symmetry

Article

IoT Vulnerabilities and Attacks: SILEX Malware Case Study

Basem Ibrahim Mukhtar [1], Mahmoud Said Elsayed [2,*], Anca D. Jurcut [2] and Marianne A. Azer [1,3]

1 School of Information Technology and Computer Science, Nile University, Cairo 12566, Egypt; mazer@nu.edu.eg (M.A.A.)
2 School of Computer Science, University College Dublin, D04 V1W8 Dublin, Ireland; anca.jurcut@ucd.ie
3 Computers and Systems Department, National Telecommunication Institute, Cairo 12677, Egypt
* Correspondence: mahmoud.abdallah@ucdconnect.ie

Abstract: The Internet of Things (IoT) is rapidly growing and is projected to develop in future years. The IoT connects everything from Closed Circuit Television (CCTV) cameras to medical equipment to smart home appliances to smart automobiles and many more gadgets. Connecting these gadgets is revolutionizing our lives today by offering higher efficiency, better customer service, and more effective goods and services in a variety of industries and sectors. With this anticipated expansion, many challenges arise. Recent research ranked IP cameras as the 2nd highest target for IoT attacks. IoT security exhibits an inherent asymmetry where resource-constrained devices face attackers with greater resources and time, creating an imbalanced power dynamic. In cybersecurity, there is a symmetrical aspect where defenders implement security measures while attackers seek symmetrical weaknesses. The SILEX malware case highlights this asymmetry, demonstrating how IoT devices' limited security made them susceptible to a relatively simple yet destructive attack. These insights underscore the need for robust, proactive IoT security measures to address the asymmetrical risks posed by adversaries and safeguard IoT ecosystems effectively. In this paper, we present the IoT vulnerabilities, their causes, and how to detect them. We focus on SILEX, one of the famous malware that targets IoT, as a case study and present the lessons learned from this malware.

Keywords: cybersecurity; cyber-attacks; IoT; security; SILEX malware; smart homes; vulnerabilities; asymmetry; symmetry

check for updates

Citation: Mukhtar, B.I.; Elsayed, M.S.; Jurcut, A.D.; Azer, M.A. IoT Vulnerabilities and Attacks: SILEX Malware Case Study. *Symmetry* **2023**, *15*, 1978. https://doi.org/10.3390/sym15111978

Received: 18 July 2023
Revised: 3 October 2023
Accepted: 23 October 2023
Published: 26 October 2023

1. Introduction

The Internet of Things (IoT) strives to connect everyday physical objects into an interconnected ecosystem of digital data that are accessible at any time and from any location. In the IoT, "things" are embedded with sensing, processing, and actuation capabilities and work independently to give smart and novel services [1]. In [2], the basic IoT architecture was described as follows: Sensors are attached to each IoT component in the first layer to record the environment, which is then sent across the network to the base station in the second layer. The data acquired at the base station are transferred to the next level for data analysis and prediction. The number of IoT devices is growing rapidly; by 2025, it is expected to reach 64 billion [3]. This continuing adoption of IoT devices around the world and the evolution of smart devices with more features than typical home devices, along with the absence of adequate security knowledge, has led to the appearance of new types of cyber-threats that have devastating and dangerous effects [4].

As a subset of IoT, the Industrial Internet of Things (IIoT) signifies a profound and transformative shift within the industrial landscape. It revolves around the seamless integration of sensors, devices, machinery, and data analytics into industrial operations. IIoT applications transcend diverse sectors, such as manufacturing, energy, healthcare, and agriculture. For instance, within manufacturing, IIoT plays a pivotal role in enabling predictive maintenance, thereby curtailing downtime and optimizing production processes. In the energy sector, it empowers intelligent grid management and resource optimization.

As more people and organizations invest in buying smart devices such as routers, CCTV cameras, sensors, and many other smart devices, cybercriminals are seeing financial chances of attacking these appliances [5]. They use exploited networks of smart devices to conduct Distributed Denial of Service (DDoS) attacks or other types of malicious attacks [6]. To know how these new types of attacks work, honeypots, which are fake networks used to attract cybercriminals and examine their behaviors [7]. Kaspersky detected more than 100 million IoT attacks from 276,000 unique Internet Protocol (IP) addresses in the first six months of 2019 [8]. This number is seven times higher than the number detected in the first half of 2018 when around 12 million were captured coming from 69,000 IP addresses [8]. The convergence of two different worlds, symmetry phenomena and the complex landscape of IoT vulnerabilities and attacks, may not be immediately apparent. However, there exist many ideas and insights that can illuminate the challenges and dynamics of safeguarding the IoT in an increasingly connected world.

In this paper, we focus on the main vulnerabilities existing in IoT devices, their causes, and how to detect them through organized detection stages. We also explore the relationship between symmetry concepts and IoT security, emphasized through a case study: the SILEX malware. By examining this complex connection, we shed light on the complexities of safeguarding IoT ecosystems in the face of asymmetrical threats. The lessons learnt from the SILEX malware are addressed, as well as recommendations for IoT security.

The contributions of this paper are as follows:

1. Present the research carried out in the area of IoT attacks.
2. Discuss the IoT security vulnerabilities, their causes, detection stages and actions.
3. Overview the IoT attacks' classification, their advantages and disadvantages from an attackers' perspective.
4. Overview the SILEX malware and the lessons learned in securing against this malware.
5. Highlight the inherent imbalance in security capabilities between IoT devices and attackers.
6. Explain the symmetrical aspect of the ongoing competition between cybersecurity defenders and attackers and emphasize the importance of addressing the inherent asymmetry in IoT security.
7. Examine the SILEX malware as an example of an asymmetric cyberattack targeting IoT devices.

The remainder of this paper is organized as follows: Section 2 presents related work on IoT attacks. Sections 3 and 4 focus on the IoT security vulnerabilities and the SILEX malware. Finally, the conclusions are presented in Section 5.

2. Related Work

In this section, we explain the asymmetry in IoT Security and present the research in the area of IoT attacks, vulnerabilities, and mitigations in Sections 2.1 and 2.2, respectively.

2.1. Asymmetry in IoT Security

In IoT networks, there is often an asymmetry in terms of security capabilities between the devices (sensors, smart appliances, etc.) and the attackers. IoT devices are typically resource-constrained and may lack robust security mechanisms, making them vulnerable to attacks. Attackers, on the other hand, have more resources and time to exploit these vulnerabilities, creating an asymmetrical power dynamic. Table 1 summarizes the key aspects of asymmetry in IoT security, highlighting the disparities between IoT devices and attackers across various dimensions.

Table 1. IoT Security asymmetry comparison: challenges faced by IoT devices versus advantages of attackers.

Aspects	**IoT Devices**	**Attackers**
Resources	Limited resources, vulnerable.	Abundant resources, exploit vulnerabilities.
Security Expertise	Lack of security focus.	Cybercriminal expertise, various attacks.
Firmware and Patching	Limited updates, exposed.	Exploit unpatched devices.
Economic Asymmetry	Cost constraints, compromises.	Economic gains from attacks.
Communication Bandwidth	Limited bandwidth, slow updates.	High speed, exploit vulnerabilities quickly.
Distributed Attack Infrastructure	Dispersed devices, coordination challenge.	Botnets, large-scale attacks.
Regulatory and Compliance	Compliance burden, resource intensive.	Operate without compliance restrictions.
Legal and Ethical	Legal constraints, ethical boundaries.	Operate outside legal norms.
Resource Scalability	Complex and costly scaling.	Quickly scale attacks.
Dependency on Third-Party Services	Rely on third-party services.	Exploit service vulnerabilities.
Time Horizon	Long device lifecycles, delayed patches.	Wait for vulnerabilities, persistent attacks.
Legal Jurisdiction	Varying legal requirements.	Operate from lax jurisdictions.
Supply Chain Vulnerabilities	Vulnerable supply chain.	Exploit supply chain weaknesses.
Budget Allocation	Limited security budgets.	Invest in IoT vulnerability exploitation.
Legal Implications	Cautious active defense.	Deploy aggressive attack techniques.
Skillset Availability	Limited IoT security skills.	Global pool of cybercriminal expertise.
Scale of Deployments	Vast device deployments, challenging management.	Target multiple devices, increase effectiveness.
Vendor and Platform Diversity	Diverse ecosystem complexity.	Exploit device and platform diversity.
Resource Allocation for Security Monitoring	Limited security monitoring.	Exploit monitoring gaps, covert operations.
Human Error and Training	Human error vulnerabilities.	Exploit misconfigurations and lack of awareness.
Publicly Available Information	Information exposure risks.	Leverage publicly available data for attacks.
Complexity of IoT Ecosystems	Complex ecosystem vulnerabilities.	Exploit complex system interactions.

2.2. Literature Review

In this section, we present the research in the area of IoT attacks, vulnerabilities, and mitigations. A survey on detecting cybersecurity attacks on the Internet of Things using artificial intelligence methods was presented in [9]. The authors in [10] examined intrusion detection systems in the Internet of Things, including approaches, deployment strategies, validation strategies, threats, public datasets, and obstacles. A survey on attacks and countermeasures of cyber-threats to industrial IoT was presented in [11]. The authors of [12] investigated the impact of several forms of DoS attacks on IoT gateways and determined that the wired-to-wireless category is the most harmful. A deep study of how a honeypot system can be used to avert a DoS attack was proposed, and verified, in [13], while a study of different attacks targeting IoT systems comparing their damage and efficiency levels was presented in [14]. In [15], the authors studied the IP camera security posture and concluded that the camera has three vulnerabilities: data are not transferred encrypted; the rTSP URL used to stream data may be brute-forced; and the credentials are saved in a mobile application in clear text. DDoS attacks on both IPV4 and IPv6 networks were tested and compared in [16]. The authors concluded that IPV6 networks have a higher risk of being attacked through DDoS attacks. The security of video surveillance systems was reviewed in [17], and the authors provided best practices to protect them. The attacks, vulnerabilities, and methods to protect smart homes and cities were summarized and presented in [18]. The attacks that threaten the confidentiality, integrity, and availability of IoT devices were analyzed in [19]. In [20], the authors analyzed two types of internet-connected cameras and found new unknown security vulnerabilities. The authors in [21] presented a taxonomy of IoT attacks. They mentioned a range of threats that can compromise the security and functionality of connected devices and systems. Common IoT attacks include DDoS attacks, which flood IoT devices with traffic to overwhelm and disrupt their operations. In addition, MQTT exploit is the de facto standard communication protocol in the IoT systems and is a widely used lightweight messaging protocol designed for efficient communication between IoT devices and systems.

The challenges and obstacles faced when implementing IoT solutions were discussed in [22]. This could include cybersecurity concerns issues, data privacy, and scalability challenges. IoT systems, often comprising diverse devices and sensors, face vulnerabilities that can be exploited by malicious actors. Common cyber-threats in IoT include malware infiltration, data breaches, DoS attacks, and unauthorized access to operational technology networks. These attacks can disrupt essential industrial processes, compromise data integrity, and pose significant risks to safety and production. Interested readers can refer to [23], and [24] for further details about IoT standards, challenges, and cybersecurity strategies.

3. IoT Security

In this section, we discuss the IoT security vulnerabilities and the IoT symmetry in attacks and defense in Sections 3.1 and 3.2, respectively.

3.1. IoT Security Vulnerabilities and Attacks

The IoT suffers from a lot of vulnerabilities. A taxonomy of IoT vulnerabilities was presented in [25]. The IoT Devices Vulnerability statistics to Palo Alto Networks 2018 mentioned in [26] reveal the weight of different categories of vulnerabilities. The users are the weakest link in the security ecosystem. The IoT security vulnerabilities are due to many reasons. Figure 1 depicts the main causes of IoT security vulnerabilities. Table 2 presents a comprehensive taxonomy of IoT vulnerabilities, categorizing them into distinct levels that are further classified into specific subcategories. Each subcategory is accompanied by illustrative examples, shedding light on the diverse range of threats that IoT systems face. Understanding and addressing these vulnerabilities is crucial in fortifying the security of IoT ecosystems and ensuring their reliable and safe operation.

The security of Internet of Things (IoT) devices is of paramount importance in today's interconnected world. As the number of IoT devices continues to proliferate, so does the potential attack surface for malicious actors. To safeguard against vulnerabilities and threats, a systematic and rigorous approach to IoT vulnerabilities' detection and analysis is essential. The OWASP identified analysis stages in order to detect the IoT firmware vulnerabilities [26]. We summarized the different stages in Figure 2.

This process involves a series of carefully orchestrated stages, each with its specific objectives and methods. These stages aim to dissect and scrutinize every aspect of an IoT device, from its firmware to its runtime behavior, to uncover potential weaknesses that could be exploited by adversaries. In this comprehensive guide, we will delve into each stage of the IoT vulnerabilities' detection process, providing in-depth insights, examples, and the tools required for effective analysis.

From the initial step of collecting information and acquiring firmware to the critical phases of dynamic analysis and exploitation, we will explore the intricacies of IoT security assessment. Each stage contributes to building a comprehensive understanding of an IoT device's security posture, enabling organizations to fortify their defenses and protect against emerging threats.

Whether for a security professional, a developer, or an organization seeking to ensure the security of IoT deployments, the following steps explain the knowledge and methodologies needed to conduct thorough IoT vulnerabilities' detection, ultimately contributing to a safer and more secure IoT ecosystem. Table 3 summarizes each step of the IoT vulnerability detection process along with the objectives, description, tools, examples, challenges, advantages, and disadvantages.

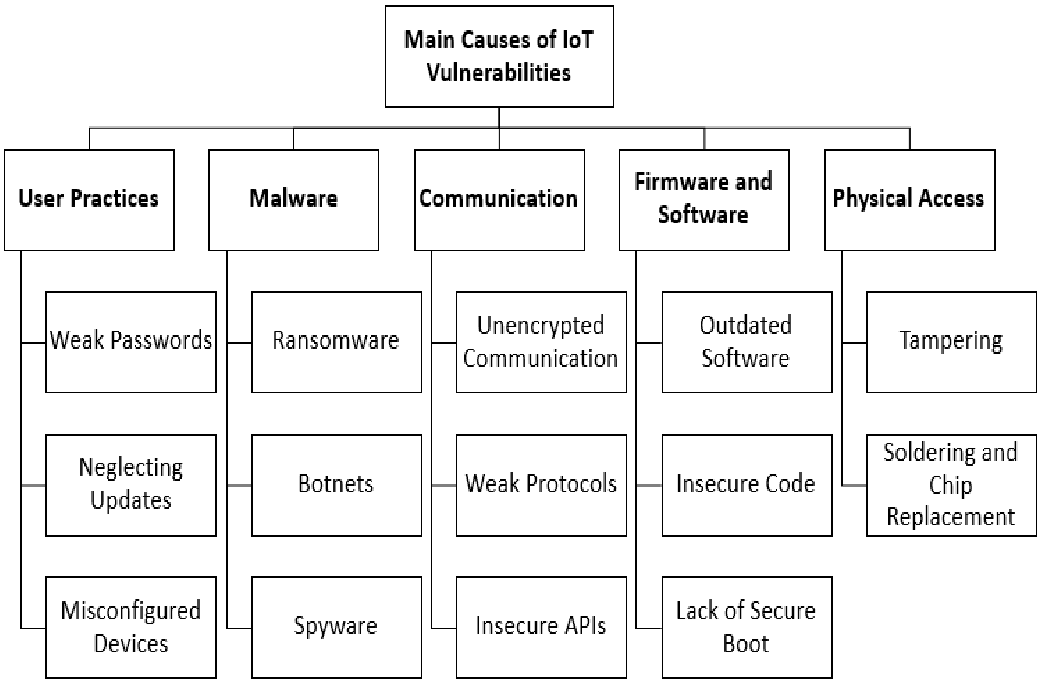

Figure 1. The main causes of IoT vulnerabilities.

Table 2. IoT vulnerabilities' taxonomy.

Vulnerability Category	Specific Subcategory	Examples
Device-Level Vulnerabilities		
Physical Device Vulnerabilities	Unauthorized Physical Access	- Unauthorized Physical Access - Device Theft - Malicious Hardware Implants
	Soldering and Chip Replacement	- Microcontroller Replacement - Memory Chip Replacement - Hardware Implants
	Physical Eavesdropping	- Wiretapping - Sensor Data Interception - Planting Listening Devices
	Environmental Attacks	- Temperature Extremes - Radiation Exposure - Humidity and Pressure
Software Vulnerabilities	Weak or Default Passwords	- Default Credentials - Weak Password Choices
	Outdated Software and Firmware	- Lack of Security Updates - Vulnerable Legacy Software - Devices running outdated, unpatched software
	Insecure Code	- Code Vulnerabilities - Inadequate Error Handling - Exploiting a known software vulnerability

Table 2. *Cont.*

Vulnerability Category	Specific Subcategory	Examples
	Lack of Secure Boot	- Boot Process Vulnerabilities - Unauthorized Bootloader - Disabling secure boot to inject malicious code
	Insecure Web Interfaces	- Weak Authentication - Lack of HTTPS - Insecure Configuration - Unauthorized access to a device's web interface
	Configuration and Management Issues	- Inadequate Security Configuration - Poorly Managed IoT Devices
Communication Vulnerabilities	Unencrypted Communication	- Lack of Encryption - Data transmitted in plain text
	Weak Communication Protocols	- Vulnerable Protocols - Exploiting a vulnerability in a communication protocol
	Insecure APIs	- Inadequately Secured APIs - Unauthorized access to an API
Supply Chain Vulnerabilities	Supply Chain Attacks	- Compromised Manufacturing - Distribution Tampering - Attackers compromise the supply chain
Network-Level Vulnerabilities		
Wireless Network Vulnerabilities	Unauthorized Access	- Wireless Hacking - Unauthorized Wireless Devices - Unauthorized access to a secured wireless network
	Signal Jamming	- Jamming Attacks - Signal Interference - Disrupting wireless signals
Data Transmission Vulnerabilities	Man-in-the-Middle Attacks	- Intercepted Data - Data Manipulation - Unauthorized interception of wireless communications
	Data Interception Attacks	- Sniffing Data Packets - Eavesdropping on Wireless Communications - Intercepting data exchanged between IoT devices
Human Factors		
Human-Caused Vulnerabilities	Weak Password Practices	- Easily Guessable Passwords using "12345" as a password
	Neglecting Software Updates	- Failure to Apply Security Patches - Not updating firmware
	Device Misconfigurations	- Poor Configuration Choices - Misconfiguring device settings

Table 2. *Cont.*

Vulnerability Category	Specific Subcategory	Examples
Social Engineering Attacks	Deceptive Social Engineering	- Phishing Attacks - Deceptive Practices - Tricking users into revealing sensitive information
Malware and Cyber Attacks		
Malware Attacks	Ransomware Attacks	- Data Encryption and Ransom Demands - IoT thermostat locked and demanding payment
	Botnet Exploitation	- IoT Device Recruitment - Distributed Denial of Service - Compromised cameras used for DDoS attacks
	Spyware and Unauthorized Data Collection	- Unauthorized Data Collection - Smart speakers recording conversations without consent
Denial of Service (DoS) Attacks	Network Overload	- Network Overload - Resource Exhaustion - Flooding a network with traffic to overwhelm IoT devices
IoT Botnet Attacks	IoT Botnet Formation and Use	- Botnet Formation - Coordinated Attacks - Using the botnet to mine cryptocurrency
Cryptographic Vulnerabilities		
Cryptanalysis Attacks	Brute Force Attacks	- Repeatedly trying different passwords to crack an encrypted file
	Differential Cryptanalysis	- Analyzing the differences between plaintext-ciphertext pairs to deduce the key
	Known-Plaintext Attacks	- Exploiting knowledge of certain parts of the plaintext and corresponding ciphertext - Cryptanalysis using known plaintext-ciphertext pairs
Web Applications Vulnerabilities		
Web Application Flaws	Web Application Flaws	- Web application vulnerabilities that lead to unauthorized access and data breaches
	Cross-Site Scripting (XSS)	- Injecting malicious scripts into web applications
	SQL Injection Vulnerabilities	- Exploiting SQL injection vulnerabilities in web applications
Privacy Concerns		
Privacy Violations	Data Privacy and Collection	- Unauthorized data collection - Unauthorized access to sensitive information

Table 2. *Cont.*

Vulnerability Category	Specific Subcategory	Examples
	Unauthorized Data Access	- Unauthorized access to user data - Violating user privacy
Regulatory and Compliance Issues		
Legal and Compliance Violations	Non-compliance with Data Protection Laws	- Violating data protection laws - Failing to adhere to GDPR or other regulations

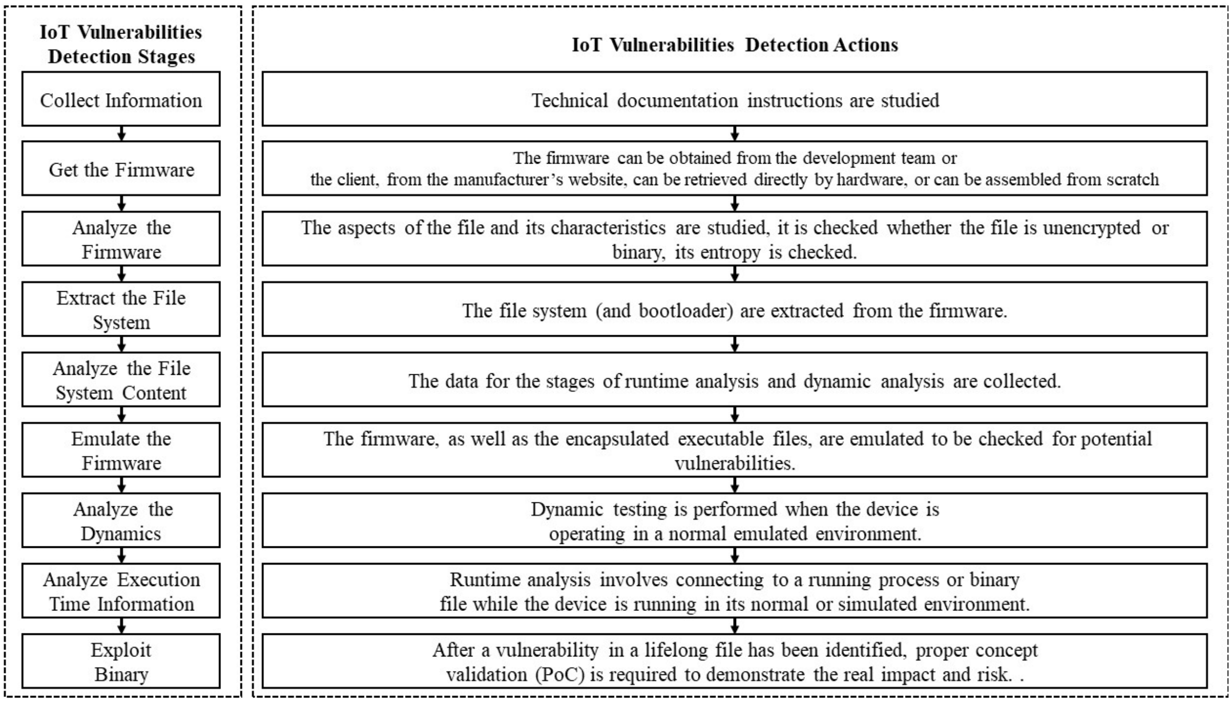
IoT Vulnerabilities Detection Stages
Collect Information
Get the Firmware
Analyze the Firmware
Extract the File System
Analyze the File System Content
Emulate the Firmware
Analyze the Dynamics
Analyze Execution Time Information
Exploit Binary
IoT Vulnerabilities Detection Actions
Technical documentation instructions are studied
The firmware can be obtained from the development team or the client, from the manufacturer's website, can be retrieved directly by hardware, or can be assembled from scratch
The aspects of the file and its characteristics are studied, it is checked whether the file is unencrypted or binary, its entropy is checked.
The file system (and bootloader) are extracted from the firmware.
The data for the stages of runtime analysis and dynamic analysis are collected.
The firmware, as well as the encapsulated executable files, are emulated to be checked for potential vulnerabilities.
Dynamic testing is performed when the device is operating in a normal emulated environment.
Runtime analysis involves connecting to a running process or binary file while the device is running in its normal or simulated environment.
After a vulnerability in a lifelong file has been identified, proper concept validation (PoC) is required to demonstrate the real impact and risk. .

Figure 2. IoT vulnerabilities' detection stages and associated actions.

Table 3. IOT vulnerability detection detailed process.

STEP 1: COLLECT INFORMATION	
• Objectives	• Understand the device's architecture, communication protocols, and potential vulnerabilities through available documentation.
• Description	• Review technical documentation, manuals, and datasheets to gather information about the IoT device.
• Tools	• None (mainly involves reading and documentation analysis).
• Examples	• Studying technical manuals to identify default login credentials for a smart home IoT device. • Analyzing a datasheet to understand the communication protocols used by an industrial IoT sensor.
• Further Details	• Collecting information is the initial step to create a foundation for further analysis and risk assessment.
• Challenges	• Incomplete Documentation: technical documentation may lack crucial details or be outdated, leaving security professionals with an incomplete picture of the device. • Limited Scope: documentation might focus primarily on functional aspects, overlooking security considerations, and vulnerabilities. • Confidential Information: access to confidential or proprietary documentation may require legal agreements or specific permissions, restricting accessibility.

Table 3. *Cont.*

Advantages	• Initial Insight: collecting information is the starting point for understanding the device's intended functionality and security posture. • Non-Intrusive: this step is non-intrusive and does not disrupt the device's operation, making it suitable for initial assessment. • Cost-Efficient: it typically requires no additional tools or resources, making it cost-efficient.
Disadvantages	• Limited to Publicly Available Information: relying solely on publicly available information may result in gaps in understanding the device's architecture and potential vulnerabilities. • Incomplete Picture: the information gathered may not reveal undocumented features, backdoors, or hidden functionalities. • Reliance on Documentation Accuracy: the accuracy and completeness of the documentation can vary, and errors or omissions can lead to misguided assumptions.
STEP 2: OBTAIN THE FIRMWARE	
Objectives	• Obtain a copy of the firmware for subsequent analysis.
Description	• Retrieve the firmware from legitimate sources, such as the development team, manufacturer's website, or directly from the device.
Tools	• Firmware flashing tools, hardware debugging equipment, or firmware download utilities.
Examples	• Downloading the latest firmware update for a smart camera from the manufacturer's official website. • Extracting the firmware from an IoT device's flash memory using a JTAG interface.
Further Details	• Care must be taken to ensure that obtaining the firmware is carried out legally and ethically.
Challenges	• Legal and Ethical Concerns: Acquiring firmware legally and ethically can be challenging, especially if it's not readily available from official sources. Reverse engineering or extraction methods may raise legal and ethical questions. • Physical Access: extracting firmware from some devices may require physical access, which may not be feasible for remote or protected environments.
Advantages	• Comprehensive Analysis: acquiring the firmware provides a complete set of data for in-depth examination, including potential security vulnerabilities. • Updates Analysis: examining firmware updates can reveal security enhancements or vulnerabilities that have been patched over time.
Disadvantages	• Legal and Ethical Complexities: obtaining firmware through reverse engineering or extraction methods may raise legal and ethical concerns, potentially violating terms of service, warranties, or intellectual property rights. • Device Accessibility: accessing and extracting firmware may not be feasible for devices in remote or highly secure locations, limiting assessment possibilities.
STEP 3: ANALYZE THE FIRMWARE	
Objectives	• Gain an initial understanding of the firmware structure and identify whether the file is encrypted or compressed.
Description	• Examine the firmware file to determine its format, check for encryption or compression, and assess its entropy.
Examples	• Identifying a firmware file as a compressed archive (e.g., ZIP) by examining its header. • Calculating the entropy of firmware sections to detect encrypted portions.
Further Details	• Identifying encryption or compression can guide further analysis steps.
Challenges	• Encrypted or Compressed Files: firmware files may be encrypted or compressed, complicating the analysis process. • Encryption Detection: identifying encryption or compression may require specialized knowledge, tools, or techniques.
Advantages	• Security Awareness: the detection of encryption or compression indicates that security measures have been implemented, suggesting the device may be protecting sensitive data. • Guidance for Further Analysis: identifying encryption or compression can guide subsequent analysis steps, such as decryption or decompression.

Table 3. *Cont.*

Disadvantages	• Decrypting or Decompressing: decrypting or decompressing sections of the firmware may require additional effort and expertise, potentially delaying analysis. • False Positives/Negatives: tools used for identifying encryption or compression can produce false positives or negatives, leading to misinterpretation of the firmware.
STEP 4: EXTRACT THE FILE SYSTEM	
Objectives	• Reveal the internal structure of the firmware by extracting the file system and bootloader.
Description	• Use specialized tools to extract file system components like configuration files, executables, and libraries.
Tools	• Binwalk or dedicated firmware extraction tools.
Examples	• Extracting configuration files containing network settings, potentially revealing security vulnerabilities. • Retrieving bootloader components to analyze boot processes and potential weaknesses.
Further Details	• Understanding the file system layout is crucial for identifying vulnerabilities.
Challenges	• Proprietary Formats: extraction may fail if the firmware uses proprietary or custom file system formats, which may not be supported by standard tools. • Bootloader Complexity: some devices have complex bootloaders with security mechanisms that make extracting the file system challenging.
Advantages	• Detailed Analysis: extracting the file system allows for an in-depth examination of configuration files, executables, and libraries, potentially revealing security vulnerabilities. • Vulnerability Identification: access to configuration files can uncover security vulnerabilities or misconfigurations that could be exploited.
Disadvantages	• Security Mechanisms: advanced security mechanisms, such as secure boot, may prevent successful extraction of the file system, requiring additional efforts or expertise. • Complex File Systems: complex file system structures can be challenging to navigate and analyze, potentially extending the analysis time.
STEP 5: ANALYZE THE FILE SYSTEM CONTENT	
Objectives	• Identify potential security weaknesses within the extracted file system, such as hard credentials or vulnerabilities in binaries.
Tools	• Static analysis tools, disassemblers, and vulnerability scanners.
	• Examples: • Scanning binary files for known vulnerabilities using tools like BinGrep or Checksec. • Inspecting configuration files for plaintext passwords or exposed sensitive data.
Further Details	• Static analysis helps identify potential issues without executing the firmware.
	• Challenges: • Time-Consuming Analysis: identifying vulnerabilities within binary files can be time-consuming and may necessitate in-depth expertise in reverse engineering. • Static Analysis Limitations: static analysis may not detect vulnerabilities that manifest only during dynamic runtime, potentially leading to missed weaknesses.
	• Disadvantages: • Missed Runtime Vulnerabilities: static analysis may not detect vulnerabilities that only become evident during runtime, such as those related to user input or system interactions. • False Positives/Negatives: vulnerability scanning tools used during static analysis may produce false positives or negatives, requiring manual verification.
STEP 6: EMULATE THE FIRMWARE	
Objectives	• Create a safe environment to analyze the firmware's behavior and check for potential vulnerabilities.
Description	• Use emulation platforms like QEMU to run the firmware in a controlled environment.
Tools	• Emulation platforms, dynamic analysis tools.

Table 3. *Cont.*

Examples	• Running the firmware in an emulated environment to observe how it interacts with different inputs and external systems. • Analyzing network traffic generated by the emulated firmware to detect potential vulnerabilities.
Further Details	• Emulation allows assessment without affecting the actual device.
Challenges	• Realistic Emulation: emulation environments may not perfectly replicate the real device's behavior, potentially leading to discrepancies in vulnerability identification. • Comprehensive Vulnerability Identification: identifying all potential vulnerabilities during emulation can be challenging, and some may only manifest in specific real-world scenarios.
Advantages	• Non-Intrusive: emulation allows for assessment without affecting the actual device, reducing risks associated with live testing. • Behavior Insights: provides insights into how the firmware behaves in various scenarios, aiding in vulnerability assessment.
Disadvantages	• Limited Realism: emulation environments may not fully simulate all aspects of the device's behavior, potentially leading to false positives or negatives. • Resource Intensive: complex emulation environments may require substantial computational resources, making them less accessible for some assessments.
STEP 7: ANALYZE THE DYNAMICS	
Objectives	• Identify vulnerabilities that may manifest during runtime by observing the device's behavior.
Description	• Conduct dynamic testing in a controlled environment where the device operates as it normally would.
Tools	• Network analysis tools, debuggers, and monitoring solutions.
	• Examples: • Monitoring system logs to detect unauthorized access attempts or unusual behavior during device operation. • Using debugging tools to analyze runtime behavior for buffer overflows or resource exhaustion.
	• Further Details: dynamic testing uncovers issues not evident in static analysis.
Challenges	• Controlled Testing: dynamic testing must be conducted in a controlled environment to prevent potential harm or disruptions to the device or network. • Resource Intensity: identifying and monitoring runtime vulnerabilities can be resource intensive, particularly in large-scale IoT deployments.
Advantages	• Real-World Scenarios: dynamic testing replicates vulnerabilities that may only manifest during real-world runtime, providing a more accurate assessment. • Mimicking Device Behavior: dynamic testing simulates how the device operates under normal conditions, aiding in vulnerability discovery.
Disadvantages	• Potential Disruptions: dynamic testing may introduce risks, such as service disruptions, security breaches, or device malfunctions, requiring careful planning and execution. • Resource Requirements: dynamic testing can be resource intensive, requiring adequate computational and monitoring resources, particularly for extensive assessments.
STEP 8: ANALYZE THE EXECUTION TIME INFORMATION	
Objectives	• Connect to a running process or binary during normal or simulated operation to analyze runtime behavior.
Description	• Attach debuggers or monitoring tools to running processes or binaries.
Tools	• Debuggers, monitoring tools, instrumentation techniques.
Examples	• Attaching a debugger to a running process to inspect memory or analyze execution flow. • Monitoring CPU and memory usage during device operation to detect anomalies.
Further Details	• Runtime analysis provides insights into real-time behavior.

Table 3. *Cont.*

Challenges	• Debugging Accessibility: attaching debuggers or monitoring tools to running processes may not always be straightforward, especially in closed or production environments. • Complex Analysis: identifying and analyzing runtime vulnerabilities can be complex and may necessitate advanced debugging techniques.
Advantages	• Real-Time Insights: provides real-time insights into the device's behavior during operation, allowing for immediate vulnerability detection. • Runtime Vulnerability Detection: enables the detection of runtime vulnerabilities, resource-related issues, or unexpected behaviors.
Disadvantages	• Device Accessibility: requires access to a running device or a controlled emulation environment, which may not be possible in all scenarios. • Potential Overhead: debugging and monitoring may introduce overhead or impact device performance, potentially affecting the accuracy of the assessment.
STEP 9: EXPLOIT BINARY	
Objectives	• Develop a proof of concept (PoC) to demonstrate the real impact and risk of identified vulnerabilities.
Description	• If vulnerabilities are identified, create a PoC exploit to validate their practical significance.
Tools	• Development of environments, exploitation of development frameworks, and debugging tools.
Examples	• Creating a PoC exploit to demonstrate how a specific vulnerability could be exploited to gain unauthorized access. • Developing an exploit to demonstrate the potential impact of a buffer overflow vulnerability in a device's firmware.
Further Details	• A PoC is essential for demonstrating vulnerabilities to stakeholders.
Challenges	• Time-Consuming Process: developing a PoC exploit can be time consuming and may require significant expertise in vulnerability exploitation, programming, and security. • Ethical Considerations: conducting exploitation without causing harm or disruptions is challenging and raises ethical questions regarding responsible disclosure.
Advantages	• Practical Validation: exploiting vulnerabilities validates their practical significance and demonstrates their potential impact on the device or network. • Stakeholder Communication: A PoC exploit effectively communicates potential risks to stakeholders, facilitating informed decision-making.
Disadvantages	• Ethical and Legal Constraints: exploiting vulnerabilities may not be possible or ethical in certain situations and can have legal implications, making responsible disclosure crucial. • Resource Intensive: developing exploits can be a complex and resource-intensive process, often requiring substantial time, expertise, and testing.

Successfully navigating these challenges while harnessing the advantages of each step is essential for conducting a thorough and effective IoT vulnerabilities' detection process. The selection of methods and tools should align with the specific context of the device, its security measures, and the overarching goals of the assessment.

Calculating the cost of each IoT vulnerabilities' detection stage involves considering various factors, including the complexity of the task, the expertise required, the time involved, and any associated tools or resources. These costs can vary widely depending on the specific context and organization. Table 4 summarizes the cost range and reasoning for the IoT vulnerabilities' detection stages.

Table 4. Cost associated with IOT vulnerabilities' detection stages.

Stage	Cost	Explanation
1-Collect Information	Low	Studying technical documentation typically involves reading existing materials and does not require extensive resources.
2-Get the Firmware	Variable	The cost can vary based on how the firmware is obtained. Obtaining it from the development team or client may be straightforward, while assembling it from scratch or reverse engineering it from hardware can be more resource intensive.
3-Analyze the Firmware	Medium	Analyzing firmware involves examining its characteristics, checking for encryption, and assessing entropy. This requires expertise and potentially specialized tools.
4-Extract the File System	Medium	Extracting the file system from firmware may require reverse engineering skills and tools. The complexity can depend on the firmware's structure and protection mechanisms.
5-Analyze the File System Content	Low to Medium	Collecting data from the file system may involve examining configuration files, scripts, and binaries. The cost can vary depending on the complexity of the file system.
6-Emulate the Firmware	Medium to High	Emulating firmware and executable files can be resource intensive. It requires specialized emulators and knowledge of the target environment.
7-Analyze the Dynamics	High	Dynamic testing involves running the firmware in an emulated environment and actively monitoring its behavior. It requires expertise and time.
8-Analyze Execution Time Information	High	Runtime analysis involves connecting to a running process or binary during normal or simulated operation. It requires expertise and tools for real-time monitoring.
9-Exploit Binary	High	Developing a proof of concept (PoC) for a vulnerability in a binary file is a high-cost task. It involves in-depth understanding of the vulnerability and its potential impact.

The actual costs can vary significantly depending on factors like the complexity of the firmware, the skills of the personnel involved, and the tools and resources available. In the following, we present other factors that can affect the cost of each stage in IoT vulnerabilities' detection in more detail.

1. Complexity of the Firmware:
    - Complexity plays a crucial role in determining costs. More complex firmware, with numerous components and intricate code structures, will generally require more time and effort to analyze. Complex firmware may also use advanced security measures, increasing the expertise and tools needed for analysis.
2. Skills and Expertise of Personnel:
    - The skill level and expertise of the individuals involved significantly impact costs: Highly skilled and experienced security analysts may be more efficient and effective in identifying vulnerabilities, potentially reducing the cost. Junior or less experienced analysts may require more time for analysis and may not catch all vulnerabilities, increasing the cost. Specialized expertise in areas like reverse engineering, cryptography, and hardware security can also affect costs.
3. Availability of Specialized Tools:
    - The availability of specialized tools can affect both the cost and efficiency of the analysis as access to advanced debugging, reverse engineering, and emulation tools can streamline the analysis process, potentially reducing costs. The cost of acquiring and maintaining these tools should be considered.
4. Time Required for Analysis:
    - Time is a significant cost factor in IoT vulnerabilities' detection: Extensive analysis and testing require more time, which can translate into higher labor costs.

Rapid detection and resolution of vulnerabilities can reduce the overall cost by preventing potential security breaches.

5. Firmware Source and Quality:
    - The source of the firmware and its quality can affect costs: Obtaining firmware directly from the manufacturer or development team can be more reliable and reduce costs compared to reverse engineering. Poorly documented or obfuscated firmware may require more effort and time for analysis, increasing costs.
6. Hardware Complexity and Protection Mechanisms:
    - If the IoT device's hardware has strong security protections, such as hardware encryption or secure boot processes, it can make firmware extraction and analysis more challenging and costly.
7. Volume and Complexity of File System Content:
    - The size and complexity of the file system content influence costs: analyzing a large and intricate file system with numerous configuration files and executables may require more time and expertise, increasing costs.
8. Emulation and Runtime Analysis Environment:
    - Creating a realistic emulation environment can be costly: Specialized hardware, software, and configurations may be needed for accurate emulation. The cost of setting up and maintaining this environment should be considered.
9. Development of Exploits and PoCs:
    - Developing proof of concept (PoC) exploits for identified vulnerabilities is a high-cost activity: Skilled programmers and security experts are required. Extensive testing and validation are necessary to ensure the PoC accurately demonstrates the vulnerability's real impact.

In summary, the cost of IoT vulnerabilities' detection is influenced by a combination of technical, personnel-, and resource-related factors. These factors can interact in complex ways, making it essential to consider them carefully when planning and budgeting for security analysis in IoT systems. The actual costs will depend on the unique characteristics of the IoT device, the security analysis process, and the organization's capabilities and resources. In case the vulnerabilities are not detected, they can be exploited, which leads to attacks on the IoT. In [27], the authors classified the different types of IoT attacks. Table 5 presents a summary of IoT attacks, including their description, subcategories, examples, advantages from the attackers' perspective, and cost factors.

3.2. IoT Symmetry in Attacks and Defense

In the context of cybersecurity, there is often a symmetrical aspect to the defense and attacks. For every security measure or defense mechanism put in place to protect IoT devices, attackers may seek out symmetrical weaknesses or vulnerabilities to exploit. This ongoing symmetry battle between defenders and attackers is reminiscent of the concept of symmetry in adversarial relationships. Table 6 provides an overview of the dynamic interplay between cyberattacks and defense strategies in the realm of cybersecurity, highlighting key aspects of symmetry and asymmetry.

Table 5. IoT attacks' classification, description, examples, advantages from attackers' perspective, and cost factors.

Description	Subcategories	Examples	Advantages from the Attackers' Perspective	Cost Factors
Physical Attacks				
Manipulate or damage physical components.	Tampering: unauthorized access, modification, or theft of physical devices, soldering. Chip replacement: replacing or modifying hardware components. Environmental attacks: exposing devices to extreme conditions.	- Tampering with a surveillance camera to disable it, replacing a device's microcontroller with a malicious one, exposing a sensor to extreme heat to disrupt its operation.	- Gain direct access to devices, facilitate data theft or implant malicious hardware, compromise the device's functionality.	1. Skill and Knowledge: individuals conducting physical attacks need to possess the skills and knowledge required to tamper with or replace hardware components effectively. 2. Equipment Cost: the tools and equipment needed for physical attacks, such as soldering tools or environmental manipulation equipment, can be expensive. 3. Physical Access: gaining physical access to IoT devices or manufacturing facilities often comes with costs related to breaking and entering or social engineering techniques.
Side Channel Attacks				
Exploit unintended information leakage for data extraction.	- Power analysis: analyzing power consumption patterns. Timing analysis: exploiting timing differences. - Electromagnetic analysis: utilizing electromagnetic emissions.	- Monitoring power fluctuations to deduce a device's cryptographic key, observing the time it takes a system to respond to specific inputs, analyzing electromagnetic radiation to extract sensitive information.	- Covert and non-intrusive methods to reveal secrets, ability to extract sensitive information without directly attacking cryptographic algorithms.	1. Specialized Equipment: acquiring specialized equipment for monitoring power consumption, timing analysis, or electromagnetic emissions can be costly. 2. Expertise and Training: conducting effective side channel attacks requires expertise in interpreting the collected data, which may involve training expenses. 3. Access to Target Devices: access to the IoT devices under attack and the ability to synchronize data collection with device operations may require additional resources.
Cryptanalysis Attacks				
Break cryptographic algorithms to reveal encrypted data.	- Brute force attacks: trying all possible keys. Differential cryptanalysis: exploiting the differences between plaintext–ciphertext pairs. Known-plaintext attacks: using known plaintext–ciphertext pairs.	- Repeatedly trying different passwords to crack an encrypted file, analyzing the differences between plaintext–ciphertext pairs to deduce the key, exploiting knowledge of certain parts of the plaintext and corresponding ciphertext.	- Successful decryption of encrypted data, potential access to sensitive information, passwords, or keys.	1. Computational Resources: the cost of computational power needed for brute-force attacks or differential cryptanalysis can be significant, especially for complex encryption algorithms. 2. Data Collection: for known-plaintext attacks, obtaining plaintext-ciphertext pairs can be expensive in terms of data acquisition. 3. Knowledge of Encryption Algorithms: understanding the encryption algorithms used and their potential weaknesses can require training and research expenses.

Table 5. *Cont.*

Description	Subcategories	Examples	Advantages from the Attackers' Perspective	Cost Factors
Software Attacks				
Exploit vulnerabilities in software components for unauthorized access.	- Malware injection: injecting malicious software into a system. Exploiting software bugs: taking advantage of software vulnerabilities. Buffer overflow attacks: overwriting data outside allocated memory.	- Distributing malware through email attachments, exploiting a known software vulnerability to gain unauthorized access, overwriting memory to execute arbitrary code.	- Potential for remote exploitation, unauthorized access, data theft, or system control.	1. Development of Malware: creating malware for software attacks may involve software development costs, including code development and testing. 2. Zero-Day Exploitation: acquiring or developing zero-day vulnerabilities can be costly, as they are often highly valuable. 3. Distribution: distributing malware to IoT devices may involve the cost of creating phishing emails or other infection vectors.
Wireless Attacks				
Target communication channels and protocols in wireless networks.	- Man-in-the-middle attacks: intercepting and manipulating wireless traffic. Eavesdropping: listening to wireless communications. Signal jamming: disrupting wireless signals.	- Intercepting and altering data between two communicating devices, secretly listening to conversations on unencrypted wireless networks, transmitting interference signals to disrupt wireless communication.	- Intercept data, disrupt network operations, impersonate devices, unauthorized access to data or network resources.	1. Wireless Equipment: acquiring the necessary wireless equipment, such as signal jammers or packet capturing devices, can be a significant cost. 2. Skill and Knowledge: conducting wireless network attacks effectively demands knowledge of wireless protocols, network security, and the skill to exploit vulnerabilities. 3. Access to Target Networks: gaining access to the target wireless networks may involve costs related to physical access, credentials, or insider collaboration.
Supply Chain Attacks				
Compromise the supply chain to introduce vulnerabilities or malicious components.	- Tampering with hardware during manufacturing, inserting malware or backdoors during production or distribution.	- Modifying a device's firmware at the production facility, adding a malicious component to a device before it reaches the end user.	- Widespread compromise of IoT devices, potential persistence of attacks over time.	1. Development Cost: this involves the creation of malicious components or backdoors and their integration into the supply chain. This may require skilled individuals or teams to develop and test the malicious elements. 2. Production Facility Access: gaining access to manufacturing or distribution facilities can be challenging and costly, either through insider collaboration or physical intrusion. 3. Secrecy Cost: ensuring the inserted vulnerabilities or malicious components remain undetected may involve additional expenses in maintaining secrecy throughout the supply chain.

Table 5. *Cont.*

Description	Subcategories	Examples	Advantages from the Attackers' Perspective	Cost Factors
Environmental Attacks				
Target IoT devices by exposing them to extreme conditions.	- Exposing devices to extreme temperature, radiation, or other environmental factors.	- Subjecting IoT sensors to extreme cold to disrupt their operation, exposing a device to strong radiation to induce faults.	- Device malfunctions or data corruption, potential exploitation of vulnerabilities.	1. Equipment Cost: purchasing or developing equipment capable of exposing devices to extreme conditions, such as temperature or radiation, can be a significant cost factor. 2. Testing and Experimentation: conducting experiments and tests to determine the right environmental conditions for disruption may incur costs. 3. Knowledge and Expertise: understanding how environmental factors affect IoT devices and their vulnerabilities may require specialized knowledge and expertise.
Denial of Service (DoS) Attacks				
Overwhelm IoT devices or networks to disrupt their availability.	- Network-based DoS attacks, resource exhaustion attacks, protocol exploitation attacks.	- Flooding a network with traffic to overwhelm IoT devices, exploiting vulnerabilities to deplete device resources, sending malformed packets to disrupt IoT protocols.	- Disruption of IoT device availability, hindrance of critical services.	1. Resources for Large-Scale Attacks: launching large-scale DoS attacks often requires access to botnets or networks of compromised devices, which can come at a significant cost. 2. Infrastructure: setting up the infrastructure for a distributed DoS attack, such as network proxies and attack tools, can be costly. 3. Skill and Knowledge: successfully conducting DoS attacks demands skill in identifying and exploiting vulnerabilities, adding to the cost.
IoT Botnet Attacks				
Hijack IoT devices to form botnets for various malicious purposes.	- Botnet recruitment, distributed denial of service (DDoS) attacks, cryptojacking, spamming.	- Infecting vulnerable IoT devices to build a botnet, using the botnet to launch DDoS attacks, utilizing the botnet to mine cryptocurrency.	- Massive computing power for malicious activities, financial gains, anonymity.	1. Botnet Recruitment: compromising a large number of IoT devices to build a botnet typically requires resources for identifying and exploiting vulnerabilities, which can incur costs. 2. Botnet Maintenance: maintaining the botnet's infrastructure, ensuring its availability, and avoiding detection may demand ongoing expenditures. 3. Skill and Resources: skilled individuals or groups are necessary to carry out the recruitment, control, and use of botnets, which can be a significant cost factor.

Table 5. *Cont.*

Description	Subcategories	Examples	Advantages from the Attackers' Perspective	Cost Factors
Data Interception Attacks				
Intercept data transmitted between IoT devices.	- Sniffing data packets, eavesdropping on wireless communications.	- Capturing unencrypted data packets from a sensor network, eavesdropping on Wi-Fi communications between smart home devices.	- Unauthorized access to sensitive data, information theft, potential exposure of sensitive user information.	1. Equipment Cost: acquiring the necessary equipment to intercept data packets or eavesdrop on wireless communications can be expensive. 2. Knowledge of Protocols: understanding and exploiting the communication protocols and encryption methods used for data transmission may require specialized knowledge. 3. Access to Transmission Channels: gaining access to the channels where data are transmitted can be costly, potentially involving physical or network-related expenses.
GPS Spoofing Attacks				
Falsify GPS signals to mislead IoT devices' location tracking.	- GPS signal manipulation, location deception.	- Sending fake GPS signals to mislead autonomous vehicles, misrepresenting the location of a tracking device.	- Misleading device or vehicle location, potential for illegal activities, evasion of tracking and monitoring.	1. Equipment Cost: generating fake GPS signals or manipulating existing ones can be costly, as it requires specialized hardware. 2. Knowledge of GPS Protocols: understanding the intricacies of GPS signal protocols and the devices that rely on them is crucial and may involve additional expenses. 3. Testing and Experimentation: conducting tests to fine-tune the spoofing equipment and ensure the desired effect can be achieved may require resources.

Table 6. Symmetry in cybersecurity: attacks versus defense.

Aspects	Attacks	Defenses	Examples
Patch and Exploit Symmetry	Attackers target vulnerabilities before patches are applied.	Manufacturers release patches to fix vulnerabilities.	Attackers exploit devices without patches.
Firewall and Evasion Techniques	Attackers bypass firewalls using evasion methods.	Network administrators deploy firewalls for protection.	Attackers use tactics to evade firewalls.
Encryption and Decryption	Attackers intercept encrypted data and attempt decryption.	Encryption secures data, making these unreadable without a key.	Attackers try to decrypt intercepted encrypted data.
Security Research and Zero-Day Exploits	Attackers exploit IoT vulnerabilities before patches are released.	Researchers identify vulnerabilities for patching.	Attackers target IoT devices with zero-day exploits.
User Authentication and Credential Attacks	Attackers trick users into revealing credentials.	Organizations implement multi-factor authentication.	Attackers compromise user credentials.
Antivirus and Malware Evasion	Attackers modify malware to evade antivirus.	Antivirus detects and quarantines malicious software.	Attackers modify malware to avoid detection.

Table 6. *Cont.*

Aspects	Attacks	Defenses	Examples
Network Segmentation and Lateral Movement	Attackers move laterally through misconfigured networks.	Organizations segment networks to limit movement.	Attackers exploit weak network segmentation.
Vulnerability Scanning and Exploitation	Attackers identify and exploit vulnerabilities before patching.	Security teams assess and patch known vulnerabilities.	Attackers exploit vulnerabilities before they are patched.
Incident Response and Evasion	Attackers cover tracks to avoid detection during incidents.	Organizations have incident response plans.	Attackers evade detection during security incidents.
AI and Machine Learning in Security	Attackers confuse AI-based security systems.	Security professionals use AI for threat detection.	Attackers manipulate data to deceive AI systems.
Web Application Security	Attackers exploit web app vulnerabilities.	Developers secure web apps against common flaws.	Attackers probe for web app vulnerabilities.
Social Engineering and Security Awareness	Attackers trick employees into revealing sensitive info.	Organizations educate employees on security awareness.	Attackers manipulate individuals for information.
Advanced Persistent Threats (APTs) and Persistence	APTs use advanced techniques to persist in networks.	Organizations detect and mitigate APTs.	A nation-state-sponsored attacker maintains control after removal.
Supply Chain Attacks	Attackers insert backdoors into software updates.	Organizations secure software and hardware supply chains.	Attackers compromise supply chains.
Insider Threats and Trust Exploitation	A disgruntled employee sabotages systems or leaks data.	Organizations monitor employees and control access.	Insiders exploit trust for malicious actions.
Cryptocurrency and Ransomware	A ransomware group demands cryptocurrency for data decryption.	Organizations protect against ransomware attacks.	Ransomware encrypts data and demands cryptocurrency.

4. The SILEX Malware

In the IoT networks, existing malware threats pose significant risks to the security and functionality of connected devices. For example, the Mirai botnet capitalizes on weak credentials to transform IoT devices into an army of bots for devastating DDoS attacks. Another example, the Stuxnet, although initially designed for industrial systems, underscores the potential dangers to IoT devices in critical infrastructure. In this article, our focus is to offer the interested reader a concise discussion of the SILEX Malware.

The use of weak passwords, default passwords, or passwords disclosed to the network has proven to be the most serious vulnerability thus far. Despite the obvious requirement for a secure password, some users continue to utilize default passwords. SILEX, an IoT malware, took advantage of this in June 2019, converting over 2000 IoT devices into "bricks" in one hour. It was built by a 14-year-old hacker, who instilled fear and began targeting and destroying IoT devices with weak passwords. SILEX is an internet-based virus designed to render IoT devices useless. It concentrated on Unix-based devices with pre-configured usernames and passwords. When SILEX discovers a susceptible device, it will overwrite all system storage with random data, erase the firewall rules and network settings, and then restart the system, rendering the device useless to users. This malware could only live for a day or two but was extremely detrimental to IoT companies in such a short time. The SILEX malware has attacked thousands of IoT devices.

The SILEX malware targets and compromises IoT devices in a systematic manner by using default credentials to log into IoT devices. But to retrieve the credentials, the malware first corrupts the IoT device storage, removing network configurations, then drops the firewall rules, and finally deletes the network configuration. As a result, the affected IoT devices become completely inoperable. SILEX achieves this by exploiting a combination of known vulnerabilities and easily guessable passwords, allowing it to infiltrate devices swiftly and spread its infection across vulnerable networks. Silex malware is so powerful that when an IoT device is affected, it is impossible to recover. However, by manually reinstalling the device firmware, one can recover the infected IoT device.

The attack steps for the SILEX malware on IoT devices generally involve the following:

1. Initial Compromise:

   Scanning for Vulnerable Devices: the malware scans the internet for IoT devices with known vulnerabilities, such as default passwords or unpatched software.

   Unauthorized Access: once a vulnerable device is identified, the malware attempts to gain unauthorized access using default credentials or known exploits.

2. Device Infiltration:

   Password Brute-Force: SILEX may attempt to guess the device's password through a brute-force attack.

   Exploiting Vulnerabilities: it may exploit known security vulnerabilities in the device's software or firmware to gain access.

3. Malware Deployment:

   Downloading and Executing Payload: once inside the device, SILEX downloads and executes its malicious payload, which includes components to target the device's critical functions.

4. Destructive Actions:

   Data Wiping: SILEX wipes the data on the device, including configurations and stored information.

   Disabling Network Interfaces: it disables network interfaces to isolate the device from the internet, preventing remote access or updates.

   Corrupting Firmware: the malware can corrupt the device's firmware, rendering it inoperable.

   Terminating Processes: it terminates critical processes, disrupting the device's normal operation.

5. Overwriting Storage:

   Overwriting Storage Devices: SILEX may write random data to the device's storage, making data recovery impossible.

6. Device Reboot:

   Forcing a Reboot: to ensure the changes take effect, the malware often forces the device to reboot.

7. Device Unusable:

   Brick the Device: after these actions, the IoT device becomes effectively "bricked" or unusable. It can no longer perform its intended functions.

8. Exfiltration Prevention:

   Terminating Data Exfiltration: SILEX may also prevent the device from exfiltrating sensitive data to external servers, limiting the attacker's access to the device's information.

9. Covering Tracks:

   Deleting Logs: to minimize detection, SILEX may delete logs and traces of its activities.

The attack steps taken by the SILEX malware are illustrated in Figure 3.

The SILEX MALWARE case study can be seen as an example of an asymmetric cyber-attack targeting IoT devices. SILEX MALWARE, which emerged in 2019, was designed to target and compromise IoT devices, rendering them inoperable by deleting their firmware. This case study highlights the vulnerability of IoT devices to relatively simple yet destructive attacks. The lack of security measures or the asymmetry in the level of security in these devices made them easy targets for an attacker who exploited this asymmetry to cause significant damage. Table 7 illustrates the asymmetry between IoT devices and the SILEX Malware across multiple critical dimensions of IoT security.

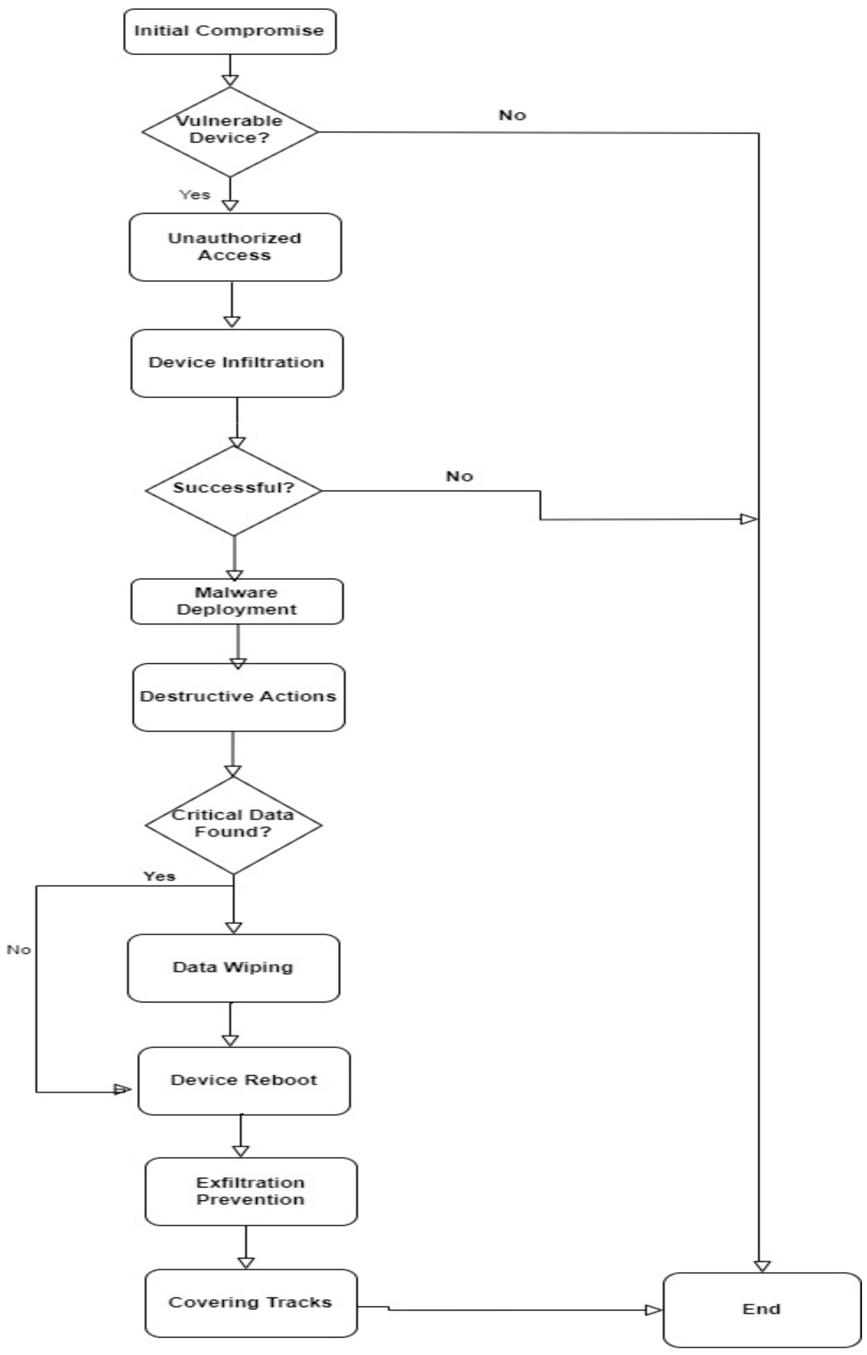

Figure 3. SILEX malware IoT attack steps.

Table 7. Asymmetry between IoT devices and the SILEX malware across multiple critical dimensions of IoT security.

IoT Security Aspect	IoT Devices	SILEX Malware	Examples
Resources	Limited computing resources, cost-effective design	More powerful attack mechanism, resource intensive	SILEX malware erases firmware on resource-constrained IoT devices
Security Expertise	Limited security practices, vulnerabilities	High level of malware development expertise	SILEX malware exploits known IoT vulnerabilities

Table 7. *Cont.*

IoT Security Aspect	IoT Devices	SILEX Malware	Examples
Firmware and Patching Asymmetry	Limited firmware updates, vulnerability exposure	Exploits unpatched IoT devices, known flaws	SILEX malware targets unpatched IoT devices
Economic Asymmetry	Cost constraints, compromises in security features	Destructive intent, economic harm to victims	SILEX malware causes economic harm to organizations and individuals
User Interface Asymmetry	Limited user-friendly interfaces, security challenges	Exploits user interface limitations	SILEX malware compromises IoT devices with basic interfaces
Attack Persistence and Duration	Extended device operation, longer attack window	Designed for persistent attacks, long-lasting damage	SILEX malware causes prolonged disruption
Resource Utilization and IoT Device Functionality	Designed for specific functions, resource constraints	Consumes significant device resources, disrupts functionality	SILEX malware disrupts IoT device functionality
Device Replacement Costs	High financial burden for replacing compromised devices	Attackers face no equivalent costs	Victims incur expenses to replace or repair devices
Detection and Attribution Challenges	Limited monitoring capabilities, difficulty in attribution	Difficult to attribute attacks, identity concealment	Identifying SILEX malware source requires sophisticated investigations
Public Awareness and Response	Limited awareness, delayed responses	Raises awareness and prompts action	SILEX malware prompts improved IoT security practices
Interconnected IoT Ecosystems	IoT devices interconnected, attack dependencies	Targets single device, disrupts interconnected networks	An attack on one vulnerable device affects entire ecosystems
Dependency on Vendor Support	Reliance on vendor for support and updates	Exploits devices with unreliable vendor support	Discontinued vendor support leaves devices vulnerable
Geographic Distribution	Global deployment, varying security levels	Targets regions with weaker security practices	Attackers target regions with relaxed IoT security standards
Legal Recourse and Liability	Legal challenges for victims, limited accountability	Operates anonymously or from jurisdictions with weak prosecution	Victims struggle to hold attackers accountable
Regulatory Response Time	Slow regulatory response, enforcement lag	Exploits regulatory gaps, launches attacks	Regulations catch up with threats like SILEX malware
Attack Variability and Evolution	Diverse IoT architectures, challenging defense	Adapts malware to target different device types	SILEX malware evolves to exploit various IoT devices

The SILEX malware attack on IoT devices highlights several important lessons and takeaways for both individuals and organizations:

1. Security Awareness:

It emphasizes the critical need for security awareness in the IoT landscape. Many IoT devices are deployed with default passwords or unpatched vulnerabilities, making them easy targets. Users and manufacturers should be proactive in addressing these issues.

2. Regular Updates and Patching:

One of the key lessons is the importance of regular software and firmware updates. Devices should be promptly updated to mitigate known vulnerabilities and protect against evolving threats.

3. Strong Authentication:

The attack underscores the necessity for strong authentication mechanisms, such as complex passwords and two-factor authentication (2FA), to prevent unauthorized access.

4. Supply Chain Security:

Securing the supply chain is crucial. Manufacturers should implement measures to verify the integrity of components and software to avoid compromise before the devices reach consumers.

5. Network Monitoring and Intrusion Detection:

   Implementing intrusion detection systems (IDPS) and network monitoring tools can help identify unusual activities early on and respond to potential threats promptly.

6. Physical Security:

   Adequate physical security measures are essential to prevent unauthorized physical access to IoT devices. This includes securing devices in locked cabinets or rooms.

7. Education and Training:

   Users and employees should receive training on IoT security best practices and be aware of social engineering threats to avoid falling victim to attacks.

8. Data Backup and Recovery:

   Regular data backups and recovery plans can help minimize data loss in the event of a destructive attack like SILEX. Having offline backups can be particularly helpful.

9. Incident Response Plans:

   Organizations should have well-defined incident response plans in place to efficiently respond to and recover from IoT security incidents.

10. Third-Party Security Audits:

    Regular security audits and vulnerability assessments, performed by third-party experts, can help identify weaknesses that might be overlooked internally.

11. Regulatory Compliance:

    Compliance with data protection laws and regulations (e.g., GDPR) is vital. Non-compliance can result in legal repercussions and penalties.

12. Sustainable IoT Ecosystem:

    Manufacturers should prioritize device longevity and support, ensuring that security updates and patches are available for a reasonable time after purchase.

13. Network Segmentation:

    The segmentation of IoT devices from critical network infrastructure can help contain the impact of an attack, isolating affected devices from more critical systems.

These lessons emphasize the importance of proactive measures, ongoing vigilance, and a multi-faceted approach to IoT security. As the IoT landscape continues to expand, it is crucial to apply these lessons to protect the increasing number of connected devices from evolving threats.

To counter these threats, security solutions encompass network segmentation to isolate IoT devices, robust authentication mechanisms, intrusion detection and prevention systems for early threat detection, regular firmware updates to patch vulnerabilities, and a security-by-design approach to embed robust safeguards into IoT devices from their inception. These multifaceted security measures are vital in safeguarding IoT ecosystems against a constantly evolving landscape of malware threats. In recent years, artificial intelligence-based approaches have yielded significant results in security detection, including IoT security [28]. Improvements required to make IoT applications more secure and resilient without compromising their dependability were mentioned in [29]. In Table 8, we summarize the main enhancements to secure IoT applications.

Table 8. Security recommendations to improve the IoT security.

IoT Security Recommendations	Rationale
1. Implement Strong Authentication and Access Control	Strong authentication ensures that only authorized users and devices can access IoT systems, reducing the risk of unauthorized access and data breaches. Access control limits permissions to minimize potential attack surfaces.
2. Regularly Update and Patch IoT Devices	Keeping devices updated with the latest security patches and firmware updates is crucial to address known vulnerabilities and protect against emerging threats. Neglecting updates leaves devices susceptible to exploitation.
3. Employ Secure Communication Protocols	Secure communication protocols, such as TLS (Transport Layer Security), encrypt data transmission, preventing eavesdropping and data interception. This safeguards data privacy and integrity.
4. Conduct Security Audits and Vulnerability Scans	Regular security audits and vulnerability scans help identify weaknesses and potential threats in IoT systems. By proactively addressing vulnerabilities, organizations can strengthen their security posture.
5. Secure the Supply Chain	Ensuring the security of the supply chain minimizes the risk of tampered or compromised devices. Supply chain attacks are a growing concern, making it vital to verify the integrity of components and software.
6. Educate Users and Employees on IoT Security	User awareness and training are essential to mitigate human-induced vulnerabilities. Educating users and employees about safe practices and social engineering threats is an effective defense.
7. Employ Intrusion Detection and Prevention Systems	Intrusion detection and prevention systems (IDPS) monitor network traffic for suspicious activities and can automatically respond to threats. This provides real-time protection against attacks.
8. Maintain Strong Physical Security	Physical security measures prevent unauthorized physical access to IoT devices. Locks, alarms, and surveillance can deter tampering, theft, and other physical attacks.

5. Conclusions and Future Work

Security in IoT is significantly important due to the increasing number of connected devices and the potential risks associated with these devices. IoT security has an inherent asymmetry, with resource-constrained devices often lacking the robust defenses required to prevent sophisticated attackers. This asymmetry creates a power dynamic in which cyber-adversaries, armed with greater resources and time, can exploit vulnerabilities in IoT ecosystems.

In this paper, we presented the state-of-the-art research efforts to secure IoT devices, and we listed and illustrated the top 10 IoT vulnerabilities as seen in the OWASP project. IoT security is an ongoing process, and it requires a combination of technical measures, policy enforcement, and user awareness to effectively mitigate the evolving threats in the IoT landscape. Moreover, we compared the symmetrical aspects of cybersecurity defense and offense, emphasizing the competition between defenders and attackers. This ongoing battle mirrors the concept of symmetry in adversarial relationships, highlighting the need for constant vigilance and adaptation in IoT security.

The case study of SILEX malware has illustrated the real-world consequences of this asymmetry, showcasing how a relatively simple but devastating attack can exploit the vulnerabilities in IoT devices lacking adequate security measures. SILEX malware serves as a reminder of the urgency to secure IoT devices, update firmware, and implement robust security practices to mitigate the risks associated with such asymmetrical attacks.

In essence, our exploration of symmetry and asymmetry within the context of IoT vulnerabilities and attacks underscores the imperative for a proactive and balanced approach to IoT security. Addressing the inherent asymmetry in IoT ecosystems is paramount to safeguarding these devices from increasingly sophisticated and destructive cyber-threats. As we move forward in the digital age, understanding and acting upon these insights will be pivotal in strengthening the grounds of IoT security and protecting the integrity of the interconnected world.

In the future, we plan to propose a new lightweight mitigation technique for the SILEX malware.

Author Contributions: Conceptualization, B.I.M.; methodology, B.I.M.; formal analysis, B.I.M., M.S.E. and A.D.J.; investigation, M.S.E.; writing—original draft preparation, B.I.M.; visualization, M.S.E. and M.A.A.; supervision, M.S.E., A.D.J. and M.A.A.; project administration, M.A.A.; funding acquisition, A.D.J. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Not applicable.

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

References

1. Harbi, Y.; Aliouat, Z.; Refoufi, A.; Harous, S. Recent security trends in internet of things: A comprehensive survey. *IEEE Access* **2021**, *9*, 113292–113314. [CrossRef]
2. Sikder, A.K.; Petracca, G.; Aksu, H.; Jaeger, T.; Uluagac, A.S. A survey on sensor-based threats to internet-of-things (iot) devices and applications. *arXiv* **2018**, arXiv:1802.02041.
3. Mian, A.N.; Shah, S.W.H.; Manzoor, S.; Said, A.; Heimerl, K.; Crowcroft, J. A value-added IoT service for cellular networks using federated learning. *Comput. Netw.* **2022**, *213*, 109094. [CrossRef]
4. Sarker, I.H.; Khan, A.I.; Abushark, Y.B.; Alsolami, F. Internet of things (IoT) security intelligence: A comprehensive overview, machine learning solutions, and research directions. *Mob. Netw. Appl.* **2022**, *14*, 1–7. [CrossRef]
5. Verma, A.; Shri, C. Cyber Security: A Review of Cyber Crimes, Security Challenges and Measures to Control. *Vision* **2022**, *17*, 09722629221074760. [CrossRef]
6. Almaraz-Rivera, J.G.; Perez-Diaz, J.A.; Cantoral-Ceballos, J.A. Transport and application layer DDoS attacks detection to IoT devices by using machine learning and deep learning models. *Sensors* **2022**, *22*, 3367. [CrossRef] [PubMed]
7. En, S.X.; Ling, L.S.; Hao, F.C. Honeypots for Internet of Things Research: An Effective Mitigation Tool. *Preprints* **2021**, 2021090461. [CrossRef]
8. IoT Under Fire: Kaspersky Detects More than 100 Million Attacks on Smart Devices in H1 2019. Available online: https://www.kaspersky.com/about/press-releases/2019_iot-under-fire-kaspersky-detects-more-than-100-million-attacks-on-smart-devices-in-h1-2019 (accessed on 24 September 2023).
9. Abdullahi, M.; Baashar, Y.; Alhussian, H.; Alwadain, A.; Aziz, N.; Capretz, L.F.; Abdulkadir, S.J. Detecting Cybersecurity Attacks in the Internet of Things Using Artificial Intelligence Methods: A Systematic Literature Review. *Electronics* **2022**, *11*, 198. [CrossRef]
10. Khraisat, A.; Alazab, A. A critical review of intrusion detection systems in the internet of things: Techniques, deployment strategy, validation strategy, attacks, public datasets, and challenges. *Cybersecurity* **2021**, *4*, 1–27. [CrossRef]
11. Tsiknas, K.; Taketzis, D.; Demertzis, K.; Skianis, C. Cyber threats to industrial IoT: A survey on attacks and countermeasures. *IoT* **2021**, *2*, 163–186. [CrossRef]
12. Lee, Y.; Lee, W.; Shin, G.; Kim, K. Assessing the impact of dos attacks on iot gateway. In *Advanced Multimedia and Ubiquitous Engineering: MUE/FutureTech*; Springer: Singapore, 2017; pp. 252–257.
13. Anirudh, M.; Thileeban, S.A.; Nallathambi, D.J. Use of honeypots for mitigating DoS attacks targeted on IoT networks. In Proceedings of the 2017 International Conference on Computer, Communication and Signal Processing (ICCCSP), Chennai, India, 10–11 January 2017; pp. 1–4.
14. Deogirikar, J.; Vidhate, A. Security attacks in IoT: A survey. In Proceedings of the 2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), Palladam, India, 10–11 February 2017; pp. 32–37.
15. Seralathan, Y.; Oh, T.T.; Jadhav, S.; Myers, J.; Jeong, J.P.; Kim, Y.H.; Kim, J.N. IoT security vulnerability: A case study of a Web camera. In Proceedings of the 2018 20th International Conference on Advanced Communication Technology (ICACT), Chuncheon, Republic of Korea, 11–14 February 2018; pp. 172–177.
16. Favaretto, M.; Tran Anh, T.; Kavaja, J.; De Donno, M.; Dragoni, N. When the price is your privacy: A security analysis of two cheap IoT devices. In *Proceedings of 6th International Conference in Software Engineering for Defence Applications: SEDA 2018*; Springer International Publishing: Berlin/Heidelberg, Germany, 2020; pp. 55–75.
17. Šimon, M.; Huraj, L. A Study of DDoS Reflection Attack on Internet of Things in IPv4/IPv6 Networks. In *Software Engineering Methods in Intelligent Algorithms. CSOC 2019. Advances in Intelligent Systems and Computing*; Silhavy, R., Ed.; Springer: Berlin/Heidelberg, Germany, 2019; Volume 984.
18. Alladi, T.; Chamola, V.; Sikdar, B.; Choo, K.K.R. Consumer IoT: Security vulnerability case studies and solutions. *IEEE Consum. Electron. Mag.* **2020**, *9*, 17–25. [CrossRef]

19. Rajendran, G.; Nivash, R.S.R.; Parthy, P.P.; Balamurugan, S. Modern security threats in the Internet of Things (IoT): Attacks and Countermeasures. In Proceedings of the 2019 International Carnahan Conference on Security Technology (ICCST), Chennai, India, 1–3 October 2019; pp. 1–6. [CrossRef]
20. Valente, J.; Koneru, K.; Cardenas, A. Privacy and Security in Internet-Connected Cameras. In Proceedings of the 2019 IEEE International Congress on Internet of Things (ICIOT), Milan, Italy, 8–13 July 2019; pp. 173–180. [CrossRef]
21. Krishna, R.R.; Priyadarshini, A.; Jha, A.V.; Appasani, B.; Srinivasulu, A.; Bizon, N. State-of-the-art review on IoT threats and attacks: Taxonomy, challenges, and solutions. *Sustainability* **2021**, *13*, 9463. [CrossRef]
22. Folgado, F.J.; González, I.; Calderón, A.J. Data acquisition and monitoring system framed in Industrial Internet of Things for PEM hydrogen generators. *Internet Things* **2023**, *22*, 100795. [CrossRef]
23. Ahmed, Y.A.; Huda, S.; Al-rimy, B.A.S.; Alharbi, N.; Saeed, F.; Ghaleb, F.A.; Ali, I.M. A Weighted Minimum Redundancy Maximum Relevance Technique for Ransomware Early Detection in Industrial IoT. *Sustainability* **2022**, *14*, 1231. [CrossRef]
24. Dhirani, L.L.; Armstrong, E.; Newe, T. Industrial IoT, Cyber Threats, and Standards Landscape: Evaluation and Roadmap. *Sensors* **2021**, *21*, 3901. [CrossRef] [PubMed]
25. Anand, P.; Singh, Y.; Selwal, A.; Alazab, M.; Tanwar, S.; Kumar, N. IoT Vulnerability Assessment for Sustainable Computing: Threats, Current Solutions, and Open Challenges. *IEEE Access* **2020**, *8*, 168825–168853. [CrossRef]
26. Liashenko, O.; Kazmina, D.; Rosinskiy, D.; Dukh, Y. Analysis of Vulnerabilities of IoT-Devices and Methods of Their Elimination. *Comput. Linguist. Intell. Syst.* **2021**, *2021*, 27–37.
27. El-Gendy, S.; Azer, M.A. Security Framework for Internet of Things (IoT). In Proceedings of the 2020 15th International Conference on Computer Engineering and Systems (ICCES), Cairo, Egypt, 15–16 December 2020; pp. 1–6. [CrossRef]
28. Fragkos, G.; Minwalla, C.; Plusquellic, J. Tsiropoulou EE. Artificially intelligent electronic money. *IEEE Consum. Electron. Mag.* **2020**, *10*, 81–89. [CrossRef]
29. ElKashlan, M.; Azer, M. Mitigating IoT Security Challenges Using Blockchain. In Proceedings of the 2020 15th International Conference on Computer Engineering and Systems (ICCES), Cairo, Egypt, 15–16 December 2020; pp. 1–6.