Categorizing IoT Software Systems Security Vulnerabilities Through Literature Studies

Clinton Hudson Moreira Pessoa
PESC/COPPE/UFRJ
Federal University of Rio de Janeiro
Rio de Janeiro - Brazil
chmp@cos.ufrj.com

Guilherme Horta Travassos
PESC/COPPE/UFRJ
Federal University of Rio de Janeiro
Rio de Janeiro - Brazil
ght@cos.ufrj.com

ABSTRACT

Despite the popularity of IoT software systems and the enormous variety of intelligent devices, there are still security challenges, considering the lack of descriptions of practices that can support the mitigation of security risks, augmenting the uncertainties on the weaknesses encompassing such systems. Therefore, this paper presents the results of two literature studies (ad-hoc and structured) that can contribute to the decision-making regarding mitigating risks associated with security vulnerabilities in IoT software systems. The ad-hoc literature study identified 27 coarse-grained security vulnerabilities from software organizations. The structured literature study identified 69 fine-grained security vulnerabilities from the technical literature, which allowed identifying and categorizing these vulnerabilities into four categories (application, network, device, and Peopleware) for better organization and understanding. The results comparison highlighted a set of 30 most impactful security vulnerabilities that should be considered by software engineers when mitigating the risks regarding the lack of security in IoT software systems.

CCS CONCEPTS

• IoT System • Security • Vulnerability

KEYWORDS

Security, Vulnerabilities, Internet of Things, Contemporary Software Systems, Literature Study, Evidence-based Software Engineering

1 Introduction

The Internet of Things (IoT) paradigm has become a crucial step in the planning and designing of contemporary software systems. It allows the integration of smart devices into a powerful network infrastructure that aids in developing modern software systems, enhancing human users' perceptual capabilities [1]. However, this growth potential contributes to such software systems becoming one of the prime targets for attackers to exploit in the cyber world, providing them with the means to access network-connected devices [2][3].

Therefore, security and privacy have been a concern in smart systems over the past years due to the high risk promoted by IoT environments regarding a lack of security and mechanisms to assess it in IoT devices [4]. The security attribute in IoT software systems requires extra care since such systems deal with data collected and shared by different devices, which usually capture various types of information [5]. In addition, we must be aware of the fragility and dangers lurking within the network, generating numerous threats and highlighting certain vulnerabilities that conventional software systems still face. However, understanding IoT vulnerabilities ensures that threats can be anticipated and mitigation strategies can be devised to minimize, for instance, the risks of intrusion or data theft [6].

Furthermore, it's essential to recognize that while many vulnerabilities identified in IoT may resemble those found in traditional software systems, the unique characteristics of IoT environments often amplify these risks. IoT devices' interconnected nature, diverse functionalities, and resource constraints introduce complexities that demand specialized security considerations [51][52]. Therefore, while some vulnerabilities may transcend both traditional and IoT software systems, the context in which they manifest within IoT ecosystems necessitates tailored mitigation strategies to address them [51] effectively.

The development of IoT software systems revealed all of these issues to our software engineering team. Among many other challenges involved in their engineering [7], the need for clearer information about security vulnerabilities regarding these modern software systems jeopardizes the decision-making in our industrial software projects. Therefore, to fill this critical information gap regarding IoT software systems and respond to concerns about the security vulnerabilities within their construction layers, this research intends to identify and categorize the known IoT security vulnerabilities described in the technical literature, thus providing an evidence-based set of information to support the software practitioners in deciding about the mitigation of risks in their IoT software projects.

Two literature studies [8] were conducted to support the results presented in this work: an ad-hoc review and a structured review. In choosing a structured review over a systematic literature review (SLR), we opted for this replicable approach due to its flexibility, the emerging nature of the topic, and the time and resources needed to filter and review studies. This approach requires less time and resources than a systematic review, involving only a few formal and rigorous steps. Additionally, it allowed us to conduct extensive work with just two researchers and produce useful results for the project.

The two studies were planned and executed to highlight security vulnerabilities in IoT software systems. The ad-hoc review identified 27 coarse-grained security vulnerabilities reported by software organizations, and the structured review identified 69 fine-grained security vulnerabilities in the technical literature. These vulnerabilities were compared and combined, evidencing the 30 major security vulnerabilities in IoT software systems requiring greater attention and treatment. Besides, from the data collected during the structured review, we identified and defined groups that categorically classify vulnerabilities based on the context in which they are recognized (network, application, device and Peopleware) for better organization and understanding. The dataset has been published at https://anonymous.4open.science/r/Files-StudyVulnerabilitiy. It provides further information on the findings, supports the acquaintance of software practitioners with the topic, and allows the software researchers to audit these results and conduct further investigations.

This paper is organized as follows: Section 2 describes the background. Next, section 3 presents the related work. Next, section 4 reports the planning and execution of the research method. Then, in the sequence, sections 5 and 6 report and discuss the results, respectively. Next, section 7 discusses the threats to validity. Finally, section 8 concludes by presenting final remarks.

2 Background

In this section, contextual and relevant information is provided to understand the topics of this work.

2.1 Internet of Things

The Internet of Things paradigm comprises intelligent technologies influencing our lives, providing smart devices designed to share information, data, and resources to meet people's needs [9]. It allows the composition of software systems from uniquely addressable objects (the things), such as fingerprint readers, gas detection systems, temperature monitoring devices, motion detection systems, and home surveillance cameras, among many others, which are equipped with identifying, sensing or actuation behaviors and processing capabilities. Therefore, these things can communicate with each other for various purposes and cooperate to reach a goal [10]. In a smart home, for instance, they assist in energy consumption optimization, cost reduction in bills, and ensuring the occupants' safety [11].

However, with this growing success, many critical security issues in software systems emerged as a menace. As approximately 24 billion devices are expected to be online in the public domain by 2025, various security vulnerabilities may be susceptible to attacks, leading to serious problems if these software systems are not properly protected or configured. In addition, different personal information is collected by various connected devices, such as name, date of birth, address, credit card information, and more [12][13], making privacy also a big concern.

2.2 Security in IoT Software Systems

Security is one of the primary pillars and one of the biggest challenges for IoT software systems. As the number of connected devices increases, the likelihood of exploiting security vulnerabilities also grows. Organizations strive to mitigate security breaches by deploying effective security tools to protect their systems from digital attacks, aiming to prevent, detect, and report attacks using cutting-edge technologies and best practices [14][15].

Unlike traditional computing environments, IoT ecosystems encompass many interconnected devices with diverse functionalities, often operating in resource-constrained environments. This introduces complexities in security management, as each device represents a potential entry point for cyber-attacks [16]. Furthermore, the security measures in IoT software systems demand a nuanced approach. Diverse communication protocols, decentralized management, and dynamic device interactions necessitate tailored security strategies to safeguard against evolving threats [49].

Many critical infrastructures rely on cyber-physical systems, including smart grids, intelligent transportation, critical infrastructure, air transportation, emergency response, and healthcare, amplifying the potential impact of security breaches. [17][18]. For instance, a compromised IoT device in a smart grid could disrupt power distribution, leading to widespread outages and economic losses [50]. It is due to these types of security breaches that there are several implementation challenges to consider in IoT software systems, including, primarily, features associated with preventing disclosure, deception, and disruptions from ensuring elements related to the key pillars of security: confidentiality, integrity, and availability of data [19].

2.3 Security Vulnerabilities in IoT Software Systems

A security vulnerability can be defined as the weakness of an asset or security mechanism that one or more threats can exploit. It can result from a design flaw or implementation defect, enabling an attacker to cause harm to the stakeholders of that asset, as described by ISO/IEC 27000 [20]. The stakeholders include the owner, users, actors, and things relying on the software system. The term "vulnerability" is often used in a very generic manner when merged with the terms “threat” or “attack” [21]. Unlike the concept applied to the term "vulnerability," "threat" is defined as the potential cause of an unwanted incident that is likely to result in harm to a software system or organization. On the other hand, an "attack" is attempting to destroy, expose, alter, steal, or gain unauthorized access to an asset [20].

Given the increasing discovery and exploitation of vulnerabilities, there has been a significant rise in research on their detection and mitigation in IoT environments. It has resulted in many academic papers and research articles addressing IoT security challenges [4][6][22][23]. This is due to the importance of security risk assessment and the development of new security strategies applied to vulnerabilities in IoT software systems [24].

By identifying vulnerabilities, we aim to provide a proactive approach to security, addressing weaknesses in IoT software systems before potential threats or attackers can exploit them. This approach allows for systematic vulnerability assessment and targeted mitigation strategies, ultimately enhancing the resilience of IoT ecosystems against emerging cyber threats. Furthermore, studying how security vulnerabilities propagate, are discovered, and remediated helps strengthen the ecosystem's health, as the delay between vulnerability discovery or the release of its fix can expose assets to threats and increase the likelihood of exploitation [23].

Based on these studies, we can consider different types of security vulnerabilities that may arise, such as Phishing. It occurs when malicious individuals manipulate email messages to lure recipients into opening them, often with the intent of tricking them into revealing sensitive information [6][25][26]. Denial of Service (DoS) happens when a system is overwhelmed with a high volume of simultaneous data requests, rendering the servers unable to handle legitimate requests [26-28]. Injection occurs when malicious code is inserted into servers using programming languages like SQL, aiming to make the server disclose confidential data or perform unintended actions [26][29][30], among others. These are just a few security vulnerabilities that can emerge in IoT software systems. Therefore, it is crucial to understand and address these and other vulnerabilities to ensure the security and integrity of IoT software systems.

3 RELATED WORKS

This section presents studies that aim to identify security vulnerabilities in IoT software systems. Each study provides an important contribution and yields results with similar insights to our work by addressing diverse approaches and scenarios and presenting their findings clearly and objectively.

Davis et al. [6] discuss the issues of adopting IoT technologies and the consequent demand for security, making many of these devices vulnerable to attacks. In their study, the authors investigate vulnerability and security posture for smart home IoT devices. The study begins with a literature review on known security vulnerability studies of IoT devices, considering four categories of attacks: 1) physical, 2) network, 3) software, and 4) encryption. It is followed by conducting experimental studies that compare the security postures between well-known and lesser-known device vendors. The authors conclude that physical, network, software, and/or encryption attacks are feasible for various IoT devices. Additionally, based on their security vulnerability studies, the authors conclude that the security posture in well-known devices is stronger than that of lesser-known ones.

In [31], Chhetri and Motti focus on security and privacy provisions for smart home devices. The authors adopt a systematic approach to extract, analyze, and categorize security vulnerabilities from these environments. The study yielded 153 security vulnerabilities, with categories based on the occurrence location or component of the smart home architecture, such as device, protocol, gateway, network, and software architecture. The authors hope the results can benefit other researchers with a comprehensive analysis and systematic categorization of smart home vulnerabilities.

Pedreira et al. [32] review security vulnerabilities, attacks, and defenses in Industry 4.0. Their review presents articles on these three main topics (security vulnerabilities, defenses, and attacks) or their intersection. The identified vulnerabilities were classified into four categories: web applications, devices, networks, and authentication. For each category, some of the associated types of vulnerabilities are presented. The results of this study also show that the number of articles focusing on vulnerabilities is relatively low compared to those focusing on attacks and defenses. Overall, the study provides insights into the current state of research on vulnerabilities in Industry 4.0 and highlights the need for further investigation.

As in the works mentioned in this section, as it is in our study, it is crucial to examine and capture an appropriate set of security vulnerabilities affecting IoT software systems using literature reviews. However, we stand out by adopting specific approaches to categorize vulnerabilities, introducing the 'Peopleware' category. Unlike other works, we focus on capturing weaknesses strictly defined as vulnerabilities—distinct from threats or attacks, aligned with ISO/IEC 27000 standards. This approach offers a more transparent and targeted perspective on the challenges to be addressed and mitigated in IoT software systems.

4 RESEARCH METHOD

There is an increasing demand for studies on frameworks that assess and quantify security vulnerabilities in IoT software systems. This demand is due to the high impact of security risk assessments and the development of security strategies on IoT [24]. However, while there are research papers that enhance our understanding of IoT security vulnerabilities, there are still certain gaps regarding those that need to be better defined due to the heterogeneity of scenarios encompassing IoT devices. Moreover, it raises questions about the key security needs that keep IoT software systems under constant surveillance. Based on this need, we aimed to investigate such security vulnerabilities by applying two literature studies, an ad-hoc review and a structured review, thus seeking to cover a wider information set.

The Ad-Hoc review can be performed without strong, descriptive rigor [33]. This review focuses on the information provided by software organizations on the web. However, to specify the steps used in identifying the security vulnerabilities, we followed the following steps: (a) to select a commonly used search engine for direct and informal searches, Google; (b) to use the search strings "vulnerabilities in IoT systems" and "Security Vulnerabilities in IoT Software Systems"; (c) to select industrial websites providing information corresponding to the search, limited to the first page of results to identify the most prominent ones.

The search yielded seven websites explicitly mentioning security vulnerabilities, presented in Table 1. We limited our analysis to these sources of information, considering the referencing factor between pages, resulting in security vulnerabilities previously indicated in referenced documents.

Table 1: Web Sites Identified in Ad-hoc Review

Source	Description
OWASP	OWASP is an open community dedicated to enabling organizations to conceive, develop, acquire, operate, and maintain applications that can be trusted.
NIST National Vulnerability Database (NVD)	The NVD is a vulnerability database maintained by the United States National Institute of Standards and Technology (NIST). It includes information about vulnerabilities in IoT devices.
FORTINET	Fortinet continues to be a driving force in the evolution of cybersecurity and the convergence of networking and security.
ZDNET	ZDNet is a business technology news website with global tech news, advice, and insights.
INFOSEC	Infosec is a website that provides the most recent information and updates on cybersecurity topics, security education trends, and cyber threats.
LINKEDIN	LinkedIn is a professional networking site designed to help people make business connections, share their experiences and resumes, and find jobs.
RESILLION	It delivers digital transformation, cyber security, and quality assurance solutions, enabling your clients to embrace and harness the power of the digital future.

Based on the investigative process in this study stage, 27 vulnerabilities were identified from these web sources. The cataloged security vulnerabilities were aggregated to avoid duplication, even if mentioned in different sources or using different names. Table 2 presents the vulnerabilities identified during the execution of the ad-hoc review.

Table 2: Security Vulnerabilities from the Ad-hoc Review

ID	Vulnerabilities
AD1	Insecure Data Transfer and Storage
AD2	Weak Passwords
AD3	Insecure Update Mechanisms
AD4	Insufficient Physical Security
AD5	Insufficient Privacy Protection
AD6	Lack of Device Management
AD7	Insecure Network Services
AD8	Insecure Ecosystem Interfaces
AD9	Manipulating the Code Execution
AD10	Lack of Encryption
AD11	Application Vulnerabilities
AD12	Incorrect Access Control
AD13	Intrusion Ignorance
AD14	Lack of Trusted Execution Environment
AD15	Outdated Software
AD16	Overly Large Attack Surface
AD17	User Interaction
AD18	Vendor Security Posture
AD19	Insecure Default Settings
AD20	Insecure or Outdated Components
AD21	TCP/IP Stacks
AD22	Account Lockout
AD23	Insecure Third-party Components
AD24	Obtaining Console Access
AD25	Lack of Two-factor Authentication
AD26	Update Location Writable
AD27	Username Enumeration

It is worth noting that among the security vulnerabilities highlighted on the identified websites, two items that were initially considered security vulnerabilities had to be excluded from the results (marked in Table 2): Denial of Service (DoS), which involves flooding and compromising services with spoofed packets, resulting in severe disruptions of the provided services; and Botnet, an array of Internet-connected devices designed to compromise networks, steal data, or send spam. This decision was made based on how we defined the term "vulnerability" in this study, where both items fall under the definition of threats rather than vulnerabilities.

4.1 Structured Review Planning

The adoption of a structured literature review required the use of a more well-defined and clear research protocol. Therefore, it inspires itself on the principles of Systematic Literature Reviews [34]. Although it does not include some steps of a complete systematic literature, a Structured Review utilizes a systematic and replicable protocol [35]. The research protocol consists of three main stages: a) Planning: In this stage, we establish the practical problem to be addressed by the review, provide the basic research question, and define the research protocol; b) Extraction Procedure: In this phase, we extract information from the selected studies based on the criteria defined in the study protocol; and c) Report: In this phase, we synthesize and present the data identified in the study's results.

By following this structured approach, we aim to ensure rigor and reproducibility in identifying and analyzing vulnerabilities in IoT software systems. However, we need a unified solution addressing security demands in developing such systems. Therefore, understanding the key vulnerability points in such systems can help mitigate a significant portion of the major and common risks associated with these software systems. Hence, this work aims to identify security vulnerabilities in software systems and IoT devices.

The formulated general research question, "What vulnerabilities affect and can be identified in IoT software systems?" is of great significance in addressing the security issue in these systems. By seeking to answer this research question, valuable insights are expected to be obtained regarding the specific security vulnerabilities that affect them.

After defining the research question, the next step is establishing the search strategy. We used the Scopus database to conduct the Structured Review and search for relevant sources of information. Scopus was selected based on its prominence and relevance as a search engine, which integrates a wide range of technical literature from various digital libraries in its collection [36]. Combined with the snowballing procedure, including cited articles or articles that mention the studies identified in the structured review, this approach can reinforce the knowledge base through a representative set of primary sources on the topic of interest and support discoveries [37].

The next step in the planning stage is to define the inclusion and exclusion criteria. As shown in Table 3, these criteria will be used in the extraction phase of the Structured Review to determine which studies contribute to addressing the practical problem.

Table 3: Inclusion and Exclusion Criteria

Criteria	Description
Inclusion Criteria	Must meet the defined research question
	Peer Reviewed
	Full text must be available
Exclusion Criteria	Duplicated studies
	Not written in English
	Not an article or a conference paper

We used all the previous information to create a search string corresponding to the criteria defined for the Structured Review. Then, the search was restricted to using specific keywords to find relevant publications. The search expression was determined following the PICOC principle [38], using the parameters "Population," "Intervention," "Outcome," and "Context".

Table 4 shows the search sequence used in the Scopus database to find related studies. It is worth noting that the searches conducted in both the ad hoc and the structured literature reviews were carried out between June and October 2022.

Table 4: Search Expression Used in Scopus Database

For Investigation by Search Expression	
Population	"ambient intelligence" OR "assisted living" OR "multiagent systems" OR "systems of systems" OR "Cyber-Physical Systems" OR "Industry 4" OR "fourth industrial revolution" OR "web of things" OR "contemporary software systems" OR "smart manufacturing" OR “digitalization” OR “digitization” OR "digital transformation" OR "smart cit*" OR "smart building" OR "smart health" OR "smart environment" OR "smart grid"
Intervention	"security" OR "vulnerability" OR "weakness" OR "Invasion" OR "threat" OR "attack" OR "anomaly" OR "malware" OR “confidentiality” OR “auditability” OR “risk”
Comparison	Not available
Outcome	“taxonomy” OR “categories” OR “classification” OR “Catalog”
Context	"internet of things" OR "Internet of Everything" OR “IoT”
Final Search String used in Scopus	
TITLE-ABS-KEY (("ambient intelligence" OR "assisted living" OR "multiagent systems" OR "systems of systems" OR "Cyber-Physical Systems" OR "Industry 4" OR "fourth industrial revolution" OR "web of things" OR "contemporary software systems" OR "smart manufacturing" OR “digitalization” OR “digitization” OR "digital transformation" OR "smart cit*" OR "smart building" OR "smart health" OR "smart environment" OR "smart grid" OR “autonomous system”) AND ("security" OR "vulnerability" OR "weakness" OR "Invasion" OR "threat" OR "attack" OR "anomaly" OR "malware" OR "confidentiality" OR "auditability" OR "risk" OR "menace") AND ("taxonomy" OR "categories" OR "classification" OR "Catalog") AND ("internet of things" OR "Internet of Everything" OR "IoT"))	

4.2 Structured Review Extraction Procedure

In this stage, we chose and extracted data from selected studies. The extraction began with defining a filter strategy to evaluate articles based on set criteria. We assessed titles and abstracts, excluding studies that did not align with the research question. Studies passing this filter were fully read, and if they addressed the research question without meeting exclusion criteria, they were included in the final list.

After defining the filter strategy, we used Scopus to retrieve 491 documents. Post title and abstract analysis, 76 papers were selected. Full-text reading further reduced this to 39. These papers initiated a snowballing process, yielding 43 forward and 86 backward papers. To minimize bias, both researchers conducted the filtering process, compared results, and reached a consensus. Ultimately, 168 papers were selected, as depicted in Fig 1.

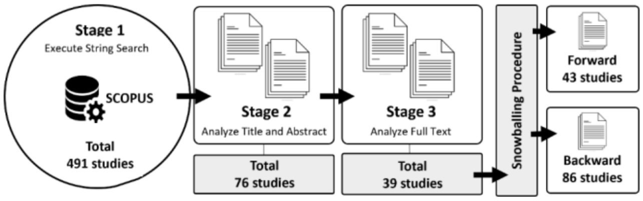

Figure 1: The filtering strategy of studies

In the last step of the extraction procedure, the data were extracted using a data collection protocol. The procedure involved using a standardized form for each selected document. Table 5 presents the form used to extract relevant information from the documents for further analysis.

Table 5: Data Collection Fields

Publication:	
Title	Indicates the article title
Author(s)	Lists the author’s name
Source	Indicates the journal or conference proceedings or book in which the article was published

Year	Indicates the year in which the article was published
Abstract	Copy of the abstract to facilitate further analysis
Data Derived from the Objective:	
Vulnerabilities	What are the vulnerabilities in IoT software systems that the study highlights?

4.3 Report

After the data extraction, the results were analyzed based on the identified responses in each data extraction form, as presented in Table 5. The subsequent analysis and discussion are detailed in the following section.

5 Structured Review Results

This section presents the results of the structured review process. The analyses are conducted to address the study's main question, identifying security vulnerabilities in IoT software systems.

The systematic planning of this review stage allowed for a more rigorous extraction of identified vulnerabilities from the selected studies. The extraction stage assessed vulnerability specifications in studies through “codings,” capturing key excerpts. This allowed for a detailed analysis of each vulnerability and comparison across studies.

The coding strategy employed in this study was based on the Grounded Theory method, which involves systematically collecting and analyzing data to develop a theory [39]. In this study, the approach systematizes data and generates related categories. This process aids in identifying and grouping vulnerabilities, which, despite varying forms and contexts, address the same security issue. This grouping also helps define and categorize vulnerabilities in IoT software scenarios: Device, Network, Application, and Peopleware.

The security vulnerabilities classified under the Device category can be exploited through physical access to the hardware. The Network category encompasses weaknesses related to communication or traffic within the IoT network. In the Application category, security vulnerabilities are associated with software system weaknesses. Lastly, the Peopleware category classifies IoT security weaknesses directly related to the human factor.

Table 6 lists the 69 identified and cataloged security vulnerabilities and their categories.

Table 6: Security Vulnerabilities from the Structured Review

ID	Vulnerabilities	Category
VUL1	Broken Authentication	Application
VUL2	Buffer Overflow	Application
VUL3	Data Inconsistency	Application
VUL4	Insecure Access Management	Application
VUL5	Insecure Interface Configuration	Application
VUL6	Insecure Management of Data	Application
VUL7	Insecure Software	Application
VUL8	Lack of Active Device Monitoring	Application
VUL9	Low-Quality Level Code	Application
VUL10	Non-repudiation	Application
VUL11	SQL Injections	Application
VUL12	Weak/lack of In-app Encryption	Application
VUL13	Malicious code in-app	Application
VUL14	Systems Low-cost	Device
VUL15	Channel Voice	Device
VUL16	Default Configuration	Device
VUL17	Device Spoofing	Device
VUL18	Electromagnetic Emanations Leaking	Device
VUL19	Energy Restraints	Device
VUL20	Heterogeneous Interaction	Device
VUL21	Insecure Data Transfer and Storage	Device
VUL22	Insecure Firmware	Device
VUL23	Insecure Initialization	Device
VUL24	Insecure Password	Device
VUL25	Insufficient Testing	Device
VUL26	Lack of Side Channel Protection	Device
VUL27	Lack of Strong Authentication	Device
VUL28	Low Computing Power	Device
VUL29	Low Data Transmission Range	Device
VUL30	Malicious Code Injection	Device
VUL31	Obtaining Console Access	Device
VUL32	Physical Damage	Device
VUL33	Physical Tampering	Device
VUL34	Sleep Deprivation	Device
VUL35	Tag Cloning	Device
VUL36	Unprotected Physical Access	Device
VUL37	Weak Access Control	Device
VUL38	Weak/lack of Encrypt	Device
VUL39	Insecure physical interface	Device
VUL40	Channel Interference	Network
VUL41	Communication Overhead	Network
VUL42	Data Leak or Breach	Network
VUL43	Eavesdropping	Network
VUL44	Fake/Malicious Node	Network
VUL45	Heterogeneous Communication	Network
VUL46	Insecure Server	Network
VUL47	Insecure Update Mechanisms	Network
VUL48	Lack of Proper Authentication Mechanisms	Network
VUL49	Lack of Strong Password	Network
VUL50	Lack of Secure Communication Protocols	Network

VUL51	Configure network repeatedly	Network
VUL52	Spoofing Signal	Network
VUL53	Unauthorized Access	Network
VUL54	Unsecured Network	Network
VUL55	Unused Ports Enable	Network
VUL56	Weak/lack of Encryption in Communication	Network
VUL57	Physical properties of the power system	Network
VUL58	Wifi De-authentication	Network
VUL59	Insecure traffic control	Network
VUL60	Centralized architecture	Network
VUL61	Access Malicious Link	Peopleware
VUL62	Identifying the Product Vendor	Peopleware
VUL63	Knowledge the System	Peopleware
VUL64	Lack of Technical Support	Peopleware
VUL65	Personal and Social Circumstances	Peopleware
VUL66	Phishing	Peopleware
VUL67	Social Engineering	Peopleware
VUL68	Untrusted Device Acquisition	Peopleware
VUL69	Vendor Security Posture	Peopleware

The QDAminer tool [53] was key in categorizing excerpts from the papers. It facilitated vulnerability analysis, category organization, and information extraction, aiding in the overall synthesis of findings, as shown in Figure 2.

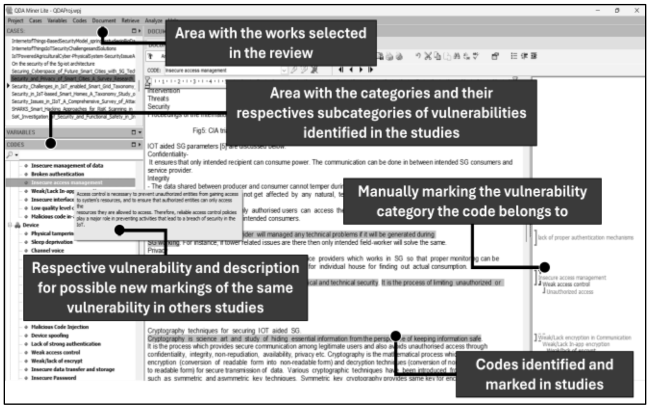
Area with the works selected in the review
Area with the categories and their respectives subcategories of vulnerabilities identified in the studies
Manually marking the vulnerability category the code belongs to
Respective vulnerability and description for possible new markings of the same vulnerability in others studies
Codes identified and marked in studies

Figure 2: Classification of Security Vulnerabilities Using the QDAMiner Lite Tool

It is important to note that some security vulnerabilities may appear repeated. However, what needs to be observed is that within each category, the risk dimensions associated with certain vulnerabilities may vary depending on the context. Therefore, it requires addressing them based on the specific needs or prioritization within the IoT software system. For example, consider the vulnerability [VUL6] "Insecure Management of Data," which relates to vulnerabilities associated with the lack of privacy or security in shared or stored data within the Application category. Similar issues may also be found in Devices and Networks contexts. To clarify the contexts where the same problem may arise, we have listed [VUL21] for the Device category and [VUL42] for the Network category, reflecting the specific contexts highlighted in the extracted studies.

Therefore, although certain vulnerabilities may have similarities, their categorization and differentiation based on contextual factors allow for a more comprehensive understanding and targeted approach to addressing them within the IoT environment. Furthermore, each vulnerability has its description, which enables an account of the problem that each one represents. The explanation for each vulnerability can be found at: https://anonymous.4open.science/r/Files-StudyVulnerabilitiy.

6 Discussion

Certain security vulnerabilities were more prominent in the extracted data, requiring attention due to their recurring presence and threat to IoT software system integrity. Based on this need, Table 7 contains the vulnerability points that overlap among the reported data from the ad hoc and structured reviews. This mapping seeks to highlight the most common vulnerabilities requiring more attention.

Table 7: Security Vulnerabilities Highlighted in the Literature Studies

***Ad-Hoc* Review**	**Structured Review**	**Vulnerabilities**	**Category**
AD13	VUL53	Unauthorized Access	Network
AD26	VUL48	Lack of Proper Authentication Mechanisms	Network
AD1, AD5	VUL21	Insecure Data Transfer and Storage	Device
AD26	VUL27	Lack of Strong Authentication	Device
AD15	VUL33	Physical Tampering	Device
AD13	VUL37	Weak Access Control	Device
AD1, AD5	VUL42	Data Leak or Breach	Network
AD11	VUL56	Weak/lack of Encryption in Communication	Network
AD14	VUL44	Fake/Malicious Node	Network
AD12, AD13	VUL4	Insecure Access Management	Application
AD11	VUL38	Weak/leak of Encrypt	Device
AD3, AD20, AD27	VUL22	Insecure Firmware	Device
AD22	VUL50	Lack of Secure Communication Protocols	Network
AD3, AD12,	VUL7	Insecure Software	Application

AD16, AD21			
AD14	VUL17	Device Spoofing	Device
AD10, AD25	VUL31	Obtaining Console Access	Device
AD12, AD26	VUL1	Broken Authentication	Application
AD1, AD5, AD12	VUL6	Insecure Management of Data	Application
AD20	VUL16	Default Configuration	Device
AD4	VUL32	Physical Damage	Device
AD2	VUL24	Insecure Password	Device
AD7	VUL54	Unsecured Network	Network
AD2	VUL49	Lack of Strong Password	Network
AD8, AD12	VUL5	Insecure Interface Configuration	Application
AD11, AD12	VUL12	Weak/lack of In-app Encryption	Application
AD6, AD12	VUL8	Lack of Active Device Monitoring	Application
AD8	VUL39	Insecure physical interface	Device
AD3	VUL47	Insecure Update Mechanisms	Network
AD18	VUL64	Lack of Technical Support	Peopleware
AD19	VUL69	Vendor Security Posture	Peopleware

The security vulnerabilities were grouped based on their relevant descriptions, which is why there is a relationship between multiple vulnerabilities across the studies, as presented in Table 7. This grouping is because the structured review had specific categories for vulnerabilities, different from those highlighted in the ad-hoc review.

The order used in Table 7 is based on the frequency of citations in the selected primary sources, as determined by the data from the structured review. Based on this, we highlight, for observation purposes, the vulnerability points that had the highest incidence in the selected works and are prominent in both literature studies: Unauthorized Access, Lack of Proper Authentication Mechanisms, and Insecure Data Transfer and Storage.

Unauthorized Access plays a crucial role in preventing activities that lead to a security breach in IoT software systems, as access control is necessary to prevent unauthorized entities from gaining access to system resources and ensure that authorized entities can only access the resources they are permitted to access [40]. The Lack of Proper Authentication Mechanisms is also viewed with great criticality in smart systems, as without strong authentication, it becomes easy for attackers to masquerade as legitimate users and use credentials or any other information that grants them access to IoT environment resources. [41]. Insecure Data Transfer and Storage in devices is also one of the characteristics that mark a significant IoT concern. The large number of devices that can collect and transfer sensitive data to databases or cloud storage poses substantial risks if any data were exposed [30]. It is important to highlight that both security vulnerabilities are directly associated with IoT services' network/communication context, emphasizing IoT's significant influence and impact on improving the global network infrastructure due to the new demands it imposes [13].

We also emphasize that the highlighted security vulnerability items have a significant impact and relevance when we observe that such results resemble the findings in the works of [6] and [31], which, despite having a contextual restriction focused on smart home devices, align a myriad of challenges and vulnerabilities to promote greater alignment regarding the challenges that affect IoT scenarios.

While these security vulnerabilities had a higher incidence, it is essential to note that all other vulnerabilities listed in Table 7 are equally important to address to achieve more secure IoT software systems. Finally, regarding the quantitative aspect of studies that cite and reinforce the highlighted arguments in the study, we provide a link[1] to a technical report that details the methodology and results of the structured review study.

By adopting a specific definition of 'Vulnerability' for item categorization, we can observe that we identify a relatively smaller set than [31]. However, it is important to highlight that this differentiation is valid for both studies. On the contrary, this more specific approach yields more precise data by restricting the inclusion of items directly associated with threats or attacks in the results.

An important item is the categories aligned with each security vulnerability identified in the structured review. The structure achieved resembles those defined in the works of [6], [31], and [32], except for the Peopleware category, which had yet to be described in previous studies addressing security vulnerabilities in IoT software systems. It indicates an important observation component that can directly impact the risks associated with security vulnerabilities related to human agents.

It is worth noting that, despite the significant impact of the human agent on software systems, few studies mention it as a point of vulnerability in the system. Moreover, those who say it usually do not emphasize this issue or directly link the weakness to the human factor. As an example, we highlight two vulnerabilities categorized as 'Peopleware' in Table 6: [VUL66] Phishing, used to induce people to enter their personal information, download malicious software capable of spreading malware or manipulation of sensor data to provide false information that can impact decision making; and [VUL67] Social Engineering, which involves manipulating users to extract private information, confidential data, or information that can be used to gain access to networks in smart environments. Based on these two vulnerabilities, we can understand the potential danger of system weaknesses. Therefore, more stringent control of the human role in systems becomes crucial to minimize significant threats and subsequent attacks.

[1] https://anonymous.4open.science/r/Files-StudyVulnerabilitiy

Analyzing the comparison data presented in Table 7, two vulnerabilities related to 'Peopleware' stood out, specifically [VUL64] Lack of Technical Support and [VUL69] Vendor Security Posture. These points are identified as major vulnerabilities in this category, mainly associated with negligence by some IoT device providers. In certain situations, these providers must provide the necessary security guidelines to users. While we cannot infer the specific impact based on our data, it is known that such neglect has consequences. Therefore, addressing them with the same importance attributed to other vulnerabilities is crucial.

Another relevant point is the specificity of vulnerabilities in IoT, as many of them can be shared by IoT and traditional software systems. When analyzing the vulnerabilities highlighted Table 6, we notice that the main difference lies in the Device vulnerabilities and some specific vulnerabilities in the other categories, as we can see in Table 8:

Table 8: Specific IoT vulnerabilities

ID	Vulnerabilities	Category
VUL15	Channel Voice	Device
VUL17	Device Spoofing	Device
VUL18	Electromagnetic Emanations Leaking	Device
VUL19	Energy Restraints	Device
VUL22	Insecure Firmware	Device
VUL26	Lack of Side Channel Protection	Device
VUL28	Low Computing Power	Device
VUL29	Low Data Transmission Range	Device
VUL31	Obtaining Console Access	Device
VUL32	Physical Damage	Device
VUL33	Physical Tampering	Device
VUL34	Sleep Deprivation	Device
VUL35	Tag Cloning	Device
VUL36	Unprotected Physical Access	Device
VUL8	Lack of Active Device Monitoring	Application
VUL40	Channel Interference	Network
VUL68	Untrusted Device Acquisition	Peopleware

The category Device is unique to IoT software systems, as it encompasses sensors, actuators, and devices directly involved in capturing and transmitting remote data, which is not characteristic of traditional systems. Furthermore, as mentioned earlier, although security vulnerabilities in IoT often share characteristics with those in traditional software systems, mitigation strategies may differ, sometimes requiring a more careful approach. It is also important to emphasize that the fact that some vulnerabilities are common in traditional scenarios should not diminish the importance of IoT-specific vulnerabilities. IoT is subject to the same weaknesses as a conventional software system, and it is essential to address them appropriately.

6.1 Solutions and Best Practices Recommendation

Several potential solutions can help mitigate the weaknesses inherent in security vulnerabilities in IoT software systems. If adopted, they can reduce the risks of threats to these systems. Table 7 highlights some solutions and best practices for the three security vulnerabilities.

Unauthorized Access [42][43][46]:

- **Strong Authentication**: Implement a strong authentication system for IoT devices. This authentication may include using strong passwords, two-factor authentication (2FA), and digital certificates to verify devices' identities.
- **Network Segmentation**: Isolate your IoT network from the rest of the IT infrastructure. Use VLANs (Virtual LANs) or separate networks to ensure that IoT devices cannot be easily accessed from other points in the network.
- **Audit and Monitoring**: Establish a continuous audit and monitoring system to track activities on IoT devices and identify anomalous behaviors.
- **Identity Management**: Use identity management solutions to control and manage the identities and privileges of users and IoT devices.

Lack of Proper Authentication Mechanisms [44-46]:

- **Device Authentication:** Implement strong authentication using complex passwords or cryptographic authentication keys. Additionally, utilize digital certificates to verify devices' identities and ensure their authenticity.
- **Physical Control:** Keep IoT devices physically secure to prevent unauthorized access. Implement physical protection measures such as locks and alarms.
- **Compliance with Standards and Regulations:** Be aware of relevant security regulations and standards, such as the GDPR (General Data Protection Regulation) in the European Union and follow applicable guidelines.

Insecure Data Transfer and Storage [46-48]:

- **Data Encryption**: Implement robust encryption to protect data in transit and at rest. Use secure protocols like HTTPS for data transmission and disk encryption for stored data.
- **Key Protection:** Keep encryption keys secure and out of reach from attackers. Use hardware like Hardware Security Modules (HSMs) to safeguard the keys.
- **Mutual Authentication:** Configure mutual authentication between IoT devices and servers to ensure both parties authenticate before exchanging data.
- **Virtual Private Networks (VPNs):** Use VPNs to create secure tunnels for data transmission between IoT devices and servers, especially on untrusted networks like the public Internet.
- **Vendor Contracts:** Ensure your IoT device suppliers implement adequate data transfer and storage security measures.
- **Secure Firmware Updates:** Keep IoT device firmware up to date to address known vulnerabilities that could impact data storage and transmission.

The recommendations are just a subset of strategies for software system protection, guiding researchers and practitioners. We emphasize the need for more studies to uncover and share mitigation strategies for other vulnerabilities. This aims to deeply understand IoT software system weaknesses and effective ways to lessen their impact.

7 Threats to Validity

Some implications and limitations regarding this study's results should be highlighted. First, we acknowledge that we do not have control over the integrity of the vulnerability listings. There may be researcher bias, where certain expectations or predispositions can influence data collection, analysis, or interpretation. For this reason, some vulnerabilities or studies may have been overlooked during the selection and extraction process.

We used a less rigorous process for the ad hoc review results in selecting security vulnerabilities. Many of them were identified from documents available on selected industrial websites, which does not guarantee the consistency of the collected data and the perception process of these security vulnerabilities. Despite its flexible and less structured nature, we acknowledge that using the Ad-Hoc review positively impacted identifying organizational vulnerabilities. It enabled a quick response to changes and new information, which is crucial in dynamic organizational environments. The absence of a rigid structure allowed for immediate adaptation to new trends and emerging standards, facilitating a more intuitive and direct detection of vulnerabilities, which later served as the basis for identifying other vulnerabilities.

We further highlight that, for the snowballing process, the bias of pre-defined data can lead to assumptions of data correlation based on the vulnerabilities identified in the previous studies (ad-hoc and structured reviews), which limits the emergence of "new information" based on different vocabularies, as for this work, vulnerabilities were delimited based on their definitions.

We also have a temporal bias due to the date of cataloging and analysis of the results, where new sources or observations may emerge based on factors unrelated to our study's intervention date. However, this is a menace that literature studies always face.

8 Conclusions

Given the significant expansion that IoT software systems have undergone in recent years, their potential for engagement in various sectors is evident. It necessitates special attention to how IoT devices manage and manipulate data, especially considering the wide variety of sensitive data. Therefore, security in IoT software systems becomes crucial for properly operationalizing this technology. Among the primary vectors associated with security, software vulnerabilities stand out as a field of study with a significant impact on mitigating the damage caused by threats or attacks that can directly interfere with the performance of IoT technologies.

This study presented a set of security vulnerabilities identified in the context of IoT software systems based on two literature studies: ad-hoc and structured reviews. Initially, our study focused on surveying security vulnerabilities based on ad-hoc reviews of seven organizations' websites, where we cataloged 27 vulnerabilities. After obtaining a baseline structure of security vulnerabilities, we proceeded to the structured literature review, where we identified a total of 70 security vulnerabilities, classified into four categories related to specific scenarios of IoT software systems: Device, Application, Network, and an additional one, Peopleware, which encompasses security vulnerabilities related to human factors. The final set of security vulnerabilities demonstrates the unity between vulnerabilities identified within the organization (using the ad-hoc review) and those found in the structured literature review, thus presenting the 30 major security vulnerabilities in IoT software systems that require greater attention and treatment. Among them, we present three vulnerabilities with the best highlight: Unauthorized Access, Lack of Proper Authentication Mechanisms, and Insecure Data Transfer and Storage, for which we also described a small set of recommendations with possible strategies and solutions to mitigate such menaces.

The results obtained in these literature studies are helping us to mitigate the risks associated with the lack of security in our IoT software systems projects. We hope they can also be useful to your software projects. As a follow-up to this research, there is an intention to explore the security vulnerabilities and findings further to develop a security vulnerability catalog with a comprehensive set of information that can guide the construction of more secure IoT software systems and inspire new investigations in the field. This catalog aims to provide valuable insights and recommendations for improving security measures in IoT environments. Additionally, to measure the impact of these findings, we would consider using models like the Common Vulnerability Scoring System (CVSS), which can significantly enhance the accuracy of assessing the severity of vulnerabilities. Although this approach was not initially considered, it would add value to future studies. Furthermore, studies are needed to determine the impacts of the human factor on the security aspects of IoT software systems, including associated risks and potential supporting countermeasures.

ACKNOWLEDGMENTS

This research was funded by Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), finance code 001. Prof. Travassos is a CNPq Researcher and CNE FAPERJ.

REFERENCES

[1] Song, L. and García-Valls, M. 2022. Improving Security of Web Servers in Critical IoT Systems through Self-Monitoring of Vulnerabilities. Sensors 22, 5004. https://doi.org/10.3390/s22135004.

[2] Siboni, S., Sachidananda, V., Meidan, Y., Bohadana, M., Mathov, Y., Bhairav, S., Shabtai, A., Elovici, Y. 2019. Security Testbed for Internet-of-Things Devices. IEEE Transactions on Reliability 68, 23–44. https://doi.org/10.1109/TR.2018.2864536.

[3] Bochie, K., Gonzalez, E., Giserman, L., Campista, M., Costa, L. 2020. Detecção de Ataques a Redes IoT Usando Técnicas de Aprendizado de Máquina e Aprendizado Profundo. XX SBSEG. SBC, Brasil, pp. 257–270. https://doi.org/10.5753/sbseg.2020.19242.

[4] Abdalla, P. and Varol, C. 2020. Testing IoT Security: The Case Study of an IP Camera. 8th IEEE ISDFS. Beirut, Lebanon, pp. 1–5. https://doi.org/10.1109/ISDFS49300.2020.9116392.

[5] Khan, M. and Salah, K. 2018. IoT security: Review, blockchain solutions, and open challenges. Future Generation Computer Systems 82, 395–411. https://doi.org/10.1016/j.future.2017.11.022.

[6] Davis, B., Mason, J., Anwar, M. 2020. Vulnerability Studies and Security Postures of IoT Devices: A Smart Home Case Study. IEEE Internet of Things Journal 7, 10102–10110. https://doi.org/10.1109/JIOT.2020.2983983.

[7] Da Silva, D., Souza, B. P., Gonçalves, T., and Travassos, G, Uma Tecnologia para Apoiar a Engenharia de Requisitos de Sistemas de Software IoT. 2020. XXIII Ibero-American Conference on Software Engineering. Curitiba, Brazil (Online), p S09 P3:14 pages.

[8] Kuhrmann et al. Kuhrmann, M., Fernández, D. M., Daneva, M. 2017. On the pragmatic design of literature studies in software engineering: an experience-based guideline. ESE 22.6.

[9] Atzori, L., Iera, A. and Morabito, G. 2010. The Internet of Things: A survey. Computer Networks, vol. 54, nº 15, p. 2787–2805, out. 2010, doi: 10.1016/j.comnet.2010.05.010.

[10] Motta, R. C., Silva, V. and Travassos G. H. 2019. Towards a more in-depth understanding of the IoT Paradigm and its challenges. JSERD, vol. 7, p. 3, ago. 2019, doi: 10.5753/jserd.2019.14.

[11] Aldahmani, A., Ouni, B., Lestable, T., Debbah, M. 2023. Cyber-Security of Embedded IoTs in Smart Homes: Challenges, Requirements, Countermeasures, and Trends. IEEE Open Journal of Vehicular Technology. 4, 281–292. https://doi.org/10.1109/OJVT.2023.3234069.

[12] Arora, A., Kaur, A., Bhushan, B., Saini, H. 2019. Security Concerns and Future Trends of Internet of Things. International Conference on Intelligent Computing, Instrumentation and Control Technologies, pp. 891–896. https://doi.org/10.1109/ICICICT46008.2019.8993222.

[13] Paes, V., Pessoa, C., Costa, V., Oliveira, L, Souza, J. 2022. IoE Knowledge Flow Model in Smart Cities. IEEE SMC, pp. 982–987. https://doi.org/10.1109/SMC53654.2022.9945275.

[14] Zanon, V., Romancini, E., Manoel, B., Lau, J., Ourique, F., Morales, A. 2022. Avaliação experimental de uma camada de segurança implementada em dispositivo vestível cardíaco para Internet das Coisas Médicas. XXII SBSEG. SBC, Brasil, pp. 97–110. https://doi.org/10.5753/sbseg.2022.224659.

[15] Torre, D., Mesadieu, F., Chennamaneni, A. 2023. Deep Learning Techniques to Detect Cybersecurity Attacks: A Systematic Mapping Study. Empirical Software Engineering 28, 76. https://doi.org/10.1007/s10664-023-10302-1.

[16] Yadav, E., Mittal, E., Yadav, H. 2018. IoT: Challenges and Issues in Indian Perspective. 3rd IEEE IoT-SIU, pp. 1–5. https://doi.org/10.1109/IoT-SIU.2018.8519869.

[17] Koziolek, H. 2011. Sustainability evaluation of software architectures: a systematic review. QoSA-ISARCS '11. Association for Computing Machinery, pp. 3–12. https://doi.org/10.1145/2000259.2000263.

[18] Sheikh, Z. and Singh, Y. 2022. A Hybrid Threat Assessment Model for Security of Cyber-Physical Systems. 7th IEEE Seventh International Conference on Parallel, Distributed and Grid Computing, pp. 582–587. https://doi.org/10.1109/PDGC56933.2022.10053332.

[19] Barisic, A. and Cunha, J. 2017. Sustainability in Modelling of Cyber-Physical Systems: A Systematic Literature Review - Intermediate Technical Report (Research Report). Universidade NOVA de Lisboa. https://hal.science/hal-03168839.

[20] ISO/IEC 27000. 2018. Information technology — Security techniques — Information security management systems — Overview and vocabulary. Accessed in 5.10.23. https://standards.iso.org/ittf/PubliclyAvailableStandards/index.html.

[21] OWASP. 2016. Category: Vulnerability. Accessed in 5.10.23. https://wiki.owasp.org/index.php/Category:Vulnerability.

[22] Kariri, E. 2022. IoT Powered Agricultural Cyber-Physical System: Security Issue Assessment. IETE Journal of Research. https://doi.org/10.1080/03772063.2022.2032848.

[23] Alfadel, M., Costa, D, Shihab, E. 2023. Empirical analysis of security vulnerabilities in Python packages. Empirical Software Engineering 28, 59. https://doi.org/10.1007/s10664-022-10278-4.

[24] Baho, S. and Abawajy, J. 2023. Analysis of Consumer IoT Device Vulnerability Quantification Frameworks. Electronics 12, 1176. https://doi.org/10.3390/electronics12051176.

[25] Sahmi, I., Mazri, T., Hmina, N. 2019. Study of the Different Security Threats on the Internet of Things and Their Applications. ACM International Conference Proceeding Series. https://doi.org/10.1145/3320326.3320402.

[26] Zhao, W., Yang, S., Luo, X., 2020. On Threat Analysis of IoT-Based Systems: A Survey. IEEE SmartIoT, Beijing, China, pp. 205–212. https://doi.org/10.1109/SmartIoT49966.2020.00038.

[27] Benzarti, S., Triki, B., Korbaa, O. 2017. A survey on attacks in Internet of Things based networks. IEEE International Conference on Engineering & MIS, pp. 1–7. https://doi.org/10.1109/ICEMIS.2017.8273006.

[28] Rahimi, H., Zibaeenejad, A., Rajabzadeh, P., Safavi, A. 2018. On the Security of the 5G-IoT Architecture. IoTSC, Mashhad Iran, pp. 1–8. https://doi.org/10.1145/3269961.3269968.

[29] Xu, H., Sgandurra, D., Mayes, K., Li, P., Wang, R. 2017. Analysing the Resilience of the Internet of Things Against Physical and Proximity Attacks. Security, Privacy, and Anonymity in Computation, Communication, and Storage. Springer International Publishing, pp. 291–301. https://doi.org/10.1007/978-3-319-72395-2_27.

[30] Sookhak, M., Tang, H., He, Y., Yu, F. 2019. Security and Privacy of Smart Cities: A Survey, Research Issues and Challenges. IEEE Communications Surveys and Tutorials 21, 1718–1743. https://doi.org/10.1109/COMST.2018.2867288.

[31] Chhetri, C. and Motti, V. 2021. Identifying Vulnerabilities in Security and Privacy of Smart Home Devices. Advances in Intelligent Systems and Computing 1271, 211–231. https://doi.org/10.1007/978-3-030-58703-1_13.

[32] Pedreira, V., Barros, D. and Pinto, P. 2021. A Review of Attacks, Vulnerabilities, and Defenses in Industry 4.0 with New Challenges on Data Sovereignty Ahead. Sensors 21, no. 15: 5189. https://doi.org/10.3390/s21155189.

[33] Silva, H. 2019. A Caixa de Ferramentas Conceituais de Richard Rorty: O Uso de Técnicas Ad hoc. Cognitio-Estudos: Revista Eletrônica de Filosofia 16, 257–267. https://doi.org/10.23925/1809-8428.2019v16i2p257-267.

[34] Biolchini, J. Mian, P. G., Natali, A. C. C. and Travassos G. H. 2005. Systematic Review in Software Engineering. Technical Report-ES 679/05. Systems Engineering and Computer Science Department COPPE/UFRJ. Access in: https://www.cos.ufrj.br/uploadfile/es67905.pdf.

[35] Moher, D., Stewart, L., Shekelle, P. 2015. All in the Family: systematic reviews, rapid reviews, scoping reviews, realist reviews, and more. Systematic Reviews 4, 183, s13643-015-0163–7. https://doi.org/10.1186/s13643-015-0163-7.

[36] Motta, R., Oliveira, K., Travassos, G. 2019. A conceptual perspective on interoperability in context-aware software systems. Information and Software Technology 114, 231–257. https://doi.org/10.1016/j.infsof.2019.07.001.

[37] Mourão, E., Pimentel, J., Murta, L., Kalinowski, M., Mendes, E., Wohlin, C. 2020. On the performance of hybrid search strategies for systematic literature reviews in software engineering. Information and Software Technology 123, 106294. https://doi.org/10.1016/j.infsof.2020.106294.

[38] Petticrew, M. and Roberts, H. 2006. Systematic Reviews in the Social Sciences. Blackwell Publishing Ltd, Oxford, UK. https://doi.org/10.1002/9780470754887.

[39] Noble, H. and Mitchell, G. 2016. What is grounded theory? Evidence Based Nursing 19, 34–35. https://doi.org/10.1136/eb-2016-102306.

[40] Alqassem, I. and Svetinovic, D. 2014. A taxonomy of security and privacy requirements for the Internet of Things (IoT). IEEE International Conference on Industrial Engineering and Engineering Management, pp. 1244–1248. https://doi.org/10.1109/IEEM.2014.7058837.

[41] Karie, N. M., Sahri, N. M., Yang, W., Valli, C. and Kebande, V. R. 2021. A Review of Security Standards and Frameworks for IoT-Based Smart Environments. IEEE Access, vol. 9, p. 121975–121995. doi: 10.1109/ACCESS.2021.3109886.

[42] Kamoru, O.K., Frank, I., & Yemi, A. 2014. Computer Security Measures, Tools and Best Practices. British Journal of Applied Science and Technology, 4, 4380-4394.

[43] Takada, T. 2017. Authentication Shutter: Alternative Countermeasure against Password Reuse Attack by Availability Control. Proceedings of the 12th International Conference on Availability, Reliability and Security.

[44] Al Abdulwahid, A., Clarke, N., Furnell, S., Stengel, I. and Reich, C. 2015. The Current Use of Authentication Technologies: An Investigative Review. International Conference on Cloud Computing (ICCC), Riyadh, Saudi Arabia, 2015, pp. 1-8, doi: 10.1109/CLOUDCOMP.2015.7149658.

[45] Patil, A., Rana, D., Vichare, S. and Raut, C. 2018. Effective Authentication for Restricting Unauthorized Users. International Conference on Smart City and Emerging Technology (ICSCET), Mumbai, India, pp. 1-4, doi: 10.1109/ICSCET.2018.8537323.

[46] Ali, R.F., Muneer, A., Dominic, P.D., Taib, S.M. and Ghaleb, E.A. 2021. Internet of Things (IoT) Security Challenges and Solutions: A Systematic Literature Review. In: Abdullah, N., Manickam, S., Anbar, M. (eds) Advances in Cyber Security. Communications in Computer and Information Science, vol 1487. Springer, Singapore. https://doi.org/10.1007/978-981-16-8059-5_9

[47] Roohi, A., Adeel, M. and Shah, M. A. 2019. DDoS in IoT: A Roadmap Towards Security & Countermeasures. 25th International Conference on Automation and Computing (ICAC), Lancaster, UK, pp. 1-6, doi: 10.23919/IConAC.2019.8895034.

[48] Wang, W., Xu, P. and Yang, L. 2018. Secure Data Collection, Storage and Access in Cloud-Assisted IoT. In IEEE Cloud Computing, vol. 5, no. 04, pp. 77-88. doi: 10.1109/MCC.2018.111122026.

[49] Olaniyi, O.O., Okunleye, O.J., Olabanji, S.O., Asonze, C.U., and Ajayi, S.A. (2023). IoT Security in the Era of Ubiquitous Computing: A Multidisciplinary Approach to Addressing Vulnerabilities and Promoting Resilience. Asian Journal of Research in Computer Science.

[50] Kimani, K., Oduol, V.K., and Langat, K. 2019. Cyber security challenges for IoT-based smart grid networks. Int. J. Crit. Infrastructure Prot., 25, 36-49.

[51] Gromov, M., Arnold, D., and Saniie, J. (2022). Tackling Multiple Security Threats in an IoT Environment. 2022 IEEE International Conference on Electro Information Technology (eIT), 290-295.

[52] Ammayappan, K., Puthuparambil, A.B., and Negi, A. (2020). Key Vulnerabilities in Internet of Things.

[53] Fortuna, B., Rupnik, J., Brank, J., Fortuna, C., Jovanoski, V., Mario, Karlovcec, Kazic, B.M., Kenda, K., Leban, G., Muhic, A., Novak, B., Jost, Novljan, Papler, M., Rei, L., Sovdat, B., Stopar, L., Grobelnik, M., Dunja, & Mladenić. 2014. QMiner: Data Analytics Platform for Processing Streams of Structured and Unstructured Data.