applied sciences

MDPI

Article

Comprehensive Study of IoT Vulnerabilities and Countermeasures

Ian Coston *, Eadan Plotnizky and Mehrdad Nojoumian

Department of Electrical Engineering and Computer Science, Florida Atlantic University, 777 Glades Road, Boca Raton, FL 33431, USA; eplotnizky2020@fau.edu (E.P.); mnojoumian@fau.edu (M.N.)
* Correspondence: icoston2016@fau.edu; Tel.: +1-561-297-3411

Abstract: This comprehensive study provides an in-depth examination of the Internet of Things (IoT), which refers to the interconnection of multiple devices through various wireless protocols that facilitate data transfer and improve operational intelligence. IoT is widely used in numerous fields, including urban infrastructure, domestic settings, transportation systems, military operations, healthcare, and agriculture. However, with its growing prevalence comes a significant increase in security risks across multiple layers, such as hardware, software, cloud infrastructure, and networks. This study categorizes these vulnerabilities and explores how adversaries can exploit weaknesses to compromise IoT systems. In doing so, it highlights the risks associated with unauthorized access, data breaches, and system manipulation, all of which pose a direct threat to confidentiality, integrity, and availability. To address these concerns, this paper examines various mitigation strategies that aim to enhance IoT security by reducing attack surfaces, improving authentication methods, and securing communication protocols. By systematically analyzing existing vulnerabilities and countermeasures, this research contributes to the ongoing effort to fortify IoT devices and infrastructure against current and emerging threats. Through this study, we seek to advance the discussion on securing IoT environments while emphasizing the importance of proactive security measures in this rapidly evolving landscape.

Keywords: internet of things; IoT security; intrusion detection system; IoT protocols; vulnerability analysis; attack mitigation

check for updates

Academic Editor: Masoud Barati and Nafiseh Kahani

Received: 29 January 2025
Revised: 8 March 2025
Accepted: 10 March 2025
Published: 11 March 2025

Citation: Coston, I.; Plotnizky, E.; Nojoumian, M. Comprehensive Study of IoT Vulnerabilities and Countermeasures. *Appl. Sci.* **2025**, *15*, 3036. https://doi.org/10.3390/app15063036

1. Introduction

The rapid expansion of the Internet of Things (IoT) has significantly changed the way technology interacts with the world, connecting billions of devices across a wide range of industries. IoT technology is at the core of modern automation, enabling smart cities, self-regulating industrial systems, interconnected medical devices, and advanced transportation networks. These devices are designed to improve efficiency and streamline operations by facilitating real-time data transfer and intelligent decision making. However, the increasing reliance on IoT introduces severe security concerns that cannot be overlooked. IoT devices, by their nature, are highly susceptible to a broad spectrum of vulnerabilities across different layers, including hardware, software, cloud infrastructure, and network communications. These vulnerabilities range from unpatched firmware and weak authentication mechanisms to unsecured wireless transmissions and exploitable network protocols. The presence of these weaknesses poses a direct risk to data confidentiality, system integrity, and operational availability, leaving IoT environments vulnerable to malicious actors. Unlike traditional computing systems, IoT devices often operate with limited security capabilities, making them attractive targets for cyber attacks. This study aims to provide a thorough investigation of IoT vulnerabilities by systematically categorizing them based on their origin and impact. Additionally, we examine potential attack vectors that adversaries may use to exploit these vulnerabilities and compromise IoT ecosystems. Understanding these weaknesses is essential in developing effective countermeasures that mitigate security risks and enhance the resilience of IoT deployments. As IoT adoption continues to grow, the need for robust security frameworks becomes increasingly critical. Through this research, we seek to highlight the importance of securing IoT environments by analyzing current security gaps, exploring potential threats, and proposing mitigation strategies that will contribute to the overall safety and reliability of IoT technology.

Organization of the Paper

The paper is structured to provide a comprehensive exploration of IoT security challenges and potential solutions. The first section, Section 2, Preliminary Research, introduces foundational IoT concepts, including an in-depth examination of the five-layer IoT architecture and its applications across various industries. This section highlights how IoT is integrated into domains such as smart cities, healthcare, transportation, industrial automation, agriculture, and military operations. By outlining these diverse applications, we establish the security challenges unique to each IoT deployment. Next, we have Commonly Used Protocols which provides an overview of the various communication protocols that enable IoT functionality. This includes both widely adopted and lesser-known protocols used in IoT ecosystems, each with its own security considerations. Understanding the strengths and weaknesses of these protocols is critical for evaluating their role in securing IoT networks. Following this, Section 3, IoT Vulnerability Layers, delves into the specific vulnerabilities present within IoT systems, categorizing them into hardware, software, network, and sensor network vulnerabilities. This section examines how these weaknesses impact the confidentiality, integrity, and availability of IoT devices and infrastructure. By understanding these vulnerabilities, we can better assess the risks that IoT environments face and identify potential attack vectors that adversaries may exploit. Then, Section 4, Significance of Our Paper, will go over other recent surveys that are similar to ours. The goal of this section is to differentiate what makes our paper unique compared to others. Finally, Section 5, Conclusions and Future Research to be Worked, summarizes the key findings of this study and discusses the importance of continuous research in IoT security. As technology advances, new vulnerabilities and attack techniques will emerge, requiring ongoing improvements to security frameworks. Future research directions will focus on refining existing countermeasures, addressing newly discovered threats, and ensuring that IoT systems remain secure in the face of evolving challenges. Overall, this paper seeks to contribute to the development of secure IoT environments by thoroughly analyzing vulnerabilities, attack methodologies, and effective mitigation strategies. By understanding the risks and implementing proactive security measures, we can enhance the resilience of IoT systems against both current and future threats.

2. Preliminary Materials

The ‘Internet of’ series of technologies has been an idea that has been developing over time for many different purposes. In today’s society, most of the new technological evolution is typically part of this thought process, whether its Internet of People (Human–Machine Interaction), Internet of Agents (Machine Learning and Artificial Intelligence), Internet of Content (Cloud), or Internet of Things (Machine-to-Machine Interaction) [1]. While the authors in [1] looked at the overall ‘Internet of’ series, we will focus on the Internet of Things. The idea of the Internet of Things is essentially when multiple devices, otherwise known as "Things", are connected together in some manner through the use of a wireless network of some sort. This network can be any wireless protocol such as Bluetooth, WiFi, 5G, ultra-wideband, radio frequency identification, and many more. Now, all these devices communicate with each other, send data to each other, and are typically used to make things smarter [1–3]. Some of the uses of IoT are seen in technology found in cities, homes, transportation, military, agriculture, and so many more. The idea is to make life easier for people who use technology by making the technology smarter. However, all IoT use cases generally fall within the general multilayer architecture ranging from three to six layers, with the four-layer model being the most popular. The five-layer architecture, as shown in Figure 1, is one that we will focus on in this investigation, as it goes a bit deeper compared to the four-layer model [4].

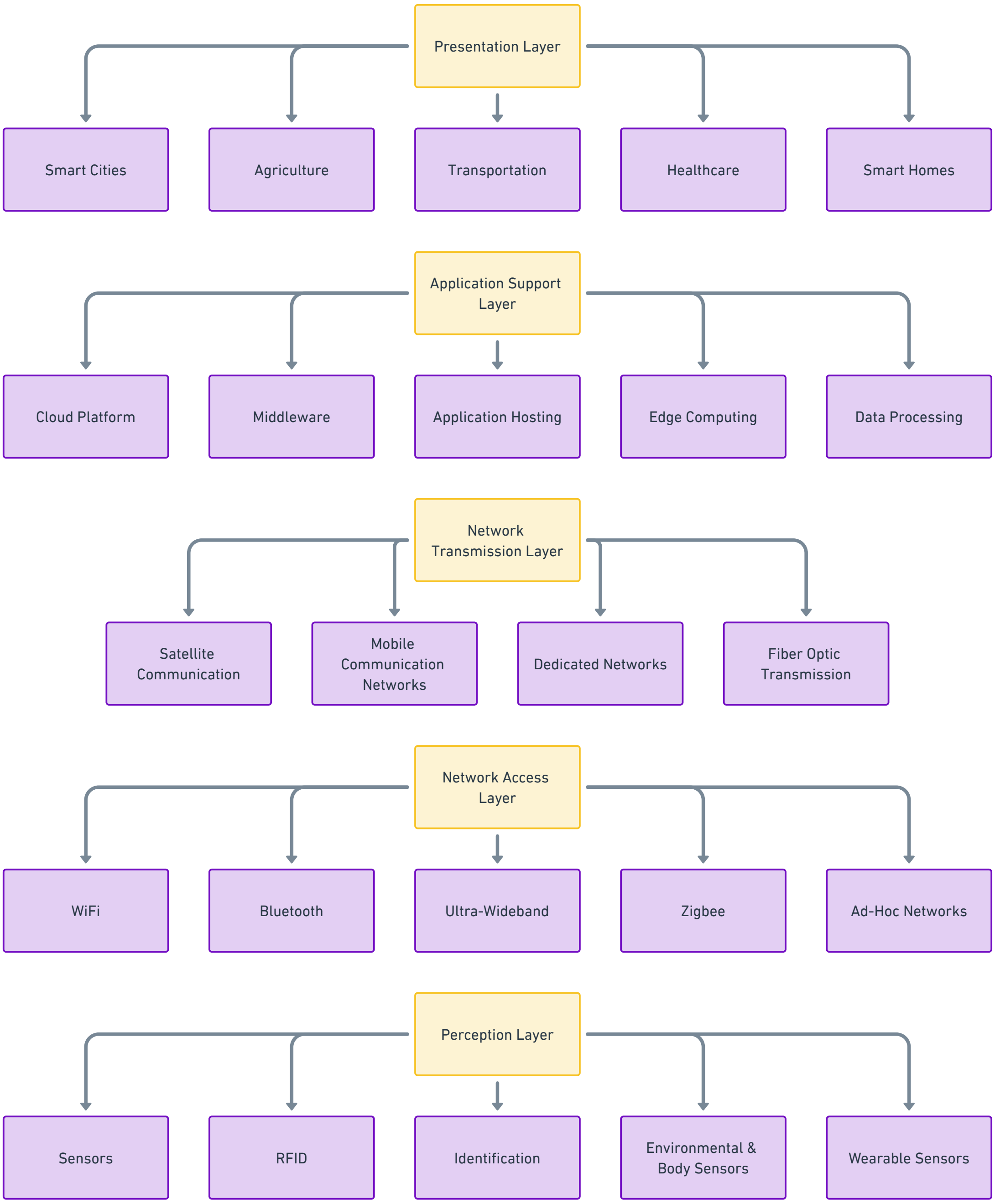

Figure 1. Five-layer architecture. This image has been completely re-modified; however, the idea of the image was inspired by [4].

The five-layer architecture consists of the perception layer, which is the layer that gathers the data. This can consist of sensors and other things that are used to collect information, such as barcodes [4]. We then move up a layer to the network access layer, which is essentially the means of communication. This is how the device sends out signals and communicates with other devices. This can be seen as wireless protocols such as ultra-wideband, WiFi, Bluetooth, etc., and it helps create the sensor network [4]. We then have the 'Network Transmission Layer' above that, which is how the network communicates as a whole. This can be seen as communication using satellites, mobile communication networks, or even a specific dedicated network [4]. The next layer would be the "Application Support Layer" which consists of middleware, cloud platforms, application host, etc. [4]. Finally, we have the "Presentation Layer", which is what IoT devices are solely used for, whether it is for a smart cities, agriculture, transportation, and many more [4]. In other words, the presentation layer is the output layer for IoT devices where they act upon the world or environment.

2.1. Internet of Things for Smart Cities and Transportation

As the Internet of Things devices become more prevalent and advanced, they are slowly being integrated into cities and vehicles in various aspects to allow for a smoother city or vehicular experience overall. The IoT can be used everywhere within cities, from security to maintenance, facilitating operations and even vehicle mobility [5]. Some of the most common uses of the Internet of Things in cities can be seen in Figure 2. As we can see, there are many avenues for IoT within smart cities that are all vary different and specific in their overall purpose.

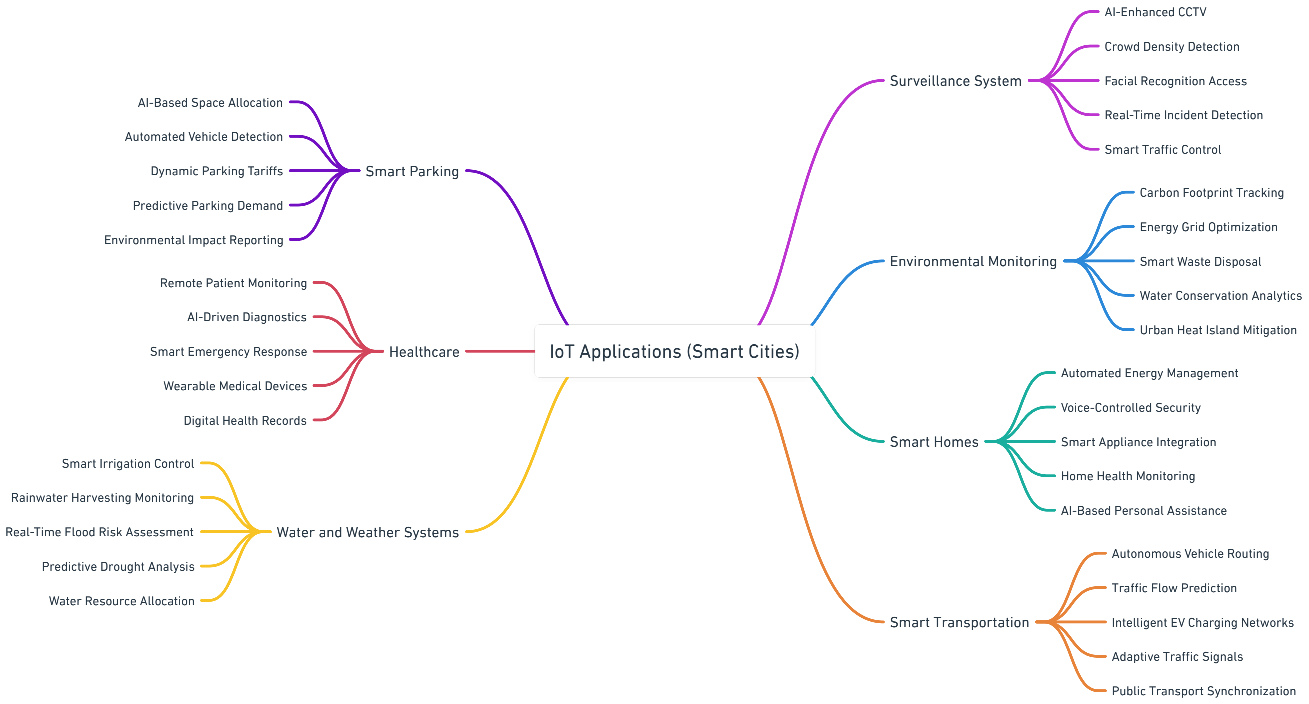
IoT Applications (Smart Cities)
Smart Parking
AI-Based Space Allocation
Automated Vehicle Detection
Dynamic Parking Tariffs
Predictive Parking Demand
Environmental Impact Reporting
Healthcare
Remote Patient Monitoring
AI-Driven Diagnostics
Smart Emergency Response
Wearable Medical Devices
Digital Health Records
Water and Weather Systems
Smart Irrigation Control
Rainwater Harvesting Monitoring
Real-Time Flood Risk Assessment
Predictive Drought Analysis
Water Resource Allocation
Surveillance System
AI-Enhanced CCTV
Crowd Density Detection
Facial Recognition Access
Real-Time Incident Detection
Smart Traffic Control
Environmental Monitoring
Carbon Footprint Tracking
Energy Grid Optimization
Smart Waste Disposal
Water Conservation Analytics
Urban Heat Island Mitigation
Smart Homes
Automated Energy Management
Voice-Controlled Security
Smart Appliance Integration
Home Health Monitoring
AI-Based Personal Assistance
Smart Transportation
Autonomous Vehicle Routing
Traffic Flow Prediction
Intelligent EV Charging Networks
Adaptive Traffic Signals
Public Transport Synchronization

Figure 2. IoT applications within smart cities. This image has been completely re-modified; however, the idea of the image was inspired by [6].

For example, you may be driving through the city in a Tesla vehicle that communicates with the surrounding smart vehicles and the traffic lights to ensure that your car and everyone else have a smooth driving experience. You may see cameras around the city that are all connected to a CCTV system that is used to prevent crime from occurring in the city. As you drive, you see a store that you want to go shopping in. You park in a parking lot and use a smart meter to monitor the time you stay there, and this may ticket you if you stay too long without paying. When you walk into the store, you see cashierless registers and cameras everywhere that track the placement of items and objects found in the store. This is the experience that a smart city wants citizens to have. All of these are used to make the city run much more smoothly and to drastically improve the lives of its citizens. With all this in mind, IoT is also being used more extensively in transportation [5]. In particular, we are seeing IoT being integrated into transportation through the use of vehicle-to-vehicle (V2V) communication protocols as well as vehicle-to-infrastructure (V2I) communication protocols. These protocols essentially allow each vehicle with IoT sensors to communicate with each other but also with the surrounding buildings within the city. Electric scooters, commonly seen around cities, as well as bus systems, are all controlled by IoT devices. All of these work together with the city to prevent citizens from being stuck in a specific location of the city, allowing for easier mobility. However, another way in which they are often used is with smart meters and self-driving capabilities. Meters use V2I technology to communicate with the vehicle to obtain an accurate time of how long the vehicle is positioned at a spot. This allows everything to be more accurate when it comes to meter tracking. Furthermore, for self-driving cars on the roads, they often use IoT technology to communicate with other cars and the infrastructure to know things such as the speed limit; the color of lights; street signs; monitor other cars, people, and objects; and so much more. Once again, the purpose is to essentially make life easier for the user.

2.2. Internet of Things for Advanced Manufacturing

The Internet of Things is not only transforming our homes and cities but also revolutionizes the manufacturing industry. Known as the Industrial Internet of Things (IIoT), this integration brings IoT technologies into advanced manufacturing processes, enhancing efficiency, reducing downtime, and improving overall productivity. As we can see, there are many different avenues where IoT is applied within manufacturing, each serving specific purposes. For example, imagine a factory floor where every machine is equipped with sensors that monitor temperature, vibration, and performance in real time. These sensors collect data and send them to a centralized system that uses advanced analytics to predict when a machine might fail or require maintenance. This predictive maintenance approach can prevent unexpected breakdowns, saving time and reducing costs. Furthermore, IoT devices enable the automation and remote control of manufacturing processes. Robots and automated systems can communicate with each other and adjust operations based on real-time data. For instance, if a production line is experiencing a bottleneck, the system can automatically reroute tasks or adjust production rates to optimize workflow. In advanced manufacturing, IoT is also enhances supply chain management. With IoT sensors tracking inventory levels, shipments, and logistics, manufacturers can maintain optimal stock levels and reduce waste. Additionally, IoT devices can ensure quality control by monitoring products throughout the production process, detecting defects early, and ensuring consistency [7]. All of these applications of IoT in advanced manufacturing aim to make the production process more efficient, reliable, and cost effective. Just like in smart homes and cities, the ultimate goal is to make life easier and more productive, not only for manufacturers but also for consumers who benefit from better products and services.

2.3. Internet of Things Smart Home System

Internet of Things devices are becoming a major component of the home environment, and their sole purpose is to make life easier for the user and people living within that environment. They first started as small gadgets or ’things’ that could be added to a home ecosystem and connected to other ’things’ with wireless signals, but as times progresses, more and more native home devices come with the ‘Internet of Things’ factor already built in [8]. The home IoT ecosystem is varied and not the same everywhere, but one of the ways it can be seen is as follows. The user may push a button in their car that sends a signal to the garage to open. From there, they tell an Alexa device that they are home, and then, Alexa would follow the routine that the user has set within the app. This routine would turn on all of the house lights, set the AC thermostat to 68 degrees Fahrenheit, set the air purifiers to a medium setting, turn on the living room TV to their favorite channel, etc. Then, when the user prepares for bed, they would tell Alexa ‘good night’. Alexa would then close all window blinds, set the air purifiers to high, turn off all lights, set the AC to a lower temperature, and activate the robot vacuum for night cleaning. This is just a sample of what home automation may look like. Everyone sets it differently and has a different use, but the overall idea of it is that it makes someone’s life easier.

2.4. Internet of Medical Things

Internet of Things devices find extensive use in the medical field, also known as the Internet of Medical Things (IoMT). IoMT is found everywhere today with wireless medical devices, widely used in hospitals, nursing homes, and homes (for personal health related use). Integrating IoT technology into the medical field has revolutionized healthcare delivery, offering numerous benefits and practical applications for healthcare services and patients. Adopting IoMT protocols offers numerous practical applications for medical device companies, health services, patients, and caregivers. For example, healthcare providers can collect real-time data on patient health conditions, such as the heart rate, blood pressure, temperature, SPO2 (blood oxygen saturation levels), and respiratory rate. This remote monitoring capability allows for spot checks, continuous patient monitoring, and the early detection of abnormalities in the patients’ health. Patients can actively participate in their care using mobile applications that connect to IoMT devices, providing instant access to their health status and facilitating the self-management of chronic diseases, saving valuable time and enabling health services to focus on other tasks. IoMT facilitates the development of innovative solutions, such as the CardiacSense watch based on IoMT [9]. The CardiacSense watch incorporates sensors and interfaces such as the pulse rate, ECG recording, continuous detection of A-fib, unlimited event reports, manually added measurement values, general arrhythmia detection, threshold configuration, detailed event report, monthly reports, and sleep time tracking [10].

2.5. Internet of Agricultural Things

We can see IoT implementation everywhere with the ongoing advances in this technology. One of the hottest implementations of IoT is in agriculture. From basic smart home gardening systems to smart farms, the IoAT offers intelligent solutions to different agricultural applications, such as precision farming, livestock monitoring, greenhouse monitoring, and agricultural drones. Smart farms often use a variety of sensors to collect real-time data on the farms status to monitor both livestock and crops, for example, with sensors that monitor oil richness, temperature, humidity, gas, air pressure, water pressure, and crop disease for field monitoring; and temperature, heart rate, and digestion for livestock [11]. One of the many innovations designed and implemented is SeeTree, an “Intelligence Platform for Trees” that allows growers to monitor the productivity and health of individual trees. It can scan and analyze hundreds of millions of trees using rich information sources, including drones, satellites, IoT sensors, climate data, and more [12,13].

2.6. Internet of Battlefield Things

Internet of Things devices are used worldwide, but an application that is becoming increasingly mroe prevalent is within the battlefield environment. This is called the Internet of Battlefield Things, otherwise known as IoBT for short. These devices are used for the sole purpose of helping the soldier while on the battlefield, whether that is by monitoring and tracking the soldier and enemies, or by actually using the IoBT device for warfare purposes [14,15]. For example, in a given field environment, the soldier will have a heart rate sensor, a salinity sensor, an optical sensor, and movement tracking, all of which are used to monitor the body and movement of the soldier. In the air, they may have an unmanned aerial vehicle (UAV) that uses video monitoring of the entire environment layout to track soldiers and enemies. Lastly, the vehicles in the field that the soldiers use will have tracking technology, as well as mine detection, sensors on weapons, etc. The Internet of Battlefield Things is an area of IoT that is extremely broad but very specific at the same time. IoT is present in signal towers, for example, but it is also used for closely monitoring the health of the soldier. Any IoT technology can be considered an IoBT device as long as it is used in the battlefield environment.

2.7. Commonly Used Protocols

Along with the evolution of IoT devices, many protocols were developed to satisfy the needs of different IoT applications. There are many different situations and needs that come with IoT devices. Some devices need protocols that are energy efficient and have a short range, some need different data rates, and some require long-range and high-speed data transport, depending on the use case. This section will give a brief introduction to some of the more common protocols as well as some of the least common protocols that you can find within IoT devices.

2.7.1. ZigBee

ZigBee is a wireless networking protocol based on the IEEE 802.15.4 technical standard [16]. It has a low data rate, low power consumption, low cost, and encrypted communication using the Advanced Encryption Standard (AES) with a 128-bit key. ZigBee is a great technology for Internet of Things devices that need a long battery life and a low data transfer rate. The ZigBee data rate is 250 kbps at 2.4 GHz (global), 40 kbps at 915 MHz (Americas), and 20 kbps at 868 MHz (Europe). Some applications of ZigBee can be found in smart homes, smart buildings, the Internet of Medical Things (IoMT), and more [17,18].

2.7.2. Dash7

Dash7 (D7AP) is an open source subGHz wireless sensor and actuator network protocol (WSAN) that complies with the ISO/IEC 18000-7 standard. It has a medium range of up to 2 Km, low power consumption (multi-year battery life), low latency, and is encrypted using AES with a 128-bit key. D7AP operates in unlicensed ISM bands of 433.92, 868, and 915 MHz [19,20]. A few use cases of D7AP can be found in agriculture IoT, IoMT, and smart cities [21].

2.7.3. WiFi

WiFi, or IEEE 802.11, is a standard of wireless LAN technology widely used in business and home environments to obtain fast and reliable Internet access. WiFi provides a common platform to connect various devices, from smart home applications to industrial sensors. The range, reliability, and security strengths of WiFi are ideal for many IoT applications [22]. WiFI encryption has evolved in recent years, and each new generation provides stronger security against attacks. WEP was the original WiFi security protocol, which used the RC4 algorithm with two sides of data communication. However, WEP is easily cracked, and it is no longer considered secure. WPA was an improved version of WEP and addressed some of the security vulnerabilities in WEP. WPA2 is the successor to WPA and is considered to be more secure. WPA2 uses the Advanced Encryption Standard (AES) cipher [23]. WPA3 is the latest WiFi security protocol, offering the strongest security to date. WPA3 uses the Simultaneous Authentication of Equals (SAE) protocol, designed to be more secure than the four-way handshake used in WPA2 [24]. IEEE 802.11 is a living standard and new generations are being developed regularly to meet the growing demands of wireless networking. Some of the most used IEEE 802.11 standards are IEEE 802.11g, which operates in the 2.4 GHz band and supports data rates up to 54 Mbps; IEEE 802.11n, which operates in the 2.4 GHz and 5 GHz bands and supports data rates up to 600 Mbps.; IEEE 802.11ac, which operates in the 5 GHz band and supports data rates up to 6.93 Gbps; and IEEE 802.11ax, which operates in the 2.4 GHz and 5 GHz bands and supports data rates up to 9.6 Gbps [25,26].

2.7.4. Cellular

Cellular networks have been around for a while. With new advancements and technologies being developed around them, new cellular networks were discussed, tested, and created to provide the best performances that can match the needs of the latest technology standards. LTE-Advanced(4G) and 5G technologies are the most recent and widely used cellular standards [27]. LTE-A (Long-Term Evolution Advanced) is a wireless cellular technology that significantly improved speed, capacity, and coverage over previous generations of cellular technology. It has a peak downlink data rate of 1 Gbps, an uplink data rate of 500 Mbps, a peak downlink spectrum efficiency of 30 bps/Hz, an uplink spectrum efficiency of 15 bps/Hz, and a bandwidth of 100 MHz [28]. LTE-A is currently being overtaken by 5G [29]. 5G is the fifth generation of wireless cellular technology. It has demonstrated improvements over previous generations of cellular networks, such as higher data rates, lower quality of service (QoS) latency, low interference, and increased capacity. Some of the 5G requirements include a maximum downlink data rate of 20 Gbps, an uplink data rate of 10 Gbps, a maximum downlink spectrum efficiency of 30 bps/Hz, an uplink spectrum efficiency of 15 bps/Hz, user plane latency of 4 ms for eMBB and 1 ms for URLLC, control plane latency of 10–20 ms, and a bandwidth of 100 MHz–1 GHz [30]. In sum, 5G is the key for advanced IoT applications, such as smart factories, smart hospitals, smart transportation, smart agriculture, smart homes and cities, etc. [30,31].

2.7.5. 6LoWPAN

6LoWPAN is a networking technology that allows IPv6 packets to be efficiently transmitted over low-power wireless networks, such as those based on the IEEE 802.15.4 standard. It supports various mesh network topologies and can fragment and reassemble packets as needed. 6LoWPAN implementations are small enough to fit 32 K flash memory parts. 6LoWPAN enables low-power mesh and sensor networks to take advantage of the benefits of IP networking [32,33]. It has frequency bands of 2.4 GHz, 868 MHz, and 915 MHz (the same as ZigBee) and data rates between 50 and 250 kbit/s [33].

2.7.6. Bluetooth

Bluetooth is a technology standard used for short-range wireless communication between mobile devices. Bluetooth operates on 79 different frequencies to transmit data from 2.402 GHz to 2.48 GHz and a range up to 100 m (330 ft). The bit rates for Bluetooth are 1 Mbps and 2 Mbps [34]. It is very useful for transmitting small fragments from different IoT sensors [35].

2.7.7. Bluetooth Low Energy (BLE)

Bluetooth Low Energy (BLE) is a wireless technology that is designed to complement both classic Bluetooth and the lowest power wireless technology possible. It is a distinct technology with different design goals and market segments than classic Bluetooth. BLE transmits data over 40 channels in the 2.4 GHz band (2.402 to 2.48 GHz) [36]. It can be used to create different types of networks, from simple point-to-point connections to complex mesh networks. This flexibility makes BLE compatible with a wide range of applications, including the Internet of Things.

2.7.8. LoRa and LoRaWAN

LoRa is an unlicensed band physical layer technology that transmits data signals in the subGHz ISM band. LoRa allows low-data-rate long-range, low-power wireless communication. LoRa has a range of up to 15 km in rural areas and up to 5 km in urban areas, with data rates of 0.3 kbps–50 kbps in Europe and 0.9 kbps–50 kbps in the US [37,38]. LoRaWAN is an open standard that was developed on top of LoRa. It consists of an end device, gateway, network server, and application server. LoRaWAN is located in the data link layer and provides a complete solution by adding a network layer that includes features such as security, authentication, and data routing [21].

2.7.9. SigFox

Sigfox is a low-power wide area network (LPWAN) technology designed for the Internet of Things. It uses ultra-narrowband technology to transmit data with a very low power consumption over long distances [39]. It operates in the 862–928 MHz frequency band and has a channel bandwidth of 100 Hz [40]. With a range of up to 50 km in rural areas and up to 10 km in urban areas, Sigfox can work well in applications that require long-range communication with battery-powered devices. Its data rate ranges from 100 to 600 Bps, depending on the region [30,41].

2.7.10. Narrowband Internet of Things (NB-IoT)

NB-IoT is a low-cost, low-power, and low-data-rate cellular technology built from LTE functions; therefore, it uses the same infrastructure as cellular networks, which makes it a scalable and reliable technology that can be deployed in a variety of locations. It has a range of up to 15 km in rural areas and 1–5 km in urban areas, a data rate up to 250 Kbps, and a 200 KHz bandwidth [21,42].

2.7.11. Near-Field Communication (NFC)

Near-Field Communication (NFC) is a short-range wireless communication protocol used by mobile devices for all kinds of applications, such as payments, digital keys for homes and cars, and data transferring. NFC provides secure communication between various devices. It has a short range of 4–10 cm, a data rate of 0.02–0.4 Mbps, and it runs on a 13.56 MHz spectrum [43]. NFC is used to enhance different IoT solutions with short-range capabilities [44].

2.7.12. Z-Wave

Z-Wave is a subGHz wireless communication protocol used by different IoT applications. It is an ultra-low-power, mesh network protocol that lets devices communicate with each other over long distances (has a range of 100 m). Its data rates are 9.6 kbps, 40 kbps, or 100 kbps, and it uses a frequency of 908.42 MHz in the United States and 868.42 MHz in Europe [45]. Z-Wave deployments can be scaled by linking together Z-Wave networks. Z-Wave is well suited for applications that require reliable, secure, and low-power communication, such as control smart home devices (lights, locks, thermostats, and security systems) [46].

2.7.13. Li-Fi

Li-Fi is a bidirectional short-range wireless technology that uses a visible light communication (VLC) system for data transmission to transfer and receive data. Li-Fi uses overhead LED lighting commonly found in homes as a means of transport and a photo-diode for decoding data. It has a maximum speed of 224 Gbps, which allows a high-definition video to be downloaded in seconds. Because Li-Fi is reliable in light use, it is limited in range since light cannot pass through objects, which makes Li-Fi effective only in closed spaces [47]. Even though its range limitation could be seen as a problem, this limitation provides an additional layer of security by prevents data from leaking into public spaces, thus preventing malicious actors from accessing your network [48].

2.7.14. Ultra-Wideband (UWB)

Ultra-wideband is a short-range, high-bandwidth and energy-efficient wireless communication protocol that can be used for radar imaging, sensor data collection, and precise location and tracking. UWB operates at frequencies from 3.1 to 10.6 GHz, has a bandwidth of 500 MHz, and a data rate of up to 1 Gbps [49,50]. UWB can be used to accurately measure the distance between two devices. This information can be used for a variety of IoT applications [51].

2.7.15. Advanced Message Queuing Protocol (AMQP)

The Advanced Message Queuing Protocol (AMQP) is a reliable and versatile M2M binary protocol. It offers two levels of QoS for the delivery of messages, uses TCP as a transport protocol, and uses TLS/SSL and SASL for security, making it a good fit for high-bandwidth, reliable, and secure networks [52]. It supports various messaging patterns, including request/response, publish/subscribe, and transactions (allowing multiple messages to be sent and received as a single unit of work) and topic-based publish-and-subscribe messaging (allows messages to be published to topics so that subscribers can receive messages that are relevant to them) [53].

2.7.16. Constrained Application Protocol (CoAP)

The Constrained Application Protocol (CoAP) is a lightweight M2M binary protocol with a fixed header of 4 bytes and small message payloads from the IETF CoRE Working Group designed for constrained IoT devices. It supports both request–response and resource–observe architectures and can be used to interoperate with HTTP and the RESTful Web API [54]. CoAP uses UDP as a transport protocol and DTLS for security, making it efficient for use on low-bandwidth and unreliable networks. It is designed to be as lightweight as possible, making it suitable for use on constrained devices with limited resources [52].

2.7.17. Message Queuing Telemetry Transport Protocol (MQTT)

The Message Queuing Telemetry Transport Protocol (MQTT) is a publish/subscribe messaging protocol used for lightweight machine-to-machine (M2M) communications in constrained networks. MQTT uses the Transmission Control Protocol (TCP) as its transport protocol that guarantees the delivery of messages. MQTT also uses TLS/SSL for security, which encrypts messages to protect them from unauthorized access. MQTT supports three levels of QoS, making it more reliable when delivering messages. It uses a small amount of bandwidth and processing power. This makes it ideal for use with small devices that have limited resources. MQTT is also suited for large networks because it can efficiently handle a large number of devices. This is because it uses a publish/subscribe model, which allows devices to receive only messages that are relevant to them [52].

2.7.18. Data Distribution Service (DDS)

The Data Distribution Service (DDS) is a machine-to-machine protocol developed by the Object Management Group (OMG) that features decentralized nodes of clients throughout a system (nodes can identify themselves as subscribers or publishers through a localization server). DDS was created to overcome the disadvantages of centralized publish–subscribe architectures. Provides many quality-of-service parameters that allow users to control the behavior of the DDS system, such as improved scalability, increased reliability, reduced latency, bandwidth, and enhanced security (provides authentication, access control, confidentiality, and integrity to the information distribution) [55,56].

2.7.19. Open Platform Communications (OPC)

Open Platform Communications (OPC) is a machine-to-machine protocol that facilitates real-time data exchange between control systems and devices and allows seamless connectivity. OPC supports classic specifications such as OPC DA (Data Access) and newer advances such as OPC UA (Unified Architecture), increasing platform security and independence. The protocol plays a key role in the Industry 4.0 environment, supporting the integration of the Internet of Things (IoT) and computer-physical systems (CPS). OPC is useful for implementing industrial systems and optimizing data exchange in heterogeneous environments [57].

3. IoT Vulnerability Layers

All IoT devices typically show some kind of vulnerability or weakness that dampens its ability to function without issue. These vulnerabilities can typically allow users to gain access to data by utilizing weakness to enter the device, or they can be used to track and manipulate the device. Either way, this is detrimental to the device and its functionality. Some of the ways that adversaries can obtain access to this is through network vulnerabilities, software vulnerabilities, or even hardware vulnerabilities, as shown below in Figure 3 [58]. It is important to know these weaknesses so that one can know the best route to actually avoid or mitigate them.

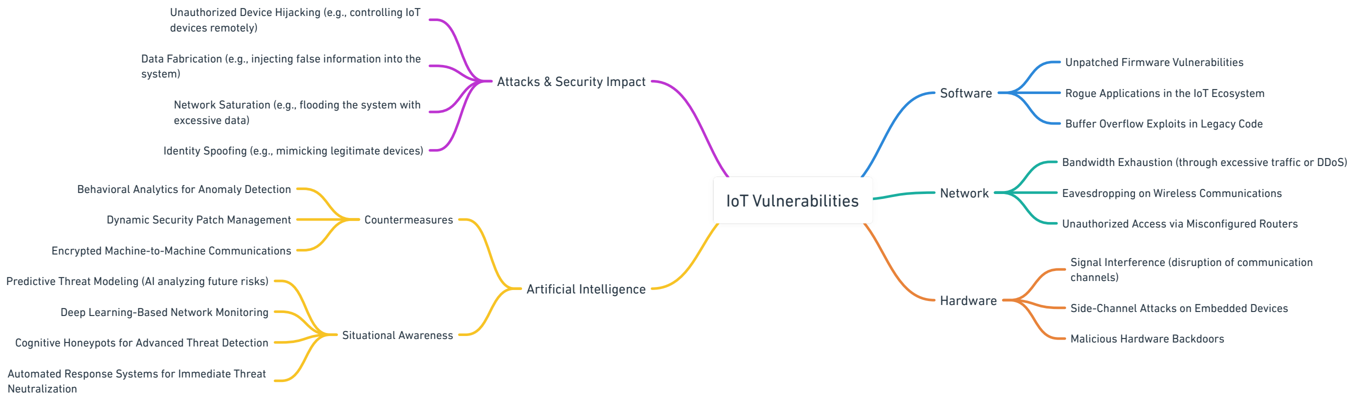
IoT Vulnerabilities
Attacks & Security Impact
Unauthorized Device Hijacking (e.g., controlling IoT devices remotely)
Data Fabrication (e.g., injecting false information into the system)
Network Saturation (e.g., flooding the system with excessive data)
Identity Spoofing (e.g., mimicking legitimate devices)
Artificial Intelligence
Countermeasures
Behavioral Analytics for Anomaly Detection
Dynamic Security Patch Management
Encrypted Machine-to-Machine Communications
Situational Awareness
Predictive Threat Modeling (AI analyzing future risks)
Deep Learning-Based Network Monitoring
Cognitive Honeypots for Advanced Threat Detection
Automated Response Systems for Immediate Threat Neutralization
Software
Unpatched Firmware Vulnerabilities
Rogue Applications in the IoT Ecosystem
Buffer Overflow Exploits in Legacy Code
Network
Bandwidth Exhaustion (through excessive traffic or DDoS)
Eavesdropping on Wireless Communications
Unauthorized Access via Misconfigured Routers
Hardware
Signal Interference (disruption of communication channels)
Side-Channel Attacks on Embedded Devices
Malicious Hardware Backdoors

Figure 3. IoT vulnerability flow. This image has been completely re-modified; however, the idea of the image was inspired by [58].

As we can see in Figure 3, these vulnerabilities are just part of the attack layer. The ways these vulnerabilities are used are represented by the other half of the attack layer. The adversary can quite easily attack any IoT device through any of the vulnerabilities, and by doing so, they can either choose to attack data integrity, confidentiality, authentication, or even availability. Confidentiality is designed to protect IoT devices and information from unauthorized access and is generally enforced with the use of encryption, access control, and also authentication of user and data [58]. Through any of these vulnerabilities above, we can see information leakage which would cause any kind of data to be releasing to anyone, whether it is the adversary or a normal person. Integrity typically guarantees the protection of unauthorized modifications to the hardware or software of a device by enforcing encryptions, input validations, interface monitoring and restrictions, and so many more [58]. These are designed to keep any portion of the device from becoming vulnerable. However, once again, during the design phase of a device, typically smaller things are overlooked, which would allow the attacker to use that device for their own malicious purposes. Accountability represents the idea of tracking actions and tasks to make sure that the device is doing what it is supposed to do. That is, it is monitoring everything the device does and limits the device to specific tasks [58]. An attacker could use the manufacturer vulnerabilities to enter a device and modify the event path. They can use this to change the purpose of devices or even reroute data to them for data monitoring. Lastly, we have the idea of availability. Availability is the idea that the device is always available for use when the user needs it [58]. When an attacker targets these vulnerabilities, they can delay the device or even make it go offline, which would render the availability weak. All of these security impacts are the results of the types of attacks that an adversary can carry out. Whether it is a physical, software, or even a network attack. The IoT device at the other end of the attack can affect one or often many of the aspects shown above.

3.1. Hardware Vulnerabilities

Most, if not all, IoT devices operate without supervision and typically have limited tamper resistant properties that make it extremely easy for an attacker to gain access to the device [58]. The attacker can modify the IoT device with respect to its services that it provides, as well as obtain data that it should not have access to and that could cause serious harm to many individuals [58]. For example, if a hacker were to gain access to a smart doorbell, it would be able to modify all the settings inside, view the data inside, or even delete the data inside. Now, let us take this step further into a very delicate environment. If an attacker were to somehow have access to a camera of a military base, they could use that access to follow where the data are being sent to. In this case, they would find a server with a lot of other camera data on it. They could then turn off the cameras and attack or even release military secrets that were caught on camera.

3.1.1. Radio Frequency Attacks

A hacker may achieve hardware-level access to a device through various methods, with the man-in-the-middle attack emerging as the most prevalent approach. Typically, discussions of man-in-the-middle attacks center on network-layer exploitations—the interception of data over wired or wireless connections. However, this perspective captures only a fraction of the technique's potential. Far more insidious is the exploitation of radio frequency (RF) emissions, which all electronic devices inherently produce. These emissions, detectable using tools such as spectrum analyzers or software-defined radios (e.g., HackRF, Ettus Research devices, or even low-cost RTL-SDR units), offer a gateway for attackers to observe and manipulate device communication at a fundamental level.

The process begins with signal identification. Every device emits RF signals characterized by unique attributes, which an attacker can analyze to discern their nature. These attributes include the bandwidth (the signal's spectral footprint), frequency (its position, such as 2.4 GHz for Bluetooth or 5 GHz for Wi-Fi), signal strength (its amplitude), and behavioral patterns (e.g., frequency hopping, intermittent bursts, or continuous transmission). Visualized on a spectrum analyzer, the signal's waveform—whether a sharp pulse or a modulated curve—further reveals its identity. With this information in hand, the attacker can classify the signal—be it Bluetooth, Zigbee, or a proprietary protocol—and assess its vulnerabilities. Wireless protocols, by their nature, possess inherent weaknesses, such as inadequate encryption, predictable timing intervals, or exploitable handshake mechanisms. Exploiting these flaws, the attacker may disrupt the signal, assume control over it, or subtly alter its content.

One effective defense against these types of attacks is the use of strong encryption mechanisms such as AES-128/256-bit encryption and elliptic curve cryptography (ECC) [59]. By encrypting data before transmission, even if an attacker successfully intercepts a signal, they would be unable to decipher its contents. Additionally, techniques such as rolling codes and nonce-based authentication prevent replay attacks by ensuring that previously captured transmissions cannot be reused. Secure key exchange protocols such as Diffie–Hellman or ECDH should also be enforced to make sure that even intercepted communications remain indecipherable [60,61].

RF-based man-in-the-middle attacks extend beyond mere disruption, enabling both the interception and manipulation of communications. The attacker might position themselves as an intermediary between two devices—say, a smartphone and a wireless peripheral—relaying messages while clandestinely monitoring the exchange. Alternatively, they could impersonate one device entirely, redirecting the signal to themselves. In this scenario, the originating device transmits data to the attacker, who may then forward them to the intended recipient, modify it, or retain it for later use. For instance, the attacker could inject malicious commands, alter transmitted data, or simply eavesdrop—all without the communicating devices detecting the intrusion. The RF medium facilitates this deception seamlessly, as the signals persist in their expected patterns, leaving no overt trace of interference. Without specialized monitoring equipment, such as an SDR wielded by a vigilant defender, the attack remains invisible [62–64].

To mitigate these attacks, frequency hopping spread spectrum (FHSS) and direct-sequence spread spectrum (DSSS) techniques can be used to prevent an attacker from isolating a specific transmission. These methods ensure that the signal dynamically shifts across multiple frequencies, making it significantly more difficult for an adversary to pinpoint, capture, or manipulate communications. Furthermore, enforcing strict firmware update policies can address known vulnerabilities in wireless protocols, preventing attackers from exploiting outdated security measures [65–67].

A practical illustration of this concept is the Bluetooth man-in-the-middle attack, depicted in Figure 4. While each wireless protocol exhibits distinct characteristics, the underlying principles of RF exploitation share common threads, making this example broadly instructive. The attack unfolds through one of two strategies. First, the attacker may deploy a wide-band jamming signal to overwhelm the Bluetooth frequency band (approximately 2.4 GHz), severing the connection between paired devices. Alternatively, they might target a specific device, flooding its RF time slots with randomized data to destabilize its communication channels. This latter approach aims to "shut down all of the piconets within the range susceptibility", compelling the user to initiate a re-pairing process out of frustration [68]. During this re-pairing, the attacker intercepts the exchange, forging messages within the input/output (IO) capabilities negotiation phase [68]. Notably, if the devices were previously paired, the attacker can bypass the need to force a reconnection, exploiting an established link directly.

Once inserted into the communication chain, the attacker establishes simultaneous connections to both devices, as illustrated in Figure 4. Acting as a relay, they forward messages between the pair while retaining the ability to monitor, modify, or inject data [68]. For example, they might alter audio streams, introduce unauthorized commands, or extract sensitive information—all executed within the RF domain. This manipulation hinges on the attacker's mastery of the signal's properties and the protocol's weaknesses, rendering the intrusion both potent and discreet. The elegance of such RF-based attacks lies in their subtlety; the devices continue their dialogue, oblivious to the adversary riding the airwaves between them.

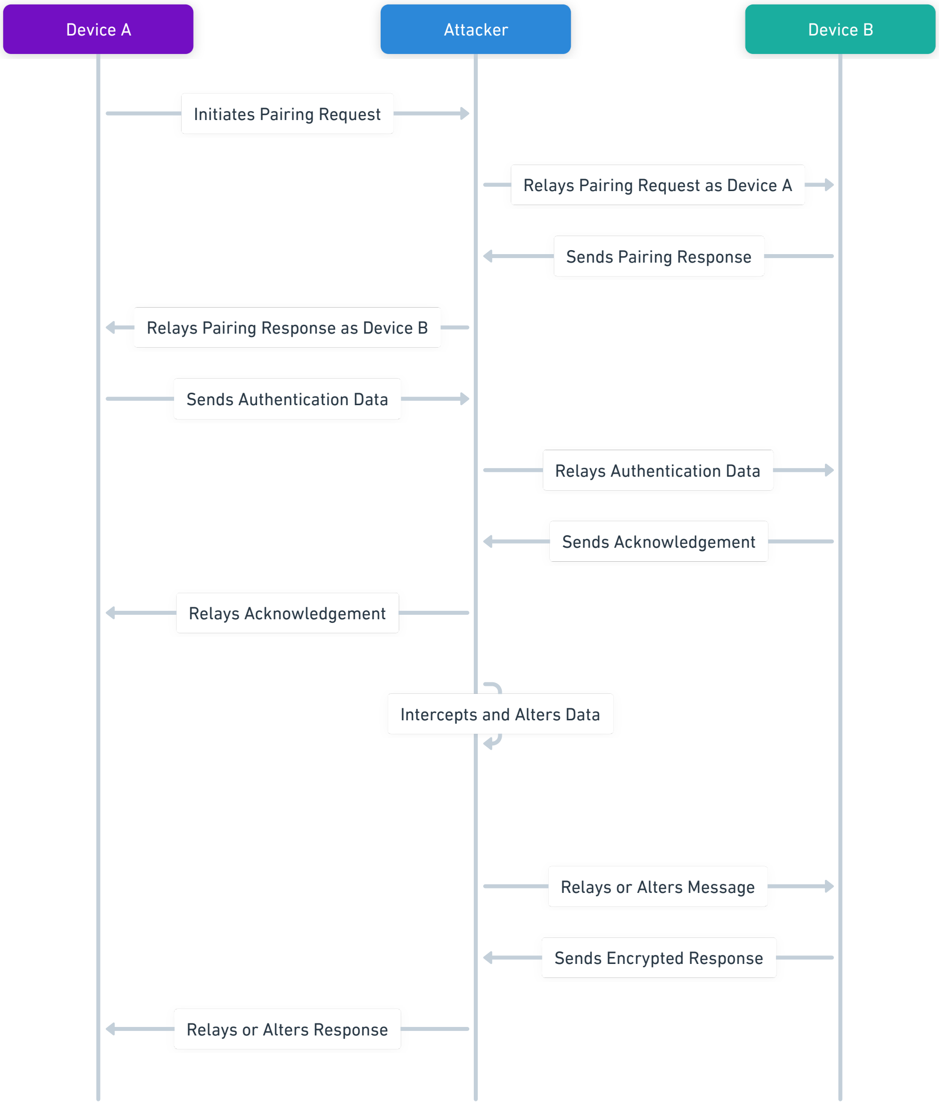
Device A
Attacker
Device B
Initiates Pairing Request
Relays Pairing Request as Device A
Sends Pairing Response
Relays Pairing Response as Device B
Sends Authentication Data
Relays Authentication Data
Sends Acknowledgement
Relays Acknowledgement
Intercepts and Alters Data
Relays or Alters Message
Sends Encrypted Response
Relays or Alters Response

Figure 4. Bluetooth man-in-the-middle attack. This image has been completely re-modified; however, the idea of the image was inspired by [68].

To counteract Bluetooth man-in-the-middle attacks, a combination of defenses at different layers of communication is necessary. Jam-resistant protocols that incorporate spread spectrum communication can prevent attackers from effectively disrupting Bluetooth signals, while adaptive power control can counteract jamming by adjusting signal strength dynamically. Additionally, secure pairing mechanisms such as Just Works Numeric Comparison or Passkey Entry, rather than legacy pairing methods, can significantly reduce the risk of adversaries intercepting the pairing process [69]. Proximity-based authentication methods, which require physical confirmation before establishing a trusted connection, can further prevent unauthorized devices from inserting themselves into the communication stream. Intrusion detection systems (IDS) designed to monitor RF spectrum anomalies can be deployed to detect unexpected transmission patterns or frequency manipulations indicative of an attack. Finally, physical-layer security enhancements, such as the use of directional antennas, shielded enclosures, and beamforming techniques, can limit RF signal exposure, making it significantly harder for attackers to intercept or manipulate communications.

3.1.2. Hardware Reverse Engineering and Micro-Probing

Reverse engineering seeks to unravel the mechanisms driving a device's operation [70]. When applied to hardware, this process equips the investigator with comprehensive insight into a device's architecture, exposing its strengths, vulnerabilities, and operational intricacies. Such attacks are categorized into three distinct types—invasive, semi-invasive, and non-invasive—as delineated in Figure 5 [71]. Each approach varies in its methodology and impact on the target device, offering attackers a spectrum of strategies to extract critical information.

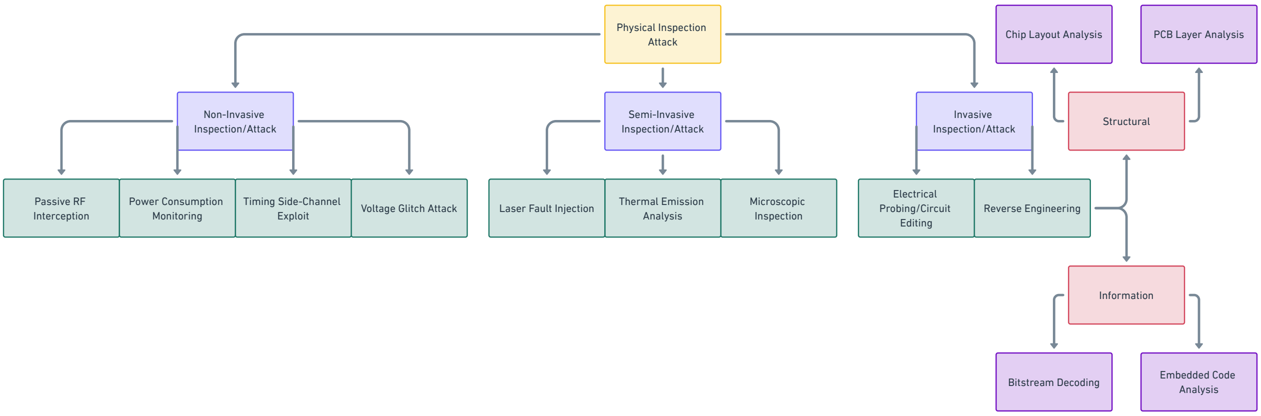
Physical Inspection Attack
Non-Invasive Inspection/Attack
Semi-Invasive Inspection/Attack
Invasive Inspection/Attack
Passive RF Interception
Power Consumption Monitoring
Timing Side-Channel Exploit
Voltage Glitch Attack
Laser Fault Injection
Thermal Emission Analysis
Microscopic Inspection
Electrical Probing/Circuit Editing
Reverse Engineering
Structural
Chip Layout Analysis
PCB Layer Analysis
Information
Bitstream Decoding
Embedded Code Analysis

Figure 5. Reverse engineering and inspection attacks. This image has been completely re-modified; however, the idea of the image was inspired by [71].

Non-invasive attacks target data extraction without altering the physical packaging or structure of the integrated circuit (IC) or printed circuit board (PCB). These attacks may be executed passively—observing emissions or behavior—or actively, through deliberate manipulation [71]. Examples include brute-force assaults and fault injection, where the attacker deploys an array of tools to probe the device's resilience. Such tools range from off-the-shelf instruments to custom-built apparatuses. A notable instance is the voltage glitch attack, wherein a precisely timed voltage spike is delivered to a specific circuit segment—often during boot—to bypass security measures or extract data [72]. Here, RF emissions play a subtle yet critical role as follows: devices under stress may leak electromagnetic signals, revealing timing patterns or internal states when monitored with a spectrum analyzer or software-defined radio (SDR). These non-invasive techniques prioritize stealth, leaving the device intact while yielding substantial insights into its operation.

To mitigate non-invasive attacks, countermeasures must focus on reducing data leakage and increasing system resilience against external probing. Implementing side-channel attack countermeasures—such as randomized execution timing, power analysis countermeasures, and shielding critical components—can help reduce the information attackers can glean from passive observations. Glitch detection circuits should be integrated to recognize abnormal voltage fluctuations and halt execution when suspicious behavior is detected. Additionally, employing hardware-based encryption with secure key storage ensures that even if attackers monitor electromagnetic emissions, the data remain indecipherable.

In contrast, invasive and semi-invasive attacks necessitate direct access to the IC or PCB's internal components, often leaving tangible evidence of interference [71]. Invasive methods involve physical interaction with the hardware—soldering wires to contact points, probing electrical buses, or even mechanically dissecting chips to expose their secrets. Meanwhile, semi-invasive approaches leverage optical techniques, such as microscopy, laser-based fault injection, or optical probing, to interact with the device at a microscopic level without fully dismantling it [71]. The toolkit for these attacks is extensive, encompassing JTagulators, logic analyzers, oscilloscopes, signal generators, power supplies, X-ray machines, and lasers, among others [71]. RF considerations emerge here as well; probing a chip's internal buses may induce unintended RF emissions, which an attacker can capture to map signal pathways or decode data flows, while such methods frequently damage the device—rendering it inoperable or visibly altered—they offer unparalleled access to its inner workings [73]. The trade-off is clear, the deeper the intrusion, the richer the data harvest, albeit at the cost of subtlety.

To defend against invasive and semi-invasive attacks, manufacturers must implement hardware security techniques at the physical layer. Anti-tamper coatings, active mesh defenses, and sensor-based intrusion detection can detect and respond to physical interference. Chip-level encryption with secure boot mechanisms ensures that even if the chip is physically accessed, its contents remain protected. Light sensors, power fluctuation detectors, and tamper-evident enclosures can be employed to thwart optical probing and laser fault injections. Additionally, fusing security-critical components after manufacturing can prevent reverse engineering by making chip-level access impractical [74].

3.1.3. Implants and Hardware Trojans

A hardware Trojan, by definition, constitutes a malicious alteration or addition to the circuitry of an Integrated Circuit (IC), introduced during its design or fabrication phases. This form of attack proves exceptionally challenging to mitigate preemptively, as the IC development pipeline lacks robust mechanisms to secure each stage—be it architectural design, synthesis, or physical layout. Compounding this vulnerability is the unpredictability of when such a Trojan might be inserted, rendering comprehensive prevention elusive [75–78]. The clandestine nature of this process ensures that the modification remains concealed until activation, posing a persistent threat to hardware integrity.

The potency of hardware Trojans lies in their capacity to operate undetected, leaving the user unaware of any compromise. This threat is universal, applicable to any device, irrespective of its purpose or complexity. An adversary wielding such a Trojan can manipulate inputs and outputs, extract sensitive information, or assume full control of the device—all without arousing suspicion. The operational diversity of these Trojans further amplifies their danger; some activate autonomously upon deployment, while others remain dormant until triggered by specific conditions, as illustrated in Figure 6 [75,76,78]. This variability in activation modes—whether time-based, signal-driven, or environmentally triggered—enhances their unpredictability, complicating efforts to detect or neutralize them.

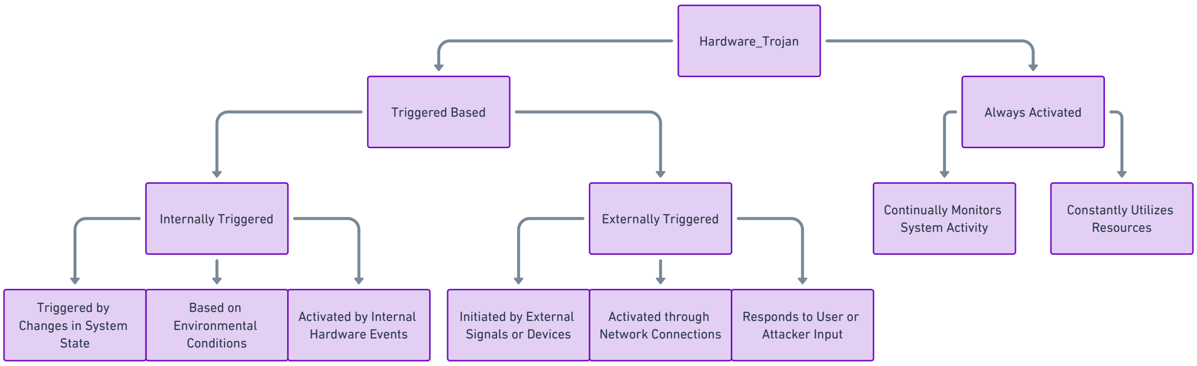
Hardware_Trojan
Triggered Based
Always Activated
Internally Triggered
Externally Triggered
Continually Monitors System Activity
Constantly Utilizes Resources
Triggered by Changes in System State
Based on Environmental Conditions
Activated by Internal Hardware Events
Initiated by External Signals or Devices
Activated through Network Connections
Responds to User or Attacker Input

Figure 6. Trojan activation modes. This image has been completely re-modified; however, the idea of the image was inspired by [75].

Mitigating hardware Trojans requires a multilayered approach spanning design-time verification, manufacturing oversight, and runtime monitoring. Implementing formal verification techniques alongside logic obfuscation during the IC design phase can significantly increase resistance to Trojan insertion [79,80]. Additionally, side-channel analysis, such as power or electromagnetic profiling, can be employed to identify anomalies indicative of unauthorized modifications. At the manufacturing stage, functional testing with hardware fingerprinting can help detect deviations from expected behavior. Post-deployment, dynamic runtime monitoring with anomaly detection algorithms can identify unauthorized data leakage or unexpected device behavior, helping neutralize threats that evade initial scrutiny.

Expanding this concept beyond the IC, hardware implants represent a broader category of malicious modifications. Unlike Trojans, which are confined to the design or fabrication stages, implants can be introduced at any point in a device's lifecycle—potentially years after its initial deployment. These implants typically manifest as discrete components or circuits integrated into the hardware, endowed with capabilities mirroring those of Trojans as follows: data leakage, operational interference, or outright control. Figure 7 exemplifies the workflow of a simple hardware Trojan, yet the principles extend seamlessly to implants. What elevates the risk of implants is their retroactive applicability; a device in active use, laden with years of accumulated data, can be retrofitted with such a modification. Once embedded, the implant grants the attacker unfettered access to stored information and ongoing operations, all while evading detection by conventional means. This adaptability underscores the profound security challenges posed by hardware-level threats, transcending the controlled environment of IC manufacturing to encompass the entirety of a device's existence.

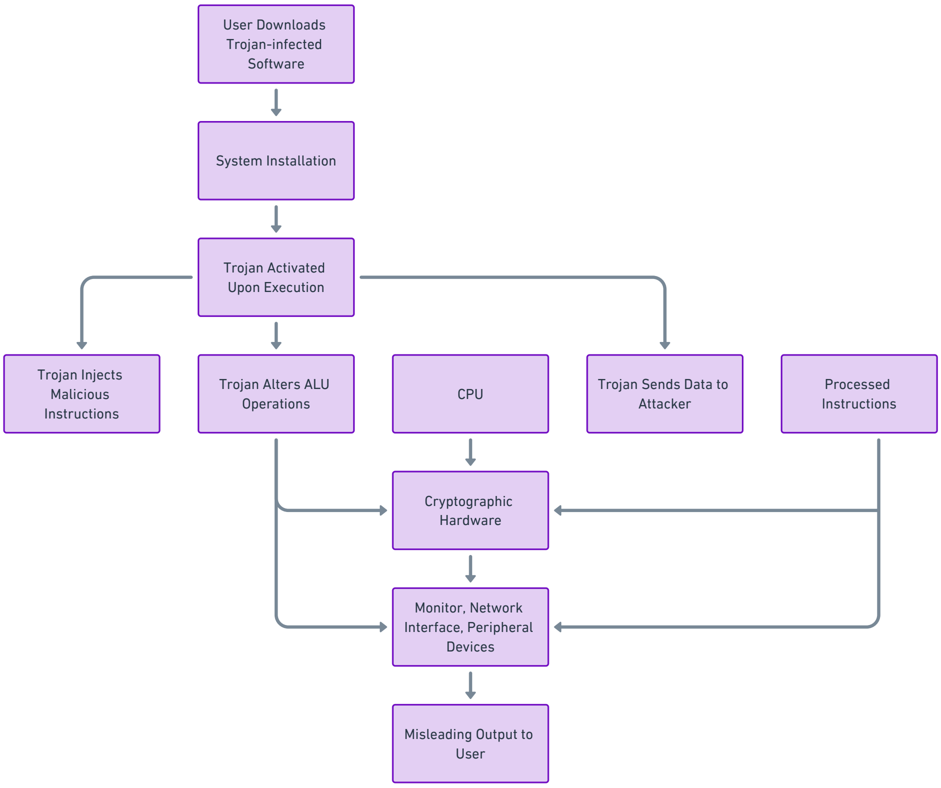
User Downloads Trojan-infected Software
System Installation
Trojan Activated Upon Execution
Trojan Injects Malicious Instructions
Trojan Alters ALU Operations
CPU
Trojan Sends Data to Attacker
Processed Instructions
Cryptographic Hardware
Monitor, Network Interface, Peripheral Devices
Misleading Output to User

Figure 7. Working process of a simple hardware Trojan. This image has been completely re-modified; however, the idea of the image was inspired by [75].

Mitigating the risks posed by hardware implants necessitates both physical security measures and continuous integrity monitoring. Device manufacturers should enforce supply chain security protocols to prevent unauthorized hardware modifications at any stage of production or distribution. Tamper-evident enclosures and active intrusion detection mechanisms can alert users to unauthorized physical modifications. Additionally, implementing periodic integrity verification through hardware attestation ensures that any post-deployment alterations are detected before they can compromise system security. Secure boot mechanisms combined with hardware-based root-of-trust validation can further reinforce a device's resistance against implanted threats [81].

3.2. Software Vulnerabilities

The functionality of Internet of Things (IoT) devices spans a wide spectrum, shaped not merely by their hardware but by the software embedded within their microcontrollers. Take, for instance, three identical IoT devices, each equipped with a Bluetooth module and indistinguishable in appearance. Their behavior hinges on the firmware as follows: one might be coded to transmit Bluetooth signals exclusively, another to receive signals only from authorized users, and a third to toggle between roles based on context. This software-driven flexibility defines the IoT's adaptability, yet it simultaneously exposes a critical Achilles' heel. When this software is compromised—whether through exposure, modification, or exploitation—the device becomes a conduit for data breaches, malicious implants, or outright control, all potentially unbeknownst to its operator.

Software vulnerabilities in IoT devices manifest in diverse forms, each presenting distinct pathways for exploitation [82]. A prevalent type is the buffer overflow, where poorly managed memory allocation allows an attacker to input data beyond a designated buffer's capacity [83]. This excess data can overwrite adjacent memory, enabling the injection of malicious code or the alteration of program execution flow. For an IoT device, this might mean hijacking a firmware update process to execute unauthorized commands. Mitigating buffer overflow vulnerabilities requires secure coding practices and memory protection mechanisms [84]. Developers should employ bounds checking and input validation to prevent excess data from overflowing memory buffers. The use of stack canaries—small security values placed on the stack to detect corruption—can help identify and block exploit attempts [83]. Additionally, firmware updates should be cryptographically signed and verified to ensure authenticity, preventing unauthorized modifications from being executed.

Another common vulnerability is weak authentication, where inadequate or hardcoded credentials—often a default username and password like "admin/admin"—grant attackers effortless entry. Once inside, they can manipulate the device's operations or siphon sensitive data, such as pairing keys for a Bluetooth connection. To counteract weak authentication, IoT devices should enforce strong password policies, mandating unique and complex credentials upon first use. Multi-factor authentication should be incorporated where feasible, adding an additional layer of security beyond just a password. Furthermore, device manufacturers should eliminate hardcoded credentials and instead implement dynamic, per-device authentication keys to prevent widespread exploitation from leaked credentials.

Insecure communication presents another major risk, where unencrypted or poorly encrypted data transmissions, for example, plaintext Bluetooth packets, expose information to interception. An attacker with a software-defined radio could eavesdrop these signals, reconstructing commands or user inputs without ever touching the device. Mitigating insecure communication requires end-to-end encryption using robust cryptographic protocols such as AES for data transmission and TLS for network communications [85]. Devices should avoid transmitting sensitive data in plaintext and implement secure key exchange mechanisms to protect encryption keys from interception. Additionally, enabling frequency-hopping spread spectrum techniques in wireless protocols can make passive eavesdropping significantly more difficult.

Physical access amplifies these risks, offering direct avenues to exploit software flaws. An attacker might connect the device to a computer or probe its pins to extract firmware from the microcontroller. In an ideal scenario, they retrieve the source code and configuration files, laying bare the device's logic. More often, they obtain only the machine code—the compiled binary. Yet, this is far from a dead end. Tools like Ghidra, IDA Pro, or a hexadecimal editor allow the attacker to dissect this code, tracing its instructions to map the device's functionality, from signal handling to security checks [86]. Sophisticated decompilation can even convert this binary into editable C code, opening the door to tailored modifications. With this insight, the attacker uncovers everything, including operational weaknesses, hidden features, or behaviors detrimental to the device's owner, for example, a tendency to broadcast identifiable data in the clear. To mitigate firmware extraction and reverse engineering, manufacturers should implement firmware encryption and readout protection mechanisms, preventing unauthorized access to microcontroller memory. Secure boot processes should be enforced, ensuring that only authenticated and untampered firmware can be executed on the device. Additionally, code obfuscation techniques can make reverse engineering significantly more challenging by disguising critical logic and security mechanisms.

Notably, the physical possession of the target device is not always necessary. IoT devices, often affordable and partially open source, invite indirect exploitation. An attacker could acquire an identical unit, extract its hexadecimal data, and reverse-engineer the firmware to mirror the target's behavior, While this may not reveal user-specific tweaks, for example, custom encryption keys, it exposes the core software architecture. Beyond mere understanding, this access enables the attacker to craft malicious alterations, such as embedding backdoors to monitor activity, adding logic for remote control, or masking their presence with stealth routines. These changes, often requiring just a handful of code lines, can be re-uploaded swiftly, turning the device into a compromised asset. To prevent malicious firmware modifications, secure firmware update mechanisms should be employed, requiring updates to be cryptographically signed and verified before installation. Rollback protection should also be in place to prevent attackers from downgrading devices to older, vulnerable firmware versions. Additionally, behavioral anomaly detection in IoT systems can help identify when a device begins exhibiting unexpected activity indicative of tampering.

Other vulnerabilities, like code injection, for example, via malformed inputs to a web interface, or privilege escalation, where an attacker exploits flawed permission checks, further broaden the attack surface, allowing attackers to escalate from limited access to full domination. In each case, the software's inherent programmability—its greatest strength—becomes the linchpin of its undoing, exposing IoT ecosystems to a multifaceted array of threats.

3.3. Network Vulnerabilities

The idea of a network can be broken down into two different subsets of networks. The typical network that is discussed is through the use of WiFi and Ethernet. This type of network has direct access to the Internet, allowing for the connection of devices through this route. The other type of network that is not commonly spoken about is the kind of network of other wireless signals. This can be a Bluetooth network, Zigbee network, UWB network, and so many more. This network is called a wireless sensor network as it typically includes many different wireless sensors, all with different wireless protocols. In this section, we will explore these two very similar but distinctly different networks and how vulnerabilities vary between the two.

3.3.1. WiFi and Ethernet Based Networks

WiFi- and Ethernet-based networks form the cornerstone of connectivity within the IoT ecosystem, linking devices through cloud platforms or localized home networks. This architecture empowers users to engage with IoT devices from virtually any location—imagine adjusting a smart thermostat from another continent. Such flexibility drives the seamless exchange of data across vast distances, yet it simultaneously exposes a critical vulnerability as follows: the broader the network's reach, the greater the opportunity for adversaries to intercept or disrupt communications. Among the most accessible threats are Denial of Service (DoS), Distributed Denial of Service (DDoS), and Energy-Oriented Distributed Denial of Service (E-DDoS) attacks, each exploiting the openness of these networks in distinct ways [87].

A typical DDoS attack, as illustrated in Figure 8, hinges on the attacker pinpointing the target device's IP address within the network. From there, they bombard it with an overwhelming volume of traffic—think a barrage of meaningless packets—aimed at saturating its bandwidth. The result is predictable; the network slows, resources divert, and eventually the system collapses, forcing the device offline [88]. Beyond this blunt disruption, E-DDoS attacks pursue a subtler goal as follows: exhausting the device's power reserves. Rather than outright crashing the network, these overload the hardware—potentially frying a PCB component or inflating energy costs—representing an insidious twist of the DoS paradigm [87]. Their commonality stems from their scattershot nature, often targeting entire device clusters rather than individual units.

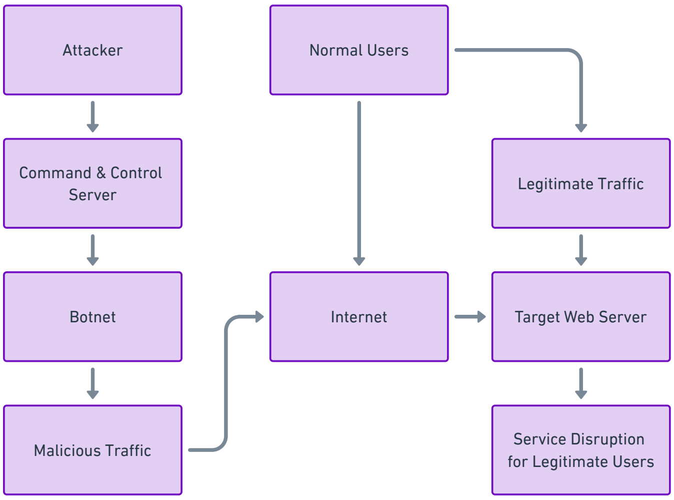

Figure 8. DDoS attack. This image has been completely re-modified; however, the idea of the image was inspired by [88].

Mitigating DoS and DDoS attacks requires network traffic filtering and rate limiting to detect and block malicious traffic patterns before they overwhelm a system. Deploying firewalls and intrusion detection systems (IDS) can help identify anomalous spikes in traffic, while blackholing and rate-limiting strategies can drop excess requests from untrusted sources. To counteract E-DDoS, IoT devices should incorporate power-aware intrusion detection, allowing them to recognize and mitigate excessive resource consumption attempts. Additionally, network segmentation can isolate critical devices from direct exposure, ensuring that a single attack does not cripple an entire IoT ecosystem.

As depicted in Figure 9, DoS-style attacks split into three categories, each exploiting a different network facet. Volume-based attacks flood protocols like TCP, ICMP, or UDP with raw data, clogging bandwidth and overwhelming internal queues. Protocol-based attacks—think SYN floods, Ping of Death, or Smurf attacks—target resource allocation. A SYN flood ties up connections by leaving them half-open, a Ping of Death delivers oversized packets to destabilize servers, and a Smurf attack amplifies traffic by echoing requests across all network nodes [87]. Then there are application-layer attacks, such as Slowloris, HTTP floods, or SMTP exploits. Slowloris sustains partial HTTP requests to monopolize server connections, HTTP floods swamp applications with legitimate-looking queries, and SMTP attacks probe mail servers for data leaks [87]. Each leverages the network's own mechanisms against it, turning openness into a liability.

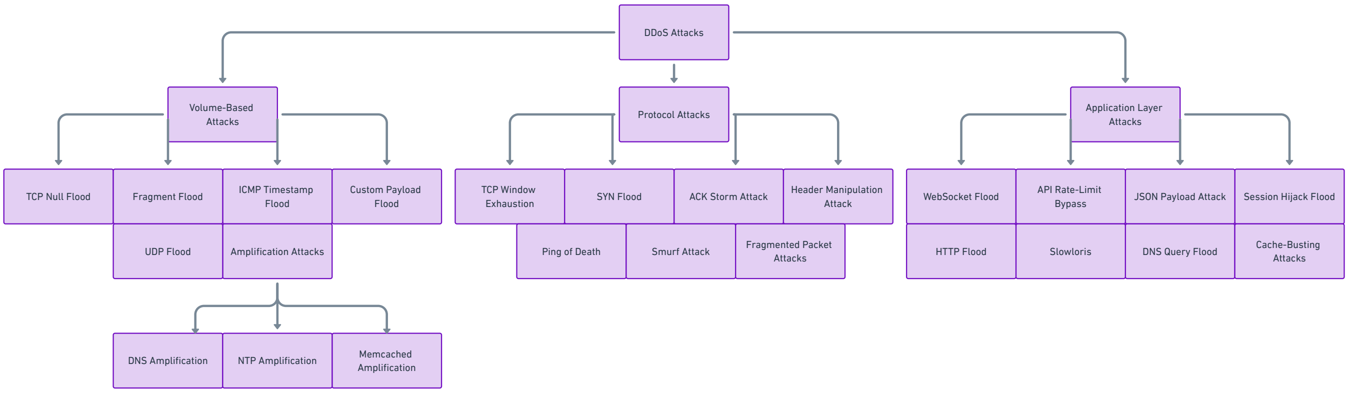

Figure 9. Types of attacks. This image has been completely re-modified; however, the idea of the image was inspired by [87].

These DoS variants, while pervasive, barely scratch the surface of network threats. Traffic monitoring attacks, like WiFi fingerprinting, offer a quieter menace. By analyzing a building's camera system bit-rate over WiFi, an attacker could infer internal movements—like tracking occupancy—without direct access [89]. These are not granular data, but they are enough to guess device roles or activities. More aggressive are packet analysis attacks, where tools like Wireshark or air sniffers capture TCP/IP traffic. These reveal communication patterns—who is talking, what is being sent—and, if unencrypted, the payload itself, like a smart bulb's status update [89]. The attacker could dissect these packets down to their bits, reconstructing them to extract detailed insights.

Mitigating traffic monitoring attacks demands end-to-end encryption for all transmitted data. Protocols like TLS for web traffic, WPA3 for WiFi security, and AES-based encryption for device communication can prevent eavesdropping. Implementing MAC address randomization can also reduce device fingerprinting risks. Additionally, firewalls with deep packet inspection (DPI) can monitor and block untrusted packets before they reach their destination.

Figure 10 outlines a sophisticated over-the-air attack, unfolding in two stages. The offline phase involves capturing traffic with a sniffer, preprocessing it and extracting features—packet intervals, sizes, or signal traits. These feed a machine learning model to profile network behavior and predict device types. In the online phase, the attacker actively eavesdrops, identifying live devices and cataloging WiFi-enabled units in range [89]. This methodical approach yields a detailed inventory without tripping alarms. Then there are network injection attacks, where tools like Kali Linux or a WiFi Pineapple inject spoofed packets or crack encryption—like brute-forcing WPA2 keys. Man-in-the-middle (MITM) attacks take it further, intercepting traffic to relay doctored messages or reroute data through malicious nodes, like a rogue DNS server [87].

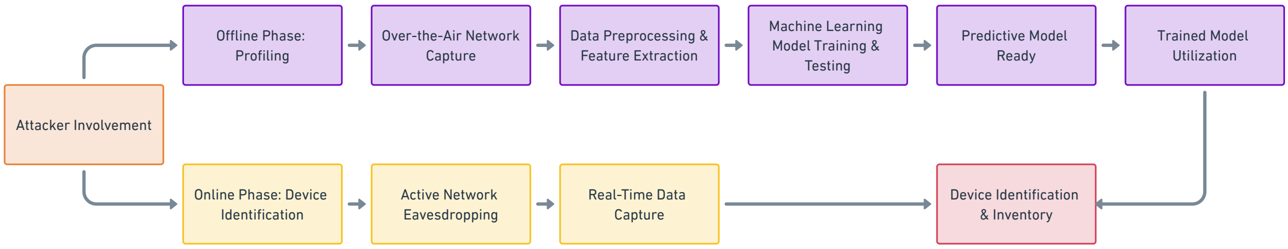
Offline Phase: Profiling
Over-the-Air Network Capture
Data Preprocessing & Feature Extraction
Machine Learning Model Training & Testing
Predictive Model Ready
Trained Model Utilization
Attacker Involvement
Online Phase: Device Identification
Active Network Eavesdropping
Real-Time Data Capture
Device Identification & Inventory

Figure 10. WiFi over-the-air attack. This image has been completely re-modified; however, the idea of the image was inspired by [89].

Defending against over-the-air attacks requires strong mutual authentication between IoT devices and their controllers. 802.1X authentication with certificate-based security ensures that only verified devices can participate in the network [90]. Additionally, regular key rotation for encrypted sessions prevents long-term eavesdropping. For network injection and MITM attacks, secure DNS (DNSSEC) and certificate pinning can validate legitimate servers, preventing redirection to malicious endpoints [91]. Enforcing firewall-based network segmentation can also reduce the spread of malicious injections.

The arsenal for these attacks is vast and accessible. Open source software—Wireshark for analysis, Aircrack-ng for cracking, or custom injectors—pairs with hardware like Raspberry Pi sniffers or Arduino jammers. Devices like the WiFi Pineapple streamline execution, with online guides detailing their construction [89]. What makes this landscape so treacherous is its evolution; as security tightens, attackers adapt, crafting new exploits to match. This relentless cycle ensures that WiFi and Ethernet networks remain a dynamic frontline in IoT security, where every connection is both a lifeline and a potential breach point. However, by implementing strong encryption, proactive monitoring, intrusion detection systems, and network segmentation, IoT devices and networks can fortify themselves against evolving threats, making it significantly harder for attackers to infiltrate and manipulate these environments.

3.3.2. Wireless Sensor Networks

A wireless sensor network (WSN) comprises spatially distributed sensors that transmit data wirelessly, forming the backbone of many IoT deployments. These networks span a diverse array of protocols—Zigbee, Z-Wave, GPS, 5G, and more—each tailored to specific sensing and communication needs. This versatility underpins the ubiquity of IoT devices, yet it also amplifies their exposure to attack. For clarity, this paper separates WiFi and Ethernet-based networks from WSNs, given the distinct breadth of vulnerabilities in the former. Within WSNs, Denial of Service (DoS) attacks present a significant threat, with their taxonomy delineated in Figure 11, encompassing a spectrum of tactics beyond those seen in traditional networking.

Figure 11 reveals a DoS framework tailored to WSNs, incorporating WiFi and Ethernet threats while introducing unique exploits. At the physical layer, jamming and node tampering reign as the most prevalent attacks. An adversary might jam a Zigbee network with interference, blocking nodes from transmitting or physically compromise a sensor to extract—or alter—its firmware [92]. Inserting a malicious node amplifies this, severing legitimate connections by drowning out signals. To mitigate jamming, frequency hopping spread spectrum (FHSS) and direct sequence spread spectrum (DSSS) can be implemented to make it harder for attackers to effectively disrupt communication [93,94]. Additionally, tamper-resistant hardware with physical shielding can reduce the risk of unauthorized access to sensors.

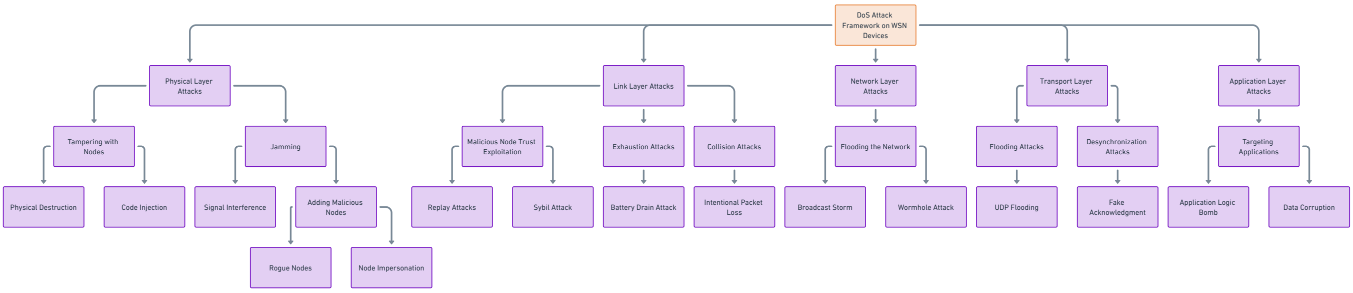

Figure 11. Wireless sensor networks DoS attacks. This image has been completely re-modified; however, the idea of the image was inspired by [92].

Link-layer attacks shift to subtler sabotage; a rogue node might masquerade as trustworthy, exhaust neighbors with incessant requests, or induce collisions to crash nodes via errors [92]. At the network layer, flooding mirrors WiFi-style drown outs, overwhelming routing paths with traffic. The transport layer hosts flooding—swamping nodes with fake peers—and desynchronization, where message timing is scrambled, leaving nodes out of sync [92]. These attacks can be mitigated by implementing rate limiting, packet authentication, and anomaly detection algorithms that identify excessive traffic patterns or suspicious communication behavior.

Beyond DoS, WSNs face routing-centric threats like the black hole attack, depicted in Figure 12. Here, a malicious node lures others to redirect data through it, acting as a sink that absorbs—or discards—transmissions, halting network flow [95,96]. Closely related to this is the wormhole attack, shown in Figure 13, where data are tunneled between two colluding nodes—say, from node A to node B—bypassing the intended routes [97,98]. This rerouting distorts network topology, granting attackers control over data paths.

To mitigate black hole and wormhole attacks, secure routing protocols such as Secure Efficient Ad hoc Distance vector (SEAD) and Ad hoc On-Demand Distance Vector (AODV) with authentication mechanisms can help verify node identities before allowing data redirection [99–101]. Time-based packet verification methods can also be implemented to detect unusual delays indicative of wormhole behavior.

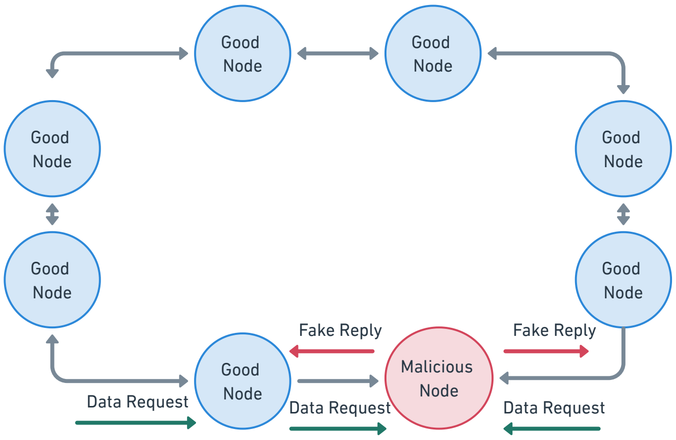

Figure 12. Blackhole attack. This image has been completely re-modified; however, the idea of the image was inspired by [102].

Figure 13 illustrates how wormhole nodes—A and B—siphon data through a tunnel, manipulating its flow. This attack manifests in four modes as follows: Packet encapsulation compresses data between nodes, evading hop-count increments typical in WSN routing [98]. Packet relay mode uses any node as a launch point, relaying traffic to disrupt paths [98]. Out-of-band channel attacks rely on a single high-power node to redirect packets via an external link [98]. Lastly, protocol distortion mode alters routing rules to attract traffic to malicious nodes [98]. Each mode exploits WSNs' reliance on decentralized coordination, turning trust into a weapon.

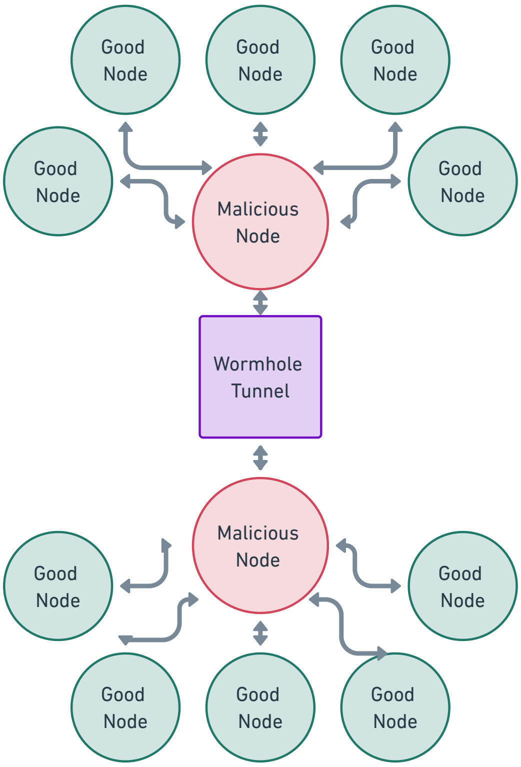

Figure 13. Wormhole attack tunnel. This image has been completely re-modified; however, the idea of the image was inspired by [98].

Additional threats include eavesdropping attacks, where sniffers—prebuilt for UWB, Zigbee, Z-Wave, or Bluetooth—capture packets over the air. Open source tools for Arduino or Raspberry Pi extend this capability to any protocol, decoding transmissions to reveal data like GPS coordinates or sensor readings [92]. Sybil attacks introduce fake node identities, tricking the network into routing through them, while sinkhole attacks lure traffic to a compromised node to manipulate or discard it [95]. These join traffic analysis attacks, where signal patterns—like a 5G node's burst rate—hint at activity without decrypting payloads [97]. Each leverages WSNs' wireless nature, where every transmission is a potential leak.

To mitigate these advanced threats, encrypted communication protocols such as AES-based encryption for data packets should be enforced. Identity-based authentication can prevent Sybil attacks by requiring node verification. For eavesdropping threats, regularly changing encryption keys and employing physical-layer security techniques such as directional antennas and shielding can reduce interception risks.

The tools for such exploits mirror WiFi's ecosystem; commercial sniffers abound, and DIY options thrive on open source code. An attacker could wield a Zigbee sniffer to log packets or craft a Raspberry Pi rig to jam Z-Wave signals. This accessibility, paired with WSNs' sprawling attack surface, ensures that vulnerabilities evolve alongside defenses, rendering these networks a persistent challenge in IoT security. However, by integrating encryption, anomaly detection, secure routing, and frequency-hopping mechanisms, WSNs can enhance their resilience against these evolving attacks.

3.3.3. Cloud Based Networks

When delving into network vulnerabilities within the Internet of Things (IoT), one must recognize the transformative role of the IoT cloud computing architecture—a recent and vital player in this domain. As standalone IoT technology struggles to meet the growing demands of users and their computational needs, the integration of IoT into cloud computing resources emerges as a powerful solution. This synergy combines the sensing and connectivity of IoT devices with the scalability and processing power of the cloud, as illustrated in Figure 14. While this union enables remarkable capabilities—imagine accessing a smart thermostat's data from across the globe—it also introduces significant security risks. Numerous studies thoroughly catalog IoT cloud vulnerabilities, so this section focuses on the contemporary Phantom-Delay Attack and expands to other pertinent threats, each highlighting the precarious balance of this hybrid ecosystem.

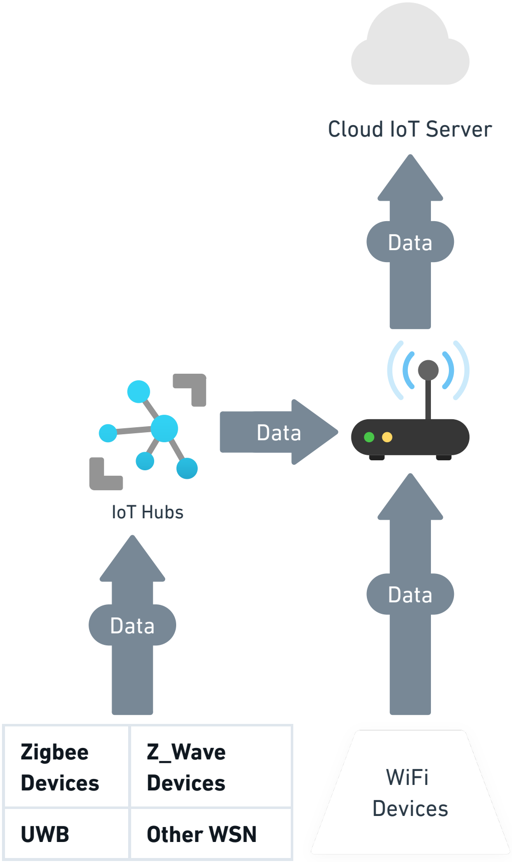

Figure 14. System model of a typical smart cloud-based home deployment. This image has been completely re-modified; however, the idea of the image was inspired by [103].

Phantom-delay attacks, a recently uncovered vulnerability, exploit the timing of communications between IoT devices and cloud servers. Unlike conventional disruptions, these attacks manipulate message delivery without discarding packets, leading to severe consequences [103]. These hinge on the following two primitives: IoT Event Message Delay (e-Delay), delaying device-to-server state updates (for example, a "motion detected" alert), and IoT Command Message Delay (c-Delay), stalling server-to-device instructions (for example, "lock the door"). Typically, transmission delays are fleeting—sub-second lags that pose no issue. Yet, a phantom-delay attack, depicted in Figure 15, extends these into minutes or hours. The cloud server, receiving a delayed "motion active" event, assumes it reflects the device's current state, acting on outdated information [104].

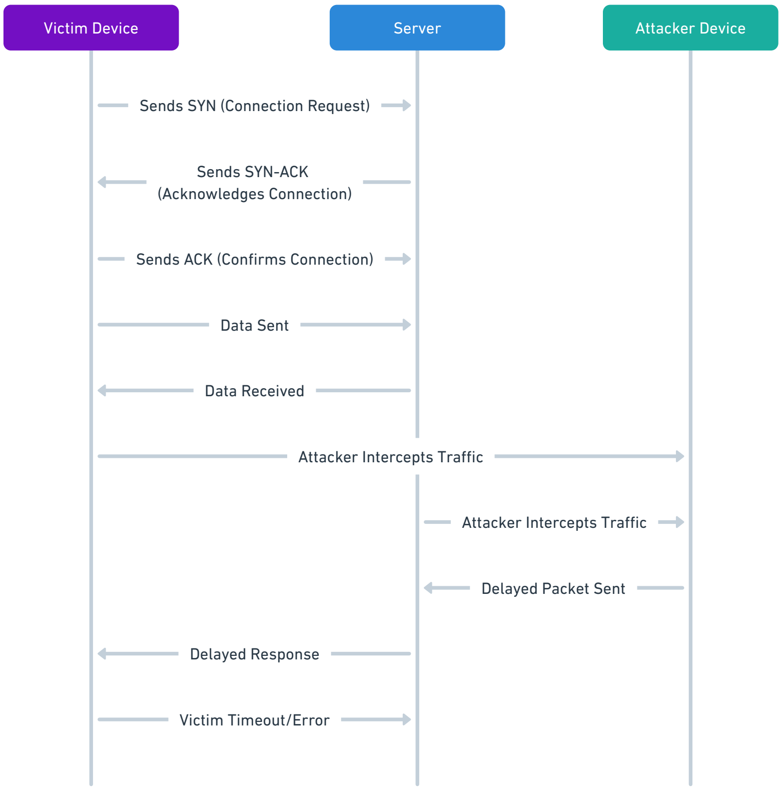
Victim Device
Server
Attacker Device
Sends SYN (Connection Request)
Sends SYN-ACK (Acknowledges Connection)
Sends ACK (Confirms Connection)
Data Sent
Data Received
Attacker Intercepts Traffic
Attacker Intercepts Traffic
Delayed Packet Sent
Delayed Response
Victim Timeout/Error

Figure 15. Phantom-delay attack. This image has been completely re-modified; however, the idea of the image was inspired by [103].

These attacks manifest in three forms. The state-update delay attack employs e-Delay to stall critical state reports—consider a medical IoT device (IoMT) or agricultural sensor lagging in notifying the server of urgent shifts, jeopardizing timely intervention [103]. The action-delay attack disrupts automation; an e-Delay might postpone an insulin pump's response to a glucose spike detected by a continuous monitor, risking health outcomes. Finally, the erroneous execution attack splits into two subtypes as follows: spurious execution, where a delayed event keeps a condition "true" past its validity, prompting unneeded actions (for example, a sprinkler activating despite ample rain); and disabled execution, where a delay holds a condition as "false", preventing necessary actions (for example, a door failing to lock despite a command) [104]. Unlike jamming, these attacks—launched from a WiFi device sniffing traffic and hijacking TCP sessions via ARP spoofing—leave no trace of dropped packets, dodging retransmission flags [105].

To mitigate phantom-delay attacks, cloud-based IoT frameworks should incorporate timestamp verification in all message exchanges, ensuring that outdated or delayed data are flagged as invalid. Secure time synchronization protocols, such as the Network Time Protocol (NTP) with cryptographic authentication, can help detect discrepancies in the expected message's timing. Additionally, introducing anomaly detection mechanisms that monitor latency variations can preemptively identify these delays, prompting system alerts or requiring revalidation before acting on stale data [106–108].

Beyond phantom delay, data injection attacks pose a stealthy threat. An adversary might feed falsified inputs—say, spoofed temperature readings—into the cloud, tricking it into erroneous decisions, like shutting down a smart HVAC system [103]. Session hijacking exploits weak authentication, such as unsecured API keys or stolen credentials, letting attackers impersonate devices and issue rogue commands—perhaps unlocking a smart door remotely. Resource exhaustion attacks target cloud endpoints, flooding them with false requests to overwhelm processing capacity, distinct from bandwidth-focused DDoS, targeting server-side logic [103]. Then there is traffic manipulation, where an attacker intercepts and alters data in transit—imagine tweaking a smart meter's usage report to inflate bills—all while the cloud assumes legitimacy [103].

To counteract data injection threats, implementing cryptographic message authentication codes (MACs) ensures that only legitimate IoT devices can submit data to the cloud. Secure firmware updates with digital signatures help prevent unauthorized modifications that might enable spoofed sensor readings. In mitigating session hijacking, enforcing multi-factor authentication (MFA) for cloud access and regularly rotating API keys reduces exposure to credential theft. Additionally, role-based access control (RBAC) minimizes the impact of compromised accounts by restricting device privileges.

For resource exhaustion and traffic manipulation attacks, rate limiting on API endpoints prevents abuse by restricting excessive requests from any single source. Encryption protocols such as TLS prevent in-transit data tampering, ensuring the integrity of communications. Advanced intrusion detection systems (IDS) equipped with behavioral analysis can identify unusual traffic patterns indicative of an ongoing attack, enabling swift countermeasures.

These vulnerabilities take advantage of the cloud's reliance on continuous IoT interactions. Using readily available tools such as WiFi sniffers, Kali Linux suites, or a Raspberry Pi configured for ARP spoofing, adversaries can easily launch attacks. The very feature that makes the cloud powerful—its ability to integrate diverse devices—also makes it susceptible, magnifying the consequences of any breach. As IoT and cloud technologies become increasingly interconnected, these risks highlight a crucial reality as follows: greater connectivity necessitates equally strong security measures to prevent the system's own architecture from becoming its greatest vulnerability.

4. Significance of Our Paper

Several recent studies have explored the security challenges associated with the Internet of Things (IoT), highlighting various vulnerabilities, attack vectors, and mitigation strategies. However, our research distinguishes itself through its depth of analysis, comprehensive categorization of vulnerabilities, and a unique focus on countermeasure implementation across multiple IoT layers. The following comparisons illustrate how our study builds upon and extends prior research in the field.

Taherdoost and colleagues [109] provide an overview of the benefits and challenges of IoT security, emphasizing network security and data integrity. Their research adopts a qualitative review methodology, synthesizing the existing literature to highlight emerging trends and challenges in IoT security. While their study discusses the importance of authentication and authorization, our research takes a more technical approach, incorporating the empirical analysis of specific attack methodologies, including radio frequency (RF) attacks, software vulnerabilities, and hardware Trojan threats. Furthermore, we present a more detailed taxonomy of IoT security risks, covering real-world attack scenarios and mitigation techniques beyond traditional encryption and access control mechanisms.

Fei and colleagues [110] conducted a systematic literature review, identifying research potential and future directions in IoT security. Their methodology involves analyzing a broad set of academic sources to categorize security threats and survey existing solutions, particularly focusing on AI-driven intrusion detection and blockchain-based security measures. While their study provides an extensive theoretical foundation, our research not only categorizes these vulnerabilities but also presents a hands-on analysis of firmware reverse engineering, hardware implants, and cloud-based attack strategies. Additionally, we introduce a layered security framework that highlights vulnerabilities and solutions at the perception, network, and application layers of IoT architecture.

Aslan and collaborators [111] offer a broad review of cybersecurity vulnerabilities, threats, and mitigation strategies, primarily focusing on general cybersecurity rather than IoT-specific threats. Their research employs a comparative analysis methodology, evaluating various cybersecurity frameworks across multiple domains. Our study expands upon their work by providing a specialized analysis of IoT networks, including wireless sensor network (WSN) vulnerabilities and IoT-specific denial-of-service (DoS) attacks. Furthermore, while their work remains largely theoretical, we discuss advanced attack techniques such as phantom-delay attacks and how adversaries exploit timing inconsistencies in cloud-based IoT deployments, incorporating empirical findings.

Sun and research team [112] investigate IoT privacy security concerns, employing a mixed-methods approach that integrates both qualitative literature reviews and quantitative evaluations of blockchain-based security frameworks. While their study addresses privacy issues, our research places greater emphasis on hardware and firmware vulnerabilities, including side-channel attacks, JTAG-based exploits, and signal interception techniques. By incorporating case studies and empirical attack simulations, we provide a more hands-on perspective on IoT security risks, bridging the gap between theoretical security principles and real-world attack scenarios.

Ul Haq and co-authors [113] focus on embedded device firmware security, employing a technical survey methodology that reviews extraction techniques and vulnerability analysis frameworks. Our research builds upon this work by incorporating practical reverse engineering scenarios, demonstrating how firmware vulnerabilities can be exploited to gain unauthorized access to IoT devices. Additionally, we propose a multilayered approach to secure IoT firmware, which includes secure boot mechanisms and the real-time monitoring of firmware integrity, combining both theoretical and experimental security assessments.

Siwakoti and contributors [114] discuss IoT vulnerabilities and criminal services enabled by IoT exploitation. Their research primarily uses a threat modeling approach, classifying attack patterns and evaluating their potential impact on IoT ecosystems. While their study highlights botnets and ransomware attacks targeting IoT devices, we extend this work by presenting mitigation techniques specific to RF jamming, protocol spoofing, and network-layer man-in-the-middle (MITM) attacks. Moreover, our work includes a detailed breakdown of attack methodologies that exploit IoT device hardware, such as micro-probing and hardware implants, reinforcing our empirical approach.

Noman and Abu-Sharkh [115] provide a comprehensive review of code injection attacks in wireless IoT networks. Their research is primarily experimental, involving the implementation and testing of various code injection attack techniques. While this study focuses on injection techniques, our research covers a broader range of threats, including physical-layer attacks, radio signal interception, and over-the-air firmware exploitation. By integrating multiple attack perspectives, we provide a more holistic understanding of IoT security risks and their countermeasures, adding a comparative dimension that contrasts real-world attack success rates.

AlSalem and colleagues [116] analyze cybersecurity risks in IoT from an economic impact perspective, using a risk assessment methodology to quantify the financial consequences of IoT security breaches. Although their study is valuable in understanding the financial implications of cyber threats, our research shifts the focus toward technical vulnerabilities and mitigation frameworks. We explore adversarial techniques such as brute-force RF attacks, hardware tampering, and signal timing exploits, providing a technical blueprint for strengthening IoT security defenses through both theoretical analysis and practical attack demonstrations.

Alqarawi and co-authors [117] present a case study on IoT security and vulnerabilities, employing a case-study-driven methodology that examines real-world implementations of IoT security frameworks. While their work is useful in assessing IoT risks in applied environments, our study introduces a more structured classification of the threats, detailing their technical execution and defense mechanisms. Our inclusion of hardware reverse engineering, cryptographic failures, and cloud-based IoT exploitation makes our research a more technical and detailed contribution to the field, with an emphasis on replicable security testing.

Aziz and research partners [118] discuss challenges and mitigation techniques for securing IoT devices, utilizing a comparative analysis of the existing security frameworks. Their work primarily focuses on securing IoT communications and authentication mechanisms. Our study extends beyond these topics to address firmware attacks, hardware vulnerabilities, and advanced network exploitation tactics. By integrating both theoretical and practical insights, we provide a more actionable guide for securing IoT environments against emerging threats, supplemented by real-world attack demonstrations.

Overall, while prior research has explored various aspects of IoT security, our study stands out by offering a multilayered, technically detailed, and practically relevant analysis of IoT vulnerabilities. We provide a unique focus on hardware, firmware, and radio frequency threats, backed by real-world attack demonstrations and countermeasure recommendations. Through this work, we aim to bridge the gap between theoretical cybersecurity research and practical IoT security implementations, offering a comprehensive and forward-looking perspective on securing IoT ecosystems.

5. Conclusions and Future Research to Be Worked

As the Internet of Things continues to expand into various sectors, the security challenges it presents are becoming more apparent and pressing. This study has demonstrated that IoT systems are inherently vulnerable across multiple layers, including hardware, software, networks, and cloud infrastructure. These vulnerabilities create opportunities for attackers to exploit IoT devices, leading to potential data breaches, system disruptions, and the unauthorized control of critical operations. The growing reliance on IoT highlights the urgent need to address these security concerns through proactive defense mechanisms and structured mitigation strategies.

The analysis conducted in this research emphasizes the necessity of strengthening IoT security by implementing best practices such as secure firmware updates, robust authentication methods, encrypted communication channels, and continuous monitoring for anomalies. By adopting these security measures, the risk of exploitation can be significantly reduced, ensuring the integrity and reliability of IoT ecosystems. Additionally, this study underscores the importance of ongoing research and development in the field of IoT security to counteract emerging threats and adapt to the evolving landscape of cyber attacks.

As IoT technology continues to evolve and integrate further into critical infrastructures, the responsibility to secure these systems must remain a top priority. Future research should focus on refining existing security measures, exploring innovative solutions to mitigate new attack vectors, and enhancing the resilience of IoT networks against sophisticated threats. By proactively addressing these challenges, we as a community can work toward building a more secure and trustworthy IoT environment that can sustain the demands of modern technology and its widespread applications.

Author Contributions: Conceptualization, I.C. and E.P.; Writing—original draft, I.C. and E.P.; Writing—review & editing, I.C. and E.P.; Supervision, M.N. 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: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author.

Acknowledgments: The authors would like to extend their gratitude to Eadan Plotnizky for their contribution to the research efforts and for writing/editing this article. Finally, we thank the anonymous reviewers for their constructive feedback and inspiring comments. The invaluable comments of the reviewers have significantly improved this document.

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

Abbreviations

A-fib	Atrial Fibrillation
AC	Air Conditioning
AES	Advanced Encryption Standard
AODV	Ad Hoc On-Demand Distance Vector Routing Protocol
AMQP	Advanced Message Queuing Protocol
API	Application Programming Interface
ARP	Address Resolution Protocol
BLE	Bluetooth Low Energy
bps	Bits per second
CCTV	Closed-Circuit Television
c-Delay	IoT Command Message Delay
CGM	Continuous Glucose Monitor
CoAP	Constrained Application Protocol
CoRE	Constrained RESTful Environments
CPS	Cyber-Physical Systems
D7AP	Dash7 Alliance Protocol
DA	Data Access
DDoS	Distributed Denial of Service
DDS	Data Distribution Service
DHCP	Dynamic Host Configuration Protocol
DH	Diffie–Hellman
DSSS	Direct Sequence Spread Spectrum
DNS	Domain Name System
DNSSEC	Domain Name System Security Extensions
DoS	Denial of Service
DTLS	Datagram Transport Layer Security
E-DDoS	Energy-Oriented Distributed Denial of Service
e-Delay	IoT Event Message Delay
ECG	Electrocardiogram
ECDH	Elliptic Curve Diffie–Hellman
ECDSA	Elliptic Curve Digital Signature Algorithm
eMBB	Enhanced Mobile Broadband
FAU	Florida Atlantic University
FHSS	Frequency Hopping Spread Spectrum
GHz	Gigahertz
Gbps	Gigabits per second
GPS	Global Positioning System
HTTP	Hypertext Transfer Protocol
HVAC	Heating, Ventilation, and Air Conditioning
IC	Integrated Circuit

ICMP	Internet Control Message Protocol
IEC	International Electrotechnical Commission
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IIoT	Industrial Internet of Things
IDS	Intrusion Detection System
IO	Input/Output
IoAT	Internet of Agricultural Things
IoBT	Internet of Battlefield Things
IoMT	Internet of Medical Things
IoT	Internet of Things
IP	Internet Protocol
IPv6	Internet Protocol version 6
ISM	Industrial, Scientific, and Medical
ISO	International Organization for Standardization
JTagulator	JTAG Analysis Tool
kbps	Kilobits per second
kbit/s	Kilobits per second
KHz	Kilohertz
LAN	Local Area Network
LED	Light-Emitting Diode
Li-Fi	Light Fidelity
LoRa	Long Range
LoRaWAN	Long Range Wide Area Network
LPWAN	Low-Power Wide Area Network
LTE	Long-Term Evolution
LTE-A	Long-Term Evolution Advanced
M2M	Machine-to-Machine
MAC	Message Authentication Code
Mbps	Megabits per second
MHz	Megahertz
MITM	Man-in-the-Middle
ms	Milliseconds
MQTT	Message Queuing Telemetry Transport Protocol
NB-IoT	Narrowband Internet of Things
NFC	Near-Field Communication
OMG	Object Management Group
OPC	Open Platform Communications
PCB	Printed Circuit Board
QoS	Quality of Service
RBAC	Role-Based Access Control
RC4	Rivest Cipher 4
REST	Representational State Transfer
RF	Radio Frequency
RTL-SDR	Realtek Software-Defined Radio
SAE	Simultaneous Authentication of Equals
SASL	Simple Authentication and Security Layer
SDR	Software-Defined Radio
SEAD	Secure Efficient Ad Hoc Distance Vector
SMTP	Simple Mail Transfer Protocol
SPO2	Peripheral Capillary Oxygen Saturation
SSL	Secure Sockets Layer
SYN	Synchronize
TCP	Transmission Control Protocol

TLS	Transport Layer Security
UA	Unified Architecture
UAV	Unmanned Aerial Vehicle
UDP	User Datagram Protocol
URLLC	Ultra-Reliable Low-Latency Communication
UWB	Ultra-Wideband
V2I	Vehicle-to-Infrastructure
V2V	Vehicle-to-Vehicle
VLC	Visible Light Communication
WEP	Wired Equivalent Privacy
WiFi	Wireless Fidelity
WPA	Wi-Fi Protected Access
WPA2	Wi-Fi Protected Access 2
WPA3	Wi-Fi Protected Access 3
WSAN	Wireless Sensor and Actuator Network
WSN	Wireless Sensor Network
XOR	Exclusive OR
Z-Wave	Z-Wave

References

1. Pico-Valencia, P.; Holgado-Terriza, J.A.; Quiñónez-Ku, X. A Brief Survey of the Main Internet-Based Approaches. An Outlook from the Internet of Things Perspective. In Proceedings of the 2020 3rd International Conference on Information and Computer Technologies (ICICT), San Jose, CA, USA, 9–12 March 2020; pp. 536–542. [CrossRef]
2. Al-Fuqaha, A.; Guizani, M.; Mohammadi, M.; Aledhari, M.; Ayyash, M. Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications. *IEEE Commun. Surv. Tutorials* **2015**, *17*, 2347–2376. [CrossRef]
3. Singh, D.; Tripathi, G.; Jara, A.J. A survey of Internet-of-Things: Future vision, architecture, challenges and services. In Proceedings of the 2014 IEEE World Forum on Internet of Things (WF-IoT), Seoul, Republic of Korea, 6–8 March 2014; pp. 287–292. [CrossRef]
4. Zhong, C.L.; Zhu, Z.; Huang, R.G. Study on the IOT Architecture and Gateway Technology. In Proceedings of the 2015 14th International Symposium on Distributed Computing and Applications for Business Engineering and Science (DCABES), Guiyang, China, 18–24 August 2015; pp. 196–199. [CrossRef]
5. Arasteh, H.; Hosseinnezhad, V.; Loia, V.; Tommasetti, A.; Troisi, O.; Shafie-khah, M.; Siano, P. Iot-based smart cities: A survey. In Proceedings of the 2016 IEEE 16th International Conference on Environment and Electrical Engineering (EEEIC), Florence, Italy, 7–10 June 2016; pp. 1–6. [CrossRef]
6. Zanella, A.; Bui, N.; Castellani, A.; Vangelista, L.; Zorzi, M. Internet of Things for Smart Cities. *IEEE Internet Things J.* **2012**, *1*, 22–32. [CrossRef]
7. Sisinni, E.; Saifullah, A.; Han, S.; Jennehag, U.; Gidlund, M. Industrial Internet of Things: Challenges, Opportunities, and Directions. *IEEE Trans. Ind. Inform.* **2018**, *14*, 4724–4734. [CrossRef]
8. Sivapriyan, R.; Rao, K.M.; Harijyothi, M. Literature Review of IoT based Home Automation System. In Proceedings of the 2020 Fourth International Conference on Inventive Systems and Control (ICISC), Coimbatore, India, 8–10 January 2020; pp. 101–105. [CrossRef]
9. CardiacSense. Heart Rate Monitor Watch. 2023. Available online: https://www.cardiacsense.com/heart-rate-monitor-watch/ (accessed on 14 June 2023).
10. Islam, S.M.R.; Kwak, D.; Kabir, M.H.; Hossain, M.; Kwak, K. The Internet of Things for Health Care: A Comprehensive Survey. *IEEE Access* **2015**, *3*, 678–708. [CrossRef]
11. Farooq, M.S.; Riaz, S.; Abid, A.; Abid, K.; Naeem, M.A. A Survey on the Role of IoT in Agriculture for the Implementation of Smart Farming. *IEEE Access* **2019**, *7*, 156237–156271. [CrossRef]
12. SeeTree. About Us. 2017.Available online: https://www.seetree.ai/about-seetree (accessed on 19 September 2023).
13. Shachar, O.; Yushchuk, M.; Salton-Morgenstern, G. Recurrent Pattern Image Classification and Registration. U.S. Patent No. 10,546,216, 28 January 2020.
14. Kott, A.; Swami, A.; West, B.J. The Internet of Battle Things. *Computer* **2016**, *49*, 70–75. [CrossRef]
15. Russell, S.; Abdelzaher, T. The Internet of Battlefield Things: The Next Generation of Command, Control, Communications and Intelligence (C3I) Decision-Making. In Proceedings of the MILCOM 2018—2018 IEEE Military Communications Conference (MILCOM), Los Angeles, CA, USA, 29–31 October 2018; pp. 737–742. [CrossRef]

16. Farahani, S. *ZigBee Wireless Networks and Transceivers*; Newnes: Newton, MA, USA, 2011.
17. Ergen, S.C. ZigBee/IEEE 802.15.4 Summary. *UC Berkeley Sept.* **2004**, *10*, 11.
18. Elahi, A.; Gschwender, A. *ZigBee Wireless Sensor and Control Network*; Pearson Educ.: London, UK, 2009.
19. Norair, J. Introduction to DASH7 technologies. In *Dash7 Alliance Low Power RF Technical Overview*; DASH7: Aberdeen, Scotland, 2009; pp. 1–22.
20. Piromalis, D.; Arvanitis, K.; Sigrimis, N. DASH7 mode 2: A promising perspective for wireless agriculture. *IFAC Proc. Vol.* **2013**, *46*, 127–132. [CrossRef]
21. Ayoub, W.; Samhat, A.E.; Nouvel, F.; Mroue, M.; Prévotet, J.C. Internet of Mobile Things: Overview of LoRaWAN, DASH7, and NB-IoT in LPWANs Standards and Supported Mobility. *IEEE Commun. Surv. Tutorials* **2019**, *21*, 1561–1581. [CrossRef]
22. Czyz, J.; Luckie, M.J.; Allman, M.; Bailey, M. Do not forget to lock the back door! A characterization of IPv6 network security policy. In Proceedings of the NDSS, San Diego, CA, USA, 21–24 February 2016.
23. Lashkari, A.H.; Danesh, M.M.S.; Samadi, B. A survey on wireless security protocols (WEP, WPA and WPA2/802.11 i). In Proceedings of the 2009 2nd IEEE International Conference on Computer Science and Information Technology, Beijing, China, 8–11 August 2009; pp. 48–52.
24. Kohlios, C.P.; Hayajneh, T. A comprehensive attack flow model and security analysis for Wi-Fi and WPA3. *Electronics* **2018**, *7*, 284. [CrossRef]
25. Pahlavan, K.; Krishnamurthy, P. Evolution and impact of Wi-Fi technology and applications: A historical perspective. *Int. J. Wirel. Inf. Netw.* **2021**, *28*, 3–19. [CrossRef]
26. Banerji, S.; Chowdhury, R.S. On IEEE 802.11: Wireless Lan Technology. *Int. J. Mob. Netw. Commun. Telemat.* **2013**, *3*, 45–64. [CrossRef]
27. Ezhilarasan, E.; Dinakaran, M. A review on mobile technologies: 3G, 4G and 5G. In Proceedings of the 2017 Second International Conference on Recent Trends and Challenges in Computational Models (ICRTCCM), Tindivanam, Tamilnadu, 3–4 February 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 369–373.
28. Akyildiz, I.F.; Gutierrez-Estevez, D.M.; Reyes, E.C. The evolution to 4G cellular systems: LTE-Advanced. *Phys. Commun.* **2010**, *3*, 217–244. [CrossRef]
29. Zhang, Y.; Årvidsson, A. Understanding the characteristics of cellular data traffic. In Proceedings of the 2012 ACM SIGCOMM Workshop on Cellular Networks: Operations, Challenges, and Future Design, Helsinki, Finland, 13 August 2012; pp. 13–18.
30. Chettri, L.; Bera, R. A comprehensive survey on Internet of Things (IoT) toward 5G wireless systems. *IEEE Internet Things J.* **2019**, *7*, 16–32. [CrossRef]
31. Zeqiri, R.; Idrizi, F.; Halimi, H. Comparison of Algorithms and Technologies 2G, 3G, 4G and 5G. In Proceedings of the 2019 3rd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT), Ankara, Turkey, 11–13 October 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 1–4.
32. Shelby, Z.; Bormann, C. *6LoWPAN: The Wireless Embedded Internet*; John Wiley & Sons: Hoboken, NJ, USA, 2011.
33. Mulligan, G. The 6LoWPAN architecture. In Proceedings of the 4th Workshop on Embedded Networked Sensors, Cork, Ireland, 25–26 June 2007; pp. 78–82.
34. Baker, N. ZigBee and Bluetooth: Strengths and weaknesses for industrial applications. *Comput. Control. Eng.* **2005**, *16*, 20–25. [CrossRef]
35. Bisdikian, C. An overview of the Bluetooth wireless technology. *IEEE Commun. Mag.* **2001**, *39*, 86–94. [CrossRef]
36. Tosi, J.; Taffoni, F.; Santacatterina, M.; Sannino, R.; Formica, D. Performance evaluation of bluetooth low energy: A systematic review. *Sensors* **2017**, *17*, 2898. [CrossRef] [PubMed]
37. Haxhibeqiri, J.; De Poorter, E.; Moerman, I.; Hoebeke, J. A survey of LoRaWAN for IoT: From technology to application. *Sensors* **2018**, *18*, 3995. [CrossRef]
38. Khutsoane, O.; Isong, B.; Abu-Mahfouz, A.M. IoT devices and applications based on LoRa/LoRaWAN. In Proceedings of the IECON 2017—43rd Annual Conference of the IEEE Industrial Electronics Society, Beijing, China, 29 October 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 6107–6112.
39. Lavric, A.; Petrariu, A.I.; Popa, V. Long range sigfox communication protocol scalability analysis under large-scale, high-density conditions. *IEEE Access* **2019**, *7*, 35816–35825. [CrossRef]
40. Fourtet, C.; Ponsard, B. An introduction to Sigfox radio system. In *LPWAN Technologies for IoT and M2M Applications*; Elsevier: Amsterdam, The Netherlands, 2020; pp. 103–118.
41. Alqurashi, H.; Bouabdallah, F.; Khairullah, E. SCAP SigFox: A Scalable Communication Protocol for Low-Power Wide-Area IoT Networks. *Sensors* **2023**, *23*, 3732. [CrossRef]
42. Ratasuk, R.; Vejlgaard, B.; Mangalvedhe, N.; Ghosh, A. NB-IoT system for M2M communication. In Proceedings of the 2016 IEEE Wireless Communications and Networking Conference, Doha, Qatar, 3–6 April 2016; pp. 1–5. [CrossRef]

43. Coskun, V.; Ok, K.; Ozdenizci, B. *Near Field Communication (NFC): From Theory to Practice*; John Wiley & Sons: Hoboken, NJ, USA, 2011.
44. Coskun, V.; Ozdenizci, B.; Ok, K. A survey on near field communication (NFC) technology. *Wirel. Pers. Commun.* **2013**, *71*, 2259–2294. [CrossRef]
45. Danbatta, S.J.; Varol, A. Comparison of Zigbee, Z-Wave, Wi-Fi, and bluetooth wireless technologies used in home automation. In Proceedings of the 2019 7th International Symposium on Digital Forensics and Security (ISDFS), Barcelos, Portugal, 10–12 June 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 1–5.
46. Sapundzhi, F. Home automation based on Z-wave technology. *Bulg. Chem. Commun.* **2022**, *54*, 92–96.
47. Bhavya, R.; Lokesh, M. A Survey on Li-Fi Technology. *Int. J. Eng. Technol.* **2016**, *3*, 1624–1625.
48. Haas, H.; Yin, L.; Wang, Y.; Chen, C. What is LiFi? *J. Light. Technol.* **2016**, *34*, 1533–1544. [CrossRef]
49. Zhuang, W.; Shen, X.; Bi, Q. Ultra-wideband wireless communications. *Wirel. Commun. Mob. Comput.* **2003**, *3*, 663–685. [CrossRef]
50. Hirt, W. Ultra-wideband radio technology: Overview and future research. *Comput. Commun.* **2003**, *26*, 46–52. [CrossRef]
51. Aiello, G.; Rogerson, G. Ultra-wideband wireless systems. *IEEE Microw. Mag.* **2003**, *4*, 36–47. [CrossRef]
52. Naik, N. Choice of effective messaging protocols for IoT systems: MQTT, CoAP, AMQP and HTTP. In Proceedings of the 2017 IEEE International Systems Engineering Symposium (ISSE), Vienna, Austria, 11–13 October 2017; pp. 1–7. [CrossRef]
53. Fernandes, J.L.; Lopes, I.C.; Rodrigues, J.J.; Ullah, S. Performance evaluation of RESTful web services and AMQP protocol. In Proceedings of the 2013 Fifth International Conference on Ubiquitous and Future Networks (ICUFN), Da Nang, Vietnam, 2–5 July 2013; IEEE: Piscataway, NJ, USA, 2013; pp. 810–815.
54. Betzler, A.; Gomez, C.; Demirkol, I.; Paradells, J. CoAP congestion control for the internet of things. *IEEE Commun. Mag.* **2016**, *54*, 154–160. [CrossRef]
55. Chen, Y.; Kunz, T. Performance evaluation of IoT protocols under a constrained wireless access network. In Proceedings of the 2016 International Conference on Selected Topics in Mobile & Wireless Networking (MoWNeT), Cairo, Egypt, 11–13 April 2016; pp. 1–7. [CrossRef]
56. OMG. *The Real-Time Publish-Subscribe Wire Protocol DDS Interoperability Wire Protocol Specification, Version 2.2*; OMG: Needham, MA, USA, 2014.
57. González, I.; Calderón, A.J.; Figueiredo, J.; Sousa, J.M.C. A Literature Survey on Open Platform Communications (OPC) Applied to Advanced Industrial Environments. *Electronics* **2019**, *8*, 510. [CrossRef]
58. Neshenko, N.; Bou-Harb, E.; Crichigno, J.; Kaddoum, G.; Ghani, N. Demystifying IoT Security: An Exhaustive Survey on IoT Vulnerabilities and a First Empirical Look on Internet-Scale IoT Exploitations. *IEEE Commun. Surv. Tutor.* **2019**, *21*, 2702–2733. [CrossRef]
59. Adeniyi, A.E.; Jimoh, R.G.; Awotunde, J.B. A systematic review on elliptic curve cryptography algorithm for internet of things: Categorization, application areas, and security. *Comput. Electr. Eng.* **2024**, *118*, 109330. [CrossRef]
60. Reddy, N.M.; Budati, A.K.; Islam, S.; Ramesh, G. Enhanced elliptic curve-diffie hellman technique with bigdata analytics for satellite image security enhancement in internet of things systems. *Earth Sci. Inform.* **2024**, *17*, 711–723. [CrossRef]
61. Sebbah, A.; Benamar, K. A Privacy-Enhanced Scheme Within The Public Key Infrastructure For The Internet Of Things, Employing Elliptic Curve Diffie-Hellman (ECDH). *Indones. J. Electr. Eng. Inform. (IJEEI)* **2024**, *12*, 65–74. [CrossRef]
62. Aoueileyine, M.O.E.; Karmous, N.; Bouallegue, R.; Youssef, N.; Yazidi, A. Detecting and mitigating MiTM attack on IOT devices using SDN. In Proceedings of the International Conference on Advanced Information Networking and Applications, Kitakyushu, Japan, 17–19 April 2024; Springer: Berlin/Heidelberg, Germany, 2024; pp. 320–330.
63. Tyagi, V.; Saraswat, A.; Kumar, A.; Gambhir, S. Securing IoT Devices Against MITM and DoS Attacks: An Analysis. In *Reshaping Intelligent Business and Industry: Convergence of AI and IoT at the Cutting Edge*; Scrivener Publishing LLC: Beverly, MA, USA, 2024; pp. 237–249.
64. Alkofahi, H.; Alawneh, H.; Skjellum, A. MitM attacks on intellectual property and integrity of additive manufacturing systems: A security analysis. *Comput. Secur.* **2024**, *140*, 103810. [CrossRef]
65. Stojanović, N.M.; Todorović, B.M.; Ristić, V.B.; Stojanović, I.V. Direct sequence spread spectrum: History, principles and modern applications. *Vojnotehnički glasnik/Mil. Tech. Cour.* **2024**, *72*, 790–813. [CrossRef]
66. Du, F.; Du, P. Micro frequency hopping spread spectrum modulation and encryption technology. *arXiv* **2024**, arXiv:2408.00400.
67. Wang, J.; Liang, Y.; Xu, X.; Wang, J.; Zhong, Y. A High Dynamic Velocity Locked Loop for the Carrier Tracking of a Wide-Band Hybrid Direct Sequence/Frequency Hopping Spread-Spectrum Signal. *Electronics* **2024**, *13*, 1794. [CrossRef]
68. Haataja, K.; Toivanen, P. Two practical man-in-the-middle attacks on Bluetooth secure simple pairing and countermeasures. *IEEE Trans. Wirel. Commun.* **2010**, *9*, 384–392. [CrossRef]
69. Fuster, J.; Solera-Cotanilla, S.; Pérez, J.; Vega-Barbas, M.; Palacios, R.; Alvarez-Campana, M.; Lopez, G. Analysis of security and privacy issues in wearables for minors. *Wirel. Netw.* **2024**, *30*, 5437–5453. [CrossRef]

70. Pratama, D.; Moon, J.; Laksmono, A.M.A.; Yun, D.; Iqbal, M.; Jeong, B.; Ji, J.H.; Kim, H. Behind The Wings: The Case of Reverse Engineering and Drone Hijacking in DJI Enhanced Wi-Fi Protocol. In Proceedings of the 2024 International Conference on Platform Technology and Service (PlatCon), Jeju, Republic of Korea, 26–28 August 2024; IEEE: Piscataway, NJ, USA, 2024; pp. 127–132.
71. Rahman, M.T.; Shi, Q.; Tajik, S.; Shen, H.; Woodard, D.L.; Tehranipoor, M.; Asadizanjani, N. Physical Inspection & Attacks: New Frontier in Hardware Security. In Proceedings of the 2018 IEEE 3rd International Verification and Security Workshop (IVSW), Costa Brava, Spain, 2–4 July 2018; pp. 93–102. [CrossRef]
72. Wurm, J.; Hoang, K.; Arias, O.; Sadeghi, A.R.; Jin, Y. Security analysis on consumer and industrial IoT devices. In Proceedings of the 21st Asia and South Pacific Design Automation Conference (ASP-DAC), Macao, China, 25–28 January 2016; pp. 519–524.
73. Bou-Harb, E.; Fachkha, C.; Pourzandi, M.; Debbabi, M.; Assi, C. Communication security for smart grid distribution networks. *IEEE Commun. Mag.* **2013**, *51*, 42–49. [CrossRef]
74. Tychola, K.A.; Rantos, K. Cyberthreats and Security Measures in Drone-Assisted Agriculture. *Electronics* **2025**, *14*, 149. [CrossRef]
75. Koley, S.; Ghosal, P. Addressing Hardware Security Challenges in Internet of Things: Recent Trends and Possible Solutions. In Proceedings of the 2015 IEEE 12th Intl Conf on Ubiquitous Intelligence and Computing and 2015 IEEE 12th Intl Conf on Autonomic and Trusted Computing and 2015 IEEE 15th Intl Conf on Scalable Computing and Communications and Its Associated Workshops (UIC-ATC-ScalCom), Beijing, China, 10–14 August 2015; pp. 517–520. [CrossRef]
76. Pan, Z.; Mishra, P. Design of AI Trojans for Evading Machine Learning-based Detection of Hardware Trojans. In Proceedings of the 2022 Design, Automation & Test in Europe Conference & Exhibition (DATE), Antwerp, Belgium, 14–23 March 2022; pp. 682–687. [CrossRef]
77. Zhang, M.; Zonouz, S. Control Corruption without Firmware Infection: Stealthy Supply Chain Attacks via PLC Hardware Implants (MalTag). In Proceedings of the 2024 ACM/IEEE 15th International Conference on Cyber-Physical Systems (ICCPS), Hong Kong, China, 13–16 May 2024; IEEE: Piscataway, NJ, USA, 2024; pp. 247–258.
78. Kokolakis, G.; Moschos, A.; Keromytis, A.D. Harnessing the power of general-purpose llms in hardware trojan design. In Proceedings of the International Conference on Applied Cryptography and Network Security, Abu Dhabi, United Arab Emirates, 5–8 March 2024; Springer: Berlin/Heidelberg, Germany, 2024; pp. 176–194.
79. Abideen, Z.U. *Reconfigurable Obfuscation Techniques for the IC Supply Chain: Using FPGA-Like Schemes for Protection of Intellectual Property*; Springer Nature: Berlin/Heidelberg, Germany, 2024.
80. Abideen, Z.U.; Gokulanathan, S.; J. Aljafar, M.; Pagliarini, S. An overview of FPGA-inspired obfuscation techniques. *ACM Comput. Surv.* **2024**, *56*, 1–35. [CrossRef]
81. Khokhar, R.H.; Rankothge, W.; Rashidi, L.; Mohammadian, H.; Ghorbani, A.; Frei, B.; Ellis, S.; Freitas, I. A Survey on Supply Chain Management: Exploring Physical and Cyber Security Challenges, Threats, Critical Applications, and Innovative Technologies. *Int. J. Supply Oper. Manag.* **2024**, *11*, 250–283.
82. Bakhshi, T.; Ghita, B.; Kuzminykh, I. A review of IoT firmware vulnerabilities and auditing techniques. *Sensors* **2024**, *24*, 708. [CrossRef]
83. Keromytis, A.D. Buffer overflow attacks. In *Encyclopedia of Cryptography, Security and Privacy*; Springer: Berlin/Heidelberg, Germany, 2024; pp. 1–4.
84. Shaw, S. Report on Stack-Based Buffer Overflows. 2024. Available online: https://cyberstan.co.uk/wp-content/uploads/2024/10/CSAO_coursework-2.pdf (accessed on 7 March 2025).
85. Mahdi, O.M.E.; Juremi, J. EFTS: An encryption file transfer system applying advanced encryption standard (AES) algorithm. In *AIP Conference Proceedings*; AIP Publishing: New York, NY, USA, 2024; Volume 2802.
86. Gaydos, M.G.; Wallace, N.L.; Brown, R.G. *Reverse Engineering and Embedded Processor Analysis*; Technical report; Sandia National Lab. (SNL-NM): Albuquerque, NM, USA, 2020.
87. Tushir, B.; Dalal, Y.; Dezfouli, B.; Liu, Y. A Quantitative Study of DDoS and E-DDoS Attacks on WiFi Smart Home Devices. *IEEE Internet Things J.* **2021**, *8*, 6282–6292. [CrossRef]
88. Ashfaq, M.F.; Malik, M.; Fatima, U.; Shahzad, M.K. Classification of IoT based DDoS Attack using Machine Learning Techniques. In Proceedings of the 2022 16th International Conference on Ubiquitous Information Management and Communication (IMCOM), Seoul, Republic of Korea, 3–5 January 2022; pp. 1–6. [CrossRef]
89. Alyami, M.; Alharbi, I.; Zou, C.; Solihin, Y.; Ackerman, K. WiFi-based IoT Devices Profiling Attack based on Eavesdropping of Encrypted WiFi Traffic. In Proceedings of the 2022 IEEE 19th Annual Consumer Communications & Networking Conference (CCNC), Las Vegas, NV, USA, 8–11 January 2022; pp. 385–392. [CrossRef]
90. Garvik, M.L.; Lindås, S.; Bø Svendsen, F. Authentication with the Use of MAC and It's Security Challenges. Bachelor's Thesis, NTNU, Trondheim, Sweden, 2023.
91. Hoffman, P.E. DNS Security Extensions (DNSSEC). *RFC 9364* **2023**. Available online: https://www.hjp.at/doc/rfc/rfc9364.html (accessed on 7 March 2025).

92. Sinha, S. Network layer DoS Attack on IoT System and location identification of the attacker. In Proceedings of the 2021 Third International Conference on Inventive Research in Computing Applications (ICIRCA), Coimbatore, India, 2–4 September 2021; pp. 22–27. [CrossRef]
93. Shi, Y.; Lu, X.; An, K.; Li, Y.; Zheng, G. Efficient index-modulation-based FHSS: A unified anti-jamming perspective. *IEEE Internet Things J.* **2023**, *11*, 3458–3472. [CrossRef]
94. Imran, M.; Zhiwen, P.; Nan, L.; Sajjad, M.; Butt, F.M. Anti-jamming for cognitive radio networks with Stackelberg game-assisted DSSS approach. *EURASIP J. Wirel. Commun. Netw.* **2024**, *2024*, 73. [CrossRef]
95. Siddiqui, M.N.; Malik, K.R.; Malik, T.S. Performance Analysis of Blackhole and Wormhole Attack in MANET Based IoT. In Proceedings of the 2021 International Conference on Digital Futures and Transformative Technologies (ICoDT2), Islamabad, Pakistan, 20–21 May 2021; pp. 1–8. [CrossRef]
96. Kumavat, K.S.; Gomes, J. Performance Evaluation of IoT-enabled WSN system With and Without DDoS Attack. In Proceedings of the 2023 International Conference for Advancement in Technology (ICONAT), Goa, India, 24–26 January 2023; pp. 1–5. [CrossRef]
97. Tatar, E.E.; Dener, M. Wormhole Attacks in IoT Based Networks. In Proceedings of the 2021 6th International Conference on Computer Science and Engineering (UBMK), Ankara, Turkey, 15–17 September 2021; pp. 478–482. [CrossRef]
98. Verma, M.K.; Dwivedi, R.K. A Survey on Wormhole Attack Detection and Prevention Techniques in Wireless Sensor Networks. In Proceedings of the 2020 International Conference on Electrical and Electronics Engineering (ICE3), Gorakhpur, India, 14–15 February 2020; pp. 326–331. [CrossRef]
99. Hu, Y.C.; Johnson, D.B.; Perrig, A. SEAD: Secure efficient distance vector routing for mobile wireless ad hoc networks. *Ad Hoc Netw.* **2003**, *1*, 175–192. [CrossRef]
100. Safari, F.; Kunze, H.; Ernst, J.; Gillis, D. A novel cross-layer adaptive fuzzy-based ad hoc on-demand distance vector routing protocol for MANETs. *IEEE Access* **2023**, *11*, 50805–50822. [CrossRef]
101. Kaddoura, S.; Haraty, R.A.; Al Jahdali, S.; Assi, M. SDODV: A smart and adaptive on-demand distance vector routing protocol for MANETs. *Peer-Peer Netw. Appl.* **2023**, *16*, 2325–2348. [CrossRef]
102. Ali, S.; Khan, M.A.; Ahmad, J.; Malik, A.W.; ur Rehman, A. Detection and prevention of Black Hole Attacks in IOT & WSN. In Proceedings of the 2018 Third International Conference on Fog and Mobile Edge Computing (FMEC), Barcelona, Spain, 23–26 April 2018; pp. 217–226. [CrossRef]
103. Fu, C.; Zeng, Q.; Chi, H.; Du, X.; Valluru, S.L. IoT Phantom-Delay Attacks: Demystifying and Exploiting IoT Timeout Behaviors. In Proceedings of the 2022 52nd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), Baltimore, MD, USA, 27–30 June 2022; pp. 428–440. [CrossRef]
104. Nassi, B.; Nassi, D.; Ben-Netanel, R.; Mirsky, Y.; Drokin, O.; Elovici, Y. Phantom of the ADAS: Phantom Attacks on Driver-Assistance Systems. *Cryptol. ePrint Arch.* **2020**, 2020/085. Available online: https://eprint.iacr.org/2020/085.pdf (accessed on 9 March 2025).
105. Whalen, S. An Introduction to ARP Spoofing. *Node99* **2001**. Available online: http://www.gbppr.net/2600/arp_spoofing_intro.pdf (accessed on 9 March 2025).
106. Gamage, K.A.; Sajid, A.; Sonbul, O.S.; Rashid, M.; Jaffar, A.Y. A Dynamic Framework for Internet-Based Network Time Protocol. *Sensors* **2024**, *24*, 691. [CrossRef] [PubMed]
107. Zhang, R.; Hu, Z.; Li, J.; Fan, F.; Wen, F. Network Time Protocol (NTP) implementation for laser inter-satellite networks. In Proceedings of the Advanced Fiber Laser Conference (AFL2023), Shenzhen, China, 10–12 November 2023; SPIE: Bellingham, WA, USA, 2024; Volume 13104, pp. 1707–1712.
108. Banerjee, P.; Matsakis, D. Network Time Protocol (NTP) and Precise Time Protocol (PTP). In *An Introduction to Modern Timekeeping and Time Transfer*; Springer: Berlin/Heidelberg, Germany, 2023; pp. 141–152.
109. Taherdoost, H. Security and internet of things: Benefits, challenges, and future perspectives. *Electronics* **2023**, *12*, 1901. [CrossRef]
110. Fei, W.; Ohno, H.; Sampalli, S. A systematic review of iot security: Research potential, challenges, and future directions. *ACM Comput. Surv.* **2023**, *56*, 1–40. [CrossRef]
111. Aslan, Ö.; Aktuğ, S.S.; Ozkan-Okay, M.; Yilmaz, A.A.; Akin, E. A comprehensive review of cyber security vulnerabilities, threats, attacks, and solutions. *Electronics* **2023**, *12*, 1333. [CrossRef]
112. Sun, P.; Shen, S.; Wan, Y.; Wu, Z.; Fang, Z.; Gao, X.Z. A Survey of IoT Privacy Security: Architecture, Technology, Challenges, and Trends. *IEEE Internet Things J.* **2024**, *11*, 34567–34591. [CrossRef]
113. Ul Haq, S.; Singh, Y.; Sharma, A.; Gupta, R.; Gupta, D. A survey on IoT & embedded device firmware security: Architecture, extraction techniques, and vulnerability analysis frameworks. *Discov. Internet Things* **2023**, *3*, 17.
114. Siwakoti, Y.R.; Bhurtel, M.; Rawat, D.B.; Oest, A.; Johnson, R. Advances in IoT security: Vulnerabilities, enabled criminal services, attacks, and countermeasures. *IEEE Internet Things J.* **2023**, *10*, 11224–11239. [CrossRef]
115. Noman, H.A.; Abu-Sharkh, O.M. Code injection attacks in wireless-based Internet of Things (IoT): A comprehensive review and practical implementations. *Sensors* **2023**, *23*, 6067. [CrossRef]

116. AlSalem, T.S.; Almaiah, M.A.; Lutfi, A. Cybersecurity risk analysis in the IoT: A systematic review. *Electronics* **2023**, *12*, 3958. [CrossRef]
117. Alqarawi, G.; Alkhalifah, B.; Alharbi, N.; El Khediri, S. Internet-of-things security and vulnerabilities: Case study. *J. Appl. Secur. Res.* **2023**, *18*, 559–575. [CrossRef]
118. Aziz Al Kabir, M.; Elmedany, W.; Sharif, M.S. Securing IOT devices against emerging security threats: Challenges and mitigation techniques. *J. Cyber Secur. Technol.* **2023**, *7*, 199–223. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.