehensive list of CVE IoT and analyzing trends of potential security weaknesses that could affect various categories of IoT components, regardless of the vendor. Such limitations pose challenges in comprehensive risk identification, especially in large-scale IoT systems with limited system knowledge. Additionally, the methodology does not consider the safety impact factor.

Sanchez et al. [42] proposed an ontology-based security risk assessment methodology for IoT environments. This framework utilizes an interoperable ontology that integrates information from various risk management methodologies, enabling dynamic adaptation to new data inputs and conditions. It offers a real-time capability to assess and manage risks in IoT environments and includes a system information module, which focuses on impact and likelihood metrics, and the assets module, which lists types of primary and supporting assets as inputs to the ontology manager. However, despite the efficiency of the proposed framework for dynamic real-time security risk assessment, both the system information module and the assets module rely on security risk assessment methodologies designed for IT organizational systems such as EBIOS, MAGERIT, MONARC, ITSRM, and CRAMM. These methodologies do not sufficiently emphasis key IoT aspects such as the intercommunication phases of coupling and bonding, the consideration of system's assets as attack platforms, resource constraints, and safety impact metrics. Indeed, by configuring and integrating this methodology with our proposed process, which takes into account key IoT aspects as inputs to the ontology manager, it could become a highly efficient tool for assessors.

Ge et al. [43] proposed a framework designed to model and assess the security of IoT systems, addressing the challenges posed by the heterogeneous nature of IoT environments. The framework utilizes the Hierarchical Attack Representation Model (HARM) in conjunction with the Symbolic Hierarchical Automated Reliability and Performance Evaluator (SHARPE) to automate the identification of potential attack paths, the assessment of security levels using proposed metrics, and evaluating the effectiveness of defense strategies. This framework efficiently automates the security analysis of IoT systems by generating attack paths and emphasizing the derivation of attack scenarios and their employment as attack platforms to some extent. However, it lacks methods for identifying security weaknesses, vulnerabilities, and attack patterns within IoT networks. This limitation is crucial for assessors conducting comprehensive risk assessments that cover various key IoT aspects, thereby restricting the analysis to the existing knowledge of the IoT network. This poses challenges in adapting to real-time changes and conducting dynamic security risk assessments in the IoT ecosystem. Furthermore, the authors do not provide a detailed analysis of the intercommunication protocols and resource constraints that can pose major security risks. Additionally, the impact of safety on risk levels is not considered.

Hassani et al. [44] introduced a novel approach that leverages the IEC 62443-3-2 and IEC 62443-4-2 standards to conduct an in-depth risk assessment, verifying the security compliance of Industrial Internet of Things (IIoT) devices and systems. It advocates the implementation of the IEC 62443 cybersecurity standard, which provides comprehensive guidelines and measures for ensuring the security and operational safety of industrial systems. However, it does not address the dynamic aspects of IoT systems for real-time security risk assessment. Additionally, the analysis overlooks dynamic attacks that involve compromising one industrial device in one zone to target devices and services in other zones of the industrial IoT system. Moreover, the safety impacts on humans and the environment are not considered, while being crucial in industrial systems.

Arat et al. [45] introduced a novel method for assessing vulnerabilities and risks within IoT systems, using a three-phase methodology: graph construction, attack path detection, and attack path filtering. This approach adopts graph-based techniques to visualize attack paths and vectors within IoT networks, applying CVE vulnerabilities from the NVD and the CVSS scoring system metrics to compute security risk levels. However, the methodology does not address the dynamic aspects of IoT systems in generating attack paths, nor does it offer structured approaches for real-time security risk assessment of CVE vulnerabilities related to the IoT domain. Additionally, the methodology overlooks the intercommunication protocols and the resource constraints of IoT networks, which may present unique security weaknesses and attack patterns. Moreover, the methodology does not account for safety impact metrics in risk level calculations, relying instead on the CVSS scoring system, originally designed for IT systems.

Duan et al. [46] proposed an automated framework for assessing the security of IoT networks, integrating machine learning (ML) and natural language processing (NLP) to predict vulnerability metric scores by training the ML model on CVE descriptions from the NVD relevant to the target system components. These scores feed into a two-layered graphical security model, consisting of an upper-layer attack graph representing network connectivity and a lower-layer attack tree detailing vulnerability information for each node to capture potential attack scenarios. The framework can identify potential attack paths and platforms originating from intercommunication processes and resource constraints through the analysis of CVE descriptions by the graphical security module. However, it fails to provide a method for conducting security assessments that abstract from the details of the target system. It requires considerable knowledge and pre-analysis of the target system's components and their associated vulnerabilities, as no efficient approaches are provided to classify security weaknesses and vulnerabilities specific to the IoT domain, posing challenges in conducting dynamic security assessments. Additionally, the framework does not consider the safety impact metric in its assessments.

George et al. [47] introduced a vulnerability-based risk as-

sessment using a multi-attacker multi-target graphical model that maps the complex relationships between attackers, targets, and existing CVE vulnerabilities within IoT networks' edge devices to generate attack path scenarios. This model assesses risks using CVSS metrics for each vulnerability, computes the likelihood of attack path scenarios, and develops corresponding mitigation strategies. However, while the methodology offers a way to generate attack scenario paths and potentially use them as platforms to conduct dynamic attacks, it lacks approaches for identifying CVE vulnerabilities while abstracting from the system's details for dynamic real-time security assessment of IoT networks. Additionally, it does not consider risks arising from IoT nodes' resource constraints and intercommunication processes, nor does it consider their impact on security attributes, business assets, and safety.

Jacobsson et al. [48] conducted a security risk assessment for smart home automation systems using the well-known Information Security Risk Analysis (ISRA) method, focusing on identifying and classifying risks through detailed analysis. The study analyzed 32 potential risks, categorizing them into low, moderate, and high risk levels based on human factors, software components, hardware components, and communication protocols. The study underscores the necessity of integrating standard security features early in the design phase to effectively mitigate these risks. The analysis includes vulnerabilities and threats that affect the intercommunication processes of protocols and resource constraints within its scope. However, the risk analysis is confined to static components of the smart home automation systems and does not address the dynamic aspects of the system. Additionally, the resultant risks do not reveal any scenarios related to the compromise of smart home connected devices and sensors used as attack platforms to target backend servers and user mobile devices. Furthermore, it overlooks the safety impact factors in risk estimation.

Mavropoulos et al. [49] proposed a comprehensive framework for security analysis of IoT systems. Recognizing the complexities of intercommunication and the dynamic nature of IoT, the framework enhances traditional security analysis methods by incorporating a new class-based notation within the *Apparatus* framework's modeling language, a security framework developed by the authors to facilitate security analysis in IoT. This adaptation captures diverse information across software, hardware, communication protocols, security, and social constructs, ensuring a holistic approach to IoT security analysis. However, while the framework addresses the dynamic and complex intercommunication of IoT systems, it lacks methods to identify security issues within the considered IoT systems. Consequently, extensive knowledge of the system's assets and security issues is required, limiting the ability to conduct risk analyses that abstract from system specifics. Additionally, the authors do not emphasize the potential use of IoT devices as platforms to conduct dynamic attacks. Moreover, the impact and likelihood metrics for risk estimation are not considered.

Table 2 summarizes the various approaches discussed in this section and the key aspects of IoT that each addresses.

4. Novel Vulnerability-oriented Security Risk Identification Framework for IoT Risk Assessment

In this section, we propose a novel security risk identification framework designed as a four-step process for IoT systems, incorporating the key aspects presented in Section 3.2.2. The first step aims to identify a list of common CWE-IoT weaknesses that potentially affect IoT components. We offer various methods for identifying such CWE, including an expert-driven approach that maps a comprehensive number of CWE weaknesses and the OWASP top ten IoT security weakness categories. The second step focuses on identifying lists of CVE-IoT vulnerabilities from the NVD. We provide in subsection 4.2 a spreadsheet table designed to identify CVE vulnerabilities based on the CWE identified in Step 1 and the affected IoT components. The third step aims to identify lists of CAPEC attack patterns that could potentially exploit the identified CVE. Similar to Step 2, we provide in subsection 4.3 a spreadsheet to identify CAPEC based on the CVE and the IoT components involved. The fourth step is dedicated to risk identification, which involves combining the previously identified weaknesses, vulnerabilities, and attack patterns through attack scenarios, as well as profiling their likelihood and severity impact factors. This process assesses impact factors on: security, business assets, and safety. The proposed process yields the following outputs:

- Lists of CWE weaknesses, CVE vulnerabilities, and CAPEC attack patterns for targeted IoT system components and functionalities (Steps 1-3),
- A list of novel attack scenarios based on CWE, CVE, and CAPEC (Step 4),
- Graded likelihood and impact risk profiling factors for security attributes of each attack scenario (Step 4),
- Graded risk impact profiling factors for business assets and safety for each attack scenario (Step 4).

In our proposed process, we omit the identification and assessment of threat sources to maintain broad applicability across diverse IoT applications, each characterized by unique threat agents. Recognizing the criticality of IoT systems in various sectors (e.g., healthcare, railways, and open environments) and their vulnerabilities to a wide spectrum of attacks, we adopt a worst-case scenario approach. This approach prepares us for diverse attacker capabilities and encourages robust *defense-in-depth* mechanisms. The process aims to design realistic attack scenarios that researchers and system designers can use to validate security measures. This helps protect applications where IoT systems are deployed, irrespective of specific threat sources.

The inputs to the process are as follows:

- A list of considered functionalities and system's components (input for all steps),
- Approaches for the identification of CWE, CVE, and CAPEC (inputs for the first three steps),

Table 2: IoT Risk Analysis Frameworks Addressing Key Aspects of IoT

Analysis Framework	Limitation (L)				
	$L_{coupling}$	L_{attack_vector}	$L_{dynamics}$	$L_{resource\_constraints}$	L_{safety}
Wang et al. [36]	✗	✓	✗	✗	✗
Casola et al. [37]	✗	✗	✗	✗	✗
Shivraj et al. [38]	✓	✓	✗	✗	✗
Kang et al. [39]	✗	✗	✗	✗	✗
Sicari et al. [40]	✓	✓	✓	✗	✗
Stellios et al. [41]	✓	✓	✗	✓	✗
Sanchez et al. [42]	✗	✗	✓	✗	✗
Ge et al. [43]	✗	✓	✗	✗	✗
Hassani et al. [44]	✓	✗	✗	✓	✗
Arat et al. [45]	✗	✓	✗	✗	✗
Duan et al. [46]	✓	✓	✗	✓	✗
George et al. [47]	✗	✓	✗	✗	✗
Jacobsson et al. [48]	✓	✗	✗	✓	✗
Mavropoulos et al. [49]	✓	✗	✓	✓	✗
Proposed Framework	✓	✓	✓	✓	✓

- Approaches for identifying attack scenarios and the corresponding grading profiling metrics for the likelihood and impact (input for Step 4),
- Cybersecurity expert knowledge is an input to Step 1 (in the case of an expert-driven approach) and Step 4.

Figure 2 illustrates a Business Process Model Notation (BPMN) diagram of the proposed process.

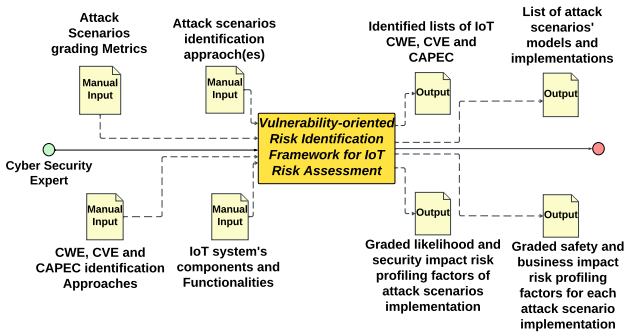

Figure 2: BNPM Diagram of the Proposed Framework

Figure 3 provides a detailed overview diagram of the steps involved in the proposed process. Each step will be detailed in the following subsections, although as previously stated Steps 1 to 3 are further elaborated in [19].

4.1. Step 1: CWE-IoT Identification

In this step, our objective is to identify common CWE weaknesses relevant to the IoT domain. To achieve this, we provide several approaches, including our proposed expert-driven, non-exhaustive mapping approach between the OWASP Top 10 categories and CWE for IoT. In addition, various other approaches in the literature could be integrated and employed to identify CWE for IoT, such as the data-driven approaches [50], model-based approaches [51], or hybrid (combination between them). This step takes the CWE identification approach as its input. The output is a list of CWE weaknesses that commonly target IoT systems, which we will focus on in subsequent steps. Note that assessors can add more CWE entries to the expert-driven mapping approach, as not all CWE entries (1,000 in total) were analyzed. Table 4 presents the proposed mapping between the OWASP Top Ten for IoT and CWE. Detailed information on each CWE is available on the MITRE Corporation's website. Detailed descriptions of each OWASP security weakness category for IoT are shown in Table 3.

4.2. Step 2: CVE-IoT Identification

The NVD database encompasses exhaustive lists of vulnerabilities across the IT, OT, and IoT domains. However, the CVE vulnerabilities are cataloged in a way that does not readily distinguish between the domains they target. They are distributed throughout the database without explicit markers indicating domain specificity. This lack of differentiation necessitates an in-depth investigation to determine whether a specific vulnerability affects a particular component within a given domain. Thus, it poses challenges for researchers attempting to isolate vulnerabilities that are specific to domain components. This motivates our proposal in this step of a cross-referenced CVE spreadsheet[11] approach to identify vulnerabilities that specifically target components within the IoT domain. We analyzed over 600 IoT-related CVE entries from the NVD database, spanning various IoT categories and their corresponding CWE weakness categories. This step utilizes the CWE identified in Step 1, in

[11] https://github.com/lounisShield/SRSEMS/blob/main/CVEIoT.pdf

Table 3: Common IoT Security Weaknesses Categorized by OWASP

OWASP IoT Weakness Category	Description
C1: Weak, Guessable, or Hard-coded Passwords	Utilizing credentials that can be easily brute-forced, are publicly accessible, or cannot be changed, as well as backdoors in firmware or client software, enables unauthorized access to deployed IoT systems.
C2: Insecure Network Services	Network services that are unnecessary or insecure and run on the IoT devices, especially those accessible via the internet that can jeopardize the confidentiality and integrity/authenticity of data. They might also enable unauthorized remote control or impact the availability of information.
C3: Insecure Ecosystem Interfaces	Insecure web, mobile, cloud, or backend API interfaces within the external ecosystems of IoT devices can allow device compromise. Common security weaknesses with these interfaces include weak encryption, improper authentication or authorization, and inadequate input and output filtering.
C4: Lack of Secure Update Mechanism	Lack of OTA update mechanisms on the IoT devices, absence of firmware validation, un-encrypted firmware updates in transit, missing anti-rollback mechanisms, and absence of notifications of security changes following updates.
C5: Use of Insecure or Outdated Components	Use of insecure or obsolete software components and libraries could allow an IoT device to be compromised by attackers exploiting known unpatched vulnerabilities. This includes insecure and outdated customized operating systems, bare-metal firmware, and third-party software or hardware components from a compromised supply chain.
C6: Insufficient Privacy Protection	Personal information stored on the IoT device or within the IoT ecosystem may be used insecurely, handled improperly, or accessed without permission.
C7: Insecure Data Transfer and Storage	Sensitive data within the IoT ecosystem lacks encryption and access control, whether it is at rest, in transit, or being processed.
C8: Lack of Device Management	Deployed IoT devices in production could lack essential security support features, such as asset management, update management, patching policies, secure decommissioning, systems monitoring, and response capabilities against security threats.
C9: Insecure Default Settings	IoT Devices or systems are often shipped with insecure default settings, or they prevent operators from modifying configurations to enhance their security.
C10: Lack of Physical Hardening	Lack of adequate physical security measures enables adversaries to gain unauthorized access to devices through insecure physical interfaces or by obtaining sensitive information by analyzing indirect information, such as power consumption or electromagnetic leaks, which could facilitate future remote attacks.

conjunction with the target system's functionalities and components, to identify relevant IoT vulnerabilities targeting specific vendor product categories. It is important to note that the spreadsheet is regularly updated as new IoT vulnerabilities are reported. Assessors can also utilize other approaches to identify IoT vulnerabilities. Precise definitions of each CVE are available on the NVD website.

4.3. Step 3: Tracing Attack Patterns

The CAPEC database provides a comprehensive mapping between CWE and CAPEC attack patterns; however, this mapping primarily targets the traditional IT domain. Therefore, to effectively identify potential risks of IoT systems, efforts should be made to re-contextualize this mapping to target the specific vulnerabilities of IoT systems. In this step, we introduce a cross-reference spreadsheet[12], akin to the one used during Step 2, which includes a list of CAPEC attack patterns for IoT, the CVE vulnerabilities associated with them, and the corresponding target IoT components. Cross-referencing is accomplished through an in-depth analysis of certain CVE vulnerabilities identified in Step 2, the affected IoT categories, and their associated potential attack patterns from the CAPEC database. The inputs for this step are the CWE identified in Step 1, the CVE from Step 2, and an attack pattern selection approach, such as the CAPEC-IoT spreadsheet. The output is a list of potential IoT CAPEC attack patterns exploitation techniques linked with CVE and CWE that could target specific system components. Efforts are underway to continuously integrate and update the spreadsheet with additional CVE and CAPEC entries, similar to Step 2.

4.4. Step 4: Identification of Attack Scenarios and Profiling of their Impact and Likelihood Factors

In this step, our goal is to identify attack scenarios derived from weaknesses, vulnerabilities, and attack patterns identified in previous steps and to profile the impact and likelihood of each attack scenario. This step is divided into three sub-

[12]https://github.com/lounisShield/SRSEMS/blob/main/CAPECIoT.pdf

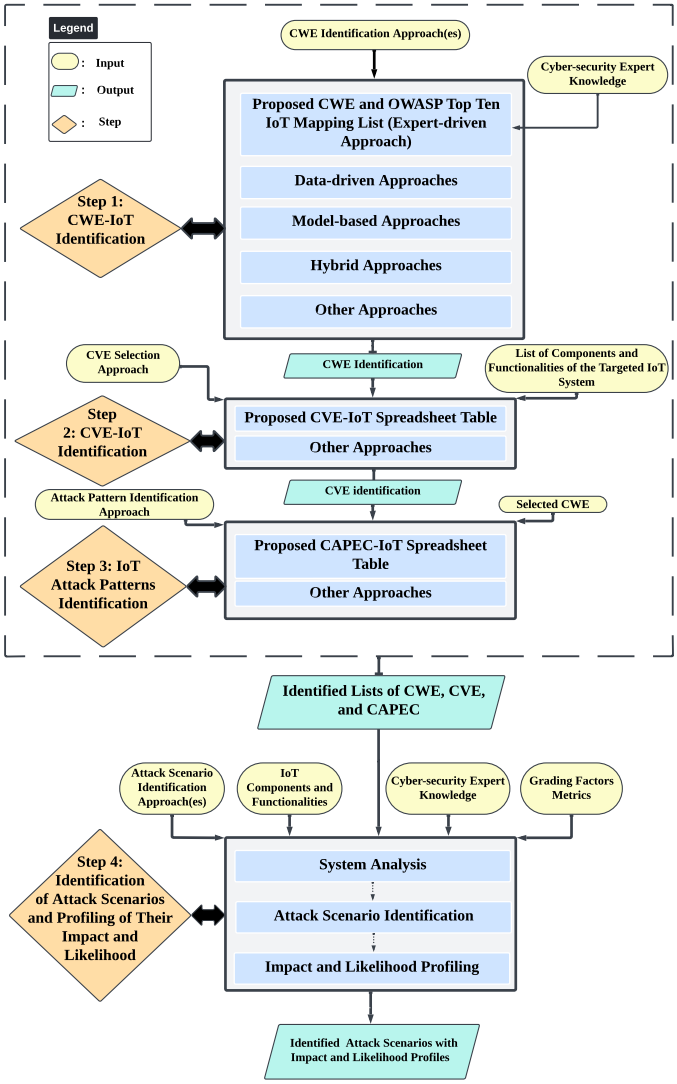

Figure 3: Four-Step Process of the Proposed Framework

modules (see Figure 3): the system analysis module, the attack scenario identification module, and the impact and likelihood profiling module. A more detailed description of each sub-module is provided in Figure 4.

4.4.1. System Analysis

This sub-module requires an understanding of the system's architectural layers, including component interconnections and network protocols specification. It utilizes security information from previous steps and information on system's functionalities and components to outline the assets associated with vulnerabilities. With this comprehensive view, cybersecurity experts can gain a better understanding of the system's weaknesses, vulnerabilities, and applicable attack patterns. This baseline facilitates the subsequent tracing of verified IoT or theoretical IoT attack scenarios.

4.4.2. Attack Scenario Identification

This sub-module employs various approaches to define diverse attack scenarios targeting the designated IoT system. Cybersecurity experts can manually combine CWE, CVE, and attack patterns to generate attack scenario pattern models (based on CWE and CAPEC) and their implementations (CVE and CAPEC). In addition, threat modeling methods such as the STRIDE model, attack tree analysis, or automated tools (with security experts' post-verification) can be integrated into our process to derive IoT-specific attack scenarios. While experts have the flexibility to device their own methods for defining realistic IoT attack scenarios, this module primarily leverages the security information identified in the previous steps and the baseline data from the system analysis module.

Table 4: Mapping between OWASP Top 10 IoT and CWE (Expert-driven Approach)

OWASP IoT Weakness Categories	CWE-IoT
C1: Weak, Guessable, or Hard-coded Passwords	CWE-261, CWE-260, CWE-521, CWE-259, CWE-257, CWE-798, CWE-522, CWE-321,CWE-256, CWE-523, CWE-307, CWE-640, CWE-255, CWE-345, CWE-287, CWE-257
C2: Insecure Network Services	CWE-287, CWE-276, CWE-255, CWE-522, CWE-269, CWE-295, CWE-120, CWE-20, CWE-598, CWE-419, CWE-22, CWE-434, CWE-1331, CWE-417, CWE-444, CWE-288, CWE-732, CWE-285, CWE-326, CWE-294, CWE-319, CWE-362, CWE-367, CWE-347, CWE-306, CWE-434, CWE-295, CWE-200, CWE-674, CWE-284, CWE-668, CWE-476, CWE-787, CWE-617, CWE-401, CWE-544, CWE-125, CWE-354, CWE-331
C3: Insecure Ecosystem Interfaces	CWE-79, CWE-20, CWE-89, CWE-377, CWE-427, CWE-352, CWE-650, CWE-287, CWE-327, CWE-601, CWE-598, CWE-307, CWE-284, CWE-319, CWE-77, CWE-78, CWE-119, CWE-295, CWE-311, CWE-325, CWE-94, CWE-125, CWE-787, CWE-416, CWE-306, CWE-112, CWE-862, CWE-427, CWE-94, CWE-330, CWE-294, CWE-322, CWE-290, CWE-434, CWE-284, CWE-918, CWE-544, CWE-200, CWE-121
C4: Lack of Secure Update Mechanism	CWE-940, CWE-15, CWE-1277, CWE-404, CWE-20
C5: Use of Insecure or Outdated Components	CWE-787, CWE-119, CWE-1233, CWE-1104, CWE-327, CWE-328, CWE-398, CWE-563, CWE-686, CWE-399, CWE-190, CWE-226, CWE-1240, CWE-693, CWE-415, CWE-476, CWE-829, CWE-334, CWE-347, CWE-306, CWE-672, CWE-295, CWE-284, CWE-120, CWE-20, CWE-125, CWE-674
C6: Insufficient Privacy Protection	CWE-359, CWE-200, CWE-295, CWE-311, CWE-312, CWE-325, CWE-326, CWE-327,
C7: Insecure Data Transfer and Storage	CWE-201, CWE-300, CWE-310, CWE-200, CWE-319, CWE-668, CWE-377, CWE-327, CWE-521, CWE-922, CWE-1240, CWE-388, CWE-323, CWE-330, CWE-326, CWE-78
C8: Lack of Device Management	CWE-909, CWE-910, CWE-920, CWE-770
C9: Insecure Default Settings	CWE-15, CWE-276, CWE-1068, CWE-269, CWE-521, CWE-1189, CWE-1231, CWE-1260, CWE-1262, CWE-1274, CWE-287
C10: Lack of Physical Hardening	CWE-1233, CWE-284, CWE-831, CWE-134, CWE-256, CWE-119, CWE-121, CWE-400, CWE-1300, CWE-1191, CWE-1244, CWE-1247, CWE-1256,- CWE-1332, CWE-1255, CWE-1384, CWE-1319, CWE-1278, CWE-1351

4.4.3. Impact and Likelihood Profiling

This sub-module incorporates inputs from the attack scenario identification module, qualitative likelihood and impact grading

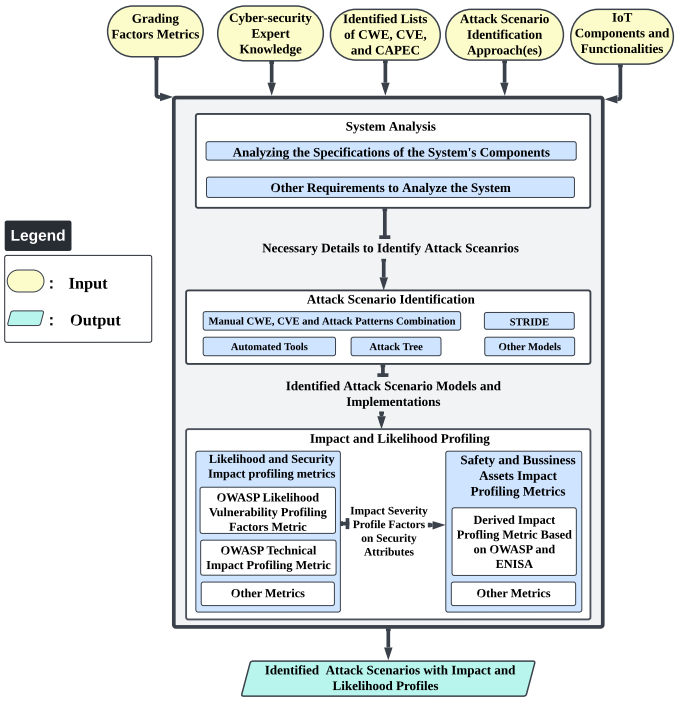

Figure 4: Modules of the Process's Fourth Step

metrics, security data from earlier steps, and experts' knowledge to assess and provide severity impact and likelihood profiles of each attack scenario. It is divided into two components: the attack scenario likelihood and security impact metrics, and the safety and business asset impact metrics. The former is used to evaluate the likelihood and severity impact profiles of attack scenarios on primary security attributes, while the latter evaluates the impact on business assets and safety in the IoT application. The evaluation begins with the outcomes of the first component, whose outcomes then serve as input for the second component, to further evaluate the impact on safety and business assets based on security attributes severity impact. Our process adopts the OWASP risk rating methodology[13] metrics for the first component to assess the likelihood and impact of attack scenarios on security attributes. Tables 5 and 6 detail the OWASP security impact and likelihood metrics used. In the second component, although the OWASP risk rating methodology provides metrics to evaluate business asset impact, it overlooks safety impact factors, which are crucial for some IoT applications. To address this problem, we have extended and integrated OWASP's business metrics with ENISA's[14] safety impact metric concerning user safety. ENISA's risk assessment method enables IoT users to evaluate the impact of a security breach on individual safety and to implement appropriate countermeasures. Furthermore, we introduce a distinct safety metric that accounts for the environmental damage factor, considering that the attacker's goal may involve material sabotage. Although environmental damage and human safety are often consolidated into the global safety attribute, we have divided it into two scales, each rated from 0 to 9: one for human safety and the other for environmental damage. In addition, we have revised the privacy violation grading metric and adapted it to evaluate personal data breaches based on their significance and the number of affected users, a measure that is more applicable to IoT contexts. This approach differs from the original assessment proposed by OWASP for IT systems, which emphasized the number of users affected by privacy violations resulting from a security breach. The severity, impact, and likelihood profiles can serve as inputs for modules dedicated to risk estimation and prioritization in IoT risk assessments. Table 7 presents the factors of the profiling metrics used to evaluate the impact on safety and business assets. The severity impact and the likelihood profiles for attack scenarios are determined by the vectors depicted in Figure 5.

[13] https://owasp.org/www-community/OWASP_Risk_Rating_Methodology

[14] https://www.enisa.europa.eu/publications/guidelines-for-smes-on-the-security-of-personal-data-processing

Technical Impact (TI)

$$\begin{pmatrix} Loss\ of\ Confidentiality \\ Loss\ of\ Integrity \\ Loss\ of\ Availability \end{pmatrix}$$

Likelihood of Vulnerability Exploitation (LVE)

$$\begin{pmatrix} Ease\ of\ Discovery \\ Ease\ of\ Exploit \\ Awarness \\ Intrusion\ Detection \end{pmatrix}$$

Impact on Safety and Business Assets (ISBA)

$$\begin{pmatrix} User\ Safety \\ Environmental\ Damage \\ Financial\ Damage \\ Reputation\ Damage \\ Non-Compliance \\ Privacy\ Violation \end{pmatrix}$$

Figure 5: Grading Metrics Vectors

Table 5: Technical Impact Grading Factors on Primary Security Attributes by OWASP

Loss of Confidentiality	Loss of Integrity	Loss of Availability
Minimal non-sensitive data disclosed (2)	Minimal slightly corrupt data (1)	Minimal secondary services interrupted (1)
Minimal critical data disclosed (6)	Minimal seriously corrupt data (3)	Minimal primary services interrupted (5)
Extensive non-sensitive data disclosed (6)	Extensive slightly corrupt data (5)	Extensive secondary services interrupted (5)
Extensive critical data disclosed (7)	Extensive seriously corrupt data (7)	Extensive primary services interrupted (7)
All data disclosed (9)	All data totally corrupt (9)	All services completely lost (9)

Table 6: Likelihood Vulnerability Grading Metric Based on OWASP

Ease of Discovery	Ease of Exploit	Awareness	Intrusion Detection
Theoratically impossible (0)	Impossible (0)	N/A	N/A
Practically impossible (1)	Theoretical (1)	Unknown (1)	Active detection in application (1)
Difficult (3)	Difficult (3)	Hidden (4)	Logged and reviewed (3)
Easy (7)	Easy (5)	Obvious (6)	Logged without review (8)
Automated tools available (9)	Automated tools available (9)	Public knowledge (9)	Not logged (9)

Table 7: Derived Grading Impact Metric on Safety and Business Assets

Safety Impact		Impact on Business Assets			
Human Safety	**Environmental Damage**	**Financial Damage**	**Reputation Damage**	**Non-Compliance**	**Privacy Violation**
Individual will not encounter inconveniences (0)	No impact (0)	No impact (0)	No impact (0)	No impact (0)	No personal data exposed (0)
Significant inconvenience without cause of injuries (3)	Minor environmental damage (3)	Less than the cost to fix the vulnerability (1)	Minimal damage (1)	Minor violation (2)	Exposure of limited raw personal data for few people (1)
Minor injuries (5)	Moderate environmental damage (5)	Minor effect on annual profit (3)	Loss of major accounts (4)	Clear violation (5)	Exposure of extensive raw personal data for tens or limited significant personal data for a few people (3)
Major injuries (7)	Major environmental damage (7)	Significant effect on annual profit (7)	Loss of goodwill (5)	High profile violation (7)	Exposure of significant personal data for tens or extensive raw personal data for hundreds of people (7)
Death (9)	Significant environmental damage (9)	Bankruptcy (9)	Brand damage (9)	No regards to security rules (9)	Exposure of extensive raw personal data for thousands or significant personal data for hundreds of people (9)

5. Addressing the Limitations of Traditional Risk Assessment Methods

This section discusses how the proposed process addresses the key IoT aspects for comprehensive risk identification that traditional security risk assessment methods fail to consider.

5.1. $L_{coupling}$

By integrating common CWE for IoT identified in Step 1 with targeted IoT components and protocols (such as gateways, sensors, actuators, and short-range wireless protocols) listed in the spreadsheet from Step 2, we facilitate the identification of potential real-world CVE-IoT vulnerabilities that target protocol implementations. Security experts can use the CWE and CVE definitions provided by their respective databases to analyze IoT-targeted CWE weaknesses and CVE vulnerabilities. This analysis helps extract preliminary key insights, opening avenues for investigating the coupling intercommunication specifications where security weaknesses and vulnerabilities are most prevalent. Synthesizing this information with the system analysis module in Step 4 allows for a comprehensive understanding of the processes and phases of device communications.

5.2. L_{attack_vector}

By identifying lists of CWE-IoT, CVE-IoT, and CAPEC-IoT that target each considered functionality and its relevant components, security experts can analyze potential attack scenarios in which a threat event takes control of a device or crafts malicious messages relayed into the IoT system. Further attack scenarios can then be developed on these compromised devices, using a previous attack scenario as a preliminary step.

5.3. $L_{dynamics}$

Our approach tackles the dynamic limitation in two ways. First, through Steps 1 and 2, the analyst can target CWE relevant to the dynamics of an IoT network, such as CWE-287: Improper Authentication and CWE-295: Improper Certificate Validation, which can arise from the mobility of IoT devices as they connect and disconnect within the same network or transition from one network to another. Second, the spreadsheet enables security experts to identify CWE typically associated with specified IoT components, based on the number of real-world CVE reported for those components, regardless of their versions or vendors. Thus, the spreadsheet can identify predictive risks for specific IoT devices and aids in real-time security assessments by abstracting from system details as the IoT domain evolves. Then, when any IoT device enters or leaves the system, the spreadsheet provides a history and prevalence of potential weaknesses or vulnerabilities that can be directly used to assess the category of the added device, facilitating the analysis of potential emerging risks. Table 8 presents an example of potential CWE-IoT security weakness trends affecting the deployment of IoT components, along with the number of relevant CVE, extracted from the proposed spreadsheet.

Table 8: Trends of CWE-IoT Security Weaknesses Potentially Affecting IoT Component Deployment

IoT Component	CWE-IoT Trend	CVE-IoT Number
Smart Hub Controllers	CWE-119: Improper Restriction of Operations within the Bounds of a Memory Buffer	41
	CWE-787: Out-of-bounds Write	10
	.....	.....
Bluetooth Low Energy (BLE) Protocol Implementation	CWE-20: Improper Input Validation	11
	CWE-287: Improper Authentication	6
	.....	.....
Medical Insulin Pumps	CWE-287: Improper Authentication	3
	CWE-522: Insufficiently Protected Credentials	2
	.....	.....
.....	.....	.....

5.4. $L_{resource\_constraints}$

The approaches provided in Step 1 are dedicated to identifying common weaknesses in the IoT context, taking into account potential weaknesses that have emerged due to the lack of robust security mechanisms and resource constraints. For instance, our proposed expert-driven mapping between the OWASP top ten IoT weakness categories and CWE includes security weaknesses related to lack of encryption, such as CWE-326: Inadequate Encryption Strength; CWE-327: Use of a Broken or Risky Cryptographic Algorithm; CWE-319: Cleartext Transmission of Sensitive Information; CWE-311: Missing Encryption of Sensitive Data; and improper authentication, such as CWE-287: Improper Authentication; and CWE-294: Authentication Bypass by Capture-Replay. Thus, by cross-referencing the IoT components with the identified CWE weaknesses in the CVE-IoT spreadsheet, we can identify a wide range of CWE weaknesses and CVE vulnerabilities

that stem from resource constraints and the implementation of lightweight protocols.

5.5. L_{safety}

By incorporating safety metrics alongside other impact metrics in Step 4, such as user safety and environmental damage, we provide a means to account for the safety factor in risk estimation and prioritization within IoT risk assessments.

6. Proposed Process's Case Study: IoT Smart Healthcare Application

We validate our process through a case study on a potential hybrid IoT smart healthcare system implementation. Such systems integrate various smart devices that communicate with an IoT gateway, enabling real-time monitoring and automation to enhance patient care with features like remote health monitoring and controlled medication delivery.

6.1. System Workflow

Our case study draws inspiration from the works of [52, 53, 54, 55, 56, 57], and is a typical architecture used in smart healthcare applications. It includes three battery-powered biosensor nodes deployed on the patient to continuously monitor temperature, pressure, and heart rate. These sensors communicate with the gateway, which implements a BLE client via the BLE protocol. This protocol has been selected for its energy efficiency, crucial in scenarios of prolonged battery life for resource-constrained sensor nodes [58]. The Message Queuing Telemetry Transport (MQTT) protocol, noted for its low power and bandwidth requirements, facilitates communication between the IoT gateway and the server side. The MQTT protocol communication comprises two key transitions [59]. In the first transition, the IoT gateway serves as the MQTT publisher client, transmitting sensor data to the MQTT broker. In the second transition, the MQTT broker forwards the data to two possible MQTT client subscribers. The server-side MQTT client subscriber is responsible for inserting this data into SQL tables. Simultaneously, medical staff can monitor patients' vital signs in real-time through a web user interface that displays the data received from the MQTT broker via a second MQTT client subscriber (e.g., a web program running on a Node.js server that implements an MQTT client) in a web browser [60]. Data stored in the database is utilized to generate various reports on patients' medical data for the medical staff, update patient information, and facilitate data analysis and visualization through charts or graphs of patients' medical histories on a web server. Communication between the web server and the user domain are supposed to be secure as traditional security techniques can be implemented to ensure integrity and confidentiality. For this reason, the web server network is excluded from the scope of our study. However, we will examine potential consequences of attacks on the IoT network as platforms for attacking the web server network. Table 9 summarizes the system's functionalities and the components related to the IoT network side.

Table 9: System's Functionalities and Components

Functionality	Components
Collecting vital measurements from the patient (F_1)	Pressure sensor (BLE slave), Temperature sensor (BLE slave), Heart rate sensor (BLE slave), IoT gateway device (BLE master)
Publishing sensor data to the application domain (F_2)	IoT gateway (MQTT client), MQTT broker
Real-time monitoring (F_3)	MQTT broker, Web browser
Storing data at the database (F_4)	MQTT broker, MQTT client, MySQL Server

The system's components and data flow were projected onto the ISO IoT architecture to comply with standardized frameworks for component interaction and to enhance the generalizability of our work. Figure 6 illustrates the system's use-case ecosystem.

6.2. Validation of the Proposed Process for the IoT Smart Heathcare system

This subsection validates our proposed process by applying it to the healthcare system presented earlier. It encompasses key steps, including the identification and assessment of CWE and CVE in IoT through the proposed OWASP/CWE mapping expert-driven approach, focusing on weakness categories that pose high criticality to IoT systems, along with their associated CAPEC attack patterns. We then focus on the BLE network within the sensing and control domain, deriving potential attack scenario models and implementations in BLE networks, and demonstrating how these scenarios could be utilized as dynamic attack platforms for launching further attacks on the application and user domains. Subsequently, we profile the impact factors of two BLE attack scenario implementations on the hospital's security attributes, patient safety, environmental impact, and business assets, including their likelihood of exploitation.

6.2.1. CWE Identification (Step 1)

In the initial step of our process, we select a comprehensive list of common CWE weaknesses pertinent to our case study. This selection utilizes our proposed expert-driven mapping of the OWASP Top Ten categories with CWE (see Table 4). From this table, we choose to focus on the top three OWASP categories that align with the unique challenges of IoT. Additionally, we consider the CWE weaknesses related to the absence of an update mechanism (C4) and the use of insecure or outdated components (C5), as they could be primary factors in the emergence of weaknesses in other OWASP categories. By doing so, we concentrate on areas of the highest risk that require minimal effort, offering a trade-off between identifying critical risks and validating our process in addressing the key IoT aspects in risk identification. It is important to note that assessors can employ more exhaustive approaches, such as data-driven methods, to study all possible risk analyses.

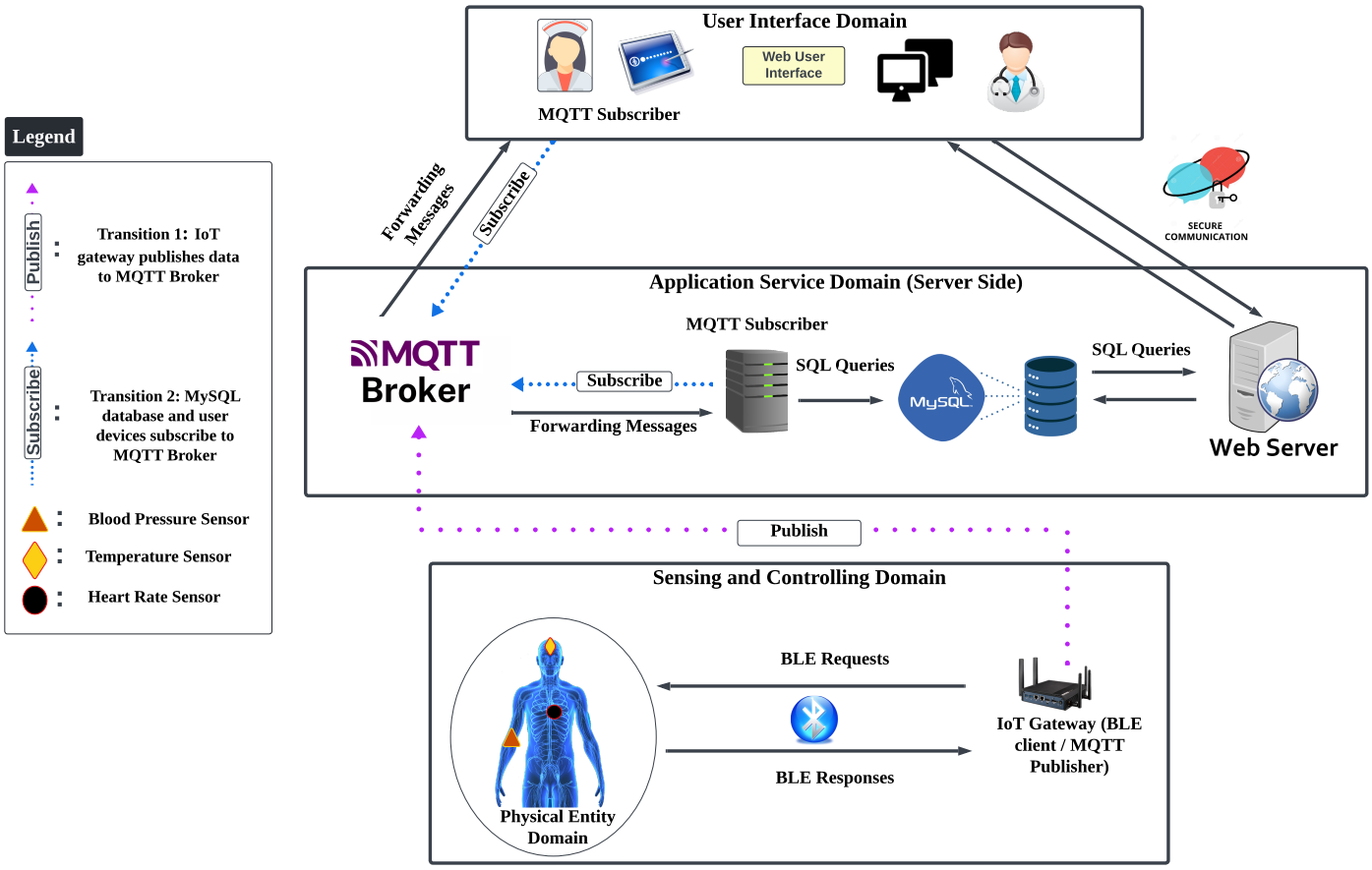

Figure 6: IoT Healthcare Application based on the ISO IoT Architecture

6.3. *CVE Identification (Step 2)*

In the second step, we classify CVE targeting the system's use cases by correlating the selected CWE from Step 1 with the system's components as listed in the CWE/CVE IoT spreadsheet. The CVE identified in this step are presented in Table 11, associated with corresponding use cases that are synthesized from the functionalities and system's components listed in Table 9. Referring to Table 11, we can observe, for example, various vulnerabilities targeting the BLE implementation include improper input validation (CWE-20), improper authentication (CWE-287), and cleartext transmission of sensitive data (CWE-319). These weaknesses highlight risks from device dynamics, resource constraints, and complex intercommunication that pave the way for investigations in subsequent steps. Note that the use cases *UC-3* and *UC-4* share the same weaknesses and vulnerabilities, as both are related to the communication between the MQTT broker and MQTT clients, one in the user domain and the other in the application domain. In addition, analysts can perform real-time risk analysis on any device or network added to the healthcare system by referring to the device or protocol category and its relevant CWE selected in the first step to pinpoint potential CVE vulnerabilities. This approach offers an efficient way to identify risks while abstracting from system details. For example, if insulin pumps, infusion pumps, medical ventilators, etc., are added temporarily to the healthcare system to administer medication to a patient, we refer directly to the 'Medical Devices and Software' category in the spreadsheet to obtain a list of weaknesses and vulnerabilities that could affect the dynamic deployment of these devices in the healthcare system. Table 10 illustrates some CWE weaknesses from the first three OWASP categories and their corresponding CVE vulnerabilities that could target the dynamic deployment of insulin pumps, infusion pumps, and medical ventilators.

6.4. *Tracing Attack Patterns (Step 3)*

In the third step, we analyze attack patterns relevant to the previously identified CVE for the considered use case by cross-referencing them with system component categories in the CVE/CAPEC IoT spreadsheet. We investigate potential attack patterns associated with the identified CVE, which are summarized in Table 12. For details on each CAPEC attack pattern, please refer to the CAPEC database.

6.5. *Identification of Attack Scenarios and Profiling of Their Impact and Likelihood on Use Case 1 (Step 4)*

In this subsection, utilizing security information gleaned from previous steps of the BLE network (*UC-1*), we identify potential attack scenarios that focus on various security weaknesses. We examine relationships between these weaknesses to construct more comprehensive attack scenarios, guided by

Table 10: Potential CWE Weaknesses and CVE Vulnerabilities Affecting the Dynamic Deployment of Medical Devices in Healthcare Systems

Dynamic Component	CWE	CVE
Insulin Pump	CWE-287: Improper Authentication	CVE-2019-10964, CVE-2020-27266, CVE-2018-14781
	CWE-522: Insufficiently Protected Credentials	CVE-2020-27258, CVE-2020-27270,
	CWE-319: Cleartext Transmission of Sensitive Information	CVE-2016-5084, CVE-2018-10634,
	CWE-330: Use of Insufficiently Random Values	CVE-2020-27264, CVE-2016-5085,
	CWE-290: Authentication Bypass by Spoofing	CVE-2020-27276
	....	....
Infusion Pump	CWE-319	CVE-2020-12037, CVE-2020-12036
	CWE-306: Missing Authentication for Critical Function	CVE-2022-26394, CVE-2017-12720
	CWE-311: Missing Encryption of Sensitive Data	CVE-2022-26390,
	....	....
Medical Ventilator	CWE-200: Exposure of Sensitive Information	CVE-2020-27290
	CWE-112: Cleartext Storage of Sensitive Information	CVE-2020-27282
	CWE-798: Use of Hard-coded Credentials	CVE-2020-27278
	....	....
....	....	....

Table 11: CVE-IoT Identification

Use Case (UC)-ID	OWASP IoT Category: CWE-IoT	CVE-IoT
UC-1: Sensors' data transmission from the Biosensor nodes (BLE slaves) to the IoT gateway (BLE client)	C2/C5: CWE-787/CWE-119	CVE-2023-28116 (V1), CVE-2020-10061 (V2), CVE-2019-13916 (V3)
	C2: CWE-287	CVE-2020-15486 (V4), CVE-2020-25183 (V5), CVE-2019-19194 (V6)
	C2: CWE-311	CVE-2018-10825 (V7), CVE-2020-11957 (V8)
	C2: CWE-78	CVE-2020-27373 (V9)
	C2: CWE-319	CVE-2020-11539 (V10)
	C2: CWE-120/CWE-20	CVE-2019-16336 (V11), CVE-2019-17520 (V12), CVE-2019-17061 (V13), CVE-2019-17060 (V14), CVE-2019-17519 (V15), CVE-2019-17517 (V16), CVE-2019-19196 (V17), CVE-2019-17518 (V18), CVE-2020-15531 (V19), CVE-2020-15532 (V20), CVE-2019-15948 (V21)
	C2: CWE-20	CVE-2019-19192 (V22), CVE-2019-19195 (V23), CVE-2020-13594 (V24), CVE-2019-19193 (V25), CVE-2022-45192(V26)
	C2: CWE-354/CWE-20	CVE-2022-45191 (V27)
	C2: CWE-617/CWE-20	CVE-2020-13595 (V28)
	C2: CWE-347	CVE-2020-13593 (V29)
	C2: CWE-125/CWE-20	CVE-2020-11114 (V30)
	C2: CWE-306	CVE-2022-45190 (V31)
	C2: CWE-287/CWE-20	CVE-2020-9770 (V32)
	C2: CWE-294	CVE-2022-25836 (V33), CVE-2020-35473 (V34)
	C2: CWE-287	CVE-2020-26558 (V35), CVE-2020-15802 (V36)
	C2: CWE-362	CVE-2021-31615 (V37)
	....	....
UC-2: Sensors' data transmission between the IoT gateway (MQTT publisher) and MQTT broker	C2: CWE-20	CVE-2024-26001 (V38), CVE-2021-21968 (V39)
	C3/C2: CWE-119/CWE-121	CVE-2018-19417 (V40)
	C3/C2: CWE-787	CVE-2021-41036 (V41), CVE-2018-8531 (V42), CVE-2021-45933 (V43)
	C2: CWE-401	CVE-2021-34431 (V44)
	C2: CWE-476	CVE-2017-2893 (V45)
	C1: CWE-798	CVE-2023-33372 (V46)
	C2: CWE-287	CVE-2023-3028 (V47)
	C3/C2: CWE-544	CVE-2023-29105 (V48)
	C2: CWE-674	CVE-2019-11779 (V49)
	....	....
UC-3: Sensors' data transmission between of the MQTT broker and the users' web browser via the MQTT client (subscriber)	C2: CWE-306	CVE-2023-49115 (V50)
	C3: CWE-79	CVE-2020-13821 (V51), CVE-2020-13932 (V52), CVE-2022-35612 (V53), CVE-2020-1941 (V54)
	C3: CWE-352	CVE-2022-35611 (V55)
	C2: CWE-119/CWE-121	CVE-2018-17614 (V56)
UC-4: Storing data from the MQTT broker to MySQL database via the MQTT client (subscriber)	....	....
....	....	....

an analysis of BLE core specifications. We then define attack scenario models that integrate CWE and CAPEC, outlining their potential implementations derived from CVE and CAPEC. These attack scenarios emerge from the intercommunication coupling processes of BLE nodes, dynamic mobility, and resource constraints. Furthermore, we derive dynamic attack scenarios using the BLE spoofing attack scenario of the sensor node as a platform, combined with the identified weaknesses and vulnerabilities in the user domain (*UC-3*) and application domains (*UC-2* and *UC-4*), aiming to launch critical attack scenarios. Subsequently, we select two attack scenario implementations for a detailed analysis of their severity impact and likelihood of exploitation.

6.5.1. System Analysis

We analyzed the BLE core specification [61] to understand the primary principles of this communication protocol. Figure 7 illustrates the message exchange, including connection parameters (see Table 13), in a simplified BLE packet exchange scenario involving sensor nodes and an IoT gateway in a star topology. This analysis helps us explore the coupling process inherent to the BLE protocol. It is noteworthy that the pairing procedure is optional and may be omitted depending on the protocol implementation. A similar analysis could be conducted for MQTT communication or other considered proto-

Table 12: CAPEC-IoT Attack Patterns Identification

CVE-IoT	CAPEC-IoT
CVE-2023-28116 (V1)	CAPEC-100
CVE-2020-10061 (V2)	CAPEC-100, CAPEC-130, CAPEC-153
CVE-2019-13916 (V3)	CAPEC-100, CAPEC-153
CVE-2020-15486 (V4)	CAPEC-94 , CAPEC-148, CAPEC-155, CAPEC-151, CAPEC-593
CVE-2020-25183 (V5)	CAPEC-114, CAPEC-151, CAPEC-148
CVE-2019-19194 (V6)	CAPEC-114, CAPEC-100
CVE-2018-10825 (V7)	CAPEC-15, CAPEC-148, CAPEC-593
CVE-2020-11957 (V8)	CAPEC-157, CAPEC-94, CAPEC-667, CAPEC-112, CAPEC-102
CVE-2020-27373 (V9)	CAPEC-88, CAPEC-115
CVE-2020-11539 (V10)	CAPEC-155, CAPEC-94, CAPEC-148
CVE-2019-16336 (V11)	CAPEC-158, CAPEC-130, CAPEC-153, CAPEC-231, CAPEC-100, CAPEC-26
CVE-2019-17520 (V12)	CAPEC-153, CAPEC-157
CVE-2019-17061 (V13)/CVE-2019-17060 (V14)	CAPEC-157, CAPEC-100, CAPEC-25, CAPEC-153
CVE-2019-17519 (V15)	CAPEC-157, CAPEC-100, CAPEC-231, CAPEC-153
CVE-2019-17517 (V16)	CAPEC-100, CAPEC-153, CAPEC-157
CVE-2019-19196 (V17)	CAPEC-100, CAPEC-114, CAPEC-157, CAPEC-153
CVE-2019-17518 (V18)	CAPEC-100, CAPEC-231, CAPEC-157, CAPEC-153
CVE-2020-15531 (V19)/CVE-2020-15532 (V20)/CVE-2019-15948 (V21)/CVE-2020-11114 (V30)	CAPEC-157, CAPEC-100, CAPEC-153, CAPEC-248
CVE-2019-19192 (V22)	CAPEC-100, CAPEC-25, CAPEC-157, CAPEC-125, CAPEC-231
CVE-2019-19195 (V23)	CAPEC-157, CAPEC-231, CAPEC-100, CAPEC-153
CVE-2020-13594 (V24)/CVE-2019-19193 (V25))/CVE-2022-45191 (V27)	CAPEC-157, CAPEC-153
CVE-2022-45192 (V26)	CAPEC-157, CAPEC-152, CAPEC-153
CVE-2020-13595 (V28)	CAPEC-157, CAPEC-153
CVE-2020-13593 (V29)	CAPEC-115, CAPEC-157, CAPEC-153, CAPEC-94
CVE-2022-45190 (V31)	CAPEC-94, CAPEC-667, CAPEC-115
CVE-2020-9770 (V32)	CAPEC-115,CAPEC-153
CVE-2022-25836 (V33)	CAPEC-157, CAPEC-114, CAPEC-94, CAPEC-112
CVE-2020-35473 (V34)	CAPEC-157, CAPEC-94
CVE-2020-26558 (V35)/CVE-2020-15802 (V36)	CAPEC94, CAPEC-157, CAPEC-114, CAPEC-667
CVE-2021-31615 (V37)	CAPEC-667, CAPEC-157, CAPEC94, CAPEC-248, CAPEC-26, CAPEC-37
CVE-2024-26001 (V38)	CAPEC-153, CAPEC-10, CAPEC-100, CAPEC-47
CVE-2021-21968 (V39)	CAPEC-100, CAPEC-94, CAPEC-165
CVE-2018-19417 (40), CVE-2021-41036 (V41), CVE-2018-8531 (V42)	CAPEC-242, CAPEC-153, CAPEC-100
CVE-2021-45933 (V43)	CAPEC-92, CAPEC-100
CVE-2021-34431 (V44)	CAPEC-100, CAPEC-153
CVE-2017-2893 (V45)	CAPEC-129
CVE-2023-33372 (V46)	CAPEC-151, CAPEC-560, CAPEC-115
CVE-2023-3028 (V47)	CAPEC-115, CAPEC-255, CAPEC-153
CVE-2023-29105 (V48)	CAPEC-153, CAPEC-230
CVE-2019-11779 (V49)	CAPEC-231
CVE-2023-49115 (V50)	CAPEC-62
CVE-2020-13821 (V51), CVE-2020-13932 (V52), CVE-2022-35612 (V53), CVE-2022-35611 (V54), CVE-2020-1941 (V55)	CAPEC-63
CVE-2018-17614 (56)	CAPEC-242, CAPEC-153, CAPEC-66
....	....

cols to examine the inherent intercommunication processes' in these protocols, thereby identifying possible attack scenarios in the attack scenario identification module.

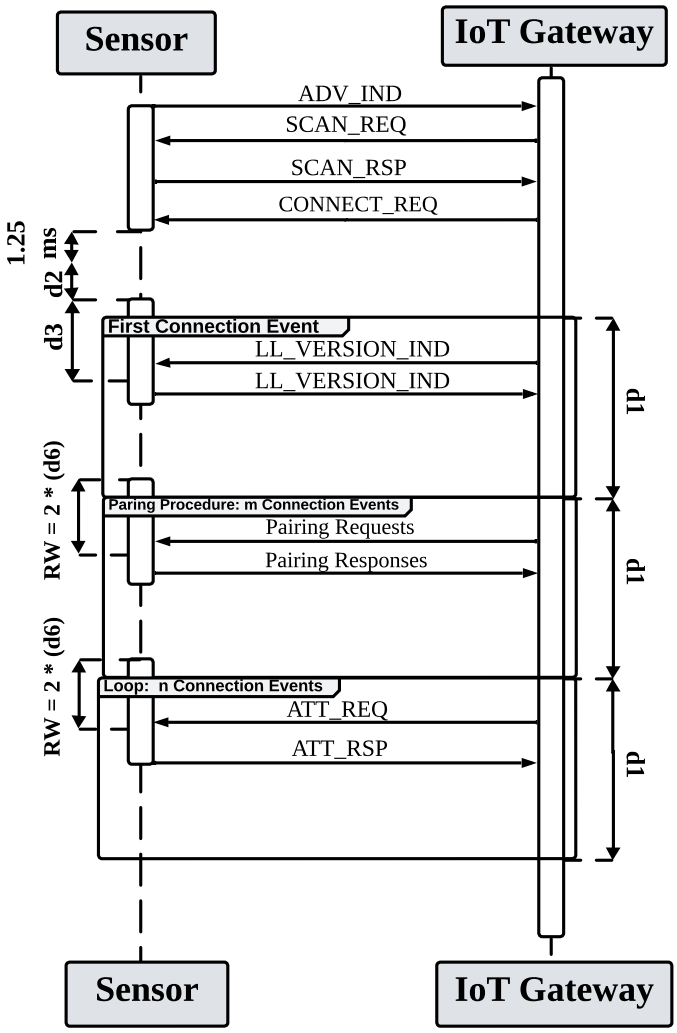

Figure 7: UML Sequence Diagram for Nominal Packet Exchanges in the BLE Protocol.
ADV_IND: Advertising Packets, **SCAN_REQ/RSP:** Scan Request/Response, **CONNECT_REQ:** Connection Request, **LL_VERSION_IND:** Version Indication Packet, **ATT_REQ/RSP:** Attribute Request/Response, **d1:** connInterval, **d2**: transmitWindowOffset, **d3:** transmitWindowSize, **d6:** windowWidening, **RW:** Receive Window.

6.5.1.1. Overview

A BLE stack consists of two primary components: the *Host*, containing high-level layers, and the *Controller*, with lower-level layers. BLE operates on 40 2-MHz channels within the ISM 2.4–2.5 GHz band. Channels 37, 38, and 39 are dedicated for *advertising mode*, broadcasting packets, while channels 0 to 36 are for the *connected mode*, transmitting *Protocol Data Unit (PDU)* data and control packets. A typical BLE packet includes *Preamble*, *Access Address*, *PDU*, and *Cyclic Redundancy Check (CRC)* fields.

6.5.1.2. Link Layer Specifications

Before sending the connection request (CONNECT_REQ) by the IoT gateway, the peripheral does not set any connection parameters (no connection events). Thus, there are two states:

1. *Advertising State:* This state involves the exchange of

advertising channel PDU packets, including *Advertising* packets (ADV_IND), potentially scan requests (SCAN_REQ) and responses (SCAN_RSP), followed by a CONNECT_REQ between the sensor node and the gateway, all without timing constraints or connection management.

2. *Connection State:* Upon the gateway's connection request, message exchanges between the sensor node and the gateway adhere to a specific timing schedule set by the connection request PDU fields. These fields dictate the timing of the first and subsequent connection events between the *Master* and the *Slave*, involving *Data Channel PDU* packets, which are divided into two categories:
    - *Link Layer (LL) Data PDU packets:* These are used to send L2CAP data, e.g., ATT_REQ/RSP (attribute requests and responses), etc.
    - *Link Layer (LL) Control PDU packets:* These are used to control the link layer connection, e.g., LL_TERMINATE_IND (terminate connection), LL_VERSION_IND (BLE controller's version packet), LL_CONNECTION_UPDATE_REQ (update connection parameters request), etc.

The *CONNECT_REQ PDU* comprises a *Header* and a *Payload*, which includes three fields: Initiator's Address (InitA), Advertiser's Address (AdvA), and the Link Layer Data field (LLData). The LLData is composed of ten fields, six of which are particularly significant for emulating the BLE protocol in our scenarios in Section 7. Table 13 presents these six fields, along with their respective parameters and range values.

- *Interval field:* This field indicates the *Connection Interval* (connInterval), denoting the time interval between two successive connection events.
- *WinOffset field:* This field specifies the *transmitWindowOffset* value, representing the time offset between the transmission of the connection request and the actual opening of the *Transmit Window*.
- *Winsize field:* This field indicates the *transmitWindowSize* value, which defines the *Transmit Window* duration. Specifically, it represents the time span during which the slave listens to the master's first packet (first anchor point) following the initiation of the connection request.
- *Timeout field:* This field indicates the *connSupervisionTimeout* value, defining the maximum allowable duration between two received *Data packet PDU* before the connection is deemed lost.
- *Latency field:* This fields specifies the *connSlaveLatency* value, reflecting the number of connection events that a *Slave* device is permitted to skip.
- *MasterSCA Field:* This field denotes the *Master's* worst-case sleep clock accuracy, which, when combined with the *Slave's* sleep clock accuracy (slaveSCA), is used to calculate the *Slave's* additional listening time, known as *Window Widening*, at each anchor point.

Figure 7 shows the *d1*, *d2*, *d3*, and *d6* connection parameters responsible for managing the connection events with *Latency* equals to zero (d5 = 0) on the considered use case packet exchange in a nominal scenario.

Table 13: Connection Request PDU Fields for Managing Connection Events

Field	Parameter	Range
Interval	connInterval (***d1***) = Interval × 1.25 ms	7.5 ms ≤ d1 ≤ 4 s
WinOffset	transmitWindowOffset (***d2***) = WinOffset × 1.25 ms	0 ≤ d2 ≤ d1
WinSize	transmitWindowSize (***d3***) = WinSize × 1.25 ms	1.25ms ≤ d3 ≤ min(10ms, d1 − 1.25ms)
Timeout	connSupervisionTimeout (***d4***) = Timeout × 10 ms	100 ms ≤ d4 ≤ 32 s, d4 > (1 + d5) × d1 × 2
Latency	connSlaveLatency (***d5***) = Latency (count)	$0 \leq d5 \leq \left(\frac{d4}{d1 \times 2}\right) - 1$, d5 < 500
MasterSCA	windowWidening (***d6***) = $\left(\frac{masterSCA+slaveSCA}{1000000}\right) \times$ timeSinceLastAnchor	Interval in μs

6.5.2. Attack Scenario Identification

We develop six novel potential attack scenario models and their implementations. The first three scenarios focus on Denial of Service (DoS) and Spoofing attacks targeting biosensor nodes in the BLE network within the sensing and controlling domains. The remaining three scenarios are based on sensor-node spoofing attacks, used as platforms to launch further attacks targeting the application service domain and the user interface domain. All scenarios are developed by synthesizing CWE-CAPEC (attack models) and CVE-CAPEC (attack implementations). Relevant CVE are highlighted in Table 11. Table 15 summarizes the derived attack scenarios and their various implementations, incorporating abbreviated CVE references from Table 11. We identify potential risks associated with the intercommunication phases of BLE—during the advertising, pairing, and connection phases—as well as risks arising from inadequate security measures due to resource constraints, and risks related to the dynamic mobility of BLE nodes, such as reconnections, as demonstrated in ASMI-1, ASMI-2, and ASMI-3. In addition, ASM-4, ASM-5, and ASM-6 illustrate how BLE sensors could serve as initial platforms for launching more complex and critical attacks targeting the user domain and application service domain. Table 14 provides a summary of the attack scenario implementations derived from Table 15 that arise from BLE nodes' intercommunication processes, dynamic mobility, resource constraints, and their roles as attack platforms.

Table 14: Mapping between the Derived Attack Scenario Models Implementation (ASMI) and IoT Unique Aspects

Addressed IoT Aspect	Phase/Weakness/Domain	Attack Scenario Model Implementation (ASMI)
Intercommunication Processes of BLE Nodes (addressing $L_{coupling}$)	Advertising Phase	ASMI-1.2, ASMI-1.5, ASMI-1.6
	Pairing phase	ASMI-1.3, ASMI-1.10, ASMI-2.1
	Connection Phase	ASMI-1.1, ASMI-1.4, ASMI-1.7, ASMI-1.8, ASMI-1.9, ASMI-1.11, ASMI-3.1, ASMI-3.2, ASMI-3.3
Dynamic Mobility of BLE Nodes (addressing $L_{dynamics}$)	Improper Authentication During Re-connection Phase (CWE-287)	ASMI-2.2, ASMI-2.3, ASMI-2.4
Resource Constraints of BLE Nodes (addressing $L_{resource\_constraints}$)	Lack of Encryption (CWE-319, CWE-311)	ASMI-1.8, ASMI-1.9, ASMI-1.11, ASMI-3.1, ASMI-3.2, ASMI-3.3
	Improper Authentication (CWE-287)	ASMI-1.1, ASMI-1.8, ASMI-1.9, ASMI-1.11, ASMI-2.2, ASMI-2.3, ASMI-2.4
	Improper Input Validation (CWE-20)	ASMI-1.1, ASMI-2.1, ASMI-1.3, ASMI-1.4, ASMI-1.5, ASMI-1.7, ASMI-1.8, ASMI-1.9, ASMI-1.10, ASMI-1.11, ASMI-2.1
Using BLE Nodes as Attack Platforms (addressing L_{attack_vector})	Attacking User Domain	ASMI-4.1, ASMI-4.2, ASMI-4.3, ASMI-4.4
	Attacking Application Service Domain	ASMI-5.1, ASMI-6.1, ASMI-6.2, ASMI-6.3

6.5.2.1. Attack Scenario Model 1 (ASM-1): DoS by Packet Injection of BLE Sensor Node

The attackers in the BLE range can sniff (CAPEC-158) the BLE communication so that they can decode the BLE exchanged messages. This is especially trivial when no encryption (CWE-319) is implemented at all. But even in the presence of encryption, due to the potential improper input validation (CWE-20) of packet fields, lack of authentication (CWE-287) of incoming packets, and possible race conditions between packets (CWE-362) that could affect the BLE implementation in sensor nodes identified in *UC-1*, the attacker can craft a malicious request packet (CAPEC-$153_{[\text{CWE-20, CWE-287}]}$) and inject it into the sensor node during an established connection (CAPEC-$26_{[\text{CWE-362}]}$). Upon receipt, the sensor node may process the malicious packet as legitimate due to improper input validation. This malformed packet has the potential to crash the sensor node or induce irregular behavior, resulting in a denial of service (CAPEC-100) or event deadlock (CAPEC-25) that cause the sensor to become unresponsive. The ultimate goal of the attacker here is to endanger the patient's life by halting the sensor node's measurements.

6.5.2.2. Attack Scenario Model 2 (ASM-2): Spoofing of BLE Sensor Node with Pairing Procedure

Attackers within BLE range can sniff (CAPEC-158) the BLE packet exchange before establishing the pairing procedure and impersonate the sensor node (CAPEC-667). This may involve changing the attacking device's name or MAC address to mimic the legitimate sensor node (CAPEC-151). Due to improper input validation (CWE-20) and improper authentication (CWE-287) identified in *UC-1*, the secure encrypted link between the sensor nodes and the gateway could be compromised, enabling sensor spoofing. These weaknesses can occur during the reconnection of a sensor node with the gateway to a previously paired connection. If a secure connection was previously established, the attacker might wait for a reconnection due to patient mobility or might disrupt the connection to force a reconnection. This disruption can be achieved either by using jamming (CAPEC-601) or by employing the ASM-1 attack scenario model, wherein the attacker injects crafted packets (CAPEC-$248_{[\text{CWE-311}]}$) into the sensor before the gateway. This exploits a race condition security weakness (CAPEC-$26_{[\text{CWE-362}]}$) and causes the DoS of the sensor nodes. Once a reconnection occurs, even with a previously established paired connection using Long Term Key (LTK) encryption/authentication, an attacker can craft and inject a packet right during the reconnection (CAPEC-153) into the gateway. Upon receipt, the gateway might mistakenly accept the spoofed, crafted packet (CWE-20, CWE-287) without requiring encryption and authentication (CWE-20, CWE-287). The attacker can then abuse the authentication mechanism (CAPEC-$114_{[\text{CWE-20, CWE-287}]}$) and transmit malicious, spoofed measurements to the IoT gateway (CAPEC-148). The ultimate goal of the attacker is to send erroneous sensor measurements to the gateway, potentially endangering the patient's life.

6.5.2.3. Attack Scenario Model 3 (ASM-3): Spoofing of BLE Sensor Node without Pairing Procedure

Building on the possibility that BLE could be implemented without encryption (CWE-319) or authentication by the sensor nodes (CWE-287) identified in *UC-1*, attackers within BLE range can sniff (CAPEC-158) the BLE communication. This enables them to impersonate the sensor node more easily than in the previous ASM (CAPEC-667). The attacker can spoof the sensor by either forcing a legitimate disconnection through the injection of crafted control packets (CAPEC-$248_{[\text{CWE-20}]}$) during an established connection or by employing the ASM-1 attack scenario model to induce a sensor DoS. This exploits a race condition weakness without breaking the connection (CAPEC-$26_{[\text{CWE-362}]}$) from the perspectives of the gateway (the gateway does not detect the connection abort). The attacker then hijacks and maintains the connection with the gateway (CAPEC-$593_{[\text{CWE-287}]}$). Subsequently, the attacker intercepts the gateway's requests and injects falsified sensor measurement responses (CAPEC-$152_{[\text{CWE-287}]}$). The attacker's aim here, as in the previous scenario, is to send erroneous sensor measurements, potentially putting the patient's health at risk.

6.5.2.4. Attack Scenario Model 4 (ASM-4): Attacking User Devices During Real-time Monitoring via Injecting Malicious Payloads from Spoofed BLE Biosensor Nodes

Utilizing the previously described attack scenario models (ASM-2 or ASM-3), an attacker can spoof a BLE sensor node and inject erroneous measurements into the IoT gate-

way. Building upon identified vulnerabilities such as Cross-site Scripting (XSS) identified as CWE-79 and improper input validation (CWE-20) in *UC-2* and *UC-3* at the MQTT broker, and considering the improper restriction of operations within the bounds of a memory buffer (CWE-119) and out-of-bounds write (CWE-787) identified at MQTT clients in *UC-2*, *UC-3*, and *UC-4*, an attacker can embed malicious XSS payloads or executable codes within BLE traffic packets as an attack vector to compromise user devices. This payload, can be relayed directly without any sanitization (CAPEC-$63_{[\text{CWE-79}]}$) by the broker and the MQTT client. This can lead to the execution of JavaScript codes within the front-end fields of dashboard browsers displaying sensor data, potentially allowing the attacker unauthorized access. The attacker can also force code execution via the injected executable code (CAPEC-$242_{[\text{CWE-119, CWE-787}]}$) at the MQTT client process to attack the user device, potentially leading to a crash (CAPEC-100) or unauthorized access (CAPEC-115). Furthermore, the XSS payload could be mistakenly treated as legitimate data by the MySQL database and stored in database tables (CAPEC-$592_{[\text{CWE-79}]}$), acting as a persistent XSS vulnerability for users later visiting the web server, potentially leading to unauthorized access. Figure 8 illustrates the workflow of the attack.

6.5.2.5. *Attack Scenario Model 5 (ASM-5): Attacking the MySQL Database via Injecting Malicious Payloads from Spoofed BLE Biosensor Node*

Similar to ASM-4, the MQTT broker and MQTT clients could be affected by an improper neutralization of query logic (CWE-943). An attacker can embed malicious SQL queries within BLE payloads as an attack vector against the database. Upon receiving the malicious payload, the MQTT broker forwards it to the MySQL database without any sanitization by either the MQTT broker or the MQTT client (CAPEC-$63_{[\text{CWE-943}]}$). These malicious SQL queries can delete entire database tables, leading to a denial of service for the MySQL database. Figure 9 illustrates the workflow of the attack.

6.5.2.6. *Attack Scenario Model 6 (ASM-6): Attacking the MQTT Broker via Injecting Malicious Payloads from a Spoofed BLE Biosensor Node*

Building on the scenarios of BLE sensor spoofing in ASM-2 and ASM-3 by attackers and considering the identified stack-based buffer overflow (CWE-121), improper restriction of operations within the bounds of a memory buffer (CWE-119), and lack of a standardized error handling mechanism (CWE-544) security weaknesses at the MQTT broker in *UC-2*, an attacker can embed malicious BLE payloads or craft malicious executable code. This code could cause remote code execution at the MQTT broker (CAPEC-$242_{[\text{CWE-119, CWE-787}]}$), leading to unauthorized access. Additionally, the attacker can also embed malicious payloads that can cause buffer overflow (CAPEC-$100_{[\text{CWE-119}]}$) or mishandle events, thereby consuming excessive memory (CAPEC$_{[\text{CWE-544}]}$) and rendering the MQTT broker unresponsive, denying service. Figure 10 illustrates the workflow of the attack.

6.5.3. *Impact and Likelihood Profiling*

Based on Table 15, we have chosen to detail the execution of two attack scenario implementations of BLE sensors node: ASMI-1.2 and ASMI-3.1. We then profile their severity impact and likelihood factors. In these scenarios, one sensor node and the IoT gateway exchange the following messages: advertisement packets, scan packets, connection requests, version packets, and ATT packets. This exchange is specifically designed to accommodate limitations in energy resources and intentionally excludes any extra message transfers in order to reduce energy consumption. Including version packets is necessary because of possible inconsistencies between the sensor nodes and the gateway in terms of their manufacturers and adherence to the BLE specification version. This is pertinent since many diverse IoT devices could operate within the same network.

6.5.3.1. *ASMI-1.2: Buffer Overflow-induced DoS in a Biosensor Node via Injection of a Crafted VERSION_IND packet*

The attacker intercepts the initial packet exchange between the sensor and gateway (CAPEC-157) and, upon detecting the connection request, crafts a malicious version request packet with a 150-byte link layer PDU length header. The attacker injects this malicious version request packet (CAPEC-153) into the communication channel between the sensor and gateway during the connection event receive window, just before the gateway, without disrupting the pre-established connection. This action exploits a race condition vulnerability in BLE specifications (CAPEC-$26_{[\text{V37}]}$). Improper handling of packet length in the sensor's BLE implementation (vulnerability V11) leads to the sensor receiving and processing the malformed packet as legitimate and allocating it to the buffer designated for original version requests. The unexpected packet size causes a buffer overflow, resulting in a denial of service for the sensor (CAPEC-$100_{[\text{V11}]}$). Figure 11 illustrates this attack scenario with a UML diagram. The profiling of the risk impact and likelihood factors in this attack scenario is summarized in Figure 12:

- **Technical Impact (TI) Profile:**
  - *Loss of Confidentiality:* Minimal non-sensitive data were lost (2). Although the primary objective of the attacker was to interrupt services, the unencrypted communications potentially allowed eavesdropping on the data transmitted.
  - *Loss of Integrity:* Data corruption was minimal and not the main intent of the attacker (3).
  - *Loss of Availability:* A significant disruption occurred in primary services due to a buffer overflow causing a denial of service in the sensor node (7). The healthcare system had to rely on remaining sensors, risking the loss of critical patient information.
- **Likelihood of Vulnerability Exploitation (LVE) Profile:**

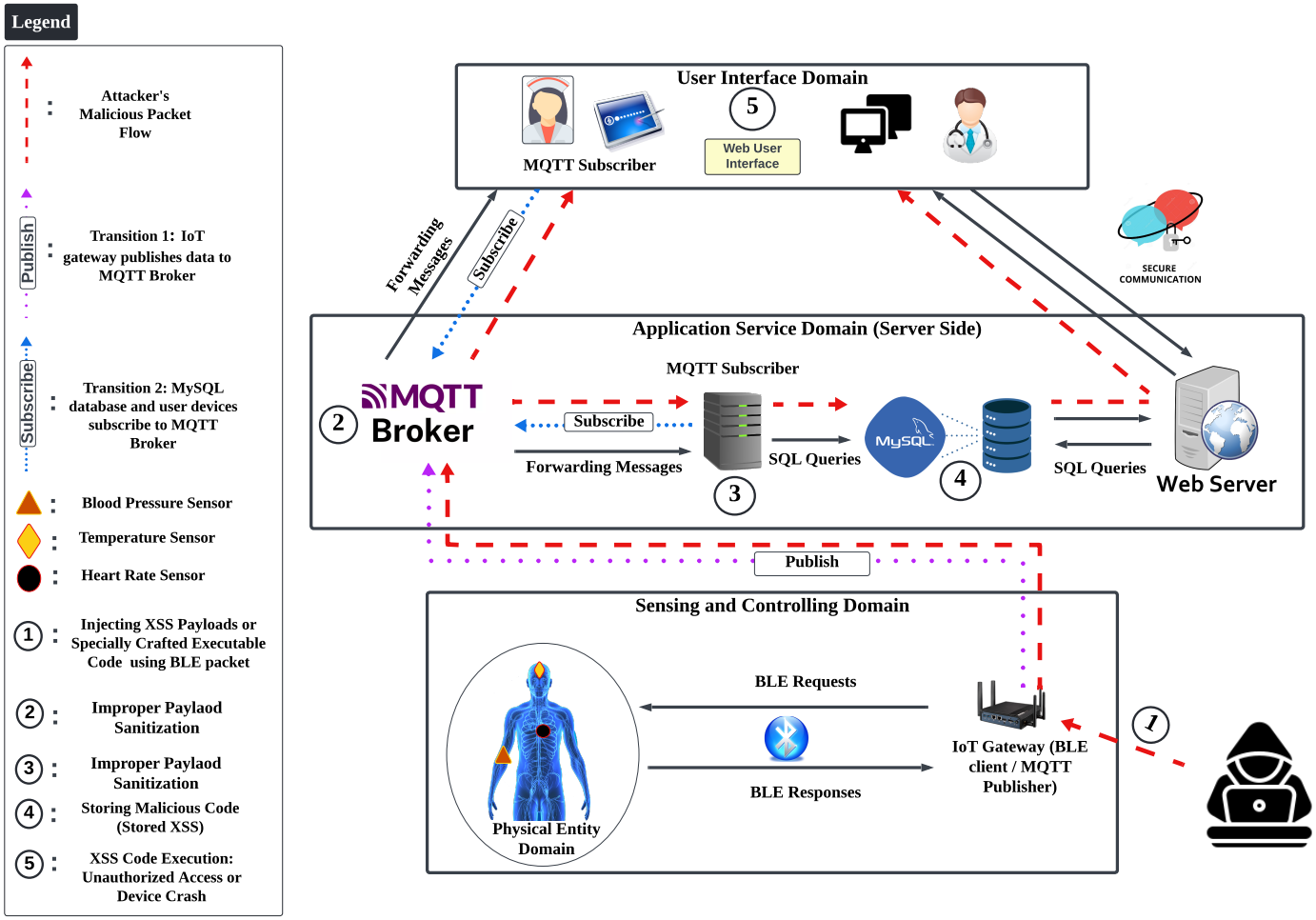

Figure 8: Attack Scenario Model 4 (ASM-4): Attacking the User Domain Using ASM-2/ASM-3 as Attack Platform

- *Ease of Discovery:* The BLE protocol vulnerabilities (V11 and V37) are well-documented, with numerous tools available for intercepting communications, making the attack surfaces easy to identify (9).
- *Ease of Exploit:* The attack involved injecting a simple maliciously crafted packet by leveraging specific knowledge about the BLE timing parameters to exploit a race condition vulnerability during a connection event (5).
- *Awareness:* The vulnerabilities are well-known and publicly reported (9).
- *Intrusion Detection:* The attack went undetected because BLE networks necessitate specialized IDS designed to analyze the internal communication protocols of BLE, which are beyond the monitoring capabilities of traditional IP-based IDS systems (9).

- **Impact on Safety and Business Assets (ISBA) Profile:**
  - *Human Safety:* If a patient in a severe health state goes unnoticed due to the sensor node's denial of service, it could lead to death (9).
  - *Environmental Damage:* There was no physical damage to the hospital's equipment (0).
  - *Financial Damage:* Potential significant financial liabilities due to insurance claims if patient harm occurred (7).
  - *Reputation Damage:* A sensor's denial of service leading to death could adversely affect the hospital's reputation and its contracts with partners (5).
  - *Non-Compliance:* The attack revealed significant regulatory non-compliance due to inadequate update and patching policies of known vulnerabilities (5).
  - *Privacy Violation:* There was no exposure of personal data as the attacker's objective is to disrupt the sensor's services (0).

6.5.3.2. *ASMI-3.1: Hijacking by Spoofing a Biosensor Node Through the Injection of an LL_TERMINATE_IND packet*

The attacker intercepts the packet exchange between the sensor and gateway (CAPEC-157) and crafts an LL_TERMINATE_IND control packet, imitating the IoT gateway device to

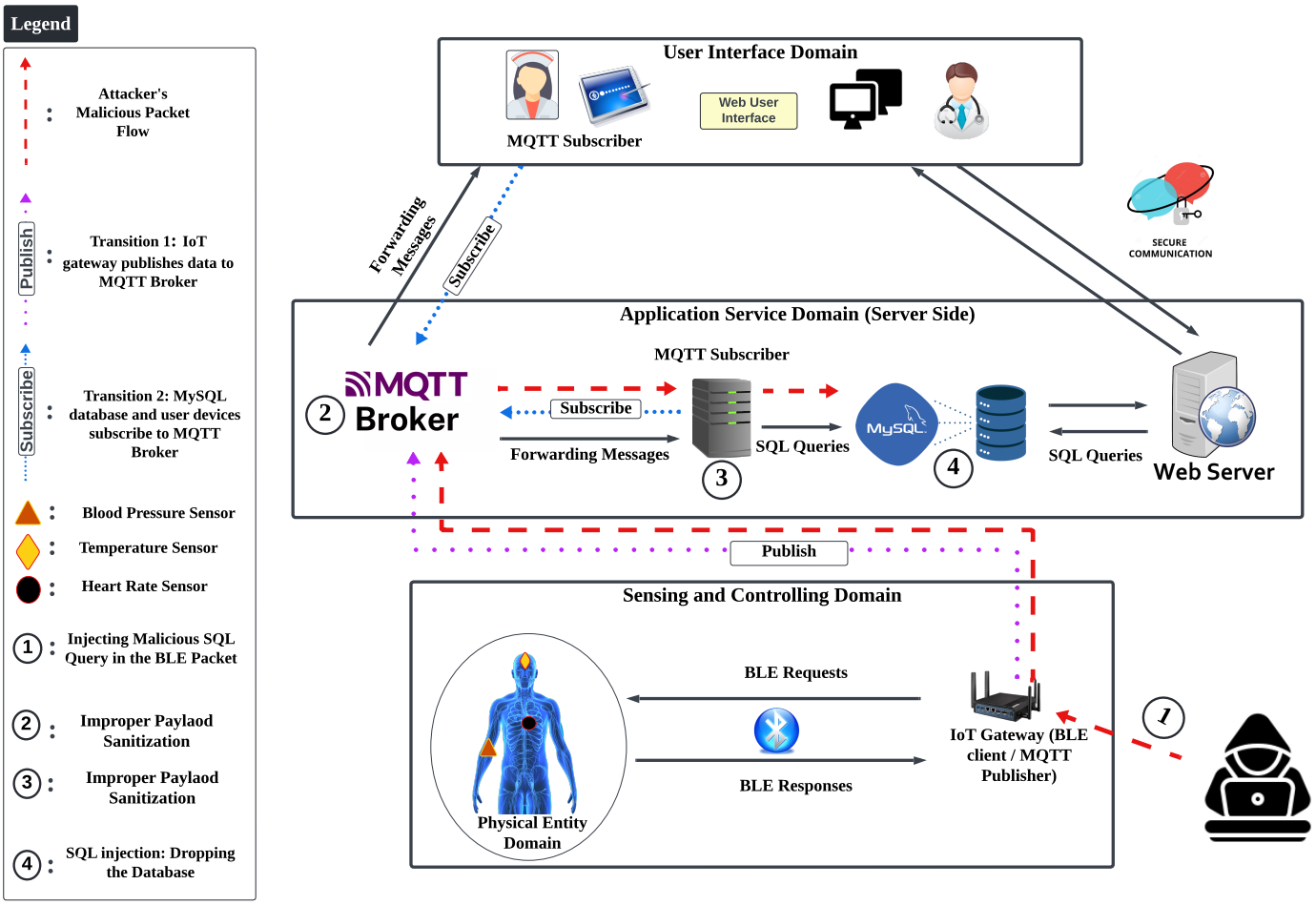

Figure 9: Attack Scenario Model 5 (ASM-5): Attacking the MySQL Database Using ASM-2/ASM-3 as Attack Platform

instruct the sensor to terminate the connection. At the start of the receive window in an established connection event, just before the gateway sends the ATT request, the attacker injects the LL_TERMINATE_IND packet into the sensor's communication channel (CAPEC-248), exploiting a race condition vulnerability in BLE (CAPEC-26$_{[V37]}$). Upon receiving the LL_TERMINATE_IND packet, the sensor node disregards the gateway's request and exits the connection. As the gateway continues attempting communication, unaware of the disconnection, the attacker exploits the lack of encryption and authentication in BLE communication to mimic the biosensor (CAPEC-667$_{[V4, V7]}$). By intercepting the gateway's read requests and crafting the necessary packet fields, the attacker hijacks the connection and injects malicious vital measurement responses during the hijacked connection events (CAPEC-593$_{[V4, V7]}$). This manipulation leads to the system processing false data for the patient. Furthermore, the attacker is capable of injecting malicious payloads to compromise the user and application domains, as delineated in the attack scenario implementations of ASM-4, ASM-5, and ASM-6. Figure 13 depicts a UML diagram of the attack scenario implementation.

The profiling of the risk impact and likelihood factors in this attack scenario is summarized in Figure 14, without considering it as an attack vector for implementing other attacks:

- **Technical Impact (TI) Profile:**
  - *Loss of Confidentiality:* Minimal non-sensitive data disclosed (2). The attacker's objectives were to spoof and corrupt the vital measurements.
  - *Loss of Integrity:* Extensive seriously corrupt data (7). The attacker spoofed the sensor's identity and can inject malicious vital readings as long as they are not detected.
  - *Loss of Availability:* Extensive primary services interrupted (7). The attacker injected malicious data posing as the biosensor, meaning the system had to rely on other sensors to detect emergencies, possibly missing critical situations.
- **Likelihood of Vulnerability Exploitation (LVE) Profile:**
  - *Ease of Discovery:* Automated tools available (9). The attacker intercepted the communication and identified the attack surfaces based on the V4, V7, and V37 vulnerabilities using automated tools and access to vulnerability information.
  - *Ease of Exploit:* Easy (5). The attacker crafted malicious packets, requiring specialized knowledge of

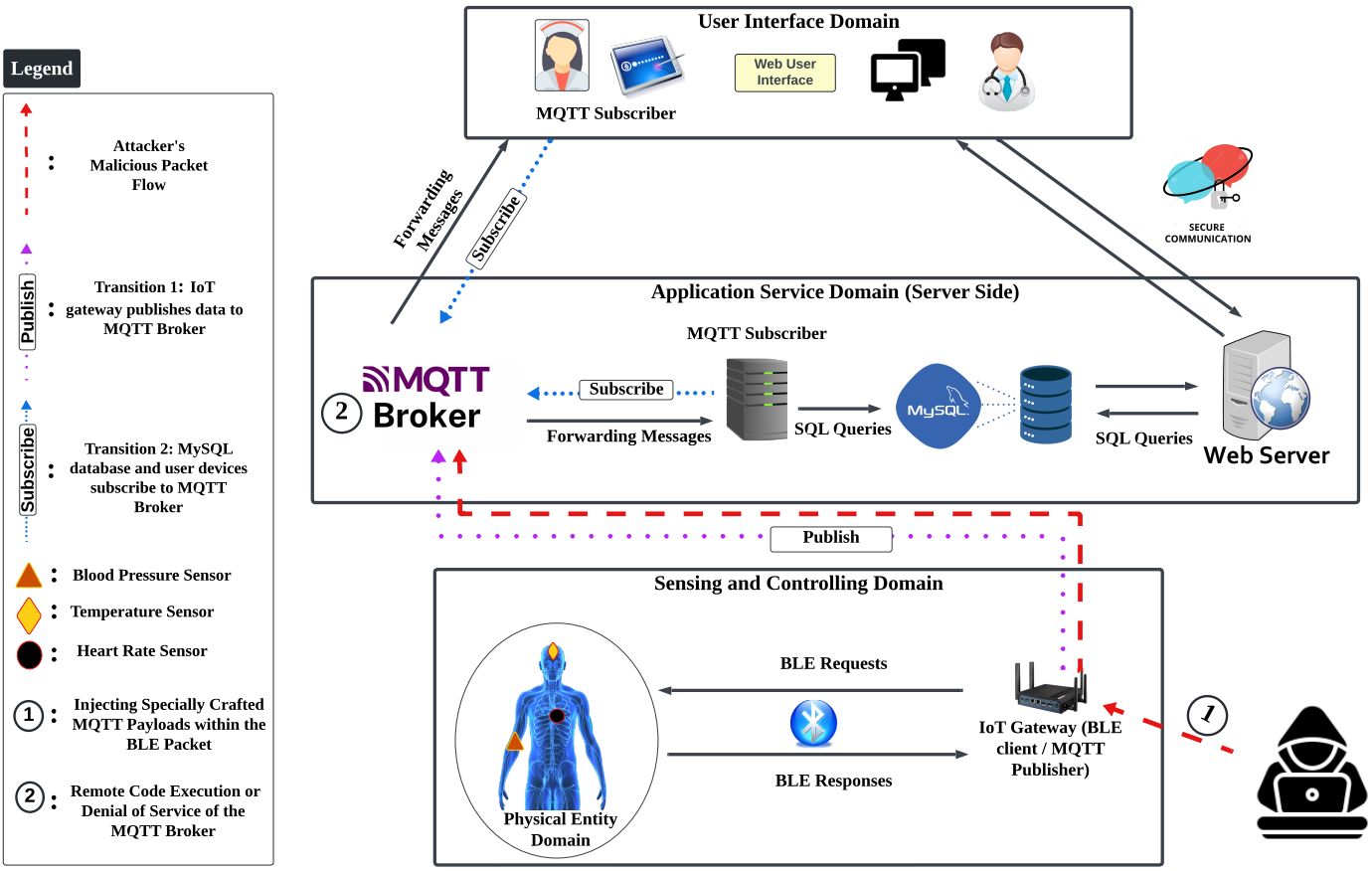

Figure 10: Attack Scenario Model 6 (ASM-6): Attacking the MQTT Broker Using ASM-2/ASM-3 as Attack Platform

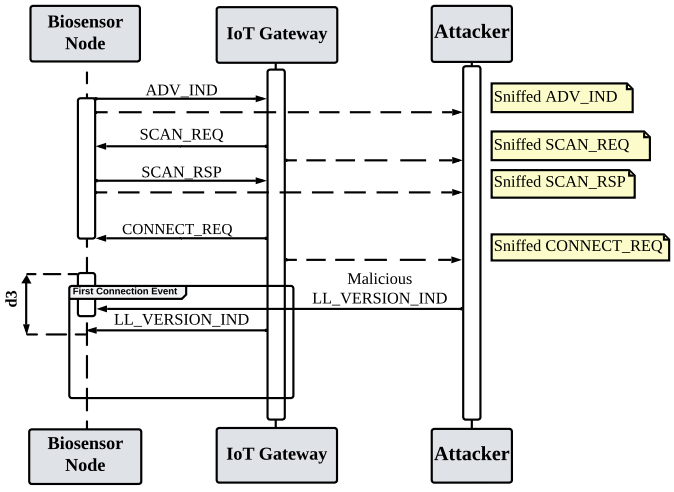

Figure 11: UML Sequence Diagram for the Attack Scenario Model Implementation 1.2 (ASMI-1.2).

ADV_IND: Advertising Packets, **SCAN_REQ/RSP:** Scan Request/Response, **CONNECT_REQ:** Connection Request, **LL_VERSION_IND:** Version Indication Packet, **d3:** transmitWindowSize.

BLE timing parameters to exploit a race condition. These packets were injected into both the sensor and the gateway, though synchronization presented certain challenges, necessitating the integration of exploit tools by the attacker to successfully exploit the vulnerabilities.

$$TI_{(ASMI-1.2)} = \begin{pmatrix} 2 \\ 3 \\ 7 \end{pmatrix}, \quad LVE_{(ASMI-1.2)} = \begin{pmatrix} 9 \\ 5 \\ 9 \\ 9 \end{pmatrix}, \quad ISBA_{(ASMI-1.2)} = \begin{pmatrix} 9 \\ 0 \\ 7 \\ 5 \\ 5 \\ 0 \end{pmatrix}$$

Figure 12: Profiling Scores for the Attack Scenario Model Implementation 1.2 (ASMI-1.2)

- *Awareness:* Public knowledge (9). The vulnerabilities are well-known and have been reported in public databases.
- *Intrusion Detection:* Not logged (9). The attack went undetected, as the BLE networks often lack IDS systems to analyze their internal communications.

- **Impact on Safety and Business Assets (ISBA) Profile:**
  - *Human Safety:* Death (9). If the patient in an emergency situation (e.g., high body temperature) were subjected to false vital readings injected by an attacker, leading to the injection of normal vital measurements, this could result in the patient's death.

Table 15: Potential Attack Scenario Models and Implementations

Attack Scenario Model (ASM)	Attack Scenarios Model Implementation (ASMI)
ASM-1	*ASMI-1.1:* DoS on a sensor node by injecting consecutive ATT read requests within a connection event (V22, V37)
	ASMI-1.2: DoS on a sensor node by injecting large version request LL header length packet at the onset of connection establishment (V11, V37, V4)
	ASMI-1.3: DoS on a sensor node by injecting malformed Security Manager (SM) public key packet during the pairing process (V37, V12)
	ASMI-1.4: DoS on a sensor node by injecting a packet with Message Integrity Code (MIC) failure field (V37, V28)
	ASMI-1.5: DoS on a sensor node by injecting a connection request with a crafted Channel Map (CM) field during connection establishment (V37, V24)
	ASMI-1.6: DoS on a sensor node by injecting a connection request with crafted Connection Interval and Timeout fields at the start of connection establishment (V37, V25)
	ASMI-1.7: DoS on a sensor node by injecting a crafted pause encryption LL_PAUSE_ENC request during a connection event (V37, V26)
	ASMI-1.8: DoS on a sensor node by injecting an empty PDU packet with invalid Next Expected Sequence Number (NESN) and Sequence Number (SN) during an established connection (V37, V2)
	ASMI-1.9: DoS on a sensor node by injecting a packet with the Link Layer ID (LLID) field set to zero during a connection event (V37, V14)
	ASMI-1.10: DoS on a sensor node by injecting a crafted pairing request packet with an oversized Long Term Key (LTK) during the pairing phase (V37, V17)
	ASMI-1.11: DoS on a sensor node by injecting a crafted L2CAP packet request with an invalid link layer L2CAP header length (V23, V16, V37)
	....
ASM-2	*ASMI-2.1:* Spoofing a sensor node by injecting a crafted out-of-order link-layer encryption request (LL_ENC_REQ) during Secure Connections pairing (V37, V6)
	ASMI-2.2: Proactive spoofing of a sensor node occurs during the patient's mobility in the paired reconnection process, wherein an 'encryption fail response' is injected into the gateway (V32)
	ASMI-2.3: Reactive spoofing of a sensor node occurs during the patient's mobility in the paired reconnection process, which involves responding to the gateway's with 'unencrypted and unauthenticated attribute request' (V32)
	ASMI-2.4: Spoofing a sensor node by injecting crafted packets (ASM-1 Implementations) or by jamming to force a sensor's reconnection, and initiating either a reactive or proactive paired reconnection with the gateway (V37, V32)
	....
ASM-3	*ASMI-3.1:* Hijacking by spoofing a sensor node during an unencrypted established connection by injecting an LL_TERMINATE_IND packet, and injecting malicious ATT measurement packets into the gateway (V37, V4, V7)
	ASMI-3.2: Hijacking by spoofing a sensor node during an unencrypted established connection by injecting a CONNECTION_UPDATE_REQ packet, and injecting malicious ATT measurement packets into the gateway (V37, V4, V7)
	ASMI-3.3: Hijacking by spoofing a sensor node by injecting crafted packets that cause denial of service (ASM-1), and injecting malicious ATT measurement packets into the gateway (V37, V4, V7)
	....
ASM-4	*ASMI-4.1:* Execution of unauthorized actions on a user device by injecting a JavaScript code (XSS payload) within a sensor BLE payload, utilizing ASM-2/ASM-3 as the attack vector during real-time data monitoring (V51, V52, V53, V54, V55)
	ASMI-4.2: Execution of unauthorized actions on a user device by injecting a JavaScript code (XSS payload) within a sensor BLE payload, employing ASM-2/ASM-3 as attack platforms and storing the malicious XSS payloads in the MySQL database (stored XSS), which are later downloaded and executed through the web server communication channel (V51, V52, V53, V54, V55)
	ASMI-4.3: Execution of unauthorized actions on a user device by injecting specially crafted executable codes within the BLE payloads, using ASM-2/ASM-3 as attack vectors via buffer overflow within the MQTT client process on the user device (V38, V41, V42)
	ASMI-4.4: DoS of the MQTT client implemented on the user device by injecting specially crafted executable codes within the BLE payloads, utilizing ASM-2/ASM-3 as attack vectors via buffer overflow within the MQTT client process on the user device (V38, V41, V42, V43)
	....
ASM-5	*ASMI-5.1:* Dropping and deleting SQL tables at the MySQL database by injecting a malicious SQL query within a sensor BLE payload, employing ASM-2/ASM-3 as attack vectors (V51, V52, V53, V54, V55)
	....
ASM-6	*ASMI-6.1:* DoS of the MQTT broker by injecting a large MQTT payload (more than 64 bytes) within a sensor BLE payload, using ASM-2/ASM-3 as attack platforms (V38, V40)
	ASMI-6.2: DoS of the MQTT broker by injecting a non-JSON MQTT payload within a sensor BLE payload, employing ASM-2/ASM-3 as attack platforms (V38, V48)
	ASMI-6.3: Executing unauthorized actions on the MQTT broker by injecting a malicious executable code within a sensor BLE payload, using ASM-2/ASM-3 as attack platforms (V38, V40)
	....
....	....

– *Environmental Damage:* No Impact (0). The attack corrupted the integrity of the biosensor readings.

– *Financial Damage:* Significant effect on annual profit (7). The death of the patient due to such an attack could financially impact the hospital, potentially involving liability and insurance payments.

– *Reputation Damage:* Loss of goodwill (5). The manipulation of data resulting in the patient's death could adversely affect the hospital's relationships and contracts with partners, as well as its reputation

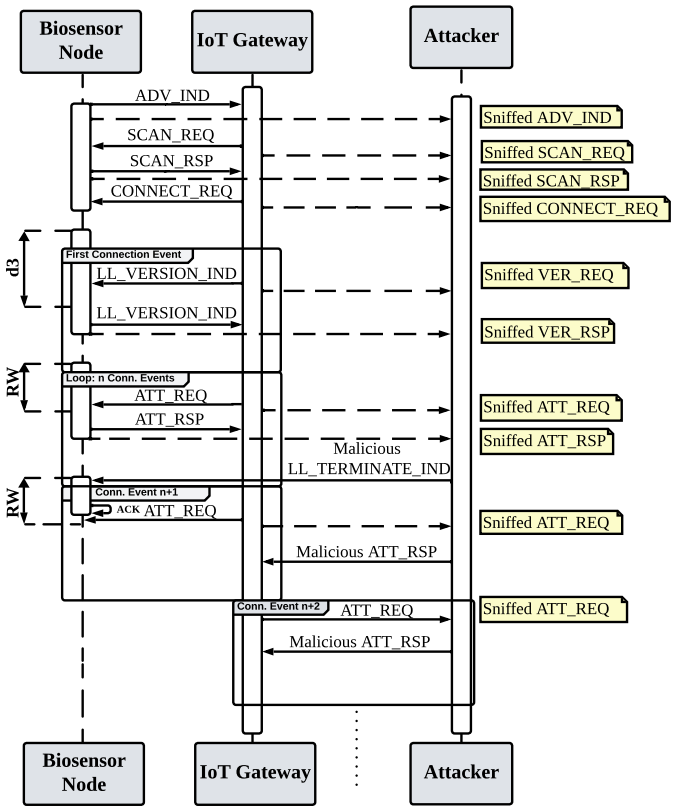

Figure 13: UML Sequence Diagram for the Attack Scenario Model Implementation 3.1 (ASMI-3.1).
ADV_IND: Advertising Packets, **SCAN_REQ/RSP:** Scan Request/Response, **CONNECT_REQ:** Connection Request, **LL_VERSION_IND:** Version Indication Packet, **ATT_REQ/RSP:** Attribute Request/Response, **LL_TERMINATE_IND:** Termination Connection Packet, **ACK:** The device's *Controller* acknowledges to the *Host* that the connection has been terminated, **d3:** transmitWindowSize, **RW (2*d6):** Receive Window.

among the public.

- *Non-compliance:* High profile violation (7). The attack leading to this data interruption constitutes a high violation of compliance with healthcare regulatory standards, due to a lack of update mechanisms and patching policies for known vulnerabilities. We consider it more severe in this scenario than in the previous one because the patient's vital measurements were incorrect rather than merely missing.
- *Privacy Violation:* No personal data exposed (0). The attacker's objective, focused on corrupting sensor measurements and injecting falsified vital readings, would not affect privacy.

$$TI_{(ASMI-3.1)} = \begin{pmatrix} 2 \\ 7 \\ 7 \end{pmatrix}, \quad LVE_{(ASMI-3.1)} = \begin{pmatrix} 9 \\ 5 \\ 9 \\ 9 \end{pmatrix}, \quad ISBA_{(ASMI-3.1)} = \begin{pmatrix} 9 \\ 0 \\ 7 \\ 5 \\ 7 \\ 0 \end{pmatrix}$$

Figure 14: Profiling Scores for the Attack Scenario Model Implementation 3.1 (ASMI-3.1)

6.6. Assessing the Framework's Generalizability Across Various IoT Systems

This subsection discusses the generalizability of our proposed risk identification framework across various IoT systems and explores how the attack scenario models, developed for the IoT smart healthcare system described in this paper, can be adapted to other IoT systems with similar components and communication protocols.

6.6.1. Framework's Generalizability to other IoT Components and Protocols

The four-step process we have developed employs a suite of general methodologies that are not confined to specific IoT components within the smart healthcare system. It incorporates approaches such as the proposed mapping between CWE and the OWASP top ten IoT security weakness categories, alongside the CVE-IoT and CAPEC-IoT spreadsheets, which include a comprehensive list of security weaknesses, vulnerabilities, and attack patterns relevant to the IoT domain. This framework can adapt to new components or protocols by incorporating them into the analyses conducted during Step 2, depending on the list of selected CWE from Step 1. This selection may focus on confined sub-cases of security weaknesses (adopting an expert-driven approach) or on a comprehensive list of CWE (adopting a data-driven approach), and then carry out modifications in the subsequent steps. For example, in a smart manufacturing context, the industrial controllers and communication protocols such as Modbus or Ethernet/IP could be added to the list of components, and their known vulnerabilities would be added to the output of Step 2 to design new attack scenario models and implementations during Step 3 and Step 4. Conversely, in smart city applications, the emphasis might shift towards IoT sensors, IP cameras, and communication technologies like Zigbee, LoRaWAN, or Sigfox.

6.6.2. Adaptability of Derived Attack Scenario Models from one IoT System to Another

The attack scenario models defined in our case study target an IoT smart healthcare system, which utilizes BLE and MQTT networks. However, they are not confined to any specific vendor or software version of devices. Instead, these models are based on common security weaknesses and CAPEC, reflecting inherent vulnerabilities that could be present in several other IoT systems implementing the same network protocols or categories of IoT components. For example, an attack scenario model that highlights a potential security weakness in BLE protocols, such as a lack of encryption or improper authentication in a healthcare application, can be equally applicable to smart home appliances like door locks and lighting systems that also use BLE. Furthermore, the identified security weaknesses and the attack scenario models related to the implementation of the MQTT protocol between the IoT gateway and the MQTT broker can similarly impact MQTT protocol other applications in smart city and industrial automation applications. Thus, attack scenario models obtained from a system during Step 4 could be used during the same step for another IoT system sharing

the same security weaknesses. In addition, the attack scenario models identified in a specific category of IoT components enable the establishment of penetration testing frameworks for IoT. These frameworks can test against potential vulnerabilities in deployed or newly developed IoT products within the same category of devices or protocol where a security weakness has been previously identified from records of vulnerabilities.

7. Simulations

We validated the practical implementation of our derived attack scenarios using the Contiki Cooja network simulator [62]. Each scenario was simulated separately, involving a temperature sensor node (node 4), an IoT gateway (node 1), and attacker (nodes 2 and 3) devices, modeled as Cooja motes. We emulated an abstract BLE protocol between the sensor and gateway to simulate link-layer packet exchanges and application-layer temperature attribute requests and responses. For the attacker, two motes were used: one (node 3) to synchronize with the sensor node and another (node 2) with the gateway, each acting as a dongle connected to a *Host* and functioning as a sniffer to eavesdrop on the communication. Figure 15 depicts the network configuration under attack. During attack execution, acknowledgments for sent, received, or sniffed messages are displayed in the Mote output window. Each packet is identified by a PDU type and payload, representing the packet's type and name, respectively. Simulation parameter values, including those for managing connection events discussed in Table 13, are provided in Table 16. Moreover, each mote is assigned a MAC address to correctly identify and connect to the intended devices.

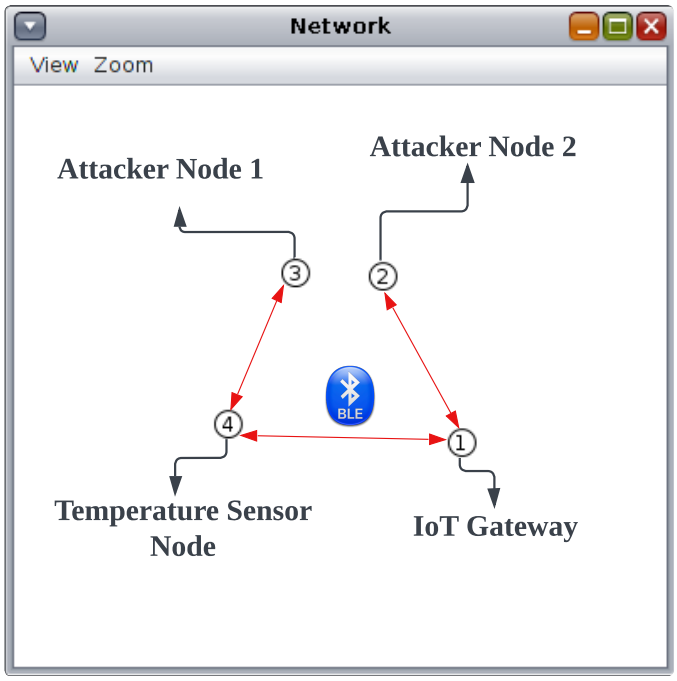

Figure 15: The Network's Components of the Cooja Simulator

Table 16: Simulation Parameters

Parameter	Range	Selected Value
d1	$7.5\,\text{ms} \leq d1 \leq 4\,\text{s}$	250 ms
d2	$0 \leq d2 \leq d1$	250 ms
d3	$1.25\text{ms} \leq d3 \leq \min(10\text{ms}, d1 - 1.25\text{ms})$	10 ms
d4	$100\,\text{ms} \leq d4 \leq 32\,\text{s}$, $d4 > (1 + d5) \times d1 \times 2$	600 ms
d5	$0 \leq d5 \leq \left(\frac{d4}{d1 \times 2}\right) - 1$, $d5 < 500$	0
d6	Interval in μs	-

We excluded the exact value of the *d6* parameter in our simulations, which represents the *Receive Window*, the listening timeframe of the slave within which the master must send its requests to be accepted. This exclusion was due to the simulator's constraints in handling clock ticks of attack injection in microseconds. Therefore, we configured the sensor to extend the listening time to ensure successful attack injection if the attacker manages to inject packets before the gateway. This adjustment makes the *Receive Window* sufficiently large for demonstration purposes.

7.1. ASMI-1.2 Validation

Figure 16 displays the results of executing the first attack scenario, where the attacker (nodes 2 and 3) intercepts the communication between the sensor and the gateway during the *Advertising State*. After seizing the gateway's connection request (CONNECT_REQ), the attacker waits for the *Transmit Window* to open, and then uses the spoofed MAC addresses to inject a malicious version indication packet (VERSION_IND) into the sensor before the gateway. This packet, with a link layer header length of 150 bytes, carries a payload replicated as 'Hacked'. Notably, when the temperature sensor node receives this packet, it treats it as a legitimate packet from the gateway. This causes a buffer overflow due to the packet's excessive length (1,214,539,115 bytes, namely more than 150 megabytes). Consequently, the sensor no longer responds to the legitimate gateway's messages.

7.2. ASMI-3.1 Validation

Figure 17 displays the results of executing the second attack scenario. Similar to the previous scenario, the attacker (nodes 2 and 3) intercepts communication between the sensor node and the gateway during the *Connection Phase*. This includes the exchange of read temperature attribute requests and responses (ATT REQ/RSP) with an average patient blood temperature of 39 degrees Celsius (emergency situation). At the beginning of connection event #1000, the attacker injects a terminate procedure indication packet (LL_TERMINATE_IND) from node 3 into the temperature sensor node, preempting the gateway's read ATT_REQ. The temperature sensor node accepts the

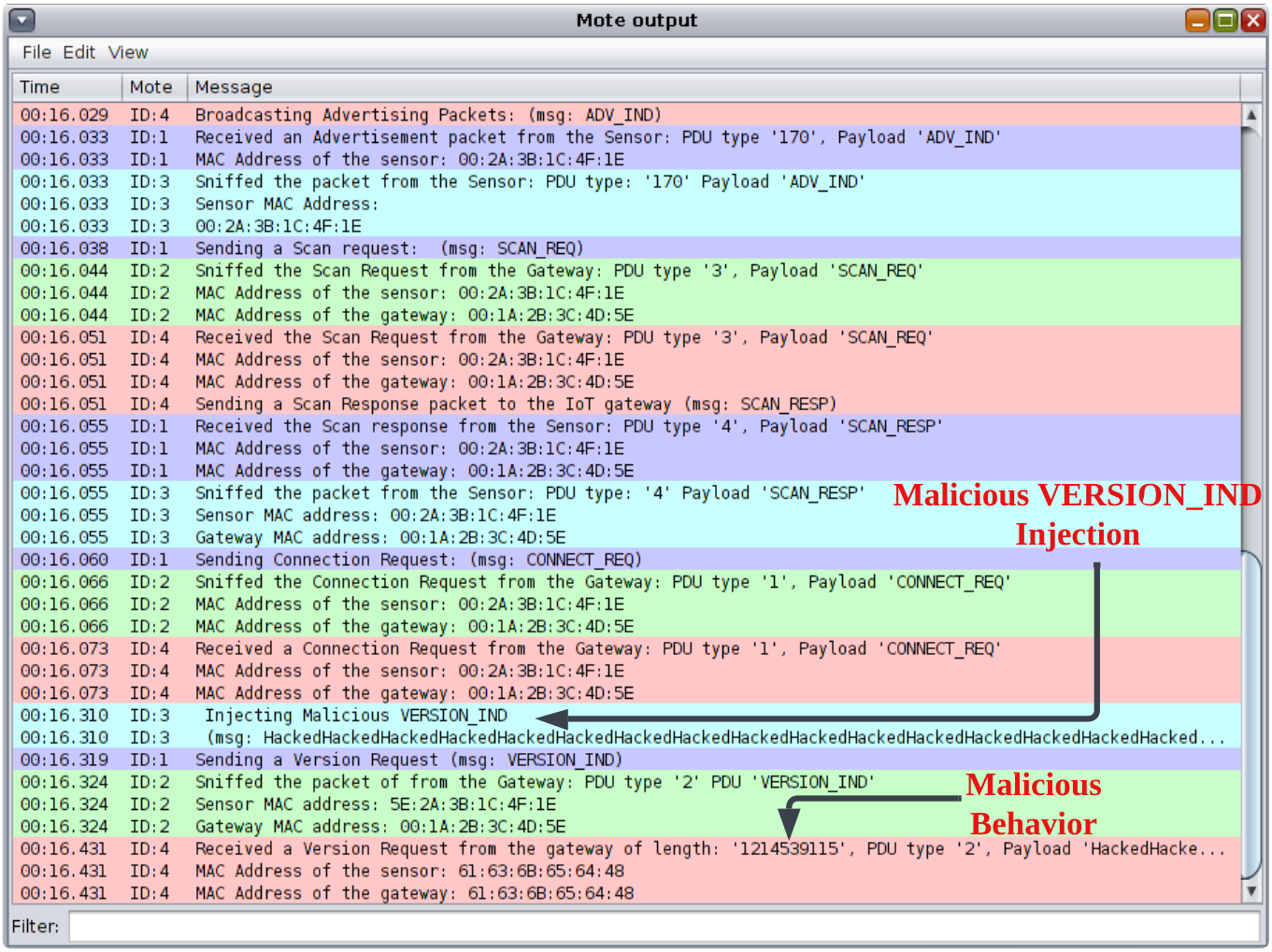

Figure 16: ASMI-1.2 Scenario Implementation Outputs

packet and exits the connection. Subsequently, node 3 transmits a notification message to node 2, informing it of the successful termination of the connection. During connection event #1000, no temperature response is transmitted to the gateway. Since the *Supervision Timeout (d4)* interval is set to 600 ms, the IoT gateway continues to send another packet at connection event #1001. At this time, attacker node 2 sniffs the ATT_REQ, takes spoofed MAC addresses, and injects a *Malicious* ATT_RSP with a fake payload temperature measurement of 37 degrees Celsius, continuing to synchronize with the gateway for the subsequent connection events. The attacker is then able to embed malicious payloads to launch dynamic attacks targeting the user domain and application domain.

8. Conclusion and Future Research Directions for IoT Risk Mitigation

In order to comprehensively identify and assess the security risks associated with IoT, it is essential to employ a dedicated security risk assessment method that addresses the key aspects of the IoT ecosystem. Traditional security risk assessment methods and existing IoT risk identification processes often fail to fully accommodate the complex and dynamic nature of IoT systems, potentially overlooking significant risks when applied to IoT. Hence, this paper has proposed a vulnerability-oriented risk identification framework, designed as a four-step process for IoT risk assessment. It begins with the identification of common weaknesses in CWE-IoT in Step 1, followed by a list of CVE-IoT in Step 2, a list of CAPEC-IoT in Step 3, and finally with the derivation of potential attack scenarios in Step 4, complete with impact and likelihood profiles. This process addresses several key aspects of the IoT ecosystem, including complex intercommunication between devices, dynamic nature of the IoT system, the potential use of devices as attack platforms, specific safety impact, and resource constraints crucial for comprehensive risk assessment. As an illustrative example, potential novel high-severity risk attack scenario models and implementations were identified in an IoT smart healthcare application use case. This was achieved through the proposed expert-driven approach, which maps the OWASP top ten weakness categories to CWE in Step 1, focusing on the first three high-criticality CWE-IoT categories. The process execution re-

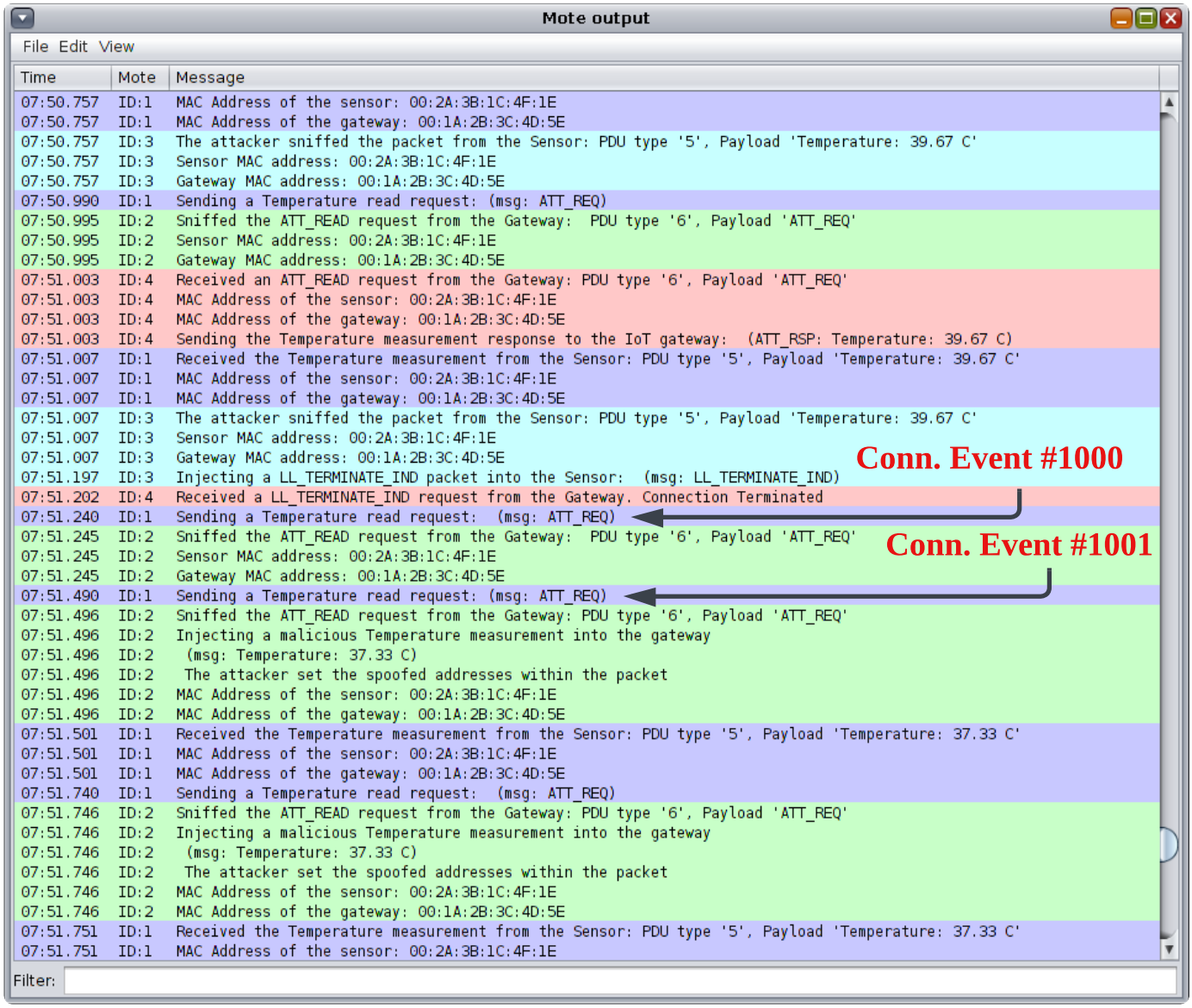

Figure 17: ASMI-3.1 Scenario Implementation Outputs

vealed critical attack scenarios that target BLE sensor nodes, possibly using them as platforms to compromise both user and backend application domains. Finally, the feasibility of the proposed process in uncovering practical implementations of derived attack scenarios has been validated through simulation of two attack scenarios. It should be noted that while expert-driven approaches are commonly used in risk assessment and are instrumental in identifying significant risks and validating the efficiency of our process, they represent a limitation in exhaustively exploring all possible risks, as the analysis is confined to particular sub-cases. Future work will aim to employ a data-driven approach or a hybrid approach to counterbalance these limitations. This will involve combining expert-driven and data-driven approaches to exhaustively explore the full range of risks, which is crucial for the development of robust defensive mechanisms capable of managing both anticipated and unanticipated IoT risks.

Considering possible future directions, the IoT sensing and control layer often consists of resource-constrained devices and present significant security vulnerabilities as evidenced in our use case study. This layer's direct link to backend services, whether centralized or decentralized (e.g., blockchain-based), heavily relies on the accuracy and integrity of its data for processing. Thus, security measures that ensure the reliability and safety of data generated from these devices is paramount. The literature proposes various lightweight security mechanisms suited to resource-constrained IoT devices. These include Number Theory Research Unit (NTRU) quantum-attack-resistant encryption schemes [63], authentication mechanisms [64], adversarial sample-based privacy protection schemes [65], and IDS systems [66]. These mechanisms can effectively mitigate various network-based risks, such as eavesdropping, spoofing, man-in-the-middle (MitM) attacks, replay attacks, and the use of compromised nodes as platforms for launch-

ing dynamic attacks. However, the heterogeneous and dynamic nature of IoT devices and protocols poses challenges in integrating homogeneous security mechanisms that are compatible with various platforms from different vendors and in having standardized tools for security testing and vulnerability patching [8]. Thus, future research directions in IoT security must focus on developing standardized and scalable lightweight security frameworks that can adapt to the heterogeneous and dynamic IoT ecosystem. Collaborative efforts across IoT sectors to share best practices such as, avoiding default and hardcoded credentials in IoT devices, preventing common coding errors that could lead to security weaknesses, conducting thorough security testing before product releases, and establishing policies and tools for vulnerability disclosure and regular patching, will also contribute significantly to reducing risks in the IoT ecosystem [67, 68].

Blockchain technology and cryptography can significantly enhance security when integrated with IoT systems. They offer robust and reliable security solutions that reduce risks significantly through secure, decentralized processing and storage of data collected from IoT devices and network participants. Additionally, blockchain provides a tamper-proof transactional database, which is more resistant to unauthorized alterations than conventional centralized methods [69]. Various frameworks for integrating blockchain and cryptography with IoT systems have been proposed in the literature. For instance, in the case study of an IoT healthcare system discussed in this paper, the blockchain and cryptography framework proposed by [70] can be utilized to significantly mitigate critical derived attack scenario, such as injecting malicious payloads to attack the application service and user domains to gain unauthorized access. Moreover, integrating blockchain with DNA-based cryptography provides robust protection against data leakage or theft even if an attacker gains access to the stored data, especially when it is stored on the cloud under third-party custody. Similar blockchain frameworks have been also proposed for applications in smart homes [71], smart cities [72], supply chains [73], and autonomous vehicles [74]. However, blockchain integration with IoT systems is an emerging topic and currently lacks standardized integration guidelines. Therefore, future research and regulatory efforts by organizations are necessary to establish guidelines and regulations for integrating blockchain into existing IoT technologies and applications, facilitating its practical implementation in the global IoT market. The IEEE has initiated a project to develop standards for blockchain in IoT, but currently, there are no globally accepted standards and protocols for blockchain integration in IoT. These are necessary to address interoperability, integration complexities, scalability, and regulatory compliance challenges in IoT [75]. Additionally, the integration of cryptographic algorithms with blockchains that are resilient to quantum attacks will be crucial, although it might amplify a particular disadvantage of blockchains which is their high energy consumption.

Acknowledgment

This work was supported by the Hauts-de-France region and the International Research Project IRP ADONIS, funded by French CNRS, Université de Technologie de Compiègne, and Lebanese University.

References

[1] P. I. R. Grammatikis, P. G. Sarigiannidis, I. D. Moscholios, Securing the internet of things: Challenges, threats and solutions, Internet of Things 5 (2019) 41–70.

[2] S. Zahid, M. S. Mazhar, S. G. Abbas, Z. Hanif, S. Hina, G. A. Shah, Threat modeling in smart firefighting systems: Aligning mitre att&ck matrix and nist security controls, Internet of Things 22 (2023) 100766.

[3] A. Bhardwaj, F. Al-Turjman, M. Kumar, T. Stephan, L. Mostarda, Capturing-the-invisible (cti): Behavior-based attacks recognition in iot-oriented industrial control systems, IEEE access 8 (2020) 104956–104966.

[4] P. Anand, Y. Singh, A. Selwal, P. K. Singh, R. A. Felseghi, M. S. Raboaca, Iovt: internet of vulnerable things? threat architecture, attack surfaces, and vulnerabilities in internet of things and its applications towards smart grids, Energies 13 (2020) 4813.

[5] J. Li, J. Jin, L. Lyu, D. Yuan, Y. Yang, L. Gao, C. Shen, A fast and scalable authentication scheme in iot for smart living, Future Generation Computer Systems 117 (2021) 125–137.

[6] I. S. Foundation, Understanding the Contemporary Use of Vulnerability Disclosure in Consumer Internet of Things Product Companies, Technical Report, IoT Security Foundation, 2018.

[7] F. Siddiqui, J. Beley, S. Zeadally, G. Braught, Secure and lightweight communication in heterogeneous iot environments, Internet of Things 14 (2021) 100093.

[8] K. Sha, W. Wei, T. A. Yang, Z. Wang, W. Shi, On security challenges and open issues in internet of things, Future generation computer systems 83 (2018) 326–337.

[9] F. Chen, Z. Xiao, L. Cui, Q. Lin, J. Li, S. Yu, Blockchain for internet of things applications: A review and open issues, Journal of Network and Computer Applications 172 (2020) 102839.

[10] S. Mathur, A. Kalla, G. Gür, M. K. Bohra, M. Liyanage, A survey on role of blockchain for iot: Applications and technical aspects, Computer Networks 227 (2023) 109726.

[11] M. A. Lawal, R. A. Shaikh, S. R. Hassan, Security analysis of network anomalies mitigation schemes in iot networks, IEEE Access 8 (2020) 43355–43374.

[12] C. Xenofontos, I. Zografopoulos, C. Konstantinou, A. Jolfaei, M. K. Khan, K.-K. R. Choo, Consumer, commercial, and industrial iot (in)security: Attack taxonomy and case studies, IEEE Internet of Things Journal 9 (2021) 199–221.

[13] B. Zhao, S. Ji, W.-H. Lee, C. Lin, H. Weng, J. Wu, P. Zhou, L. Fang, R. Beyah, A large-scale empirical study on thevulnerability of deployed iot devices, IEEE Transactions on Dependable and Secure Computing (2020).

[14] F. Hashmat, S. G. Abbas, S. Hina, G. A. Shah, T. Bakhshi, W. Abbas, An automated context-aware iot vulnerability assessment rule-set generator, Computer Communications 186 (2022) 133–152.

[15] J. R. Nurse, S. Creese, D. De Roure, Security risk assessment in internet of things systems, IT Professional 19 (2017) 20–26.

[16] S. El Jaouhari, E. Bouvet, Secure firmware over-the-air updates for iot: Survey, challenges, and discussions, Internet of Things 18 (2022) 100508.

[17] A. Ur-Rehman, I. Gondal, J. Kamruzzaman, A. Jolfaei, Vulnerability modelling for hybrid industrial control system networks, Journal of grid computing 18 (2020) 863–878.

[18] R. Ross, Guide for Conducting Risk Assessments, Technical Report, National Institute of Standards and Technology (NIST), 2012.

[19] M. Beyrouti, A. Lounis, B. Lussier, A. Bouadallah, A. E. Samhat, Vulnerability and threat assessment framework for internet of things systems, in: 6th Conference on Cloud and Internet of Things (CIoT), IEEE, 2023, pp. 62–69.

[20] H. HaddadPajouh, A. Dehghantanha, R. M. Parizi, M. Aledhari, H. Karimipour, A survey on internet of things security: Requirements, challenges, and solutions, Internet of Things 14 (2021) 100129.
[21] ISO/IEC 30141:2018, Internet of Things (IoT) — Reference architecture, Technical Report ISO/IEC 30141:2018, International Organization for Standardization and International Electrotechnical Commission, 2018.
[22] K. Kandasamy, S. Srinivas, K. Achuthan, V. P. Rangan, Iot cyber risk: A holistic analysis of cyber risk assessment frameworks, risk vectors, and risk ranking process, EURASIP Journal on Information Security 2020 (2020) 1–18.
[23] V. Malamas, F. Chantzis, T. K. Dasaklis, G. Stergiopoulos, P. Kotzanikolaou, C. Douligeris, Risk assessment methodologies for the internet of medical things: A survey and comparative appraisal, IEEE Access 9 (2021) 40049–40075.
[24] P. Napolitano, G. Rossi, M. Lombardi, F. Garzia, M. Ilariucci, G. Forino, Threats analysis and security analysis for critical infrastructures: Risk analysis vs. game theory, in: International Carnahan Conference on Security Technology (ICCST), IEEE, 2018, pp. 1–5.
[25] A. Ur-Rehman, I. Gondal, J. Kamruzzuman, A. Jolfaei, Vulnerability modelling for hybrid it systems, in: IEEE International Conference on Industrial Technology (ICIT), 2019, pp. 1186–1191.
[26] B. F. Zahra, B. Abdelhamid, Risk analysis in internet of things using ebios, in: IEEE 7th Annual Computing and Communication Workshop and Conference (CCWC), IEEE, 2017, pp. 1–7.
[27] B. Ali, A. I. Awad, Cyber and physical security vulnerability assessment for iot-based smart homes, Sensors 18 (2018) 817.
[28] M. H. Bhuyan, N. A. Azad, W. Meng, C. D. Jensen, Analyzing the communication security between smartphones and iot based on coras, in: Network and System Security: 12th International Conference (NSS), Springer, Hong Kong, China, 2018, pp. 251–265.
[29] C. Hankin, P. Malacaria, et al., Attack dynamics: An automatic attack graph generation framework based on system topology, capec, cwe, and cve databases, Computers & Security 123 (2022) 102938.
[30] N. M. Karie, N. M. Sahri, W. Yang, C. Valli, V. R. Kebande, A review of security standards and frameworks for iot-based smart environments, IEEE Access 9 (2021) 121975–121995.
[31] W. Kang, J. Deng, P. Zhu, X. Liu, W. Zhao, Z. Hang, Multi-dimensional security risk assessment model based on three elements in the iot system, in: 2020 IEEE/CIC International Conference on Communications in China (ICCC), 2020, pp. 518–523.
[32] I. Stellios, P. Kotzanikolaou, M. Psarakis, C. Alcaraz, Risk assessment for iot-enabled cyber-physical systems, Advances in Core Computer Science-Based Technologies: Papers in Honor of Professor Nikolaos Alexandris (2021) 157–173.
[33] S. Taubenberger, J. Jürjens, Y. Yu, B. Nuseibeh, Problem analysis of traditional it-security risk assessment methods–an experience report from the insurance and auditing domain, in: Future Challenges in Security and Privacy for Academia and Industry: 26th IFIP TC 11 International Information Security Conference, SEC 2011, Lucerne, Switzerland, June 7-9, 2011. Proceedings 26, Springer, 2011, pp. 259–270.
[34] J. Fei, K. Chen, Q. Yao, Q. Guo, X. Wang, Security vulnerability assessment of power iot based on business security, in: 1st International Conference on Control, Robotics and Intelligent System, 2020, pp. 128–135.
[35] F. Hashmat, S. G. Abbas, S. Hina, G. A. Shah, T. Bakhshi, W. Abbas, An automated context-aware iot vulnerability assessment rule-set generator, Computer Communications 186 (2022) 133–152.
[36] H. Wang, Z. Chen, J. Zhao, X. Di, D. Liu, A vulnerability assessment method in industrial internet of things based on attack graph and maximum flow, IEEE Access 6 (2018) 8599–8609.
[37] V. Casola, A. De Benedictis, M. Rak, U. Villano, Toward the automation of threat modeling and risk assessment in iot systems, Internet of Things 7 (2019) 100056.
[38] V. Shivraj, M. Rajan, P. Balamuralidhar, A graph theory based generic risk assessment framework for internet of things (iot), in: IEEE international conference on advanced networks and telecommunications systems (ANTS), IEEE, 2017, pp. 1–6.
[39] W. Kang, J. Deng, P. Zhu, X. Liu, W. Zhao, Z. Hang, Multi-dimensional security risk assessment model based on three elements in the iot system, in: 2020 IEEE/CIC international conference on communications in China (ICCC), IEEE, 2020, pp. 518–523.
[40] S. Sicari, A. Rizzardi, D. Miorandi, A. Coen-Porisini, A risk assessment methodology for the internet of things, Computer Communications 129 (2018) 67–79.
[41] I. Stellios, P. Kotzanikolaou, C. Grigoriadis, Assessing iot enabled cyber-physical attack paths against critical systems, Computers & Security 107 (2021) 102316.
[42] C. Sánchez-Zas, X. Larriva-Novo, V. A. Villagrá, D. Rivera, A. Marín-Lopez, A methodology for ontology-based interoperability of dynamic risk assessment frameworks in iot environments, Internet of Things (2024) 101267.
[43] M. Ge, J. B. Hong, W. Guttmann, D. S. Kim, A framework for automating security analysis of the internet of things, Journal of Network and Computer Applications 83 (2017) 12–27.
[44] H. L. Hassani, A. Bahnasse, E. Martin, C. Roland, O. Bouattane, M. E. M. Diouri, Vulnerability and security risk assessment in a iiot environment in compliance with standard iec 62443, Procedia Computer Science 191 (2021) 33–40.
[45] F. Arat, S. Akleylek, A new method for vulnerability and risk assessment of iot, Computer Networks 237 (2023) 110046.
[46] X. Duan, M. Ge, T. H. M. Le, F. Ullah, S. Gao, X. Lu, M. A. Babar, Automated security assessment for the internet of things, in: 2021 IEEE 26th Pacific Rim International Symposium on Dependable Computing (PRDC), IEEE, 2021, pp. 47–56.
[47] G. George, S. M. Thampi, Vulnerability-based risk assessment and mitigation strategies for edge devices in the internet of things, Pervasive and Mobile Computing 59 (2019) 101068.
[48] A. Jacobsson, M. Boldt, B. Carlsson, A risk analysis of a smart home automation system, Future Generation Computer Systems 56 (2016) 719–733.
[49] O. Mavropoulos, H. Mouratidis, A. Fish, E. Panaousis, Apparatus: A framework for security analysis in internet of things systems, Ad Hoc Networks 92 (2019) 101743.
[50] G. Bakirtzis, B. J. Simon, A. G. Collins, C. H. Fleming, C. R. Elks, Data-driven vulnerability exploration for design phase system analysis, IEEE Systems Journal 14 (2019) 4864–4873.
[51] T. Jungebloud, N. H. Nguyen, D. Seong Kim, A. Zimmermann, Hierarchical model-based cybersecurity risk assessment during system design, in: IFIP International Conference on ICT Systems Security and Privacy Protection, Springer, 2023, pp. 30–44.
[52] F. Famá, J. N. Faria, D. Portugal, An iot-based interoperable architecture for wireless biomonitoring of patients with sensor patches, Internet of Things 19 (2022) 100547.
[53] H. H. Alshammari, The internet of things healthcare monitoring system based on mqtt protocol, Alexandria Engineering Journal 69 (2023) 275–287.
[54] P. Alqinsi, I. J. M. Edward, N. Ismail, W. Darmalaksana, Iot-based ups monitoring system using mqtt protocols, in: 2018 4th International Conference on Wireless and Telematics (ICWT), IEEE, 2018, pp. 1–5.
[55] K. Grgić, I. Špeh, I. Hei, A web-based iot solution for monitoring data using mqtt protocol, in: 2016 international conference on smart systems and technologies (SST), IEEE, 2016, pp. 249–253.
[56] K. Medhi, N. Ahmed, M. I. Hussain, Dew-based offline computing architecture for healthcare iot, ICT Express 8 (2022) 371–378.
[57] S.-H. Chang, R.-D. Chiang, S.-J. Wu, W.-T. Chang, A context-aware, interactive m-health system for diabetics, IT professional 18 (2016) 14–22.
[58] Z. Abbas, W. Yoon, A survey on energy conserving mechanisms for the internet of things: Wireless networking aspects, Sensors 15 (2015) 24818–24847.
[59] E. Riedel, Mqtt protocol for sme foundries: potential as an entry point into industry 4.0, process transparency and sustainability, Procedia CIRP 105 (2022) 601–606.
[60] T. Wu, F. Wu, C. Qiu, J.-M. Redouté, M. R. Yuce, A rigid-flex wearable health monitoring sensor patch for iot-connected healthcare applications, IEEE Internet of Things Journal 7 (2020) 6932–6945.
[61] B. SIG, Bluetooth Core Specification, Technical Report, Bluetooth Special Interest Group (SIG), 2023. Version 5.4.
[62] G. Oikonomou, S. Duquennoy, A. Elsts, J. Eriksson, Y. Tanaka, N. Tsiftes, The Contiki-NG open source operating system for next generation IoT devices, SoftwareX 18 (2022) 101089.
[63] B. Bi, D. Huang, B. Mi, Z. Deng, H. Pan, Efficient lbs security-preserving

based on ntru oblivious transfer, Wireless Personal Communications 108 (2019) 2663–2674.
[64] W. Alnahari, M. T. Quasim, Authentication of iot device and iot server using security key, in: 2021 International Congress of Advanced Technology and Engineering (ICOTEN), IEEE, 2021, pp. 1–9.
[65] G. Xie, G. Hou, Q. Pei, H. Huang, Lightweight privacy protection via adversarial sample, Electronics 13 (2024) 1230.
[66] M. Eskandari, Z. H. Janjua, M. Vecchio, F. Antonelli, Passban ids: An intelligent anomaly-based intrusion detection system for iot edge devices, IEEE Internet of Things Journal 7 (2020) 6882–6897.
[67] M. G. Samaila, J. B. Sequeiros, T. Simoes, M. M. Freire, P. R. Inacio, Iot-harpseca: a framework and roadmap for secure design and development of devices and applications in the iot space, IEEE Access 8 (2020) 16462–16494.
[68] A. Karale, The challenges of iot addressing security, ethics, privacy, and laws, Internet of Things 15 (2021) 100420.
[69] M. A. Uddin, A. Stranieri, I. Gondal, V. Balasubramanian, A survey on the adoption of blockchain in iot: Challenges and solutions, Blockchain: Research and Applications 2 (2021) 100006.
[70] H. Kaur, R. Jameel, M. A. Alam, B. Alankar, V. Chang, Securing and managing healthcare data generated by intelligent blockchain systems on cloud networks through dna cryptography, Journal of Enterprise Information Management 36 (2023) 861–878.
[71] M. Ammi, S. Alarabi, E. Benkhelifa, Customized blockchain-based architecture for secure smart home for lightweight iot, Information Processing & Management 58 (2021) 102482.
[72] C. Li, X. Li, P. Liu, W. Qiu, C. Yao, B. Yuan, Efficient and traceable data sharing for the internet of things in smart cities, Computers and Electrical Engineering 103 (2022) 108389.
[73] T. K. Agrawal, V. Kumar, R. Pal, L. Wang, Y. Chen, Blockchain-based framework for supply chain traceability: A case example of textile and clothing industry, Computers & industrial engineering 154 (2021) 107130.
[74] C. Oham, R. A. Michelin, R. Jurdak, S. S. Kanhere, S. Jha, B-ferl: Blockchain based framework for securing smart vehicles, Information Processing & Management 58 (2021) 102426.
[75] T. R. Gadekallu, Q.-V. Pham, D. C. Nguyen, P. K. R. Maddikunta, N. Deepa, B. Prabadevi, P. N. Pathirana, J. Zhao, W.-J. Hwang, Blockchain for edge of things: Applications, opportunities, and challenges, IEEE Internet of Things Journal 9 (2021) 964–988.