electronics

MDPI

Article

Detection of Vulnerabilities in Smart Buildings Using the Shodan Tool

Sofía Mulero-Palencia [1,*] and Victor Monzon Baeza [2]

[1] CARTIF Technology Centre, Parque Tecnológico de Boecillo, 47151 Boecillo, Valladolid, Spain
[2] SigCom Group in SnT, University of Luxembourg, L-1855 Luxembourg, Luxembourg; victor.monzon@uni.lu
* Correspondence: sofmul@cartif.es

Abstract: Smart buildings, integral components of modern urban landscapes, are confronted with diverse vulnerabilities that jeopardize system robustness, cybersecurity, data confidentiality, and the well-being of the occupants. This work aimed to identify and evaluate vulnerabilities specific to smart buildings, introducing an innovative assessment approach leveraging the Shodan tool. The analysis comprised three stages: information collection, result extraction using Shodan, and vulnerability identification, culminating in a comprehensive evaluation. This study pioneers the use of Shodan for smart building vulnerability detection, together with databases and associated nomenclature, to serve as a robust foundational tutorial for future research. The findings yielded a meticulous analysis of primary security risks inherent in building systems, advocating for implementing targeted measures to mitigate potential impacts. Additionally, this study proposes an evaluation methodology encompassing metrics to gauge the effect of vulnerabilities on integrity, availability, and scope. By addressing insecure configurations, deployment inadequacies, and suboptimal cybersecurity practices, this framework fortifies smart buildings against potential threats. This study's originality lies in its Shodan-centric framework, revolutionizing the approach to smart building applications and vulnerability detection. This research contributes to the field by identifying critical vulnerabilities and proposing effective mitigation strategies, thereby elevating the overall security and safety of interconnected smart spaces.

Keywords: smart building; cybersecurity; Shodan; vulnerabilities; smart cities

Citation: Mulero-Palencia, S.; Monzon Baeza, V. Detection of Vulnerabilities in Smart Buildings Using the Shodan Tool. *Electronics* **2023**, *12*, 4815. https://doi.org/10.3390/electronics12234815

Academic Editors: Juan-Carlos Cano and Zheng Xu

Received: 13 October 2023
Revised: 14 November 2023
Accepted: 23 November 2023
Published: 28 November 2023

1. Introduction

New demands in the urban environment that have emerged in recent years have increased the importance of the term Smart City. The significant increase in population in these areas has led to a scenario in which buildings must improve the quality of life of their inhabitants through new strategies that guarantee intelligent, efficient, and safe management of resources. The benefits offered by smart buildings range from energy and maintenance savings and resource optimization, to increasing the amount of services provided to users and improving the security and availability of information. However, the introduction of new technologies at both the building and city level brings with it new challenges and risks that need to be identified and addressed. Smart buildings suffer from many vulnerabilities that can compromise system availability, system security, and information confidentiality, and even endanger the safety and health of residents. This situation motivated an analysis of the main security risks associated with various smart building systems, their impact, and what measures can be taken for their mitigation. Furthermore, possibilities offered by existing tools such as Shodan can help to identify and uncover risks and vulnerabilities in key elements, and to understand and predict the characteristics of cybersecurity vulnerabilities of technologies associated with different systems.

Vulnerability and threat are two terms that will be cited a number of times in this section. Therefore, it is important to be aware of the difference between the two and to avoid interchanging and confusion between vulnerability, threat, and risk [1]:

- A **vulnerability** refers to a weakness or flaw in a system that puts information security at risk. An attacker could exploit it to compromise the integrity, availability, or confidentiality of information, hence the need for detection and elimination.
- A **threat** is related to an action that exploits a known vulnerability to attack the security of an information system. They can come from provoked attacks (viruses, theft), physical events (fire, power failure), or negligence in the use of systems (unencrypted connections, missing passwords). Vulnerabilities can always be exploited, hence threats.
- A **risk** refers to the likelihood of a security incident occurring and materializing as a threat.

1.1. Motivation

The Internet of things (IoT) can create vulnerabilities in smart buildings due to various inherent characteristics of IoT devices and systems.

1. **Increased Attack Surface**: The proliferation of IoT devices in smart buildings leads to a larger attack surface. Each connected device represents a potential entry point for cyber threats.
2. **Lack of Standardization:** The diversity of IoT devices often results in a lack of standardized security protocols. This variability makes implementing consistent and adequate security measures across all devices challenging.
3. **Insecure Devices:** Many IoT devices are designed with a focus on functionality and cost, rather than robust security. This makes them susceptible to exploitation by malicious actors.
4. **Insufficient Authentication and Authorization:** Weak authentication mechanisms and inadequate authorization processes in IoT devices can make it easier for unauthorized entities to access sensitive systems and data.
5. **Data Privacy Concerns:** IoT devices often collect and transmit sensitive data. These data can be intercepted or manipulated if not adequately secured, leading to privacy breaches and potential misuse.
6. **Supply Chain Risks:** Security vulnerabilities may be introduced at various IoT device supply chain stages, from manufacturing to deployment. Malicious actors can exploit these weaknesses for their benefit.
7. **Limited Update Mechanisms:** Some IoT devices lack robust mechanisms for receiving security updates. This makes them vulnerable to known exploits, as patches and fixes may not be promptly applied.
8. **Interconnectedness:** The interconnected nature of IoT systems means that a compromise in one device can have a cascading effect, potentially affecting the entire smart building ecosystem.
9. **Insufficient Encryption:** Inadequate encryption practices in communication between IoT devices and central systems can expose sensitive information to eavesdropping and unauthorized access.

Understanding these nine factors is essential for implementing robust cybersecurity measures in smart buildings, mitigating potential risks, and ensuring the safe and secure operation of IoT-enabled systems. To enact these measures, it is imperative to initially detect and precisely identify vulnerabilities. Hence, our work emphasizes this preliminary phase, which we deem essential before formulating cybersecurity measures.

Furthermore, the rapid proliferation of the IoT has led to an exponential surge in network traffic, underscoring the critical need for effective vulnerability detection techniques. These methods are crucial for identifying potential weaknesses in information flows and for safeguarding against network attacks and vulnerabilities in various procedures, resolving the previous factors. As smart cities continue to evolve, there is a growing emphasis on fortifying critical infrastructure to ensure urban security [2]. Within this context, smart buildings play a pivotal role, constituting an integral part of the urban landscape to contribute to developing sustainable objectives in Smart Cities [3].

Detecting vulnerabilities in smart buildings is crucial for ensuring the safety, security, and privacy of occupants and system integrity. As smart technology becomes increasingly integrated into our daily lives, it is imperative that we stay ahead of potential threats. By proactively identifying vulnerabilities, we can preemptively address security risks, mitigate potential breaches, and implement robust protective measures. This not only safeguards the well-being of individuals within these spaces, but also contributes to the overall resilience and reliability of the infrastructure. Ultimately, the motivation lies in creating an environment where cutting-edge technology harmoniously coexists with unwavering security, providing peace of mind for all stakeholders involved. In Section 2, we provide an overview of the current state of vulnerability detection within the context of smart buildings, in conjunction with work carried out to date. This will serve to establish the necessity for the research presented in this work.

1.2. Our Contributions

This work presents an analysis of the main vulnerabilities in terms of cybersecurity in smart buildings and seeks solutions to attain safe environments. The research contributions in this field include

- A detailed study of the primary risks, threats, and vulnerabilities found in intelligent buildings and their components;
- Vulnerability classification databases and associated nomenclature are reviewed, to serve as a tutorial for future research works;
- Analysis and guidelines for Shodan as a potential tool to detect vulnerabilities;
- Based on previous analysis, we propose a three-step methodology utilizing the Shodan search engine to gather information from the real world. This is applied to smart building environments for smart cities;
- The obtained results are thoroughly analyzed, to determine which of these vulnerabilities can be potentially exploited.

Through these contributions, we introduce a framework centered around Shodan, a readily available tool that is both free and accessible. This framework, not only facilitates the advancement of research in this field within the scientific community, but also offers a novel approach to smart building applications and vulnerability detection, emphasizing the essential analysis that must be considered. Previous studies have not scrutinized whether the vulnerabilities identified in specific devices, as per the literature and the Common Vulnerabilities and Exposures (CVE) database, can be detected in the devices disclosed via Shodan. Furthermore, these studies and identification of vulnerabilities were carried out from the perspective of the network, not from the point of view of smart building requirements. We also highlight the union with Shodan as a noteworthy novelty that can enhance future research in these academic and scientific fields.

This paper is structured as follows: Section 2 explores the origin of the term "smart building" and the technologies commonly employed in this field, as well as an overview of different techniques and tools for detecting vulnerabilities. Section 3 analyses the primary vulnerabilities, threats, and attacks that smart building systems may encounter. In Section 4, an overview of the Shodan tool, its potential applications, and a review of the literature related to its use in various contexts is provided. Furthermore, Section 5 outlines the usage of the Shodan search engine for obtaining real-world information and defines an action methodology. Moreover, the obtained results are analyzed to identify exploitable vulnerabilities in Section 6. Finally, Section 7 evaluates the validity of the initial approach and analyses the research findings. Due to the sensitivity of the subject matter, Section 8 includes a brief reflection on ethical issues.

2. State of the Art: Smart Building and Technologies

Smart buildings play an important role in the development of smart cities. Among the elements they bring are improved connectivity, accessibility, energy efficiency, and even security, with the latter understood as improvements in terms of surveillance and moni-

toring. This chapter begins by examining the meaning and evolution of the term smart building. Subsequently, the most common smart building technologies are analyzed, which are useful for improving the operation of smart buildings.

2.1. Smart Buildings

The term "smart building", coined by the Intelligent Building Institution (IBI) in the 1980s, refers to a structure integrating diverse systems for efficient resource management. Originally focused on technical performance and cost savings, the concept was broadened in the 1990s. The International Council for Research and Innovation in Building and Construction (CIB) expanded the definition in 1995, characterizing a smart building as a dynamic and responsive architecture optimizing conditions for inhabitants through continuous interaction among places, processes, people, and management [4].

From an industrial perspective, entities like CABA (Continental Automated Buildings Association), IBM (International Business Machines Corporation), and Siemens have provided varied descriptions, which were analyzed and compiled in [5]. The European Commission, in its 2017 report, also contributed to the definition of smart buildings as a system of communication technologies enabling objects, sensors, and functions within the structure to exchange information [6]. Therefore, over time, the idea that the building is sensitive and must meet the needs of the user and the energy system has become increasingly important. Finally, the definitions and terminologies related to smart buildings were brought together in the IEEE 2785-2023 standard [7], although this standard refers to smart homes. This standard presents a fundamental framework and requirements for designing and modeling this type of system.

From a technological standpoint, the literature presents innovative approaches for building renovation utilizing cutting-edge technologies like digital twins [8] and 6G communications [9]. However, IoT stands out as the quintessential technology. Given the substantial data managed by various sensors, effective integration is crucial, and this is achieved through the definition of building information modeling (BIM). When delving into Internet-related technologies, it is imperative to consider potential cyber threats. A study by [10] examined existing BIM and IoT data integration research, exploring the associated challenges, benefits, and opportunities.

Nowadays, a building's awareness of its surroundings [11–14], user-centric functionality, and ability to autonomously adapt to recognized needs are pivotal. This adaptability optimizes the comfort, safety, and well-being of occupants, as well as energy usage [15–17].

2.2. Technologies in Smart Buildings

Information and communication technologies (ICT) are used in smart buildings to enable control and automation of operations. Smart buildings can cover different domains, such as air conditioning systems, lighting, solar power generation, energy supply, temperature sensors, humidity sensors, energy consumption sensors, and surveillance cameras [18]. Therefore, the core components of building components include extensive sensor and actuator systems, network and communication systems, software platforms, building conditioning systems, and intelligent control devices [18]. The volume of smart home and smart building devices is expected to grow from 233 million to 980 million units by 2025. Introducing new technologies and devices also means increased security risks in terms of system availability and control, data protection, and occupant well-being. As the number of IoT devices increases, new potential attack surfaces (surface attacks) and mechanisms for launching these attacks (attack vectors) are introduced.Furthermore, according to the 2017 EC report [6], there are significant discrepancies in the use of the communication protocols traditionally used in smart buildings. Harmful practices, as well as intentional events, lead to a complex scenario, where consequences range from the reputation of device and system suppliers to the health and safety of building occupants, potentially resulting in significant financial losses as a result of equipment replacement [19,20]. The main risks,

attacks, and vulnerabilities that systems linked to a smart building may face must be further examined.

2.3. Vulnerability Detection and Tools

As the significance of vulnerability detection in smart buildings is paramount, there is a notable body of research dedicated to this endeavor. One such study by [21] offers a comprehensive survey of software vulnerability detection benchmarking, shedding light on various benchmarking approaches. However, it does not delve into specific tools. Another innovative approach proposed by [22] leverages multi-objective techniques and neural networks for detecting vulnerabilities in smart contracts. Although highly effective, this approach focuses exclusively on smart contracts within the Ethereum network and does not address vulnerabilities in smart buildings.

Within the broader scope of IoT security, researchers have explored cutting-edge techniques like a dual-channel convolution neural network with spider monkey optimization (DCCNN-SMO) [23]. This deep learning method employs visual analysis of colored images to detect intrusions in IoT networks, exemplifying the potential of advanced algorithms in bolstering security. Additionally, supervised machine learning (ML) techniques have gained traction in intrusion-detection systems for IoT [24]. This line of inquiry aims to enhance the overall understanding in the field, though it does not specifically target vulnerabilities in smart buildings. Furthermore, advancements in machine learning have demonstrated impressive accuracy rates. For instance, studies employing techniques like artificial neural networks (ANN) and support vector machines (SVM) achieved noteworthy accuracies of 94.02% [25] and 99.8% [26], respectively. However, it is important to note that decision tree methods have shown a comparatively lower accuracy, at 89% [27].

In a distinct approach, ref. [28] explored vulnerability detection through code slicing, successfully identifying a substantial number of potential vulnerabilities. This study identified 118 potential vulnerabilities, of which 94 were found to have been silently patched. From the remaining cases, three were confirmed and designated with a CVE designation. This underscores the efficacy of code-centric vulnerability detection methods.

As the network becomes more accessible and flexible, the connected systems are increasingly exposed to risks. There are a variety of tools currently available for Internet scanning, with different capabilities, as outlined below. Some tools allow users to interact directly with the system to receive targeted results. In this category, we have Nessus [29], Skipfish [30], and Vigilant [31] as examples of such tools. Nessus offers a free, limited-use version. However, it is unsuitable for conducting large-scale network scans. On the other hand, Skipfish is open-source and also free, but it lacks a user interface and is challenging to navigate. The last tool is Vigilant, a vulnerability detection tool against fault-injection attacks targeting the hardware implementation of locking techniques, which aids designers in identifying critical nets susceptible to key leakage attacks. However, this tool was not designed for vulnerabilities in smart buildings. Alternatively, there are additional tools available that facilitate larger-scale automatic scanning, which increases the overall network coverage. Censys, Zoomeye, Thingful, and Shodan are some examples that provide such features and subsequently share the results publicly. Among these options, Censys [32] offers a user-friendly interface, but its use of the API has limitations. On the other hand, ZoomEye [33] allows different search criteria, although it is necessary to know the key term to be inspected. It can be complementary to Shodan and offers a limit of results per month in its free account, which cannot be downloaded. Thingful [34] collects extensive data and offers a vast index of multi-domain information. However, not being an open-source tool, it does have limitations on the public availability of the collected data.

On the other hand, for learning about IoT cybersecurity, Shodan [35] seems to be a better choice, due to its user-friendly web and API interfaces (application programming interface), and the numerous tools with easy-to-use interfaces offered. It also provides various types of licenses, with ample opportunities for academic use. In recent years, various researchers have utilized Shodan to assess the security of different IoT devices,

as detailed in Section 4. It is noteworthy that, in [36], a comparison of the various tools mentioned above emphasized the importance of having such tools to alert about device vulnerabilities before they become dangerous in a preventive manner. However, these tools can also allow malicious individuals to manipulate information, launch planned attacks, and cause damage to computer systems. Table 1 summarizes the features of the different alternatives.

Table 1. Summary of internet scanning tools.

Tool	Type	Features	Limitations
Nessus [29]	Interactive	Free version, Limited use	Unsuitable for large-scale scans
Skipfish [30]	Interactive	Open-source, Free	No user interface, Challenging to navigate
Vigilant [31]	Interactive	Fault-injection attacks hardware implementation	Limited to simulations
Censys [32]	Automatic	User-friendly interface	API limitations
ZoomEye [33]	Automatic	Different search criteria	Specific keyword search, results can not be downloaded
Thingful [34]	Automatic	Extensive data, Multi-domain information	Limited public data availability
Shodan [35]	Automatic	User-friendly web and API	Various license options Academic use opportunities

3. Vulnerabilities in Connected Buildings

The range of possibilities offered by intelligent buildings allows the user to enjoy numerous benefits, such as energy savings and reduced maintenance costs, better safety and security conditions for occupants, optimization of resources, and a wider range of services for the tenant. These services focus on improving the quality of life of occupants and managers, increasing their knowledge level and participation in the building management process.

However, it is not all advantages: introducing new technologies and devices also means increased security risks in terms of system availability and control, information privacy, and occupant well-being. As the number of IoT devices increases, new elements are introduced that can be exposed to attacks (surface attacks) and the mechanisms that trigger those attacks (attack vectors). The consequences of such events range from the reputation of the device and system suppliers to the health and safety of building occupants, and can result in significant financial losses as a result of equipment replacement [19]. The consequences can be much worse if an organization is affected.This chapter takes a closer look at the main risks, attacks, and vulnerabilities that systems linked to a smart building may face.

3.1. Common Threats in Smart Buildings

Once the main components of a smart building's systems have been identified, the typical threats, vulnerabilities, and exposures present in both the device software and associated gateways and web services can be identified. Usually, vulnerabilities are exploited through a series of threats materialized in the form of intentional attacks. Various classifications can be established to list the most common issues, either by focusing on their source, intention, nature, or which part of the architecture is under attack. First, threats in the smart building environment can originate internally or externally. Internal threats arise from issues within the system itself, such as incorrect network configuration or incorrect passwords, while external threats come from sources outside of the building, such as wireless protocols or the use of radio frequencies [37]. Continuing with the intentionality of the action, a threat can be considered intentional when there is a premeditated intention to cause harm, such as

information theft or malicious code injection. Conversely, unintentional threats might arise as a consequence of not implementing an adequate security policy, leading to undesired outcomes, such as a lack of backup power systems in case of power failure.

On the other hand, passive and active attacks can be differentiated. In a passive attack, the attacker obtains unauthorized access to private data after verifying or listening to a transmission without altering it. Within this category, attacks that affect the privacy of an intelligent building (eavesdropping attacks, aiming to monitor the user's traffic without his knowledge) and traffic analysis attacks (the attacker observes and analyses the exchanged information, which is a very difficult problem to trace) can be mentioned. Finally, in active attacks, the attacker can alter information and introduce harmful elements into the building's network by changing the information being transmitted or inserting new data flows. This group includes seven different subcategories: masquerading attacks, replay attacks, message modification attacks, denial-of-service attacks, interception attacks, and session-stealing attacks [37].

From [38], we can obtain different information related to the common attacks, impacts, and affected architecture elements. Table 2 provides a classification of this information according to which element of the architecture is threatened (hardware, software, or connectivity of the elements involved), including details of the characteristics of the nature of each attack and the specific impact produced by each of them.

Table 2. Attack categories according to the type of element in the architecture (own elaboration).

Classification	Characteristics	Specific Attack	Attack Impact
Hardware attacks	Related to vulnerabilities that affect the hardware part of the systems.	Physical attacks	Limited or no physical protection to restrict access to internal parts of the device.
		System batteries/energy use	Reduce uptime.
		Denial of service (DoS)	Many machines attack a device at the same time and the device may become unresponsive.
		Reverse engineering	Analyzes the architecture and technology of a device to replicate or modify it.
Software attacks	Linked to software bugs and misbehavior that compromise the security of systems.	Inconsistent software and firmware updates	Devices not patched and updated correctly.
		Reverse engineering	Find security risks in programs.
		Malicious software injection	Purposes such as accessing restricted information or increasing user privileges.
		Improper device integration	Systems are not integrated correctly.
		Incorrect configurations	Bad configuration due to simple/default passwords, etc.
		Poor cryptographic key management	Information can be easily intercepted.
Connectivity attacks	Problems related to attacks focused on intercepting data exchanged between different elements of a network, or impersonating an authorised third party.	Denial of service/distributed denial of service (DoS/DDoS)	Requests are generated to a service, consuming its resources and reducing its response capacity.
		Man-in-the-middle (MiTM)	Impersonation and eavesdropping on the network.
		Insecure interfaces	Configurations lacking security (web interfaces, APIs, cloud services).
		Jamming	Alter or cancel a communication, so that the receiver cannot interpret the message correctly.

3.1.1. Communication Protocols: Common Characteristics and Threats

This section covers the communication protocols commonly used in smart buildings, as their implementation may lead to a scenario where new risks arise. To provide the common functionalities in a smart building, standardized and open technologies are becoming more and more common, both through wired and wireless networks. Some related protocols are specific for this purpose, such as BACNet (building automation and

control networks) and KNX, both of which have become standards. Among the most common wireless protocols are Zigbee or EnOcean [39].

On the one hand, the evolution of building systems to incorporate wireless technology, thus improving the possibilities and flexibility of communications and simplifying installation and maintenance, has increased the vulnerabilities of building systems. In this context, protocols such as EnOcean, Zigbee, and those used by Bluetooth and WiFi (wireless fidelity) technology are common [40,41]. These protocols present higher levels of security than other wired protocols used for equivalent purposes in smart buildings but do not guarantee the same reliability, in terms of availability or latency [19,42,43]. On the other hand, some of the most common wired network protocols include BACnet, KNX, and LON (local operating network) [39]. The problem with many of these is that they were not designed with security as a primary requirement, so most have poor or non-existent security implementations, and this can cause significant problems within larger systems [41]. Both alternatives have advantages and disadvantages, and details on the most relevant protocols of each type are summarized in Table 3.

Table 3. Common characteristics and threats in communication protocols.

Technology	Protocol		Key Aspects
Wired technology	KNX	Benefits	Open standard that can be used as a backbone to connect several networks together [44].
		Drawbacks	Simple access control mechanisms that transmit passwords in plain text over the network. Security layer is based on TCP or UDP.
	LONTalk	Benefits	Used in the distributed automation system called LONWorks, lighting, and HVAC systems.
		Drawbacks	No data encryption, but sender authentication is available. It suffers from DoS attacks, and the lack of encryption technology allows eavesdropping on network traffic [40,44].
	BACnet	Benefits	Standard protocol that aims to support the interoperability between vendors.
		Drawbacks	Message security is applied at the network level, and authentication is used. It can suffer spoofing attacks, DoS, write ownership, or disabling of network connections [19,40].
	Modbus	Benefits	Serial communications protocol mainly used in industrial control applications.
		Drawbacks	Modbus/RTU offer neither encryption nor authentication. Modbus/TCP sends messages as unencrypted text over the network and can be easily intercepted, but TLS cryptography prevents MiTM attacks [19,45].
	HTTP	Benefits	Connectionless client/server protocol that is transaction-oriented, useful for sending large amounts of information.
		Drawbacks	The use of SSL/TLS over HTTP is recommended to ensure the correct transmission of information over an encrypted channel and avoid servers in IoT devices [19].
	MQTT	Benefits	Message transport protocol based on publications and subscriptions, widely used in communications linked to IoT devices.
		Drawbacks	Should be used in conjunction with TLS to secure communication over TCP (no authentication or encryption) [45].

Table 3. *Cont.*

Technology	Protocol		Key Aspects
Wireless technology	Bluetooth	Benefits	Short-range technology widely used in smart homes and buildings.
		Drawbacks	Authentication, authorization, confidentiality, data integrity, and pairing. It can suffer from reverse and social engineering attacks, passive eavesdropping, MiTM, Bluebugging, and FalseTiming attacks.
	EnOcean	Benefits	Proprietary protocol mainly used in home and building automation.
		Drawbacks	Communications can be protected by authentication code.
	WiFi	Benefits	Used in intelligent buildings for high-performance audio or video, centralized management applications, and connection between multiple devices.
		Drawbacks	Compatible encryption mechanisms susceptible to radio interference, risk of DoS network, eavesdropping and packet sniffing in traffic, Evil Twins or Hotspots [46].
	Zigbee	Benefits	Communication protocol for wireless personal area networks.
		Drawbacks	Data packet encryption, susceptible to injection, wormhole, DDoS, and eavesdropping attacks [47].

3.1.2. Threats Related to Specific Systems

This section analyses the threats associated with the use of specific implementations. Taking into account the different domains within the scope of a smart building and the associated key components as outlined in Section 2.2, a selection of specific systems has been made whose implementation is increasingly common, even at the domestic level: the use of smart thermostats, smart plugs, smart cameras, and smartphones, and their management possibilities through building applications (smartphones), and finally, smart lighting.

Smart thermostats: A significant amount of smart thermostats have been installed by users who are unaware of their most prevalent security flaws. As a result, a community has arisen to exploit these issues. These security breaches pose potential threats to occupants' privacy, malfunctions, and security concerns regarding authentication, network communication, and user information access. Thermostats typically feature a temperature sensor, a communication port with the HVAC system of the building (heating, ventilation and air conditioning), and an interface for transmitting and receiving data. They communicate through WiFi, radio frequency, or Bluetooth. Each procedure has its own pros and cons. For instance, RF is beneficial when no wireless network is available, but it can potentially disrupt other systems in the smart building. On the other hand, WiFi is widely used and enables remote control, but it may still experience some interference from external sources.

These devices are typically battery-operated and will automatically restart in the event of a power outage. However, network failures can cause inaccuracies due to the initial learning time, which can result in undesired behavior. Additionally, these devices have internal memories that store collected data, making them susceptible to attacks on potential servers and local attacks, although this also reduces the necessary connections. Smart thermostats are designed to learn user behavior patterns, to optimize energy consumption. However, if this information is accessed, it can be used track an occupant's lifestyle, violating their privacy and intimacy [48].

Thermostats from brands like EcoBee, Nest, and Honeywell are commonly used in this environment. In terms of connectivity, it is worth noting the following:

- Concerning connectivity, it is noteworthy that all three options endorse **WiFi security protocols**. The device may face harm if the network is attacked via weak security measures in the wireless protocol. With regards to the Honeywell range of thermostats, there may be some issues initially. During setup, an attacker may be able to register the thermostat by collecting information provided by the thermostat's web server before the user has a chance to complete registration. This vulnerability can be prevented by

using a secret value for registration and implementing a protocol such as WPA2 (WiFi protected access);

- In terms of **cloud connection**, all three options employ encryption techniques, ensuring that no sensitive data are exposed. In the case of Nest, the device can utilize CA to authenticate the identity of remote web servers by establishing TLS connections, thereby rendering it impervious to man-in-the-middle attacks;
- Access to this form of home thermostat can be obtained through either a **web portal** or a **mobile application**, which facilitates its management. If the password has a low level of security, an unauthorized user may commandeer the system without prior consent. Additionally, Ecobee thermostats offer an API for device interaction, which presents a new possible vulnerability.

The Nest device is detailed in [49], explaining how encryption methods prevent remote access attacks. However, the device could be compromised if known weaknesses are used to gain access or install malware.The article also investigates several unsuccessful attacks of this kind, where appropriate access credentials are unknown. Despite this, it is improbable that such an attack would happen without some form of social engineering to obtain the user's credentials.

Smart Plug: Smart plugs represent a simple and low-cost alternative for adopting smart technologies in a building. These electrical devices that plug into an outlet are intended to allow control of the connected device through the use of some kind of commonly used application, such as a smartphone. Thus, the user can monitor the energy usage of the devices, set on/off schedules, and receive alerts for any undesirable behavior [50].

Some of the leading consumer electronics manufacturers have entered the market with various devices, such as Belkin, TP-Link, and B-Link. The solutions they offer for monitoring and controlling consumption are not unified, and the cost they present is quite high. Among the wireless technologies used are WiFi, Bluetooth, BLE (Bluetooth low energy), Z-wave, and Zigbee, although the use of the latter is not very widespread [51]. In general, issues commonly arise regarding network communication and access to information. A number of potential attacks are outlined below [50]. First, if communication protocols do not employ cryptographic mechanisms, capturing traffic and reverse engineering it to obtain user credentials is possible. Furthermore, it is worth mentioning that, if the remote server with which the application that allows its management communicates does not perform a proper authentication, four types of attacks can occur:

- *Device scanning*: The MAC (medium access control) address space can be monitored to determine the status of all online sockets. This could reveal which users are using default passwords;
- *Brute force attack*: If password changes are made, they may disrupt processes;
- *Spoofing attack*: The remote application may be manipulated to send a user's authentication credentials, which can then be intercepted for login purposes;
- *Firmware attack*: The goal here is to have malicious firmware loaded on the socket, even allowing root access to the system.

It is, therefore, important for manufacturers to implement a number of defensive measures. First, the use of cryptography to encrypt communication (TLS/SSL or HTTPS, with encryption); meanwhile, in terms of authentication, mutual authentication systems should be used between sockets and servers to avoid a man-in-the-middle attacks. In addition, the use of intrusion detection systems allows identifying anomalous behavior, such as in the case of a phishing attack, while anti-bot mechanisms allow dealing with brute force attacks; for example, by limiting the number of login attempts. On the user side, it is also important that default passwords are not used [50]. Extensive research has been conducted on the susceptibility of commercially available devices. Commercial smart plugs from brands such as TP-Link, Hive, anz Meross have been analyzed for vulnerabilities in [50], and some of them were classified as high impact. To cite a few examples, the Meross device could reveal the wireless network password due to a critical issue with user WiFi passwords not being encrypted during device configuration, and on

the other hand, for the TP-Link Kasa an attacker could take full control of the plug due to a weak encryption mechanism.

Smart camera: The use of security cameras to monitor buildings is a widespread and widely adopted practice. They are typically connected to the internet, perform standard surveillance functions, and include motion sensors to trigger alarm systems within the building, all while reporting relevant data to the authorities. Notifications are sent to the user's smartphone app whenever an event occurs. Common vulnerabilities in cameras range from issues with insufficient authentication/authorization to insecure web interfaces and insecure software/firmware. Smart cameras can capture user information, including images, phone numbers, personal identification, and addresses. This unauthorized access to images or videos, and the capture of personal data, can pose significant risks of being linked to specific individuals, resulting in serious confidentiality concerns.

Some common vulnerabilities are associated with the practices listed below [52,53]: first of all, video transmission should be made more secure using SRTP protocols instead of RTP (secure real-time transport protocol). Moreover, encrypted video transmission with weak key management should be avoided, while URLs that can access system information, particularly during configuration, should be secured with authentication methods. Default or insecure passwords must not be used, and security update mechanisms should be established to mitigate risks. Finally, exposure of video on unencrypted removable physical storage should be avoided, and contextual information, such as the camera's exact location or specific device data, should not be exposed.

Smartphone: Smartphones, like other building-connected systems, are vulnerable to cyber-attacks due to the valuable information they store and due to their crucial role in ensuring the needs of occupants and managers are met. The main threats to these systems include network performance issues (crashes, communication), authentication and authorization problems, and malfunctioning software. Some of these threats are linked to the following types of attacks [54,55]:

- *Denial of service (DoS)*: With this type of attack, access to monitoring and management of building resources can be blocked through the mobile interface, posing a risk to the building;
- *Power and internet supply failure*: For the former, the presence of backup batteries can minimize its occurrence. For the latter, it is advisable to have a backup service;
- *Malicious code injection*: This requires the execution of code or scripts via an exploit. The attacker aims to gain complete access to the home control application and potentially steal the user's personal data;
- *Software failures*: These malfunctions can be resolved through consistent updates and manufacturer support;
- *Attacks using weak encryption mechanisms and passwords*: The confidentiality of user data is jeopardized during authentication by weak encryption mechanisms and passwords;
- *Eavesdropping attack*: Real-time information transmitted by smart devices through the network can be obtained;
- *Man-in-the-middle attack*: Data are first sent to an intermediary, which can manipulate them. Appropriate communication protocols can reduce these risks.

Smart Lighting: Connected lighting systems can be controlled remotely, usually via mobile devices. An application provided by the manufacturer or a third party enables users to switch on, switch off, and adjust the color or brightness of the system. Additionally, there are remote controls that enable direct device commands, without the need for an app. Smart lighting devices often rely on Bluetooth, Zigbee, and WiFi. The low-power network version of ZigBee, Light Link (ZLL), is utilized by connected lighting systems, including Philips Hue, Osram Lightify, and GE Link [56].

Currently, there is no standardized protection for this type of IoT device. These elements are susceptible to various attacks, primarily linked to the communication mechanisms, authentication and authorization issues, and the insecurity of their software or firmware [56,57]:

- *Denial of service*: Blocking the radio frequency spectrum can result in the denial of remote access. This can be achieved through radio or Bluetooth interference. Real-time traffic analysis is currently employed to detect this issue, and channel hopping is a widely used measure to avoid it;
- *Lack of authentication and encryption*: Devices using the same communication standard as the lighting system can easily connect to it, as there are no encryption or authentication schemes for incoming connections. This opens up the possibility of unauthorized users modifying the behavior of the lighting. In the specific case of ZLL, relying solely on the use of a network master key is critical;
- *Malicious firmware updates*: Attackers can exploit the microcontroller for over-the-air updates, to facilitate undesirable actions. The firmware's physical access port can also be used for malicious purposes, including brute force or denial-of-service attacks.
- *Side channel attack (SCA)*: This technique exploits physical parameters and cryptographic implementation weaknesses. This method can obtain details about the communication process and circumvent encryption by altering the physical parameters of the device.

In addition, it should be noted that hardware tamper resistance is not a priority in IoT devices, and reverse engineering attacks are possible.

3.2. Global Mitigation Strategies

It is necessary to, not only be aware of the attacks taking place in smart buildings, but also understand the approach to be taken to protect the systems. In addition, there is often a lack of training among users or inconsistencies in standard security practices [19]. There are a number of mitigation measures that can be undertaken to try to minimize the risks:

- *Increase security awareness and develop security know-how:* different actors need to be aware of the issues involved in exposing buildings to the threats described, from developers, integrators, or suppliers to maintenance personnel and end users. It is important to make good choices when selecting the devices to be used, configuring the network, updating the different elements, etc. Small training sessions adapted to the profile of the specific actor can contribute to this training, progressively reduce the existing margin for improvement, avoid unnecessary risks, and permit dealing with emergency situations [40,44]. Simple operations, such as modifying default user credentials or not executing and downloading files from unknown sources, should become common practice [58]. Likewise, from a hardware perspective, device configuration should not be manipulated without prior knowledge, as this can become a gateway to a new threat, resulting from a lack of knowledge [54];
- *Having a matrix of responsibilities:* if the specific activities that should be reviewed and executed by different actors have been previously determined, it is possible to have greater control over the actions carried out in the building in terms of security [44];
- *Ongoing system maintenance:* Regularly updating the operating system and firmware by applying security patches is essential. To do this, it is necessary to select devices that support this type of update in advance and for vendors to offer this alternative. In addition, the standard approach should be to remove all services that are not clearly needed and close ports that will not be used intentionally [40]. In this regard, it is important to perform regular audits, to be aware of threats, system hot spots, and vulnerabilities already identified [44];
- *Device software updates:* vendors should offer these patches but also inform customers about them. On the other hand, customers must be able to apply them, i.e., the update mechanism must not be too complex for the intended user. From the software point of view, it is up to developers to bring trusted applications to the market using strong and secure encryption [44,54].
  Use of secure encryption techniques: e.g., two-factor authentication [44,54]. From a hardware point of view, an improvement in the processing power of devices would allow the use of more secure encryption keys, thus also protecting data privacy [58,59];

- *Network monitoring mechanisms:* to be able to identify intruders and detect potential threats in advance [44,54]., by first capturing packets to have a database that characterizes normal activity, and then being able to identify abnormal activity with good results [39]. For intrusion detection, physical measurements can be used as an alternative to unmask anomalies in the system if they are meaningless.
- *Protecting the system from denial of service attacks:* routing access restrictions, detection of false routing information, and detection of wormholes can be configured [39];
- *Provide some degree of modularity between the elements of different systems:* this allows preventing an attack on one entity from compromising all the others. A good practice is to configure the global system so that different networks exist, avoiding attackers having access to more if one network is compromised. Conversely, it should be noted that monitoring different networks may require additional maintenance efforts [40,44,59];
- *Having backup systems for critical elements of the systems* [40].

4. Shodan Tool Overview

In this section, we delve into the core functionalities of Shodan, exploring how it operates, what information it gathers, and how it has emerged as an invaluable asset in identifying potential points of vulnerability within various networked systems. In addition, we provide general concepts to understand the potential of this tool.

4.1. The Shodan Tool

Shodan is a search engine that enables one to explore the Internet and locate connected devices. Unlike traditional search engines such as Google or Bing, it provides information about available systems and services connected to the network, such as webcams, routers, servers, and smart devices. It can be used to perform cybersecurity audits on IoT (Internet of things) systems and devices, without the need to scan the devices directly [38]. Two common types of cybersecurity issues in the IoT can be uncovered: weak security mechanisms, and a lack of proper security configuration.

Through Shodan, it is possible to answer questions such as which devices are connected to the internet, their location, and even who is using them or whether they are running a particular software, to give a few examples. This kind of information makes it a powerful tool for crawling the Internet and indexing discovered services [60]. On the other hand, there are other web-based search engines for scanning generic vulnerabilities, such as Zmap, Censys, and the Thingful tool, which collect information from IoT devices, but Shodan is easier to use [38].

The returned information is stored in a database accessible online via the web interface provided in [35] (Figure 1) or via the API (application programming interface). In order to use the API, an account is required (access is free for students), and in this way, filters can be applied and search results can be restricted, e.g., by specifying country, city, IP or specific port, or by searching for keywords, such as "thermostat". Depending on the type of search, the information obtained ranges from the most popular software version on specific devices to vulnerabilities in IoT networks and smart devices to common security flaws. For each service discovered, Shodan repeatedly scans and stores the results, creating a time series of the results available and accessible to security experts for processing and analysis. Shodan's crawlers run 24 h a day and update the database in real time) [61].

The open data that Shodan collects are banners: textual information describing a service on a device (e.g., for a web server, this would be the headers that are returned). Their content differs depending on the service type, and they are accompanied by device metadata such as geographic location or host name (some fields can only be extracted using the developer API). Search results can be downloaded in JSON (JavaScript Object Notation), CSV (comma-separated values), or XML (eXtensive Markup Language) format. By default, the search query analyzes data collected in the last 30 days, unlike the old shodanhq.com website, which searched the entire Shodan database [60]. Another possibility offered by Shodan on its website includes the generation of reports based on a search query as a

snapshot. Once created, it is not modified when updating the information in the database and can be useful for tracking purposes. With the map tool, it is possible to visually explore the search results (Figure 2), and Shodan Exploit gathers vulnerabilities and exploits so that they can be searched through the web interface (an exploit is a code or program that exploits a security hole to be used by an attacker for his own benefit). Finally, it is also possible to identify the screenshots collected by Shodan through a screenshot filter and to query them from Shodan Images [60].

Numerous research studies have used the Shodan tool to assess the security in complex networks of smart devices exchanging information over the Internet. The growth of IoT as a technological paradigm in recent years has made security aspects increasingly critical. In addition, the amount of sensitive data shared over the network is increasing, as are the possibilities for remote access to its constituent elements. Understanding the nature of security attacks is essential and allows better protection measures to be built into systems from the outset. This is extensively explored in Nawir's paper [62], which analyzed network security issues in the domains of smart homes, healthcare, and transportation. Ten different types of attacks are listed, some of them described in considerable detail, and which can occur in more than one of the analyzed domains. In [63], an attack taxonomy was also created to classify and describe common threat scenarios, focusing on embedded systems. The study aimed to help the analysis and design of systems contained or based on embedded devices, minimizing the level of risk. To obtain information on vulnerabilities and exposures, the description provided in CVE (Common Vulnerabilities Exposures) is taken into account. The literature also includes studies on the application of the tool in specific locations. This is the case in [64], where Shodan was used to scan for vulnerabilities in IoT devices in Jordan, in order to alert the community about IoT security issues and to raise awareness about exposure to potential attackers. The authors concluded by stressing the importance of disabling both vulnerable and unused services and the need for proper configuration and regular software updates. In this study, numerous open webcams and industrial control systems were found to be compromised. On the other hand, an analysis of IoT applications, this time focusing on the domains of the smart home, smart city, and smart health, was carried out in [65]. The study aimed to identify the types of IoT applications most suitable for use in the United Arab Emirates, their possible security threats, and their potential impacts. Continuing in the field of medical security, another work identified medical devices from primary vendors such as Omron Corporation and Bionet as susceptible to attacks such as remote code execution. The lack of security in the authentication of these systems opens a potential window of attack through remote control [66].

As more and more smart elements become invisibly integrated into every aspect of our lives, the reliability of the systems built on top of them becomes more critical in terms of security. In [67], a remote security assessment for IoT devices was designed and performed on an extended dataset of initial public Shodan queries (searches associated with specific devices). The results confirmed the existence of a large number of publicly available and Internet-accessible elements vulnerable to simple access control issues, such as the use of default credentials or simple passwords (e.g., routers, firewalls, and webcams). In [49], the analysis focused on evaluating the security of a smart thermostat using Internet resources for attacks at the physical and network level. Although the vulnerabilities are publicly disclosed, according to the authors, it can be difficult for inexperienced attackers to use the available information effectively. Nevertheless, if an attack were successful, the damage done would have very negative consequences. Not only Shodan has been used to scan for vulnerable devices. In [68], three different methods were followed to find security problems associated with them: on the one hand, a search for vulnerable DSL routers (digital subscriber line) was performed using Shodan. Thanks to Masscan, a quick search of a large address space of devices potentially susceptible to a Heartbleed-type error (a weakness that allows the theft of protected information due to the SSL/TLS encryption

used associated with vulnerable versions of OpenSSL) was carried out. Finally, Nmap (Network mapper) was used to find information about vulnerable printers.

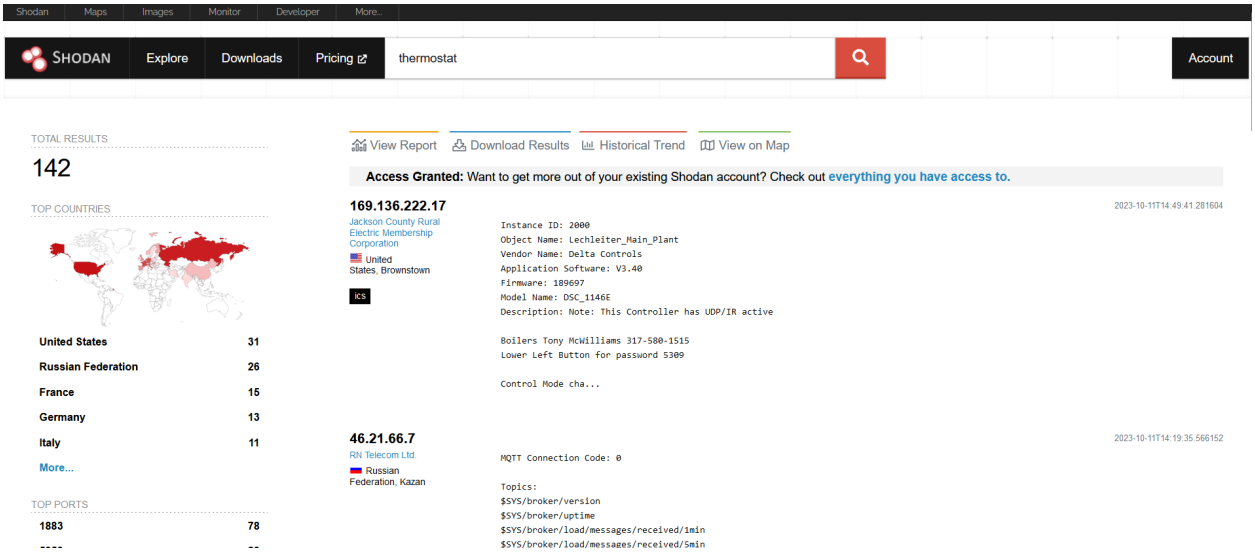

Figure 1. Example of results obtained with the Shodan browser. As can be seen in the top center, it was possible to access a report, download the results, and consult the historical trend and the results on a map.

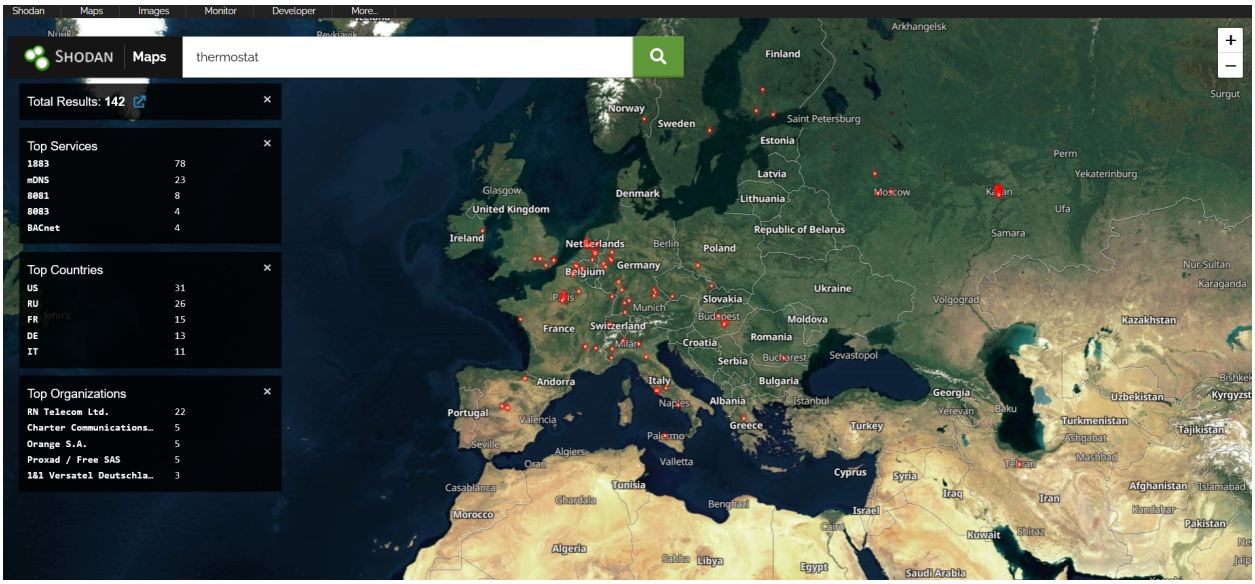

Figure 2. Display of results on the interactive map for thermostat search.

Thanks to their flexibility and adaptability, IoT systems are useful for a wide range of industrial, civil, and commercial applications. Shodan has been used to identify thousands of devices associated with industrial control systems, raising serious security concerns. In [69], a deployment was conducted to evaluate Shodan's indexing and querying capabilities (scanning frequency, web database identification accuracy), revealing that the tool was able to find devices within 19 days.

In [66], an analysis of supervisory control and data acquisition (SCADA) systems used to control critical infrastructures was carried out. As vendors further integrate internet technology, their vulnerability increases. The study identified the SCADA devices included in the Shodan database, and based on data mining techniques, vulnerabilities were assessed, reaching conclusions such as that more than half a million devices from major SCADA

vendors can be accessed and that thousands of them have critical vulnerabilities, such as outdated software or problems associated with the use of default credentials.

Finally, ref. [70] discussed the possible use of Shodan to find vulnerabilities on a large scale, and to study whether the real system exposure could be quantified. The study focused on SCADA devices, printers, and webcams, revealing the possibility of accessing a multitude of them without the need for authentication.

4.2. General Concepts about the Use of Shodan

Shodan is a search engine that makes it possible to passively scan the Internet for available text banners when a connection to a device is established. These banners provide details about the associated functions. If this information is analyzed in detail, it can help to detect potential threats. Some interesting considerations on the use of the tool are detailed below. First of all, it is preferable to have a user account to start using the tool. By applying for the student option, it is possible to work from a completely free academic developer profile, which has access to a wider range of functionalities than if you use an unregistered user account, for which the options are rather more limited (more queries allowed, report generation, and use of filters). In addition, not all the devices found by Shodan are real, as there are a significant number of honeypots distributed across the network as IT security tools to gather information on the strategy followed by cyberattackers, find out about the latest malware in use, track infections, and anticipate attacks. Shodan has a functionality that can roughly identify whether the IP address provided may or may not be a honeypot based on a calculated score.

Furthermore, the search for devices can be performed through the web browser or from the API, with a user account being mandatory in the latter case. The filters for searching information allow the consultation of exact terms and the use of Boolean operators. The syntax consists of indicating the filter to be used, followed by ":" and the key term in inverted commas. The results can be downloaded from the corresponding tab, and the reports tab creates summaries of results. There is also a section to report graphs and an interactive view on maps to navigate and visualize the geographical distribution of the results. Finally, another interesting feature of the tool is the exploit database, in which a code can be provided for a specific vulnerability. If there is a result, a link is provided [67], where there is extensive documentation on it.

In particular, in this research, an academic account with developer privileges was used to perform queries through the browser and the Shodan API, free of charge. Use of the API requires registering an account to use the API, with limited queries in its free version. With an academic account, there are no restrictions on the use of filters, the number of possible daily queries increases, and credits are available (not available for the free version). Moreover, the possibilities of accessing data depend on the amount of credits available in the account, which can be of two types: query credits, and exploration credits. Finally, three different payment plans are available: freelancer, small business, and corporate. The main differences are the number of results that can be obtained, the number of IPs that can be scanned, and the number of IPs for which network monitoring is offered. All of them have access to most filters, and the API and can be used for commercial purposes. It is considered that a freelancer plan may be sufficient to put in place a system that would allow regular audits to be carried out in a building and to have control over the security of the building.

5. Detecting Vulnerabilities with Shodan

In this section, the Shodan search engine was used to obtain knowledge and information about vulnerabilities of devices present in a smart building.

5.1. Detection Methodology

The methodology to be followed for the analysis of vulnerabilities in real cases consists of three main blocks: the collection of information, the extraction of results, and the

identification of vulnerabilities. This last step allowed obtaining a series of results to be analyzed in detail.

1. **Data collection**: different sources of information were analyzed. A series of key terms were identified to be used in the searches, to obtain data on the devices of interest. The terms used included the different types of devices considered within a smart building (connected camera, smart meter, smart thermostat, smart home, etc.) and variants, such as the proper names of specific devices and their manufacturers. In the first phase, tests were carried out manually via the web interface offered by the tool, using simple filters. Once interesting results had been found, it was necessary to consult additional information on the type of device, which could be provided through the manufacturer's documentation, user manuals, etc.;
2. **Results extraction**: a set of scripts in Python programming language was programmed to interact with the Shodan API. The Shodan interface allowed us to perform very general queries on specific devices, but it was more appropriate to automate the analysis of detailed cases using the API. The query strings used the same simple terms and filters as explored in the previous block. The returned data were processed and labeled to facilitate the results extraction block;
3. **Vulnerability identification**: the results obtained were analyzed to identify security vulnerabilities. The results were checked against the information available in the Common Vulnerabilities and Exposures (CVE) database for equivalent terms and other selected search terms listed below, which allowed completing the particular analysis of each case. After compiling the results, manual checks were carried out to ensure their validity, and the most frequent vulnerabilities were also consulted in the National Vulnerability Database, to complete the information and obtain the severity score for each.

In addition to the analysis method described above, the results of manual searches that have provided interesting information are included.

5.2. Data Sources

As indicated in the previous section, the first phase of the study identified a number of smart building systems and associated key terms to be considered. In addition to the information provided by the literature, additional sources were consulted to complete the documentation on specific devices.

The Shodan datasets consulted via the API were dynamically generated in response to the specified search criteria. These records included information on the type of device, possible exploits, and their significance. On the other hand, and in parallel, records were generated from the Common Vulnerabilities and Exposures database using similar search criteria. The information from both sources was preprocessed and tagged for extraction and analysis. In both cases, therefore, the information was updated and allowed analysis of the results at a given time, which, if repeated, would provide information on the evolution of risks associated with the systems detected.

5.3. Vulnerability Classification

This section explains the common vulnerability score system, which will be used to identify the problems found. It is one of the most widely used systems and is common in databases of publicly known vulnerabilities such as CVE or NVD. CVE was defined by the MITRE corporation (Bedford, MA, USA) (MITRE, 1999–2023), while NVD represents the US repository of vulnerability information, which in turn uses the nomenclature included in CVE. This is an open and universally used framework that establishes a set of metrics to communicate the characteristics, impact, and severity of vulnerabilities. The first version appeared in 2004, followed in 2007 by CVSS 2, and in 2012 work began on version 3.0. In 2019, version 3.1 was released, which did not introduce major changes in wording compared to version 3.0; its main objective was to clarify and improve the existing standard. The latter comprises three main metric groups: base, temporal, and environmental. Changes

between versions focused on improving definitions and the accuracy of assessments and mainly affected the base metrics group [71–73].

For version 3.1, the scoring calculations are performed similarly to version 2.0, such that the base metrics will define a value that can be made more accurate thanks to the metrics in the temporal and environment groups.

- **Base metrics:** these refer to intrinsic qualities of a vulnerability that are not dependent on time and environment, and present in all user environments. In turn, they consist of two groups of metrics: those of exploitability, and those of impact. Exploitability reflects the ease and technical means with which a vulnerability can be exploited (AV, AC, PR, UI), while impact reflects the direct consequence of using an exploit successfully (C, I, A). Finally, the scope (S) metric complements the global assessment of the previous metrics, providing a higher or lower value to the result, depending on the resources affected (Table 4);
- **Temporal metrics:** these refer to vulnerability characteristics that change over time but are constant in a user's environment. They are a set of optional metrics, whose value can be omitted;
- **Environment metrics:** refer to vulnerability characteristics related to a particular user's environment. Optional set of metrics, which may not be used in the final assessment if no particular metric exists.

Depending on the values achieved by a set of variables, a score between 1.0 and 10.0 is obtained for each group, following a series of formulas set out in the CVSS specifications. The calculation process is shown in Figure 3. This score is evaluated on the basis of a described scale that establishes the severity value of the vulnerability. The final numerical value is accompanied by a text string called "vector string", which starts with the CVSS tag. A calculator is available on the NVD website to obtain the scores of each metric according to the CVSS vector string associated with the vulnerability for CVSS v3.1 [74].

Table 4. CVSS base metrics.

Group	Metric	Possible Values	Meaning
Exploitability	Attack Vector (AV)	[N, A, L, P] (Network, Adjacent, Local, Physical)	It reflects the context in which the vulnerability can be exploited.
	Attack Complexity (AC)	[L, H] (Low, High)	Describes the conditions that must be present for the vulnerability to be exploited.
	Privileges Required (PR)	(N, L, H) (None, Low, High)	Checks whether privileges are required to perform the attack.
	User Interaction (UI)	(N, R) (None, Required)	Describes whether a user interaction is necessary for the attack to succeed.
Impact	Confidentiality (C)	(N, L, H) (None, Low, High)	Assesses the extent to which a component's vulnerability compromised info confidentiality.
	Integrity (I)	(N, L, H) (None, Low, High)	Assesses the effect on the integrity of a vulnerability that has been successfully exploited.
	Availability (A)	(N, L, H) (None, Low, High)	Assesses the effect on the accessibility of the element by a vulnerability successfully exploited.
Scope	Scope (S)		Captures whether a vulnerability in one vulnerable component impacts resources in components beyond its security scope.

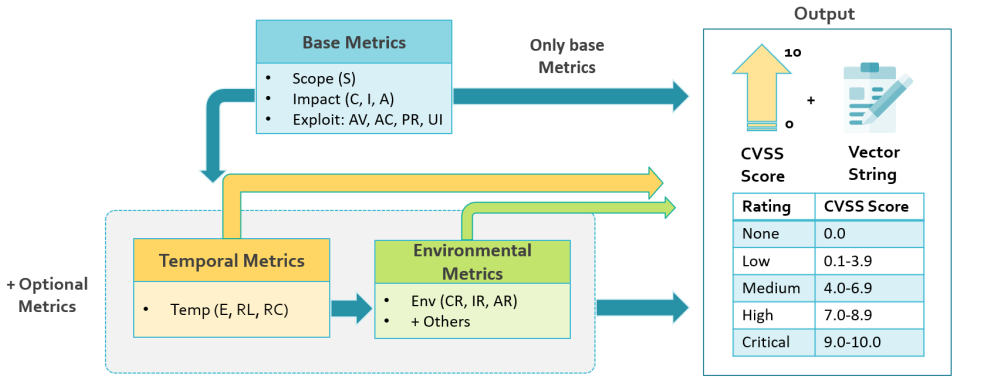

Figure 3. CVSS metrics and relation.

6. Results

This section analyses the results obtained using Shodan. This has been divided into two blocks: the first focuses on analyzing queries through the browser as a first approximation of use and aims to detect devices that facilitate access to information or even manipulate it; the second focuses on the use of the API. The recommended methodology was employed in the latter, and the tool was also used through the browser to compare results. After a first analysis of the literature focusing on smart buildings, an attempt was made to determine what information is openly accessible through the Shodan search engine, and a series of scripts were programmed to extract it. This information was then checked against the information extracted from the CVE database. The last step was to analyze the results obtained, to better understand which vulnerabilities associated with the system can be exploited.

6.1. Relevant Manual Searches

The objective of this manual analysis was mainly to locate devices that are easily accessible, either due to a lack of access credentials configuration or the use of default authentication parameters. It was necessary to perform numerous queries and to focus on specific devices. Thanks to Shodan, it was possible to find a significant number of IoT devices in a smart building that could be targeted. Following a similar approach to the previous theoretical analysis, smart thermostats, webcams, and digital video recording (DVR) systems were analyzed. In addition, network elements such as a router, centralized building management applications, and even a solar farm were manipulated.

- **Webcams**. It is common to find video surveillance systems that use default credentials or passwords that are easy to crack. This is the case with Blue Iris, the IP camera management software for Windows. It is possible to view all content via the integrated UI3 web server and using the term "ui3 -" yielded 193 results. The lack of authentication on the devices allowed access to live cameras without specifying any parameters. Similarly, running the query "IP CAMERA Viewer" yielded 338 devices, many of them offering access to live cameras without authentication.
- **Smart thermostats**. Information associated with two types of devices was found in the search for smart thermostat management applications that were not password protected. The ICY Clever thermostat could be accessed through the query title: "ICY Clever Thermostat", while the Heatmiser WiFi thermostat could be accessed through the filter title: "Status & Control", obtaining 12 and 164 results, respectively.
- **Network elements**. The query "default password" returned a significant number of devices with default passwords. By combining that information with specific filters to find specific devices, such as with the query "default password" product: "TP-LINK WR841N WAP http config", we could access that router model.
- **Smart home systems**. Providing access to the management software of individual smart devices, such as a thermostat, is dangerous, and even more so at the building

level. This is the case if access to home applications, such as those offered by FHEM, OpenHAB, and HomeMatic, is not properly protected. FHEM can be controlled directly via the web or a smartphone, and the query "FHEM home automation" returned 47 results. HomeMatic offers a similar solution for different activities in the home. The search query "homematic" returned 751 results. Finally, OpenHAB allows connecting devices and services from different providers. The default search for OpenHAB returned four results, some of them fully accessible.

- **Grid elements** The solar country: CZ search identified several IPs associated with solar farms. Some of them were not protected, and through the WATTrouter Mx tool, which is a programmable controller for optimizing the self-consumption of the energy produced by a PV plant, it was possible to obtain the information of the installations. Figures 4–6 show some of the results obtained for the devices mentioned in this sub-section.

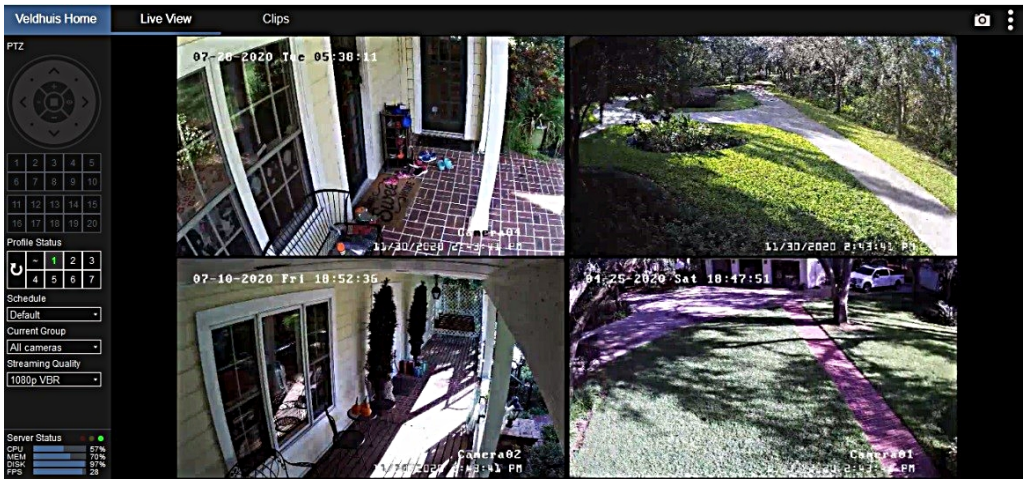

Figure 4. Example result of security issues in surveillance systems. The "IP CAMERA Viewer" search allowed real-time access to the management screen of video surveillance systems without authentication.

6.2. Manual and API Searches

Once the API key had been obtained to allow use of this communication interface, work began on the design of the Python program. These results were contrasted with the Shodan web interface to check their coherence through the use of bounded filters.

This study considered the different building systems analyzed in the previous section: smart thermostats, smart plugs, smart cameras, smartphones, and smart lighting. First, a selection of keywords related to each of them was made, and a set of filters were applied. Exploits identified for specific vulnerabilities in the items returned by Shodan were analyzed. Shodan integrates with exploit databases to provide information on known exploits for specific devices or services, allowing them to assess the potential associated risks. Second, for the same key elements, related information was searched in the CVE database, in order to identify the associated vulnerabilities (element, vulnerability, and description). In addition to using equivalent terms in the CVE database search, other vulnerabilities of a more general nature were consulted, related to typical protocols used in this type of architecture and analyzed in the previous section (Bluetooth, Zigbee, Modbus, BACnet, etc.); service protocols (NTP, network time protocol; RDP, remote desktop protocol; Telnet, Teletype Network, etc.); and other elements such as routers, firewalls, storage devices connected to the exposed network (NAS), common servers (Apache; NGINX; FTP, File Transfer Protocol), common databases (mySQL, MongoDB, etc.), and operating systems.

Table 5 shows, on the one hand, the set of terms analyzed for each type of device in Shodan and the number of results obtained. A global query was performed for all categories, using a term that allows searching for devices in the group of interest and a second search focused on a specific name as an example of a device in the category. All

of these were performed through the API and the search engine, in order to compare the results, which seem consistent. On the other hand, the second part of Table 5 lists the search terms used in the CVE database to relate the common vulnerabilities associated with the type of device to the results obtained using Shodan. In addition to the terms directly related to the established categories, other terms related to elements linked to other problems that were considered relevant for tagging the final results were collected.

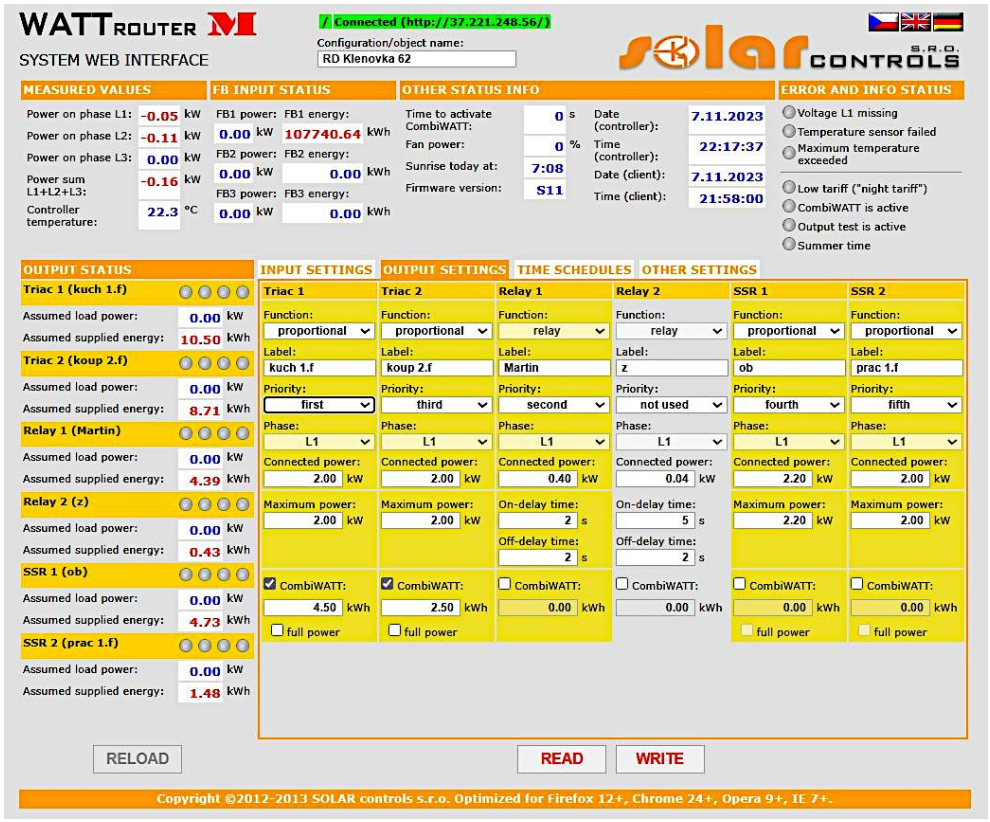

Figure 5. Example result of security issues in a photovoltaic installation management application. The search with the "country:CZ" tag provided access to the WATTrouter Mx application's configuration panel associated with a solar farm without any authentication

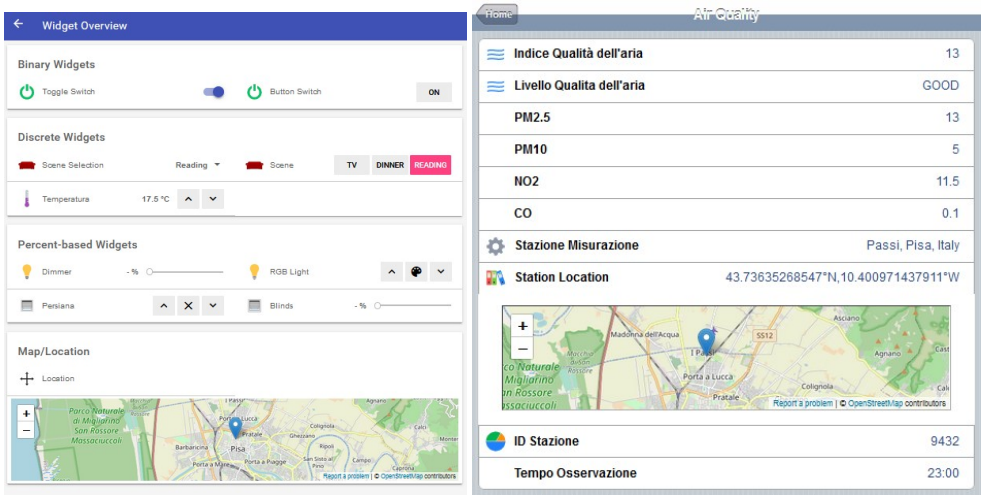

Figure 6. Example result of security issues in a smart home management application (OpenHAB) without access credentials. The "OpenHAB default" search enabled the detection of smart home management applications that lack access credentials, permitting the unrestricted adjustment of parameters linked to the various systems.

The results obtained for each of the categories are analyzed below, highlighting some vulnerabilities with a higher presence in the results (Table 6). The information from Shodan was cross-checked with that from the CVE database (Table 7). In the case of specific devices belonging to each category, an information contrast was carried out, firstly, by relating the information coming from the device-specific CVE, and second, by taking into account the general information of the category.

Table 5. Terms analyzed using the Shodan tool and terms analyzed in the CVE database related to the selected categories.

Category	Keyword (Shodan)	Results	Keyword (CVE DB)	Vulnerabilities
Smart thermostat	Thermostat	109	Smart thermostat	1156
	Heatmiser Wifi Thermostat	350	Heatmiser/EcoBee/ Radio Termostat	3
Smart plug	Smart plug	98	Smart plug	3
	WeMo Switch	10	Meross device/ WeMo device	12
Smart Camera	Smart Camera/ Connected Camera	1843/83	Connected Camera/Network Camera/Smart Camera/IP Camera	168
	DCS-2121	1237	D-Link DCS	43
Smartphone	Smart home phone	31	Smart home phone	121
	HomeMatic	751	OpenHAB/Homematic	34
Smart Lighting	Smart light	146	Smart bulb	8
	Philips Hue	169	Philips Hue/ Osram Lightify	11

- **Smart thermostat**. For the smart thermostat group, the specific device selected was a Heatmiser Wifi Thermostat. No vulnerabilities associated with the specific device type were obtained in either of the two cases analyzed. On the other hand, it was observed that the problems with the highest incidence were related to communication protocols and the use of specific servers and databases. Two examples of vulnerabilities obtained for the global category and one example for the specific thermostat model were selected. Their information is shown in Table 7. The three selected vulnerabilities have a critical severity, and all of them have associated exploits to exploit them.
- **Smart Plug**. The specific device selected was a WeMo Switch, and no vulnerabilities were found in any of the directions found in Shodan. No vulnerabilities associated with this type of specific device were found in the global case. It was also observed that the problems with the highest incidence were related to communication protocols, and the use of specific servers and databases. Two examples of vulnerabilities obtained for the global category were selected. This information is shown in Table 7. The first vulnerability has a medium severity, while the second one is critical. In addition, both have an associated exploit to exploit the vulnerability in the system.
- **Smart Camera**. The specific device selected was the D-Link DCS-2121. No vulnerabilities associated with the specific type of device were found in either of the two cases analyzed. On the other hand, it can be seen that the problems with the highest incidence were related to communication protocols, the use of specific servers, and the operating system. One example of a vulnerability was selected for the global category, and another for the chosen camera model. This information is shown in Table 7. The first vulnerability has a medium severity, while the second one is high. In addition, both have an associated exploit. In the case of the global category, the results included vulnerabilities already analyzed in previous sections, specifically: CVE-2018-1312 with 13 occurrences, CVE-2017-7679 with 12 occurrences, and CVE-2017-15906 with 5 occurrences. For the specific device analyzed, the vulnerability CVE-2019-0220 was also found 2 times and CVE-2018-1312 another 2 times.

- **Smartphone and applications**. The specific device selected for the management of different systems in the building was HomeMatic. No vulnerabilities associated with this specific type of device were found in either of the two cases analyzed. On the other hand, it can be seen that the problems with the highest incidence were related to communication protocols, the use of specific servers, and databases. Two examples of vulnerabilities obtained were selected, one for the global category, and the other for the selected application model. This information is shown in Table 7. The first vulnerability has a medium severity, while the second one has a high severity. In addition, both have associated exploits (1 for the first, and 2 for the second). In the case of the global category, the results included vulnerabilities already analyzed in previous sections, specifically: CVE-2017-15906 10 times, CVE-2018-1312 with 5 occurrences, CVE-2017-7679 with 5 occurrences, and CVE-2019-0220 with 5 occurrences. For the specific device analyzed, the vulnerability CVE-2019-0220 was also found 7 times, CVE-2018-1312 6 times, CVE-2017-15906 5 times, and CVE-2018-15919 5 times.
- **Smart Lighting**. For the smart lighting group, the results obtained are included in Table 6. The specific device selected within the category was a Philips Hue. No vulnerabilities associated with the specific device type were obtained in either of the two cases analyzed. On the other hand, it can be seen that the problems with the highest incidence were related to communication protocols, the use of specific servers, the operating system, and databases. An example of vulnerability obtained for the global category has been selected. This information is shown in Table 7. The vulnerability analyzed has a medium severity and also has 2 associated exploits. In the case of the global category, the results included vulnerabilities already analyzed in the previous sections, specifically CVE-2019-0220 with 4 occurrences; as well as CVE-2018-17199 and CVE 2018-1312, CVE-2019-0211 on 3 occasions; and on 2 occasions, CVE-2018-15919 and CVE-2017-15906. For the specific category, we again found twice the vulnerabilities CVE-2019-0220, CVE-2019-0211 and CVE-2018-15919, while 1 time we found CVE 2018-1312.

Table 6. Vulnerabilities obtained for the terms analyzed by category.

Category	Type	Vulnerabilities					
		Shodan	Device		Protocol		Other
Thermostat	Global	205	0	62	HTTP, FTP	63	Apache HTTPD, ProFTPD, OpenSSH, mySQL, PostgreSQL
	Heatmiser Wifi (sp.)	16	0	2	HTTP	2	Apache HTTPD
	Heatmiser Wifi (gl.)	16	0	2	HTTP	2	Apache HTTPD
Plug	Global	152	0	38	SSH, HTTP, FTP, SNMP	37	OpenSSH, Apache HTTPD, PostgreSQL, mySQL
	WeMo Switch (sp.)	0	0	0	-	0	-
	WeMo Switch (gl.)	0	0	0	-	0	-
Camera	Global	125	0	61	HTTP, FTP, SSH	83	Apache HTTPD, OpenSSH, Linux, ProFTPD
	DCS-2121 (sp.)	40	0	32	RDP, HTTP, FTP	30	Apache HTTPD, ProFTPD
	DCS-2121 (gl.)	40	0	32	RDP, HTTP, FTP	30	Apache HTTPD, ProFTPD
Home System	Global	344	0	84	HTTP, SSH, SNMP, FTP	80	OpenSSH, Apache HTTPD, ProFTPD, PostgreSQL, mySQL
	HomeMatic (sp.)	206	0	78	HTTP, FTP, SSH	77	Apache HTTPD, OpsenSSH, mySQL, NGINX, PostgreSQL
	HomeMatic (gl.)	206	0	78	HTTP, FTP, SSH	77	Apache HTTPD, OpsenSSH, mySQL, NGINX, PostgreSQL
Lighting	Global	205	0	32	HTTP	52	Apache HTTPD, OpenSSH, Linux, PostgreSQL
	Philips Hue (sp.)	28	0	25	HTTP	27	Apache HTTPD, OpenSSH
	Philips Hue (gl.)	28	0	25	HTTP	27	Apache HTTPD, OpenSSH

Table 7. Vulnerabilities obtained for the terms analyzed in the different categories.

Category	Type	Code	Overall CVSS	Occurrences	Exploits	Description
Thermostat	FTP, ProFTPD	CVE-2019-12815	9.8	1	1	Allows remote code execution and information disclosure without authentication.
	HTTP, Apache HTTPD	CVE-2018-1312	9.8	10	1	In common Digest authentication configuration, an attacker could replay HTTP requests.
		CVE-2017-7679	9.8	1	2	Read one byte beyond the end of a buffer when sending malicious content.
Plug	SSH, OpenSSH	CVE-2017-15906	5.3	4	1	An OpenSSH function allows attackers to create zero-length files.
	PostgreSQL	CVE-2017-15098	8.1	1	1	Some calls may crash the server or reveal some bytes of server memory.
Camera	HTTP, Apache HTTPD	CVE-2019-0220	5.3	10	1	A vulnerability was found in Apache HTTP Server related to consecutive slashes.
		CVE-2018-17199	7.5	2	1	The session expiry time is checked before decoding the session, and is ignored in some sessions.
Smartphone	SSH, OpenSSH	CVE-2018-15919	5.3	10	1	A behaviour observed in OpenSSH could even be used by remote attackers to detect the existence of users on a target system when GSS2 is in use.
		CVE-2019-0211	7.8	7	2	Executed code in secondary process/subprocessed having limited privileges could execute random code with root privileges.
Lighting	Linux	CVE-2019-9193	7.2	1	2	A functionality is enabled by default and can be abused to execute arbitrary operating system commands on Windows, Linux, and macOS.

7. Discussion

The term smart building has evolved significantly over the years. From the efficient management of resources and the proper functioning of a building, ensuring a certain level of automation, to the current concept, in which the building not only takes into account the environment in which it is located but centers its functioning around the user, in order to recognize needs, effectively manage resources, and ensure the well-being of the occupant. Technological advancements have facilitated this development, resulting in complex buildings requiring seamless coexistence and resource sharing, without compromising user safety.

The incorporation of new sensors and actuators in smart buildings increases the level of complexity. This has led to the progressive emergence of new communication mechanisms and an increase in the use of devices, triggering a greater number of unwanted interactions and creating a scenario that is increasingly exposed to more vulnerabilities. On the other hand, the growing number of smart buildings in the home environment brings multiple smart home devices and applications with simple functions into play. The remote control possibilities offered by such devices widen the range of possible attacks, as many of them lack basic security mechanisms, due to their simplicity and complexity.

This is, therefore, a scenario in which the number of interfaces for accessing information continues to increase, as do the vulnerabilities. Cybersecurity is essential at all levels, from application-level encryption through authentication systems, firewalls, private networks, and the physical layer. It is also important to be cautious about the privacy of individuals: much of the data collected must be handled and displayed with great care, so as not to violate users' rights. On the other hand, many of the common security problems on devices are related to human failures. These range from system usage errors to configuration problems, often due to a lack of cybersecurity knowledge or awareness. Facilitating the implementation of security mechanisms by simplifying their use or encouraging maintenance tasks could help increase user engagement.

Finally, it is possible to draw some conclusions from the practical analysis of the Shodan tool. The development of a program that facilitates the collection of open data on different devices linked to a building opens up the possibility of knowing details about them, such as their location, IP address, whether there are ports open to the connected elements, and even the firmware version used or the specific model of the product. In addition, Shodan provides information on associated vulnerabilities, many of them related to insecure configurations or insufficient authentication mechanisms. A manual inspection through a search engine allows detection of these types of flaws and easily gaining access to many of these systems, and even manipulating them, without going through high-security barriers, thus violating the privacy of the users of these systems.

A second type of analysis, through the API offered by the tool, and crossing these results with those obtained for similar searches and other relevant terms in the CVE vulnerabilities database, made it possible to identify some of the most common problems in the systems studied. As a relevant conclusion, none of the vulnerabilities specific to the particular devices analyzed were detected as a vulnerability present in the set of results returned by Shodan. Neither were they found in the analysis of searches for specific commercial devices of each type. However, the most frequent vulnerabilities seemed to , mainly be related to problems linked to protocols such as SSH, HTTP, and FTP; a specific server such as OpenSSH or Apache HTTPD; or databases.

8. Conclusions

This work aimed to identify and evaluate vulnerabilities specific to smart buildings, introducing an innovative assessment methodology and approach that leverages the Shodan tool as a potential framework. These vulnerabilities range from risks to system availability, cybersecurity, and data privacy, to the well-being of occupants. Through a focused effort to identify and assess these vulnerabilities, this study provides invaluable insights into fortifying the security of smart building environments. Central to this endeavor is the pivotal role played by Shodan, a specialized tool adept at network scanning and vulnerability detection.

The contributions achieved in this paper are multifold. First, it meticulously examines the primary risks, threats, and vulnerabilities inherent in intelligent buildings and their constituent components. The definitions of threat, risk, and vulnerability have been thoroughly examined, to understand the differences between them, as well as the security concerns that may be present in the situation under analysis. Subsequently, this document explored how threats manifest as attacks on specific systems in a smart building. This can serve as an invaluable tutorial for future research endeavors in this domain.

Second, the Shodan search engine was utilized to identify security issues on particular devices, recognize critical hazards and vulnerabilities, and reduce overall risks. To conduct the research and analyze the outcomes, a three-step methodology was proposed and executed to detect exploitable vulnerabilities in the context of smart buildings. The proposed methodology allowed the collection of open data on connected devices in buildings, to assess their potential risks. Using Shodan, either through the API or manual inspection, it was easy to access details of the associated vulnerabilities. These data were compared with results from similar searches and relevant terms found in the CVE vulnerability database, to identify common system issues. The Shodan results did not reveal any specific vulner-

abilities for the devices under review, including a smart thermostat, smart plug, smart camera, smartphone, and smart lighting. Furthermore, no specific commercial devices were vulnerable, according to searches for particular vulnerabilities. The most common issues were typically associated with using specific protocols, servers, or databases, as well as configuration and authentication issues.

As a result, the importance of enforcing security mechanisms within buildings to safeguard occupants' well-being was discussed. In addition, several mitigation strategies were proposed that could help prevent many of the problems listed when applied in particular ways.

As a final thought, the challenge posed by cybercrime is increasing. Organizations must deal with increasingly complex techniques to prevent attacks and equip building systems with technologies to ensure their protection. Cybersecurity audits help detect a system's weaknesses and vulnerabilities that could be exploited by malicious users or attackers targeting a particular organization or set of devices. They also help prevent information theft. These procedures should be documented in detail and contain specific recommendations to address individual issues.

This study's originality lies in using Shodan as a tool that can perform such checks on smart building systems, detect security flaws, and even find exploits or malicious code that can leverage weaknesses to trigger a particular desired behavior. The result is a framework for future advances in academic research, as well as by startups, for identifying security issues in devices commonly found in smart buildings and unlocking their full potential. Furthermore, the document provides instructions on evaluating whether the identified security issues require immediate attention because of their severity. The lack of real-world testing in many implementations means that the consequences of a severe system issue are not as well understood as they should be.

Moving forward, this analytical approach could be extended to scrutinizing various facets of intelligent systems. Expanding the analysis to encompass the security of prevalent architectural elements situated at diverse levels would offer a comprehensive understanding of potential vulnerabilities. Moreover, integrating artificial intelligence (AI) techniques into the design of building defense strategies presents a promising avenue for future research. This could encompass a range of practices, such as refining intrusion detection systems, enhancing access control measures, and automating configuration processes. This integration of advanced AI methodologies holds the potential to significantly elevate the overall security posture of smart buildings and similar IoT-enabled environments.

Author Contributions: Conceptualization, methodology, investigation, validation, software, formal analysis, and data curation, resources, writing—original draft preparation and visualization, S.M.-P.; writing—review and editing, S.M.-P. and V.M.B.; supervision, V.M.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Publicly available datasets were analyzed in this study. This data can be found here: [https://www.shodan.io/].

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AI	Artificial Intelligence
ANN	Artificial Neural Network
API	Application Programming Interface
BACnet	Building Automation and Control networks
BIM	Building Information Modeling
BLE	Bluetooth Low Energy
BPIE	Buildings Performance Institute Europe
CABA	Continental Automated Buildings Association
CIB	International Council for Building Research and Innovation in Building and Construction
CSV	Comma-Separated Values
CVE	Common Vulnerabilities Exposures
CVSS	Common Vulnerability Space System
DoS/DDoS	Denial of Service/Distributed Denial of Service
DSL	Digital Subscriber Line
DCCNN	Dual Channel Convolution Neural Network
DVR	Digital Video Recorder
EC	European Commission
FTP	File Transfer Protocol
HTTP	Hypertext Transfer Protocol
HVAC	Heating, Ventilation and Air Conditioning
IBM	International Business Machines Corporation
IEEE	Institute of Electrical and Electronics Engineers
IBI	Intelligent Building Institution
IoT	Internet of Things
IP	Internet Protocol
JSON	JavaScript Object Notation
LON	Local Operating Network
MAC	Medium Access Control
ML	Machine Learning
MiTM	Man-in-The-Middle
MQTT	Message Queuing Telemetry Transport
NAS	Network Attached Storage
Nmap	Network mapper
NTP	Network Time Protocol
NVD	National Vulnerability Database
RDP	Remote Desktop Protocol
RTP/SRTP	Real-time Transport Protocol/Secure Real-time Transport Protocol
SCA	Side-Channel Attack
SCADA	Supervisory Control And Data Acquisition
SMO	Spider Monkey Optimization
SNMP	Simple Network Management Protocol
SSH	Secure Shell
SSL/TLS	Secure Sockets Layer/Transport Layer Security
SVM	Support Vector Machine
TCP	Transport Layer Security
Telnet	Teletype Network
ICT	Information and Communication Technology
UDP	User Datagram Protocol
WiFi	Wireless Fidelity
WPA	WiFi Protected Access
XML	EXtensive Markup Language
ZLL	Zigbee Light Link

References

1. Beyrouti, M.; Lounis, A.; Lussier, B.; Bouadallah, A.; Samhat, A.E. Vulnerability and Threat Assessment Framework for Internet of Things Systems. In Proceedings of the 6th Conference on Cloud and Internet of Things (CIoT), Lisbon, Portugal, 20–22 March 2023; pp. 62–69. [CrossRef]
2. Villar Miguelez, C.; Monzon Baeza, V.; Parada, R.; Monzo, C. Guidelines for Renewal and Securitization of a Critical Infrastructure Based on IoT Networks. *Smart Cities* **2023**, *6*, 728–743. [CrossRef]
3. Tarazona Lizarraga, C. *Análisis de las Necesidades de una Smart City en el Marco de un Desarrollo Sostenible*; Universitat Oberta de Catalunya: Barcelona, Spain, 2020.
4. Omar, S. Intelligent building, definitions, factors and evaluation criteria of selection. *Alex. Eng. J.* **2018**, *57*, 2903–2910. [CrossRef]
5. Mulero Palencia, S. *Vulnerabilidades en Edificios Inteligentes*; Universitat Oberta de Catalunya: Barcelona, Spain, 2021.
6. Commission, E. Smart Building: Energy Efficiency Application. 2017. Available online: https://ati.ec.europa.eu/sites/default/files/2020-06/Smart%20Building-%20Energy%20efficiency%20application%20%28v1%29.pdf (accessed on 1 October 2023).
7. *IEEE Std 2785-2023*; IEEE Standard for Architectural Framework and General Requirements for Smart Home Systems. IEEE: Piscataway, NJ, USA, 2023; pp. 1–50. [CrossRef]
8. Eneyew, D.D.; Capretz, M.A.M.; Bitsuamlak, G.T. Toward Smart-Building Digital Twins: BIM and IoT Data Integration. *IEEE Access* **2022**, *10*, 130487–130506. [CrossRef]
9. Ma, G.; Dang, S.; Alouini, M.S.; Shihada, B. Smart Buildings Enabled by 6G Communications. *IEEE Internet Things Mag.* **2022**, *5*, 181–186. [CrossRef]
10. Mohammed, B.H.; Sallehudin, H.; Mohamed, S.A.; Satar, N.S.M.; Hussain, A.H.B. Internet of Things-Building Information Modeling Integration: Attacks, Challenges, and Countermeasures. *IEEE Access* **2022**, *10*, 74508–74522. [CrossRef]
11. Kumari, P.; Gupta, H.P. An Energy-Efficient Smart Building System using Autonomous Networks. *IEEE Commun. Stand. Mag.* **2022**, *6*, 32–36. [CrossRef]
12. Nguyen, D.H. Enhancing Building Energy Efficiency Through Its Windows. In Proceedings of the 2023 10th International Conference on Power and Energy Systems Engineering (CPESE), Nagoya, Japan, 8–10 September 2023; pp. 141–146. [CrossRef]
13. Kim, D.; Yoon, Y.; Lee, J.; Mago, P.J.; Lee, K.; Cho, H. Design and Implementation of Smart Buildings: A Review of Current Research Trend. *Energies* **2022**, *15*, 4278. [CrossRef]
14. Yagüe García, S. *Análisis del rol de las Casas Inteligentes en Smart City*; Universitat Oberta de Catalunya: Barcelona, Spain, 2021.
15. Aliero, M.S.; Qureshi, K.N.; Pasha, M.F.; Jeon, G. Smart Home Energy Management Systems in Internet of Things networks for green cities demands and services. *Environ. Technol. Innov.* **2021**, *22*, 101443. [CrossRef]
16. Apanavičienė, R.; Shahrabani, M.M.N. Key Factors Affecting Smart Building Integration into Smart City: Technological Aspects. *Smart Cities* **2023**, *6*, 1832–1857. [CrossRef]
17. Habash, R. 4 - Building as a smart system. In *Sustainability and Health in Intelligent Buildings*; Habash, R., Ed.; Woodhead Publishing Series in Civil and Structural Engineering; Woodhead Publishing: Sawston, UK, 2022; pp. 95–128. [CrossRef]
18. Qolomany, B.; Al-Fuqaha, A.; Gupta, A.; Benhaddou, D.; Alwajidi, S.; Qadir, J.; Fong, A.C. Leveraging Machine Learning and Big Data for Smart Buildings: A Comprehensive Survey. *IEEE Access* **2019**, *7*, 90316–90356. [CrossRef]
19. IoTSF. Can You Trust Your Smart Building? Understanding the Security Issues and Why They Are Important to You. 2019. Available online: https://www.iotsecurityfoundation.org/wp-content/uploads/2019/07/IoTSF-Smart-Buildings-White-Paper-PDFv2.pdf (accessed on 1 October 2023).
20. Li, G.; Ren, L.; Fu, Y.; Yang, Z.; Adetola, V.; Wen, J.; Zhu, Q.; Wu, T.; Candan, K.; O'Neill, Z. A critical review of cyber-physical security for building automation systems. *Annu. Rev. Control* **2023**, *55*, 237–254. [CrossRef]
21. Bi, Y.; Huang, J.; Liu, P.; Wang, L. Benchmarking Software Vulnerability Detection Techniques: A Survey. *arXiv* **2023**, arXiv:2303.16362.
22. Jain, V.K.; Tripathi, M. Multi-Objective Approach for Detecting Vulnerabilities in Ethereum Smart Contracts. In Proceedings of the 2023 International Conference on Emerging Trends in Networks and Computer Communications (ETNCC), Windhoek, Namibia, 16–18 August 2023; pp. 1–6. [CrossRef]
23. Mariappan, U.; Balakrishnan, D.; Rajendran, S.; Alagusundar, N.; Sheriff, A.A.; K, A. Cyber Security Threat Detection in Internet of Things Using Optimized Deep Learning Technique. In Proceedings of the 2023 3rd Asian Conference on Innovation in Technology (ASIANCON), Ravet, IN, India, 25–27 August 2023; pp. 1–6. [CrossRef]
24. Saidin, S.B.; Hisham, S.B.I. A Survey on Supervised Machine Learning in Intrusion Detection Systems for Internet of Things. In Proceedings of the 2023 IEEE 8th International Conference On Software Engineering and Computer Systems (ICSECS), Penang, Malaysia, 25–27 August 2023; pp. 419–423. [CrossRef]
25. Khatri, A.; Khatri, R. DDoS Attack Detection Using Artificial Neural Network on IoT Devices in a Simulated Environment. In *International Conference on IoT, Intelligent Computing and Security: Select Proceedings of IICS 2021*; Agrawal, R., Mitra, P., Pal, A., Sharma Gaur, M., Eds.; Springer: Singapore, 2023; pp. 221–233.
26. Abdaljabar, Z.H.; Ucan, O.N.; Ali Alheeti, K.M. An Intrusion Detection System for IoT Using KNN and Decision-Tree Based Classification. In Proceedings of the 2021 International Conference of Modern Trends in Information and Communication Technology Industry (MTICTI), Sana'a, Yemen, 4–6 December 2021; pp. 1–5. [CrossRef]
27. Talita, A.S.; Nataza, O.S.; Rustam, Z. Naïve Bayes Classifier and Particle Swarm Optimization Feature Selection Method for Classifying Intrusion Detection System Dataset. *J. Phys. Conf. Ser.* **2021**, *1752*. [CrossRef]

28. Salimi, S.; Kharrazi, M. VulSlicer: Vulnerability detection through code slicing. *J. Syst. Softw.* **2022**, *193*, 111450. [CrossRef]
29. Tenable. Nessus Scan Tuning Guide. Available online: https://docs.tenable.com/quick-reference/nessus-scan-tuning/Content/PDF/Nessus_Scan_Tuning_Guide.pdf (accessed on 28 October 2023).
30. Kali. Skipfish Home Page. Available online: https://www.kali.org/tools/skipfish/ (accessed on 28 October 2023).
31. Mankali, L.; Patnaik, S.; Limaye, N.; Knechtel, J.; Sinanoglu, O. VIGILANT: Vulnerability Detection Tool Against Fault-Injection Attacks for Locking Techniques. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **2023**, *42*, 3571–3584. [CrossRef]
32. Censys. Censys Official Web Page. Available online: https://censys.io (accessed on 28 October 2023).
33. ZoomEye. ZoomEye Official Web Page. Available online: https://www.zoomeye.org (accessed on 28 October 2023).
34. Thingful. Thingful Official Web Page. Available online: https://www.thingful.net (accessed on 28 October 2023).
35. Shodan. 2013. Available online: https://www.shodan.io/ (accessed on 1 October 2023).
36. Tundis, A.; Mazurczyk, W.; Mühlhäuser, M. A Review of Network Vulnerabilities Scanning Tools: Types, Capabilities and Functioning. In Proceedings of the 13th International Conference on Availability, Reliability and Security, New York, NY, USA, 27–30 August 2018. [CrossRef]
37. Safavi, S.; Meer, A.; Keneth Joel Melanie, E.; Shukur, Z. Review and Solutions. In Proceedings of the Cyber Resilience Conference (CRC), Putrajaya, Malasia, 13–15 November 2018.
38. Fernández-Caramés, T.; Fraga-Lamas, P. Teaching and Learning IoT Cybersecurity and Vulnerability Assessment with Shodan through practical use cases. *Sensors* **2020**, *20*, 3048. [CrossRef]
39. Ciholas, P.; Lennie, A.; Sadigova, P.; Such, J. The Security of Smart Buildings: A systematic literature review. *arXiv* **2019**, arXiv:1901.05837.
40. Brooks, D. Security threats and risks of Intelligent Building Systems: Protecting facilities from current and emerging vulnerabilities. In *Security Threats and Risks of Intelligent Building Systems: Protecting Facilities from Current and Emerging Vulnerabilities*; IGI Global: Hershey, PA, USA, 2012; pp. 1–16. [CrossRef]
41. Graveto, V.; Cruz, T.; Simões, P. Security of Building Automation and Control Systems: Survey and future research directions. *Comput. Secur.* **2022**, *112*, 102527. [CrossRef]
42. Seferi, R.; Giangiacomi, S.; Berberi, K. Vulnerabilities and Attacks in a Smart Buildings Scenario. In Proceedings of the 2019 IEEE 23rd International Symposium on Consumer Technologies (ISCT), Ancona, Italy, 19–21 June 2019; pp. 296–298. [CrossRef]
43. Smart Home/Smart Building Connectivity Options and Their Cybersecurity. 2020. Available online: https://society5.com/smart-cities/smart-building-smart-home-connectivity-cybersecurity/ (accessed on 1 October 2023).
44. Wendzel, S.; Tonejc, J.; Kaur, J.; Kobekova, A. Cyber Security of Smart Buildings. In *En Security and Privacy in Cyber-Physical Systems: Foundations, Principles, and Applications*; John Wiley & Sons Ltd.: Hoboken, NJ, USA, 2018; pp. 327–351.
45. Mekala, S.H.; Baig, Z.; Anwar, A.; Zeadally, S. Cybersecurity for Industrial IoT (IIoT): Threats, countermeasures, challenges and future directions. *Comput. Commun.* **2023**, *208*, 294–320. [CrossRef]
46. Ignacio Porro Sáez, I.C. IoT: Protocolos de Comunicación, Ataques y Recomendaciones. 2019. Available online: https://www.incibe-cert.es/blog/iot-protocolos-comunicacion-ataques-y-recomendaciones (accessed on 1 October 2023).
47. Zohourian, A.; Dadkhah, S.; Neto, E.C.P.; Mahdikhani, H.; Danso, P.K.; Molyneaux, H.; Ghorbani, A.A. IoT Zigbee device security: A comprehensive review. *Internet Things* **2023**, *22*, 100791. [CrossRef]
48. Keşkişoğlu, A.; Turhan, C. Challenges on smart thermostat systems in Intelligent Buildings. In Proceedings of the 4th International Energy and Engineering Congress, Gaziantep, Turkey, 24–25 October 2019.
49. Moody, M.; Hunter, A. Exploiting known vulnerabilities of a smart thermostat. In Proceedings of the 2016 14th Annual Conference on Privacy, Security and Trust (PST), Auckland, New Zealand, 12–14 December 2016; pp. 50–53. [CrossRef]
50. Ling, Z.; Luo, J.; Xu, Y.; Gao, C.; Wu, K.; Fu, X. Security Vulnerabilities of Internet of Things: A Case Study of the Smart Plug System. *IEEE Internet Things J.* **2017**, *4*, 1899–1909. [CrossRef]
51. Suryadevara, N.K.; Biswal, G.R. Smart Plugs: Paradigms and Applications in the Smart City-and-Smart Grid. *Energies* **2019**, *12*, 1957. [CrossRef]
52. Bugeja, J.; Jönsson, D.; Jacobsson, A. An Investigation of Vulnerabilities in Smart Connected Cameras. In Proceedings of the 2018 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops), Athens, Greece, 19–23 March 2018; pp. 537–542. [CrossRef]
53. Alharbi, R.; Aspinall, D. An IoT analysis framework: An investigation of IoT smart cameras' vulnerabilities. In Proceedings of the Living in the Internet of Things: Cybersecurity of the IoT—2018, London, UK, 28–29 March 2018; pp. 1–10. [CrossRef]
54. Teixeira, D.; Assunção, L.; Paiva, S. Security of Smart Home-Smartphones Systems. In Proceedings of the 2020 15th Iberian Conference on Information Systems and Technologies (CISTI), Seville, Spain, 24–27 June 2020; pp. 1–5. [CrossRef]
55. Karimi, K.; Krit, S. Smart home-Smartphone Systems: Threats, Security Requirements and Open research Challenges. In Proceedings of the 2019 International Conference of Computer Science and Renewable Energies (ICCSRE), Agadir, Morocco, 22–24 July 2019; pp. 1–5. [CrossRef]
56. Morgner, P.; Mattejat, S.; Benenson, Z. All Your Bulbs Are Belong to Us: Investigating the Current State of Security in Connected Lighting Systems. *arXiv* **2016**, arXiv:1608.03732.
57. Gui, Y.; Siddiqui, A.S.; Tamore, S.M.; Saqib, F. Investigation of Vulnerabilities on Smart Grid End Devices. In Proceedings of the 2019 IEEE CyberPELS (CyberPELS), Knoxville, TN, USA, 29 April–1 May 2019; pp. 1–6. [CrossRef]

58. Yaacoub, J.P.A.; Noura, H.N.; Salman, O.; Chehab, A. Ethical hacking for IoT: Security issues, challenges, solutions and recommendations. *Internet Things Cyber-Phys. Syst.* **2023**, *3*, 280–308. [CrossRef]
59. Hammi, B.; Zeadally, S.; Khatoun, R.; Nebhen, J. Survey on smart homes: Vulnerabilities, risks, and countermeasures. *Comput. Secur.* **2022**, *117*, 102677. [CrossRef]
60. Matherly, J. *The Complete Guide to Shodan: Collect. Analyze Visualize. Make Internet Intelligence Work for You*; Leanpub: Victoria, BC, Canada, 2016.
61. Fagroud, F.Z.; Ajallouda, L.; Ben Lahmar, E.H.; Toumi, H.; Achtaich, K.; Filali, S.E. IOT Search Engines: Exploratory Data Analysis. *Procedia Comput. Sci.* **2020**, *175*, 572–577. [CrossRef]
62. Tundalwar, D.S.; Pandhare, R.A.; Digalwar, M.A. A Taxonomy of IoT Security Attacks and Emerging Solutions. In Proceedings of the 2023 2nd International Conference on Paradigm Shifts in Communications Embedded Systems, Machine Learning and Signal Processing (PCEMS), Nagpur, India, 5–6 April 2023; pp. 1–5. [CrossRef]
63. Papp, D.; Ma, Z.; Buttyan, L. Embedded systems security: Threats, vulnerabilities, and attack taxonomy. In Proceedings of the 2015 13th Annual Conference on Privacy, Security and Trust (PST), Izmir, Turkey, 21–23 July 2015; pp. 145–152. [CrossRef]
64. Al-Alami, H.; Hadi, A.; Al-Bahadili, H. Vulnerability scanning of IoT devices in Jordan using Shodan. In Proceedings of the 2017 2nd International Conference on the Applications of Information Technology in Developing Renewable Energy Processes & Systems (IT-DREPS), Amman, Jordan, 6–7 December 2017; pp. 1–6. [CrossRef]
65. Ahamed, J.; Rajan, A.V. Internet of Things (IoT): Application systems and security vulnerabilities. In Proceedings of the 2016 5th International Conference on Electronic Devices, Systems and Applications (ICEDSA), Ras Al Khaimah, United Arab Emirates, 6–8 December 2016; pp. 1–5. [CrossRef]
66. McMahon, E.; Williams, R.; El, M.; Samtani, S.; Patton, M.; Chen, H. Assessing medical device vulnerabilities on the Internet of Things. In Proceedings of the 2017 IEEE International Conference on Intelligence and Security Informatics (ISI), Beijing, China, 22–24 July 2017; pp. 176–178. [CrossRef]
67. Albataineh, A.; Alsmadi, I. IoT and the Risk of Internet Exposure: Risk Assessment Using Shodan Queries. In Proceedings of the 2019 IEEE 20th International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM), Washington, DC, USA, 10–12 June 2019; pp. 1–5. [CrossRef]
68. Markowsky, L.; Markowsky, G. Scanning for vulnerable devices in the Internet of Things. In Proceedings of the 2015 IEEE 8th International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS), Warsaw, Poland, 24–26 September 2015; Volume 1, pp. 463–467. [CrossRef]
69. Bodenheim, R.; Butts, J.; Dunlap, S.; Mullins, B. Evaluation of the ability of the Shodan search engine to identify Internet-facing industrial control devices. *Int. J. Crit. Infrastruct. Prot.* **2014**, *7*, 114–123. [CrossRef]
70. Patton, M.; Gross, E.; Chinn, R.; Forbis, S.; Walker, L.; Chen, H. Uninvited Connections: A Study of Vulnerable Devices on the Internet of Things (IoT). In Proceedings of the 2014 IEEE Joint Intelligence and Security Informatics Conference, The Hague, The Netherlands, 24–26 September 2014; pp. 232–235. [CrossRef]
71. Antonio López. Métricas de Evaluación de Vulnerabilidades: CVSS 3.0. 2015. Available online: https://www.incibe.es/incibe-cert/blog/cvss3-0 (accessed on 1 October 2023).
72. (INCIBE), H.R.S. Midiendo la Severidad de las Vulnerabilidades: Cambios en CVSS 3.1. 2019. Available online: https://www.incibe-cert.es/blog/midiendo-severidad-las-vulnerabilidades-cambios-cvss-31 (accessed on 1 October 2023).
73. FIRST. Common Vulnerability Scoring System v3.1: Specification Document. 2019. Available online: https://www.first.org/cvss/v3.1/specification-document (accessed on 1 October 2023).
74. NVD. Common Vulnerability Scoring System Calculator. 2019. Available online: https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator (accessed on 1 October 2023).