frontiers | Frontiers in The Internet of Things

TYPE Systematic Review
PUBLISHED 23 January 2024
DOI 10.3389/friot.2023.1306018

OPEN ACCESS

EDITED BY
Oktay Cetinkaya,
University of Oxford, United Kingdom

REVIEWED BY
Mohamed Rahouti,
Fordham University, United States
Rashid Amin,
University of Chakwal, Pakistan

*CORRESPONDENCE
Marwa Salayma,
msalayma@gmail.com

RECEIVED 03 October 2023
ACCEPTED 14 December 2023
PUBLISHED 23 January 2024

CITATION
Salayma M (2024), Risk and threat mitigation techniques in internet of things (IoT) environments: a survey.
Front. Internet. Things 2:1306018.
doi: 10.3389/friot.2023.1306018

Risk and threat mitigation techniques in internet of things (IoT) environments: a survey

Marwa Salayma*

Faculty of Engineering, Department of Computing, Imperial College London, London, United Kingdom

Security in the Internet of Things (IoT) remains a predominant area of concern. Although several other surveys have been published on this topic in recent years, the broad spectrum that this area aims to cover, the rapid developments and the variety of concerns make it impossible to cover the topic adequately. This survey updates the state of the art covered in previous surveys and focuses on defences and mitigations against threats rather than on the threats alone, an area that is less extensively covered by other surveys. This survey has collated current research considering the dynamicity of the IoT environment, a topic missed in other surveys and warrants particular attention. To consider the IoT mobility, a life-cycle approach is adopted to the study of dynamic and mobile IoT environments and means of deploying defences against malicious actors aiming to compromise an IoT network and to evolve their attack laterally within it and from it. This survey takes a more comprehensive and detailed step by analysing a broad variety of methods for accomplishing each of the mitigation steps, presenting these uniquely by introducing a "defence-in-depth" approach that could significantly slow down the progress of an attack in the dynamic IoT environment. This survey sheds a light on leveraging redundancy as an inherent nature of multi-sensor IoT applications, to improve integrity and recovery. This study highlights the challenges of each mitigation step, emphasises novel perspectives, and reconnects the discussed mitigation steps to the ground principles they seek to implement.

KEYWORDS

internet of things, defence in depth, threat mitigation, life-cycle, device self-defence, continuous monitoring, microsegmentation, redundancy and resilience

1 Introduction

Although we refer to *the* Internet of Things (IoT) as if it formed a system, this terminology is not entirely correct and does reflect modern developments. Precursors of the IoT enjoyed various names such as Pervasive Computing, Ubiquitous Computing or Machine-to-Machine Communications (the latter taking a more network centric view) and sought to convey the idea that computation could occur anywhere. Indeed, the IoT is not just a collection of "things", nor a well defined system formed of things, but the instrumentation of the entire physical space surrounding us with an Internet connected digital interface and computational capabilities that increasingly comprise decision making and even learning. We often talk of "*smart*" things whether it is, for example, smart-meters, smart-buildings or, smart-toys. Again this only reflects that all objects that we are accustomed to in our physical spaces now comprise a digital component able to perceive the physical world through sensors and to control it through actuators. This

point is particularly important when it comes to security. Compromising the security of the digital interface of a physical object impacts its physical behaviour and security, and the threats to be considered do not all originate in the digital (cyber) space but may start by exploiting their physical vulnerabilities or the trusting nature of their human users. Having made this point, this paper adopts commonly accepted terminology and refers to IoT Systems, Devices, Networks or Environments, bearing in mind that it is only a digital (cyber) perspective on the entirety of the physical world that surrounds us, which interconnects the physical world to the resources of the digital space.

The number of IoT devices is continuously increasing, and this trend is set to continue. In their latest report, IoT Analytics estimated that in 2022, the global number of connected IoT devices grew to 14.4 billion, which is a 18% increase compared to 2021, and by 2025, IoT Analytics predicts that there likely to be around 27 billion IoT connections (Sinha, 2021). In some respects, this may turn out to be an underestimate. On one hand the size of the devices is continuously reducing as well as their power consumption. On the other the (wireless) network connectivity is increasing, e.g., the deployment of 5G (Wasicek, 2020). Finally, devices are increasingly capable of learning and autonomous decision making. These trends will lead to more devices being used to monitor the physical world at a finer level of granularity and provide increasingly complex systems that optimise our usage of resources, personalise the services that are offered to us and, hopefully, increase our quality of life.

However, adding an IoT device to a system is also adding an opportunity to compromise that system for a malicious actor. Any device connected to the Internet can be attacked from any other Internet location. Furthermore, in contrast to traditional computers or cloud servers securely hosted in offices or secure physical locations, IoT devices are deployed in the physical environment and can also be subjected to direct connections and physical attacks. Considering their vulnerability, a direct consequence of adding many IoT devices to our systems is that the attack surface of the IoT systems is also increasing exponentially. Faster interconnections, and rapid response also mean that compromises can spread faster and wider within the systems making them more difficult to protect and dependent on rapid response to a compromise to maintain their resilience. As well as making systems more robust to adversarial threats "by design", it is also necessary to deploy response techniques that can hinder the progress of an attack as well as responses that enable an adaptation (re-configuration) of the system and its recovery to maintain the system's function even when the systems have been partially compromised.

The security and resilience of IoT environments is a complex topic that spans across the entirety of their life-cycle from design and realisation to their deployment, operation and decommissioning. In contrast to the other related surveys which fall short of outlining concrete coherent steps to mitigate the spread of attacks in an IoT environment, this survey explores security measures in IoT system that are applied throughout the life-cycle of the IoT devices starting from their design, to the moment when a device joins a network, while the IoT device operates in the network, and until it leaves (or is/removed) from the network and decommissioned. The survey discusses threat mitigation techniques applied across different IoT application contexts, and elaborates on how to apply each mitigation technique, its benefits and limitations and the extent to which progress has been reported in the literature. In essence, the discussed measures provide answers to the questions of how IoT device(s) connect and communicate with new devices and systems safely, starting from the moment when the IoT device(s) join the new environment, whilst operating in it, when an attack occurs, and until the device is removed/decommissioned or leaves the environment. This paper adopts a" defence in depth" strategy in discussing mitigation techniques proposed for the aim of controlling and slowing down the spread of threats in the IoT environment throughout the life-cycle of the IoT device. A defence in depth strategy to securing systems uses measures that aim to reduce systems vulnerabilities, contain threats, and mitigate attack effects if they occur, such that if an attacker manages to overcome one layer of defence, they still need to overcome the subsequent defence layers to compromise the system (Vacca, 2012). The challenges are being addressed in the design of individual devices and in the design and operation of deployments. Like in the case of enterprise or more traditional computing environments, new techniques are being developed to make devices more trustworthy and new techniques are being developed to make systems more resilient and trustworthy by detecting, mitigating and responding to threats at run-time.

The contributions of this survey are summarised as follows: i) This survey discusses the state of the art covered in previous surveys, whilst focusing on defending against threats rather than on the threats alone. A summary of the discussed topics in the prior surveys along with examples of mitigation techniques suggested are presented in Table 1. ii) This survey collates current research into risk and threat mitigations in the dynamic IoT environment, considering the mobility if the IoT systems, which is composed of several devices that join and leave a network dynamically. To achieve this, this survey provides an overview and presents these mitigation techniques uniquely throughout the life cycle of the IoT device, starting from its design, to the moment when a device joins a network, while the IoT device operates in the network, and until it leaves (or is removed) from the network and is eventually decommissioned, hence adopting a "defence-in-depth" approach. A taxonomy for the mitigation techniques discussed in this survey, and which are applied throughout the life-cycle of an IoT device is presented in Table 2. iii) This survey takes a more comprehensive and detailed step by analysing a broad variety of methods for accomplishing each of the mitigation steps, and elaborates on how to apply each mitigation technique across different IoT application contexts, its benefits, as well as highlighting their challenges, limitations, difficulty of implementation, and the extent to which progress has been reported in the literature. iv) This survey sheds a light on a rarely discussed method in literature, that is, exploiting the redundancy as an inherent nature of multi-sensor IoT applications to improve integrity and recovery, and discusses different methods on harnessing redundancy in inter-connectivity as a mitigation technique to reconfigure networks in response to security events and isolate compromised devices, whilst enabling the rest of the network to operate normally. v) The survey emphasises novel perspectives for the discussed mitigation steps, and reconnects them to the ground principles they seek to implement.

This survey is structured according to the life-cycle of an IoT device in a dynamic context, i.e., before a device joins a system, when

TABLE 1 A taxonomy of the topics and countermeasures covered in prior related surveys.

Research theme	Topics covered	Examples on mitigation techniques covered	Related surveys
IoT application domains	Industrial Control Systems (SCADA), SmartGrids, Intelligent Transportation Systems, E-Health and Medical IoT Systems, Smart Home and Automation IoT Systems	- Threats related, e.g., Secure remote access	Stellios et al. (2018)
		- Vulnerability related, e.g., Tamper resistance	
		- Connectivity related, e.g., Network segmentation	
IoT stack layers	1- Physical Layer, Data Link Layer, Network Layer, Transport Layer, Application Layer	1- Physical: spread-spectrum communication, MAC: Error correction codes, Network: Multi-path routing, Transport: security policies, Application: CoAPs	1-Butun et al. (2019)
	2- Sensing layer, Network layer, Middleware layer, Application layer	2- Using blockchain, using fog Computing, using machine learning, using edge computing	2-Hassija et al. (2019); Lu and Xu (2019)
	3- Perception layer, Network layer, Application layer	3-Perception: intrusion detection, Network: IPv6 and IPSe	3-Mohamad Noor and Hassan (2019)
IoT technologies (practical and technical)	1- ZigBee, BLE, 6LoWPAN, LoRaWAN.	1- ZigBee: trust centre, BLE: pairing using elliptic curve cryptography, 6LoWPAN: RPL, LoRaWAN: different keys to verify Message Integrity Code (MIC)	1-Meneghello et al. (2019)
	2- Case studies: EV charger, Itron Centron CL200 smart meter, Fitbit Aria, g Google's Nest Thermostat, Tesla Model S, Chamberlain MyQ, Parrot AR 2.0 Quadcopter, Edimax IP camera system	2- Additional parameter validation, tamper-resistant, encryption, chain-of-trust secure boot, Random Number Generators, Strong password, access restrictions, identity management	2-Alladi et al. (2020)
	3- Physical-based, e.g., RFID, Protocols-based, e.g., NFC,Bluetooth, Wifi, ZigBee, Network-based, e.g., RPL, 6loWPAN, TCP-UDP.	3- Physical: lightweight cartographic mechanisms, protocol based: error control mechanisms, network based: intrusion detection	3-Abdul-Ghani et al. (2018)
IoT device life cycle	1- Smart home environment: security of the development of IoT devices, integration of devices in home networks, usage until end-of-life	1- Minimum reliability, trust infrastructure, network segmentation, use gateways, vulnerability survey, software updates, remote protection, secure backup	1-Cedric Levy-Bencheton (2015)
	2- IoT supply chain: actors, processes and technologies	2- Product design: sabotage prevention, semiconductor fabrication: scrap management, component manufacturing: defective components, component Assembly: firmware access control, device programming: coding practices, distribution: tracking for registration, technical support: patches, recovery: data removal	2-Christina Skouloudi (2020)
	3- Smart environments: security Critical Information Infrastructures, Policies, Technical Measures	3- Security by design, trust management, firmware updates, authentication, cryptography, logging, end-of-life support	3-ENISA (2017)
Other themes	1- Certification	1- IoT device life cycle support	1-Matheu et al. (2020b)
	2- Artificial Intelligence	2- Naïve Bayes for intrusion detection	2-Kuzlu et al. (2021)
	3- Software Defined Networks	3- vFirewalls, vIoTHoneyNet, traffic filtering and isolation, vChannelProtection	3-Liu et al. (2023b)
	4- Blockchain	4- Trust management	4-Molina Zarca et al. (2019)

a device wants to join a system, while the device is in the system, when a cyber attack occurs, if the device has been compromised, and when the device leaves or is removed from the system. This structure is depicted in Table 2.

After summarising the aspects covered in prior surveys in Section 2, this survey discusses aspects of self-protection and self-defence in Section 3, in particular, techniques to secure the IoT device before and when a device join a new system. Techniques based on mediation are discussed in Section 4 as techniques to secure the IoT device and a system while the device operates in the system. Segmentation techniques are discussed in Section 5 as techniques to mitigate the impact of the attack on the system when a cyber attack occurs. Techniques based on redundancy and recovery are discussed in Section 6 as mitigation techniques to be applied when the device leaves or is removed from the system, before discussing and drawing the conclusions in Section 7.

2 Prior surveys

Many surveys have been published on IoT threats, attacks and countermeasures (Abdul-Ghani et al., 2018; Stellios et al., 2018; Butun et al., 2019; Hassan et al., 2019; Hassija et al., 2019; Meneghello et al., 2019; Alladi et al., 2020; Matheu et al., 2020a; Hamza et al., 2020), at least twelve have been published very recently, between 2021–2023 (Choo et al., 2021; Kuzlu et al., 2021; Najmi et al., 2021; Abdel-Basset et al., 2022; Rayes and Salam, 2022; Swessi and Idoudi, 2022; Liu et al., 2023a;

TABLE 2 A taxonomy for mitigation techniques discussed in this survey that are applied throughout the life-cycle of an IoT device.

IoT Device life cycle	Mitigation technique	Related studies and proposed work
Before a device joins a system	Device self-protection and self-defence	Sidhu et al. (2019); Hamadeh et al. (2017); Lu et al. (2020); Eldefrawy et al. (2012); Mohan et al. (2018); Dhavlle et al. (2021); Demme et al. (2013)
	- Hardware self-protection	
	- Software self-protection	Ravi et al. (2004); Zavalyshyn et al. (2020); Frank et al. (2018); Choi et al. (2018); Ankergård et al. (2021); Mercado-Velázquez et al. (2021); Navas et al. (2020, 2021)
	- Moving Target Defence	
If a device wants to join a system	Certification	Matheu et al. (2020b), Matheu et al. (2020b)
While the device is in the system	Mediation	Leo et al. (2014); Mahmoud et al. (2015); Davies et al. (2016); Chio et al. (2019)
	- IoT Edge	Zarpelão et al. (2017)
	- Continuous monitoring	Franco et al. (2021); Vetterl (2020); Pa et al. (2015)
	*Intrusion detection	Kuzlu et al. (2021); Chaabouni et al. (2019); Kumar et al. (2021); Meidan et al. (2018); Pacheco et al. (2019); Pauna et al. (2019)
	*Honeypots	
	*AI techniques for monitoring and detection	
When a cyber attack occurs	Device self-protection and self-defence	Sidhu et al. (2019); Hamadeh et al. (2017); Lu et al. (2020); Eldefrawy et al. (2012); Mohan et al. (2018); Dhavlle et al. (2021); Demme et al. (2013)
	- Hardware self-protection	Ravi et al. (2004); Zavalyshyn et al. (2020); Frank et al. (2018); Choi et al. (2018); Ankergård et al. (2021)
	- Software self-protection	Mercado-Velázquez et al. (2021); Navas et al. (2020, 2021)
	- Moving Target Defence	
If the device has been compromised	Device(s) isolation and system segmentation	Stellios et al. (2018); Xing (2021); Stergiopoulos et al. (2020); Luiijf and Klaver (2021)
	- Identifying and documenting IoT dependencies	Wasicek (2020); Osman et al. (2020); Mämmelä et al. (2016)
	- Micro-Segmentation	Baldini et al. (2020); García et al. (2019); Zarca et al. (2019)
	- Software Defined Networks (SDN)	
When the device leaves or is removed from the system	System availability and resilience	Laszka et al. (2018); Venkatakrishnan and Vouk (2016)
	- Diversity	Salayma et al. (2017); Illiano and Lupu (2015); Illiano et al. (2017)
	- Exploiting data correlation	Rosenberg and Reinhardt (2021)
	- Exploiting data overhearing	Li et al. (2017); Ar-Reyouchi et al. (2020); Liao et al. (2019); Lee and Lee (2020)
	- Network Coding (NC)	

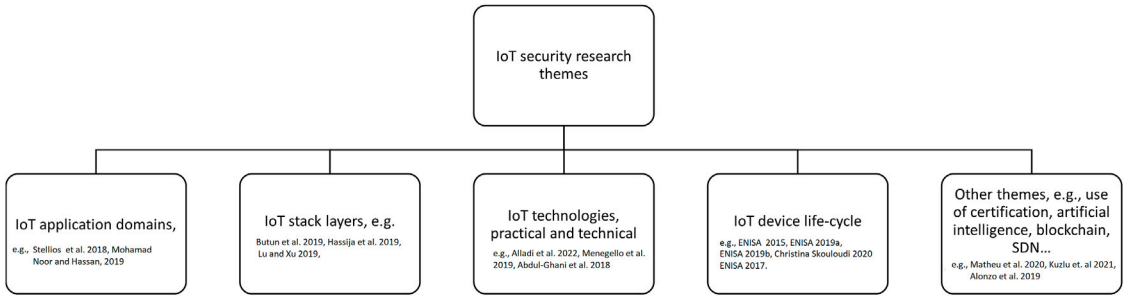

FIGURE 1
Research themes of the surveys on security in IoT environment.

Gerodimos et al., 2023; Hromada et al., 2023; Kamalov et al., 2023; Kumar et al., 2023; Stergiou et al., 2023). They introduce different classifications of IoT security challenges from varying perspectives. Examples of approaches taken by surveys are first described, before a figure representing themes is shown in Figure 1. The topics discussed in the prior surveys and presented in this section are summarised in Table 1, which lists the surveys according to the categorised themes. Due to space limitation, Table 1 shows only examples of mitigation techniques presented in the prior surveys, highlighting the distinctive mitigation techniques that differentiate between the discussed surveys.

Stellios et al. (2018) classify IoT attacks as well as mitigation techniques across different application domains from 2010 until 2018. The attacks discussed comprise real-world incidents, as well as attacks that have been implemented and published as proof-of-concept, both are referred to in the survey as "verified attacks". The application domains considered include industrial control systems, smart power grids, intelligent transportation systems, and medical applications. The authors highlight the potential impact of attacks on critical systems when IoT devices are connected to them directly or indirectly. The IoT attacks considered include those that occur even when no IoT devices connected to the critical infrastructure. Such hidden paths of attack are termed "subliminal attack paths" by the authors. The survey sheds light on the gaps in current security controls applied in each sector, and emphasises the inadequate implementation of current security controls owing to a lack of regulation and security policies that would force operators to use security tested, but usually more expensive IoT devices.

In a slightly earlier paper (Mohamad Noor and Hassan, 2019), discuss the research trends in IoT security and IoT security control strategies (mainly focussing on 2016–2018). The security controls discussed are applied according to the threat vectors. The authors observe that authentication is the most popular method at the application layer (60%) followed by access control mechanisms, that involve trust evaluation. The authors observe that trends in the development of IoT security controls mainly focus on improving lightweight authentication and encryption for power and resource constrained devices. The survey concludes that IoT security mitigation should target all architecture layers, including perception, network, and application, whereas most of approaches focus primarily on the network layer.

The surveys by Abdul-Ghani et al. (2018); Meneghello et al. (2019); Alladi et al. (2020) classify IoT security risks and mitigation techniques from a practical, technical, consumer and application perspective and focus on the practical implications of IoT security. For example, Meneghello et al. (2019) discusses IoT security from a more practical perspective compared to others and focuses on the security controls adopted in popular IoT communication protocols, such as: ZigBee, Bluetooth Low Energy (BLE), 6LoWPAN, and LoRaWAN, whilst highlighting the weaknesses of these controls. The survey also discusses other security mechanisms including the use of encryption, both standard and light-weight, random number generation, secure hardware, and Intrusion Detection Systems (IDS). The survey points out the importance of security by design and the systematic use of standard security mechanisms, which are often poorly implemented due to the heterogeneity of IoT devices.

Other surveys (Butun et al., 2019; Hassija et al., 2019; Lu and Xu, 2019) also adopt a taxonomy for IoT threats and countermeasures distinguishing between the sensing, network and application layers. For example, Lu and Xu, (2019), proposes a four layers IoT architecture: sensing, network, middleware, and application, and discusses potential attacks and threats that target each layer. The survey classifies IoT attacks into eight categories, and briefly introduces common countermeasures that apply not only to specific layers but also to "intelligent" objects and the entire network. The work discusses RFID-Based Authentication Measures, as well as measures that apply in Wireless Sensor Networks (WSN) and identifies a number of trends and emergent developments towards more secure IoT systems including: cloud service security, 5G, Quality of Service-Based Design, IoT forensics and self-management.

Although the taxonomies proposed in the surveys mentioned above provide a good summary of the work done so far in this space, and adopt different perspectives on IoT cybersecurity, they fall short of outlining concrete coherent steps to mitigate the spread of attacks in an IoT environment. In fact, threat mitigation strategies in IoT system should be considered and applied throughout the life-cycle of the IoT devices starting from their design, to the moment when a device joins a network, while the IoT device operates in the network, and until it leaves/removed from the network and is eventually decommissioned. To address these aspects, the European Union Agency for Network and Information Security (ENISA) published good practices that can applied in an IoT environment and guidelines that apply to every step of a product's life-cycle: its development, its integration in the system, and its usage and maintenance until end-of-life. The first guidelines were published in 2015 and targeted the IoT product life-cycle in the context of smart home environments (Cedric Levy-Bencheton, 2015). More recently, in 2020, ENISA published guidelines on securing the IoT supply chain, that discuss the entire lifespan of IoT devices: from requirements and design, to end use, delivery maintenance, and disposal, recommending security measures for each step (Christina Skouloudi, 2020). This work also builds on guidelines proposed by ENISA in 2019 (ENISA, 2019b). With these studies, ENISA complements their baseline security recommendations for IoT (ENISA, 2017), which are combined in one tool for securing different "smart environments", such as smart hospitals, smart airports, smart cars and smart cities (ENISA, 2019a). Much of the state of the art established by ENISA is based on surveys and interviews with cybersecurity experts, IoT devices manufacturers, network operators, and standards groups.

Although the guidelines proposed by ENISA are comprehensive, they often assume that the user has control over the integration and usage of the IoT devices. In contrast, IoT networks often have to integrate devices that are outside of operator control. Another shortcoming is an insufficient consideration of the dynamic aspects of a system. In many cases, systems are composed of several devices that join and leave a network dynamically or that may be intermittently connected. Such devices are often mobile, either because they are mobile themselves, e.g., autonomous vehicles, drones, or because they are instrumenting objects that are physically mobile, e.g., body sensor networks for healthcare. This dynamicity requires security controls, such as authentication, access control, risk and trust evaluation and countermeasures to be applied continuously rather than at specific points in time. For example, the risk to a network can vary depending the type and number of devices

joining the network. Finally, IoT environments can leverage redundancy to improve integrity and recovery. The same physical reality is often sensed through multiple sensors making it more difficult to compromise the perception of reality by compromising a single (or a few) devices. Redundancy in inter-connectivity permits to reconfigure networks in response to security events and isolate compromised devices, whilst enabling the rest of the network to operate normally. Finally, redundancy of functionality can help enable the recovery through adaptation and reconfiguration. But it is often not sufficient to consider the attacks alone. The impact of the attacks must be considered to determine the levels of redundancy required. This is particularly important in safety-critical systems and critical infrastructures.

A classification for the prior surveys according to their focus, is depcited in Figure 1. The survey distinguishes between a focus on: IoT application domains, stack layers, technologies, device life-cycle and other themes. Each one of these areas discusses main topics commonly covered in literature that relate to that theme. For example, studies vary in the number of layers considered, but most commonly discuss the: *sensing*, *networking*, and *application layers* (Lu and Xu, 2019).

Based on the presented related work, and their shortcomings, two questions are raised:

1. How could IoT device(s) be enabled to, join, operate, and leave new environments in a secure way, such that the current IoT device(s) and systems in the environment stay protected from possible threats and attacks that might be caused by any newly joined infected device(s)?
2. How can newly joined device(s) be protected from being infected or compromised by compromised devices that already exist in the network?

To answer these questions, this study explores below additional security measures that can be applied across different IoT application contexts, and elaborates on how to apply each mitigation technique, its benefits and limitations and the extent to which progress has been reported in the literature. By contrast to the vast majority of the research directions presented in literature which is related to this work, this survey does not discuss the mitigation techniques in the IoT domain from the layered network architecture of an IoT system, such as IoT-Cloud/Edge, IoT-IoT, and IoT-gateway. In another words, the discussions in this work is not structured based on each network layer of the IoT device or its associated protocols. This is because discussing the topic from the layering architecture perspective does not help to answer the two questions presented above: the layering discussion does not account for the dynamicity of the IoT environment, one of which is the mobility of the IoT device itself when joining or leaving a network, which is one of the main reasons for the fast propagation of a cyber attack in the IoT environment. Rather, this survey provides answers to the questions of how IoT device(s) connect and communicate with new devices and systems safely, from the perspective of discussing the life cycle of the device(s), starting from the moment when the IoT device(s) join the new environment, whilst operating in it, when an attack occurs, and until the

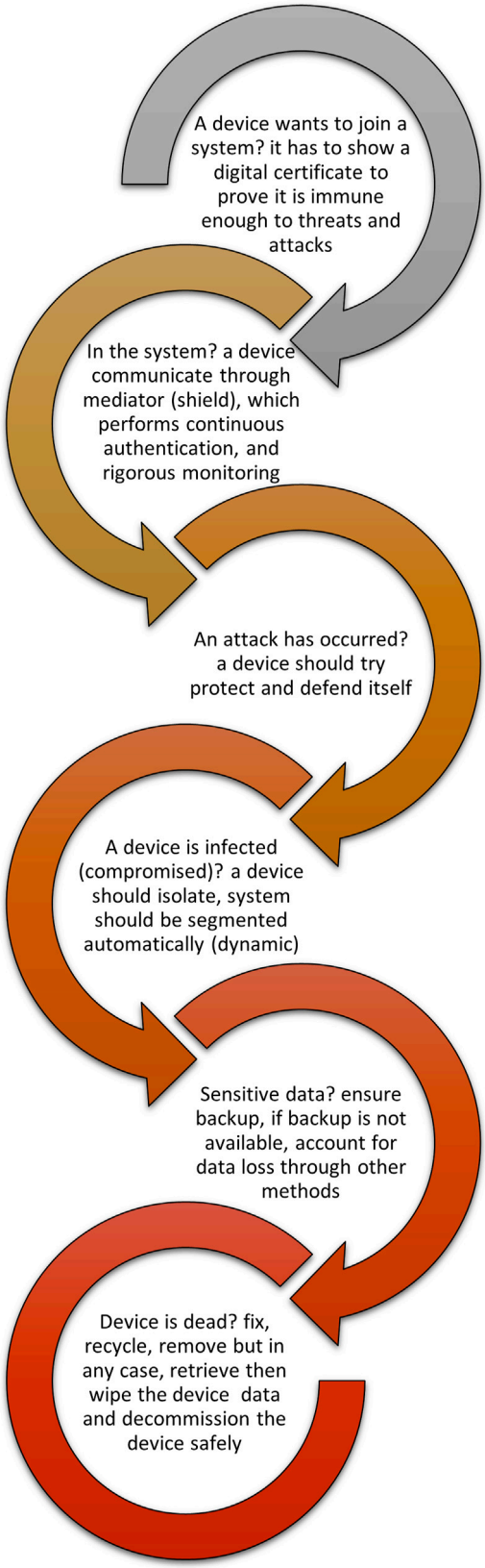

FIGURE 2
Mitigation techniques throughout the life-cycle of an IoT device: a defence in depth approach.

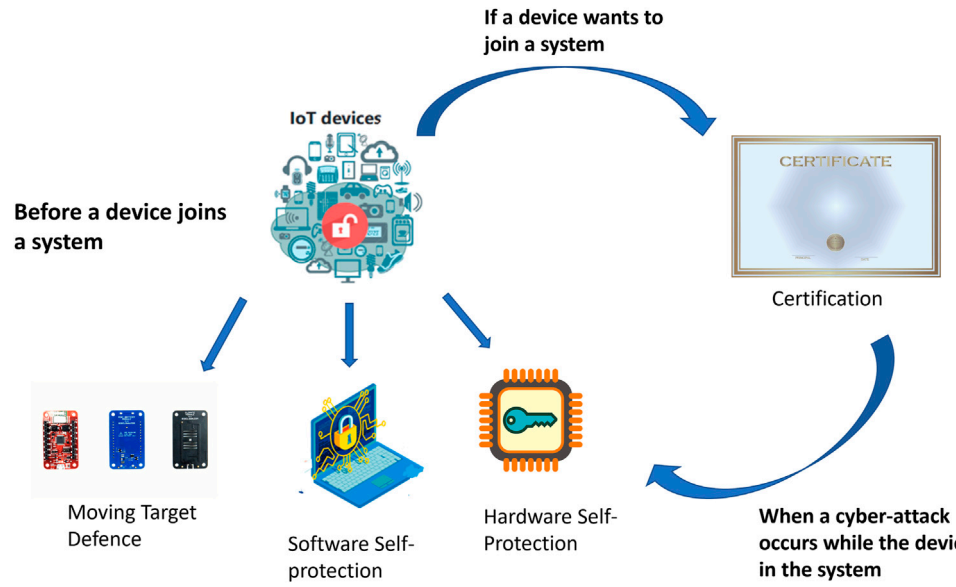

FIGURE 3
IoT device self protection and self defence.

device is removed/decommissioned or leaves the environment, and hence the discussed measures address the dynamicity of the IoT environment. Those steps are depicted in Figure 2, which shows the mitigation steps as a continuous process run in a "defence in depth" approach starting when the device tries to join a system until it dies or leaves. Those mitigation techniques are summarised in Table 1 and are discussed in detail in the following Sections.

3 Device self-protection and self-defence

Securing IoT environments starts with securing the devices themselves. Given that IoT devices are typically exposed to both physical and cyber threats, a "zero trust" assumption is often a safe default, i.e., a device should not be connected unless it has been authenticated, authorised and reasonable steps can be taken to trust the device's integrity and even its capability to defend itself. If a device is able to prove its integrity and show its level of robustness to external threats, then it may be allowed to join a network without increasing substantially the risk to the network itself. Although some studies argue that such proofs could take the form of digital certificates, this only raises other questions: who it issue such certificates? based on which verification processes? what are the cost implications? Even then TOCTOU (time-of-check, time-of-use) issues remain, i.e., how to ascertain that nothing has interfered with the device's integrity or trustworthiness since the last check. After securely joining the network by providing a certificate, software and hardware self protection and defence mechanisms including adopting advanced techniques to make it even harder for the attacker to perform the attack, all help the IoT device continue operating in the environment even after being infected from newly joined devices and *vice versa*. Those mechanisms are depicted in Figure 3, which shows how those techniques are involved during the IoT life cycle, and in particular, before the IoT device joins a system, if the IoT device wants to join a new system, and once a cyber attack occurs while the IoT in the new system. Those techniques are discussed in the following subsections.

3.1 Hardware self-protection

More emphasis is often put in the related work, on software and communication trustworthiness than on device hardware protection, although a number of research groups have focused on hardware aspects. The hardware of an IoT device can serve as a "root-of-trust" for subsequent integrity verifications, so ensuring its security is particularly important. It needs to be robust to physical attacks, supply-chain attacks and side-channel attacks amongst others. Physical attacks, include modification of the hardware circuit, e.g., by removing the component's physical packaging and/or modifying the hardware. Supply-chain attacks refer to modification of the components in a hardware circuit or the circuit itself by suppliers (with or without their knowledge). Side-channel attacks refer to attacks conducted by monitoring or using the properties of the hardware such as power consumption, execution time, behaviour under faults (Ravi et al., 2004).

An example of a physical attack is a Hardware Trojan (HT)—a circuit inserted into a larger one to alter its function and implement malicious behaviour. Hardware Trojans are discussed thoroughly in Sidhu et al. (2019), within the broader context of challenges of securing IoT hardware.A number of methods are discussed in Sidhu et al. (2019) to detect the modifications, which are classified into *pre-silicon* i.e., validating the integrated circuit (IC) at the design stage, and *post-silicon* i.e., verifying the fabricated IC at the manufacturing stage techniques. Post-silcon detection is further divided into *destructive techniques* such as "depackaging" the IC using reverse engineering techniques, and *non-destructive techniques* such as verifying the fabricated IC using testing methods including

functional tests, side channel analysis and automatic test-pattern generation (ATPG). Sidhu et al. (2019) also discuss other HT countermeasures to complement HT detection, as they advocate embedding HT prevention methods during the design phase to implement the concept of *Design for Trust (DfT)* (Sidhu et al., 2019). DfT involves methods to prevent HT insertion, such as obfuscation. However, despite its benefits, run-time monitoring incurs significant additional overhead in a resource environment that is already heavily constrained. "Split manufacturing" stands as a third countermeasure against HT, aiming to hide the design intent of the IC to prevent malicious insertion (Sidhu et al., 2019).

Other hardware security controls seek to identify and authenticate devices uniquely. These include the use of Key Injection, Physically Uncloneable Functions (PUF) and the use of Hardware Security Modules (HSM) (Hamadeh et al., 2017; Lu et al., 2020). HSMs as well as Trusted Platform Modules (TPMs), and more generally Trusted Computing plaforms seek to establish a hardware root of trust on the platform that can then be used to ascertain the platform's integrity, e.g., through attestation, authenticate the device or securely perform cryptographic functions.

However, TPMs (including TPM 2.0) are costly to use in a resource constrained devices, due to the additional space and power consumption they impose. To address this resource issues, the Trusted Computing Group (TCG) created Device Identifier Composition Engine (DICE), a security standard with lightweight hardware requirements, that can be used for hardware-based cryptographic device identity, data encryption, attestation of device firmware, and safe deployment. It can be implemented in a small size micro-controller, and hence is suitable to use in resource-constraint IoT devices. DICE has been first to be adopted by Microsoft for Azure IoTs.

Another hardware based technique to provide security in the IoT context is the Remote Attestation (RA). RA presents a security technique in which a trusted verifier assures the integrity of a prover, i. e., the untrusted device. There are a number of research efforts that propose secure and lightweight hardware architectures such as SMART (Eldefrawy et al., 2012) and TrustLite (Koeberl et al., 2014) to provide a secure remote attestation for embedded and IoT devices. Such architectures adopt minimal hardware components such as simple memory protection units (MPU) (Mohan et al., 2018). However, such architectures fail to provide secure attestation to large number of IoT devices such as drones.

Physical attacks cannot be mitigated without designing circuits that are tamper resistant. Different approaches have been proposed in the literature and are discussed in Ravi et al. (2004). These are divided into i) attack prevention techniques, such as components packaging, and designing hardware circuits with independent power and timing properties, ii) attack detection, such detecting illegal memory accesses by un-trusted software at run-time, iii) attack recovery, for example, displaying a security warning and rebooting the system, iv) and tamper evident design.

Tamper proof and tamper evident design can be used at different levels of complexity to meet different levels of physical security requirements from minimum protection (e.g., seals or enclosures) to environmental failure protection and testing, which is the highest level of protection. IBM's 4758 PCI cryptographic adaptor is an example of adoption of such techniques, as the chip includes internal tamper circuits and sensor components, to detect and respond to physical penetrations, temperature and voltage attacks. However, applying such security levels of protection is expensive, far beyond the cost of typical IoT devices.

Mitigation techniques against side-channel attacks are also discussed in Ravi et al. (2004). These include methods, such as randomisation, to reduce the system's exposure to monitoring and analysis of side-channel information, such as power, timing, and electromagnetic radiation. Randomisation is effective approach as it imposes a significant extra burden on the attacker, and is not entirely unrelated to Moving Target Defence techniques employed at the higher layers. Other methods proposed by the authors against power analysis attacks, aim to increase the number of samples needed to conduct the attack by applying *data masking*, introducing noise into the power measurement data and the use of reduced signal amplitudes. Methods for detecting *fault injection attacks* and preventing transient fault attacks on cryptographic hardware are suggested to harness sensors that can monitor environmental properties to detect fault injection attacks, leveraging error detection methods to prevent such attacks.

From another angle, hardware information has been proven to assist in malware detection, e.g., by applying Machine Learning (ML) techniques to the low-level micro-architectural features captured by Hardware Performance Counters (HPC) with Machine Learning (ML). This has led to the so called *Hardware-assisted Malware Detection (HMD)*, which has been shown to offer improvements on traditional software-based malware detection techniques (Demme et al., 2013; Dhavlle et al., 2021) were the first to harness HPC for malware detection. However, it was not long before evasion techniques were proposed. An adversarial attack on HMDs through which the malware detection accuracy is reduced is proposed in Dhavlle et al. (2021). In response, Dhavlle et al. (2021) aims to improve HMDs to be robust to adversarial attacks through Adversarial Training (AT). The IoT device hardware protection and defence approaches discussed in this section are summarised in Figure 4.

Despite its benefits, hardware protection can have significant implications in terms of increased cost and increased overhead. For example, adding dedicated crypto-processors adds significant cost to the device, whilst monitoring techniques impose high system overhead. Protection against side-channel attacks also introduces significant overheads as the observability of the channel needs to be reduced, e.g., by redundant or irregular use. The overhead is not only in terms of processing but also in terms of power consumption, reducing the applications in which the IoT devices can be used. Whilst effective, hardware protection can only be deployed in applications where the additional cost is justifiable and the additional overhead can be tolerated. This contrasts with the broad use and adoption of IoT devices that mainly leverages their reduced cost and power consumption.

3.2 Software self-protection

Software-based approaches to improve robustness against attacks are categorised in Ravi et al. (2004) according to three design considerations: i) ensuring the integrity and privacy of sensitive code and data at every stage of software execution, ii) ensuring security when executing a given program, and iii)

FIGURE 4
IoT device hardware protection and defence approaches.

removing software vulnerabilities that make the system vulnerable to attacks. Software attacks typically performed by malware exploit flaws in the program and its execution. Such flaws are vulnerabilities when they enable malicious actors to gain privileges in the system, access its content and control (at least partially) its execution (Ravi et al., 2004). Software self-protection and self-defence approaches, can be two fold: i) tamper-resistance techniques that deal with attacks when they occur, resist and counter their effect, and ii)

hardening techniques that aim to reduce the number of vulnerabilities.

Ravi et al. Ravi et al. (2004) discussed techniques to ensure software integrity such as adding hardware to support tamper resistance, secure bootstrapping, enhancing OS security, ensuring integrity of software with safety checks, as well as software authentication and validation. They describe an example of secure bootstrapping tailored to the IBM PC architecture, which exploits the layered nature of the boot process, starting from turning on the system and moving from layer to layer verifying software integrity at each one.Common approaches for hardware support rely on physical isolation, e.g., secure co-processors and memory subsystems for secure storage to which access is only allowed to trusted system components. Dedicated hardware for bus monitoring can also be used to protect the memory from illegal accesses. A bus monitoring system allows to detect illegal access to sensitive memory regions, and take suitable actions in return, such as zeroing memory areas.

Methods to enhance the security of OS, such as applying strong process isolation and attestation are also discussed in Ravi et al. (2004). Process isolation involves protecting a process' private resources from one another, whilst attestation aims to guarantee the integrity of a process before it runs. For example, a common approach is to compute a hash code of the software and verify it against a pre-computed value before it runs.

Process isolation can be achieved through sandboxing, restricted control transfers, and code origin checks. Other techniques aim to ensure that a process will not violate its security policies. These include proof-carrying code but also process shepherding, which monitors all control transfers with the aim at detecting and stopping malicious code from being executed. Programming language and software verification techniques can be used to verify the implementations and are commonly used for security protocols. They aim to detect software flaws (including vulnerabilities), ensure correctness and verify the implementation. Although such formal verification techniques have difficulty scaling to larger software implementations they can be more readily used in IoT devices where the code-base tends to be smaller.

Software and systems hardening is defined by the National Institute of Standards and Technology (NIST) (Paulsen, 2018) as the process of eliminating reasons that could attract the attackers commit attacks by turning off non-essential services and patching vulnerabilities. It is particularly important in IoT environments especially when the devices are connected to the physical world (Cyber-Physical Systems) and are used in the context of decision making for the physical world. In such circumstances safety considerations also apply. Hardening techniques for IoT devices are discussed across several studies in the literature. For example, it is recommended as good practice for IoT manufacturers to increase the robustness of their products by adding safety algorithms to overcome accidental or intentional faults and decrease their impact. This is also applied in the context of cryptographic software. For example, Zavalyshyn et al. (2020) studied the effectiveness of five common software hardening techniques applied to a lightweight block cipher, called PRESENT aimed at resource constrained environments, such as WSN and RFID (Bogdanov et al., 2007). The hardening techniques considered include classic loop hardening, variable duplication, function duplication, decryption in place, and statement based counters, and were evaluated in preventing sensitive data leaks, realising their security and their impact on software performance. Their study revealed that redundancy hardening techniques on a functional level generally provide a good balance between fault tolerance, and software security, whereas classic techniques, such as classic loop hardening, are more vulnerable. The study also indicates that, generally, hardening alone is not sufficient to mitigate faults, and needs to be combined with other techniques to increase software robustness.Systems hardening techniques are crucial to minimise attack surfaces that provide open doors or possibilities for attackers to launch attacks. Default factory usernames and passwords, and open ports are often scanned by malware to enable compromise; the Mirai botnet is only one example. An analysis of the Mirai botnet, and software hardening techniques to protect IoT devices against it are presented in Choi et al. (2018); Frank et al. (2018)) also propose a framework that leverages system hardening and security monitoring to minimise vulnerabilities in IoT devices that do not implement security by design. The authors have analysed various IoT vulnerabilities defined by the Open Web Application Security Project (OWASP), and implemented a service prototype to detect malware in infected IoT devices.

Hardening typically involves removing un-necessary functionality. However, in IoT devices, what is or is not essential is more difficult to predict at the outset, and (Zavalyshyn et al., 2020) show that some software hardening techniques can negatively impact system software and its performance. Hardware components, e.g., for tamper resistance can be effective but they add to the cost and the design of the device and also need to managed and used judiciously. On another direction, the literature presents a number of software based RA techniques that do not rely on specialized hardware components. The survey in Ankergård et al. (2021) discusses multiple software-based RA techniques, the potential and challenges of applying them to provide security in the IoT context, for example, it discussed methods to integrate RA with Blockchain to provide secure real time IoT and embedded systems (e.g., decentralization, traceability, anonymity and non-repudiation) such as Vehicle-to-Vehicle communications (Xu et al., 2018; Fortino et al., 2020).

Figure 5 summarises the main techniques mentioned so far in this section. These techniques can help considerably secure an IoT environment but often have high performance costs. Hardware based techniques can establish significant roots of trust, however this comes at a higher monetary cost to the realisation of the IoT device. Moreover, they also need to be managed.

3.3 Certification

Certificates in the form of, e.g., public or secret key certificates are a well adopted mechanism for providing a cryptographically signed proof that a certain process (e.g., verification of identity in the case of identity certificates has taken place). It would be relatively straightforward to use them to prove that a particular device has been through a certification process to prove that it has been checked for compliance with a number of requirements and has a certain level of robustness against attacks. However, the challenge is to define the process and the criteria for such certification.

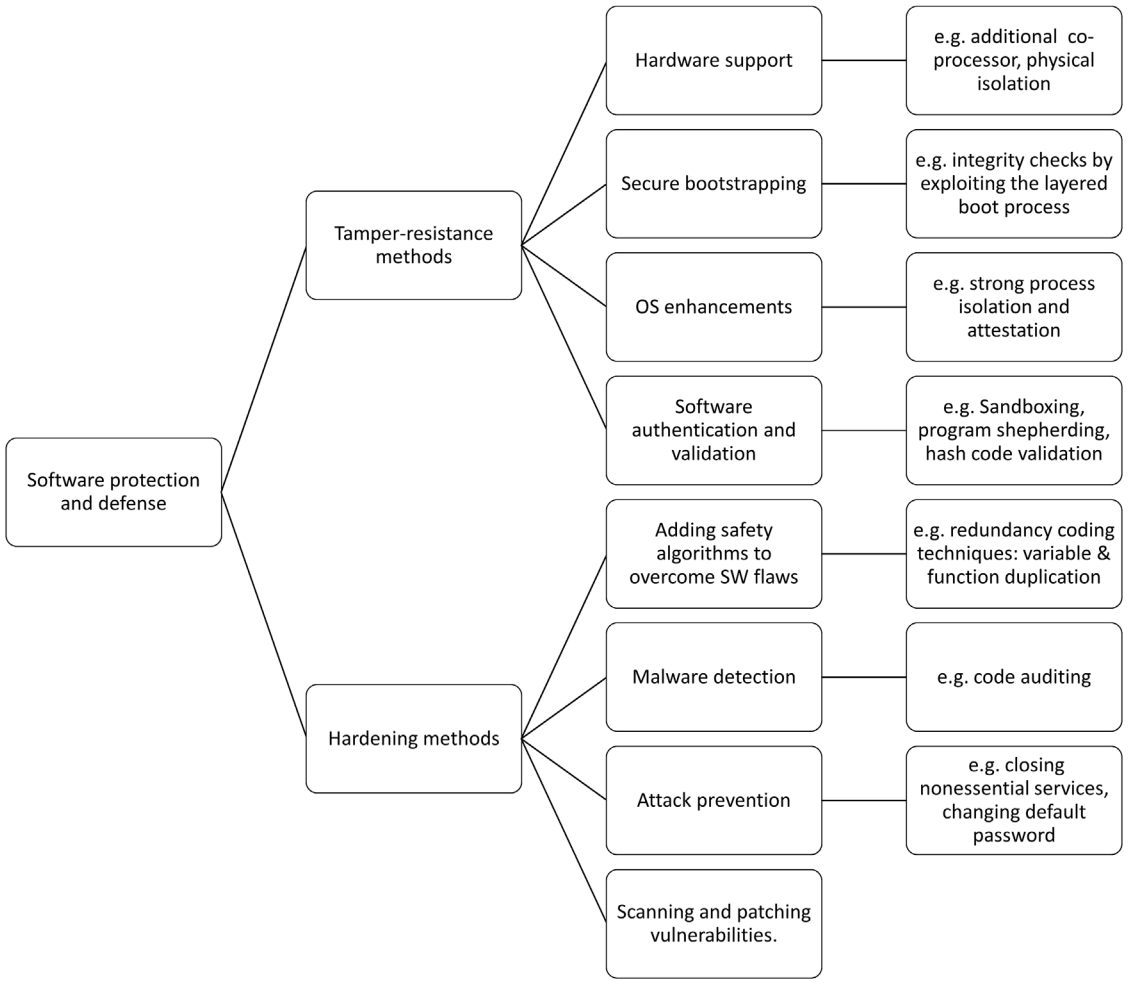

FIGURE 5
IoT device software protection and defence approaches.

Cybersecurity digital certification is defined by the Committee on National Security Systems (CNSS) a "Comprehensive evaluation of an information system component that establishes the extent to which a particular design and implementation meets a set of specified security requirements." (CNSSI, 2015). However, certification is not a trivial process, and certifying devices remains a significant challenge. In fact, there are many major questions that need to be answered in this arena, such as "who does the certification and what to certify", as discussed in Voas and Laplante, (2018); Voas and Laplante, (2018) highlight that both security and privacy might not be certifiable properties because "IoT" environments cannot be defined generically, so far there is even no clear definition for what IoT is. Moreover, there is no standardised methodology for certification, and due to the number of sectors and different environments in which IoT devices are used, defining generic certification processes would be a significant challenge. It may be possible to design such certification processes in specific sectors and for specific categories of devices, e.g., consumer IoT, however, the aim to make such certification processes generic conflicts with the need to have well specified requirements for certification as only features expected to be common can be certified. Certification is also usually a very costly and time consuming process (see, for example, certification of medical devices) thus delaying adoption and imposing upon the manufacturers singicant additional delays and costs in an ever evolving environment where the time-to-market is short and the profit margins are thin (Matheu et al., 2020b).

(Matheu et al., 2020b) provide an overview of the main cybersecurity certification schemes and discuss the challenges of adopting them for the IoT. They further discuss the efforts related to the basic building blocks processes of cybersecurity certification, as proposed by the European Telecommunications Standards Institute (ETSI) (ETSI, 2015), which include the risk assessment and testing frameworks. The survey provides recommendations to enable applying cybersecurity certification to emerging technologies, such as the IoT, and propose a multidisciplinary IoT cybersecurity certification framework that integrates research and technical tools with policies and governance structures. Going further (Matheu et al., 2020b) suggest three approaches for certification: certifying products, certifying the processes that produce the products, and certifying the people who produced the product.

Certification for IoT can have significant benefits, but it remains significantly challenging. As stated in Voas and Laplante, (2018)

"When it comes to dealing with the IoT certification quagmire, good enough is better than nothing, although good enough remains in the eye of the beholder."

3.4 Moving Target Defence (MTD)

System self-protection is crucial to improve the "immunity" of a device against threats, especially known threats, however, the static nature of the design of the software and the hardware makes it still vulnerable to attacks. For example, some IoT applications require numerous IoT devices to be deployed for environmental monitoring around the globe. Even if those devices are hardened before deployment, they will typically employ static and identical software configurations and hardware components. If an attacker succeeds in exploiting a vulnerability in one application of one device, the same vulnerability can be exploited in the many thousands of other devices that are deployed. Things get even worse as it takes in practice a long time to update and patch the software across a large number of devices, which means that the whole IoT system will be subject to attacks using that vulnerability for long time (Shrobe et al., 2018).

Moving Target Defence (MTD) has been proposed in recent years as a new type of defence that aims to leverage the attacker's lack of knowledge of the system topology and operation and aims to make it difficult for an attacker to perform the attack, by making the attack process exhausting and time consuming. To achieve this, MTD can be applied through three techniques, which all present a deterrent in space and time for the attacker to perform the attack. Those techniques are: i) randomisation, or non-determinism in the internal system while ensuring it achieves same functionality, ii) diversity, which refers to deploying non-homogeneous system components, to avoid a component from being breached by the same attack, iii) dynamism, and refers to changing the system properties regularly so to hinder same attack from compromising it in the future. Shrobe et al. (2018) describe these techniques in detail, and discuss possible methodologies to apply them with respect to the generic layered design of a computing system including the hardware, operating system, run-time environment, and data applications. Their chapter highlights the benefits, challenges and limitations of adopting MTD defence techniques in general. They propose three criteria to evaluate the effectiveness of applying MTD techniques: timeliness, unpredictability and coverage. Based on these criteria, Shrobe et al. (2018) identify several issues that need to be addressed when implementing MTD in practice. For example, one needs to consider the balance between preserving the system's performance requirements and the cost incurred by applying MTD, especially in IoT systems.

A very recent study (Mercado-Velázquez et al., 2021) targets specifically this trade-off, and proposes a framework that employs MTD by randomly shuffling the communication protocols between IoT nodes and an IoT gateway. The strategy proposed aims to balance the increased overhead resulting from the use of an MTD approach and its impact on system availability with minimising the chance attack success. The proposed framework addresses the MTD design issues, which are: "What to move," "When to move," and "How to move" aiming to use the framework as a guideline for applying MTD in IoT environments. Navas et al. (2020) go further proposing a generic modular MTD framework for IoT environments, that can be adapted to the requirements of a given IoT network. The framework adopts two MTD strategies; one that targets UDP port-hopping (having evaluated its effectiveness in a scenario where nodes were exposed to a remote Denial-of-Service attacks), and another that targets the Constrained Application Protocol (CoAP) resource URIs. Navas et al. (2021) provide a systematic review of the existing IoT MTD techniques up to July 2020. The survey categorises the techniques based on entropy-related metrics and validates the suitability of the MTD techniques to improve the resilience of IoT systems.

Overall, MTD techniques have been applied in enterprise systems but their use in an IoT environments remains tentative and requires more research. MTD techniques rely on an important assumption, that the defender has sufficient asset and configuration control that they can change their network configuration without causing any mis-configurations in the process. This is a tall order in large scale and heterogeneous environments and is very likely to be orders of magnitude more difficult for IoT in practice.

4 Mediation

Mediation mechanisms allow secure communication while the IoT devices operate in the environment by providing a communication shield between entities. First introduced by Saltzer and Schroeder in 1975 as a one of eight design principles in the context of engineering secure multi-user operating systems to support confidentiality properties for use in government and military organisations (Martin et al., 2021), *mediation* is an important concept to ensure control of access, detect anomalous behaviour and address a whole variety of issues including performance and re-configuration. In IoT environments mediation of interactions plays a significantly more important role given the heterogeneity of devices, their resource constraints and the frequent changes to the system configuration caused by new devices joining, leaving or dynamically connecting to the system. Devices such as IoT Gateways are currently deployed, and the process of interaction mediation goes far beyond the simple traffic filtering commonly encountered in other environments (Davies et al., 2016).

Given the inherent distributed nature of IoT environments distribution aspects are important. The traditional design dichotomy between: i) centralised mediation and ii) distributed mediation is more present than ever as system designers struggle to resolve the tensions between autonomy, compositionality, control and coordination across a wide array of applications. Centralised mediation is often encountered in systems that see IoT as a separate sub-system that needs to be integrated into the broader Enterprise or Internet architecture. De-centralised solutions emphasise autonomy of control, scaleability and, increasingly, transparency. Naturally, each of the paradigms has its advantages and disadvantages. Centralised mediation is preformed by an entity, usually a resource rich device, called an IoT gateway or an edge device managing the IoT environment and mediating interactions with it. The gateway is responsible for performing varied and complex operations, which cannot be distributed easily to the IoT devices without draining their resources, however, it introduces a

requirement for the devices to communicate frequently with the gateway, which also can have a high energy cost. Distributed architectures tend to be aimed for devices with more computational and power resources that can coordinate and control their interactions with the environment according to well defined (but possibly dynamically changing) policies.

4.1 IoT gateway or an edge

Several studies discuss a centralised mediation in IoT systems, mostly from the perspective of designing a gateway (IoT edge device) that is not only designed to provide management, interoperability (for example, through semantic mediation) and a bridge to allow communication between heterogeneous IoT device, but also to be responsible for securing the IoT system. For example, Leo et al. (2014) introduce a Secure Mediation Gateway (SMGW) for smart home environments that acts as middleware solution to control communication between heterogeneous IoT devices. The SMGW provides secure communications among heterogeneous IoT nodes by means of a federated network comprising two groups, the intraSMGW group and the interSMGWs group. The SMGW acts as a boundary entity between the groups, and allows secure communications among entities within the intraSMGW group and among interSMGW domains. SMGW enables secure remote access to remote devices, and allows remote devices to access a device within the intraSMGW using a secure end-to-end link that adopts a public key and a digital signature encryption schemes. The security services provided by the SMGW include the authentication, authorisation, confidentiality, and integrity for publication and subscription of services in intraSMGW and interSMGW domains. The work in Mahmoud et al. (2015), which is based on a previous studies (Castrucci et al., 2012) similarly proposes a gateway that provides an abstraction to allow information mediation. Its functionality is further extended to enable heterogeneous Critical Infrastructures (CIs) to predict service failures.

Gateways are also proposed as a solution to address privacy concerns. Davies et al. (2016) address the issue of privacy and the lack of user control over their local data by proposing a locally-controlled software component called a *privacy mediator* located in the local domain of the sensor devices. The privacy mediator dynamically enforces the privacy policies of the owners of the IoT devices within their local domain, before data is released from the user's local control. The proposed approach delivers a secure solution at the edge of the cloud, while reducing the privacy burden on the application developers, by involving a small set of trusted parties that provide the mediation code. The framework provides a scaleable solution that can support IoT applications with high data rates.

Gateways are also seen as a means to address the challenges of heterogeneous communication. Chio et al. (2019) proposes a mediation-based architecture that supports protocol translation to bridge the heterogeneity gap in the IoT network. Although the work does not address issues related to cybersecurity in particular, it tackles the challenge of mediator placement in IoT system. This issue is crucial to solve in order to achieve a timely data exchange between IoT devices. The authors suggest an adaptive placement of mediators through an integer linear programming algorithm that takes into account network resources and IoT device attributes, such as bandwidth, and data size constraints.

4.2 Continuous monitoring and analysis

Mediation of communication does not only enable controls to be enforced but also to monitor interactions to identify anomalies and identify illegal intrusions through the operation of the system. Continuous monitoring is required to make sure a device continue operating in a safe environment. Frequently, and increasingly this is done with the help of Artificial Intelligence (AI) techniques, such as machine learning techniques that can learn "normal" behaviour and identify anomalous patterns.

4.2.1 Intrusion detection systems (IDS)

System monitoring typically involves integrating Intrusion Detection Systems (IDS), which are usually combined with IoT Security analytics to monitor the network against malicious system traffic at the network level. There are numerous studies and surveys summarising the work conducted in this area.

(Debar et al., 1999) provides a comprehensive taxonomy for the legacy IDS(s) in networks in general. However, it is important to point out that intrusion and anomaly detection in IoT systems is more challenging than in traditional networks. One reason for this is the diversity of devices and protocols employed, another is the diversity of contexts in which the devices operate, thus leading to large variations in the traffic patterns. Finally, the dynamic evolution of the system, e.g., through new devices being added or leaving introduces a further dimension of variation.

(Zarpelão et al., 2017) provide a comprehensive survey and taxonomy of IDS(s) in IoT. Generally speaking, IDS(s) are classified as Network based IDS (NIDS) and Host based IDS (HIDS), where NIDSs monitor network traffic connecting one or more network groups or segments and HIDSs are integrated into end devices and monitor system activities. Zarpelão et al. (2017) further classify IDS(s) according to the IDS detection and placement approaches and the detection methods being applied.

4.2.2 Honeypots

Applying IDS should be combined with applying honeypots, to form what is called the "Intrusion Detection Honeypot". A honeypot is a trap used to attract attackers to exploit fake vulnerabilities in order to gain access to a fake target. Not only does this provides early warnings once a threat is detected, but also helps in collecting, analysing and identifying information about attackers and their activities. Spitzner in Spitzner, (2003) best defined a honeypot as "a resource whose value lies in being probed, attacked or compromised". There are different types of honeypots, and they can be categorised in different ways; a common approach is to consider their level of interaction with the attacker. High-interaction

honeypots allow full access to a "real" system, i.e., one that reproduces with a high fidelity an actual real system's operation and data. Low and medium interaction honeypots reproduce the system's behaviour and data with lower fidelity but as a result are easier to set-up and maintain (Vetterl, 2020; Franco et al., 2021). This makes them very convenient to use for detecting and gathering information about large scale attacks.

Honeynets comprise multiple honeypots deployed within the same system. Franco et al. (2021) provide a comprehensive review of the literature related to honeypots and honeynets for IoT environments, Industrial Internet of Things (IIoT), and Cyber-Physical Systems (CPS) over the period 2002–2020. The survey introduces a taxonomy of honeypots and honeynets based on their: purpose, role, level of interaction, scaleability, resource level, availability of source code and target (IoT, IIoT, or CPS application). Vetterl, (2020) also investigate low and medium interaction honeypots proposed in literature but also design a high interaction honeypot for IoT and Customer-premises equipment (CPE) called *Honware*. *Honware* can be implemented and deployed rapidly as it extracts firmware images automatically to emulate the device's behaviour in a virtual environment. Its evaluation revealed that it is able to detect known and unknown attacks effectively. *IoTPOT* is another newly developed high-interaction honeypot system designed to mimic IoT devices, and is used to analyse the behaviour of botnets Pa et al. (2015).

In contrast to IDS, a honeypot does not have any legitimate traffic, as there is no reason for legitimate traffic to connect to the honeypot. This means that malicious activities are the only ones honeypots can detect, and any activity with the honeypot is likely to be a threat or an intrusion attempt, leading to a low false positive rate. This stands in contrast to the traditional IDS which can generate a high level of false alerts. This also means that with using honeypots, the real asset may not be touched at all by the attackers leading to a more secure system. Additionally, honeypots provides a mean for distracting the attacker from going forward towards the actual target by spending more time in trying to exploit the wrong system. Not only this would exhaust the attacker's resources, but would give more time for the security experts and system administrators to identify attacker's intent and techniques and to isolate the actual target. However, there are no guarantees that attackers will "fall for it".

If implemented or deployed inadequately, honeypots may actually increase the risk to the system as they can attract attackers. For this reason and to combine their efficacy at isolating attacks with the need to reduce the false positive rates of IDS(s), honeypots should always be introduced in conjunction with rather than as an alternative to other security approaches such as IDS and firewalls.

Note also that low interaction honeypots are easily detected by attackers. For example, Vetterl, (2020) report that their study was able to fingerprint a large internet-scale low and medium interaction honeypots, which use off-the-shelf libraries for protocols implementation. The study also found that a large number of honeypots are out of date, and their operators use a standardised deployment scripts, which facilitates detecting them. For these reasons, the study recommends the development of a new generation of honeypots, which focuses on the lower levels of the network stack.

4.3 Artificial Intelligence (AI) techniques for monitoring and detection

Nowadays, combining AI with security methods is a favoured approach to dynamically and accurately secure systems against threats. This is due to rapid evolution of attacks in the IoT environment in particular, and the need to protect these systems intelligently and in real-time. For example, it is increasingly popular to combine IDS(s) with machine learning (ML) approaches, that help updating the system autonomously in order to detect and prevent new type of attacks (Kumar et al., 2021).

(Kuzlu et al., 2021) discuss the role of incorporating AI in IoT cybersecurity, and the use of machine learning methods such as decision trees, K-nearest neighbours, support vector machines, and neural networks to detect attacks in IoT environments. Similarly, Chaabouni et al. (2019) discuss NIDS techniques employing ML for IoT environments and provide a comprehensive review of NIDS(s) that integrates different aspects of learning techniques for IoT. Kumar et al. (2021) propose a *unified intrusion detection system for IoT environment (UIDS)* to defend the network from attacks also based on ML approaches. The proposed IDS model acts as the watchdog in the IoT based system to prevent the system from internal and external malicious attacks. Meidan et al. (2018) present a NIDS method that uses deep auto-encoders to detect anomalies in the IoT traffic. The approach is evaluated empirically on different commercial IoT devices considering IoT botnets such as *Mirai* and *BASHLIT*. Pacheco et al. (2019) propose an IoT threat framework with a neural network model that is able to detect potential attacks against each layer of the framework. They create a reference model for end nodes automatically and perform behaviour analysis at run-time. The authors claim that the framework recognises both known and unknown threats with high detection rate and low false positive alarms.

Honeypots have been evolved to include AI and ML techniques to interact with the attacker and decide the actions to be taken, instead of relying on human pre-programmed traps. These are referred to as *"self-Adaptive" honeypots*. For example, Pauna et al. (2019) propose a self-adaptive IoT honeypot system that interacts with the attackers based on a set of actions triggered by a reinforcement learning algorithm. They test two behaviours that partially map the Mirai botnet.

Despite their benefits, AI techniques usually requires complex operations and therefore requires resource rich hosts, which limits their usage in the endpoints of the network. Moreover, AI techniques vulnerable to attacks, and can even "weaponise" the AI itself against the system. Methods that exploit AI to attack IoT systems are discussed in Kuzlu et al. (2021).

5 Device(s) isolation and system segmentation

When an attack is detected, in order to eliminate it from spreading further into the rest of the IoT system and other connected systems, then the infected devices and all (or part) of the devices reachable from the compromised nodes should be isolated from the network. Putting devices into quarantine thus seeks to eliminate the threat from infecting the rest of the network

and further the damage, whilst allowing the rest of the network to continue working after having being partially compromised (Castiglione and Lupu, 2020).

This is, however, a delicate approach, as inter-dependencies between devices within the IoT system and to other parts of the system can aggravate the effect of isolation. Identifying the dependencies between IoT system components and other systems, and modelling and analysing the system risk, are required to evaluate the level of threat across the entire network, and prioritise which reachable devices need to be isolated and which are not. This helps to perform device isolation without unduly degrading the performance of the system as a whole. Segmenting and segregating the network is a always a good practice to mitigate risks when designing a network, but one has to make sure that the benefit gained from isolating the infected devices exceeds the losses (Soikkeli et al., 2019). Therefore, the first design requirement to be considered when constructing the system, and before segmenting the network is to identify and document IoT dependencies.

5.1 Identifying and documenting IoT dependencies

IoT attacks typically have cascading effects on the system, due to the inter-dependencies between the system's devices and components. In many cases, IoT systems are basically systems of systems or systems connected to critical infrastructure (CI). Thus, failures in one system leads to disruptions in other systems, which can unpredictably proceed further to causing severe unexpected or undesirable consequences (Stellios et al., 2018; Xing, 2021).

Despite their critically, especially in CI or with systems connected to CI, systems inter-dependency and possible failure mechanisms are not well understood and have only recently begun to receive attention in the IoT community (Little, 2003). identifies three classes of infrastructure failure effects due to inter-dependencies between systems: i) The cascading effect, when a disruption in one system causes a disruption in another, ii) the escalating effect, when a disruption in one system exacerbates an independent disruption of another, and iii) the Common cause effect, when a disruption of two or more systems occurs at the same time because of a common cause. Xing, (2021) reviews the literature on cascading failures in IoT systems, their major causes (one of which is the cyber-attacks), their mathematical models and simulations, and mitigation techniques in IoT systems that take into account cascading failures from a reliability and resilience perspective. They consider how IoT system inter-dependencies can affect the resilience of the system. The article also sheds light on the differences and relation between reliability and resilience and states that reliability is the end goal of system design while resilience is the way in which the end goal can be achieved.

(Stergiopoulos et al., 2020) model the inter-dependencies between assets and devices in IoT environments, and develop a framework that presents those connections in a company's business processes. The framework discusses automation in security risk analysis and restructuring IoT systems, by leveraging dependency risk graphs, graph minimum spanning trees, and network centrality metrics to model the networks' dependencies. The framework was tested on a real-world company, and proved its ability to automatically identify critical components and dependency structural risks, prioritise assets based on their impact on business processes and propose network topologies with the optimum number of asset sub-nets, while preserving business operations. Luiijf and Klaver, (2021) discuss Critical Information Infrastructure (CII) disruptions, dependencies, and cascading effects based on empirical data.

5.2 Network segmentation

Identifying inter-dependencies helps to identify critical system devices, prioritise assets and document inter-connections between those components and other systems and critical infrastructures. Such information is critical to determine the impact of isolating a device or a group of devices when threats occur, or in order to avoid cascading disruptions to the system. Isolating groups of devices in a network, in its broader context, is known as *network segmentation*. Generally speaking, segmentation is partitioning the network, system or environment into smaller groups or networks (segments), sometimes even down to the host itself.

Network segmentation is a well known practice commonly applied in legacy networks before the emergence of IoT, and can help contain security breaches to a single (or a few) infected segments, thus preventing the threat from spreading further into the other parts of the network. The Information Assurance Directorate (IAD) identifies two approaches for network segmentation (Wagner et al., 2016).

1. Segregate Networks and Functions (SNF), which refers to partitioning the network into groups or segments based on their functionality or intended services, and limiting the communications between the segments. This hinders the attacker, who managed to gain access into the network, from causing harm by acquiring further access into the rest of the network.
2. Limit Workstation-to-Workstation Communication (LWC), which refers to controlling communications on a more granular level inside a segment, between segments and between segments and the Wide Area Network (WAN). LWC grants communication privileges only when necessary by enforcing the least privilege principle.

Legacy practical approaches to network segmentation include both physical and logical methods. The physical approach uses multiple firewalls, for example, to separate financial applications from medical devices and IT applications within a hospital. This approach remains expensive and requires complex separations with thousands of firewall rules to segment internal networks. Logical segmentation on the other hand, is conducted with VLANs, whose configuration is equally labour intensive. Network segmentation is traditionally conducted as a proactive approach once the network is constructed. For the enterprise networks, a number of studies have investigated means of automating network segmentation. Wagner et al. (2017c), Wagner et al. (2016), Wagner et al. (2017b); Wagner et al. (2017a); Hemberg et al. (2018).

Due to the unpredictable and highly dynamic nature of IoT environments, IoT network segmentation offers a promising approach to defending them by adapting the system in real time and implementing reactive approaches through *dynamic segmentation*. However, very few work studies tackle dynamic network segmentation, e.g., Wagner et al. (2017b), and even then are oriented to Enterprise systems. The approach proposed in Wagner et al. (2017b) generates the segmentation architecture dynamically when threat level changes, based on a nature-inspired process that investigates different solutions using Simulated Annealing (SA). In their proposed technique, Wagner et al. (2017b) adopts a continuous time Markov chain model (CTM) to evaluate the risk in a network environment, and generates architectures adaptively by adding, removing and altering the existing services in order to combine or split enclaves containing the devices. However, the work only considers attacks originating from the Internet, whilst in the context of IoT, new (compromised or infecting) devices could dynamically join and leave the network enclaves.

As mentioned earlier, network segmentation has not been applied broadly to IoT networks. There are a number of reasons that discouraged the implementation rate of network segmentation in IoT systems. One of which is the administrative complexity of applying segmentation by the average user, specially in personal settings, such as personal or home networks. Another, is that network segmentation approaches require significant manual and expert effort. Dynamic segmentation techniques such as those mentioned earlier cannot easily scale to large network and they cannot cope with the dynamicity of the IoT environments. IoT devices frequently exchange information with each others and with the cloud, breaking the perimeters usually secured by residential firewalls or gateways (Wasicek, 2020).

More importantly, IoT devices are always active in a highly connected environment, where they are hardly managed or patched against the latest security updates, which all lead to an increased system attack surface. If an attacker manages to pass the residential firewall, which in many cases is the only firewall in the network, and succeeds in exploiting one entry point in the system device, the attacker then perform lateral movements to connect and infect other devices in the system without restriction. Lowering the manual effort to perform segmentation is key, but research approaches on dynamic network segmentation still rely on complex and highly extensive network simulations that do not scale well.

5.3 Host segmentation (microsegmentation)

Classical network segmentation approaches based on firewalls, VLANs, *etc.*, have difficulty coping with the dynamic nature of IoT environments where devices can frequently join and leave the network dynamically. At the same time the network infrastructure is becoming more programmable, e.g., through Software Defined Networking (SDN) and Network Function Virtualisation (NFV), while edge networks such as home networks are increasingly supported by an *Edge Cloud Network* such as residential gateways in 5G networks. By adopting microsegmentation, devices in IoT systems can be individually characterised and isolated to prevent a weak device from serving as an entry point to the entire network. While both network segmentation and microsegmentation control the flow of traffic between network segments and application components based on security rules, microsegmentation works at a finer level of granularity segmenting the individual host workloads throughout the life-cycle of the device from it first joins the network until it is disconnected. Therefore, microsegmentation is often referred to as "host-based segmentation" rather than "network segmentation".

Micro-segmentation establishes a network inventory of all the devices. When a device joins the network it is fingerprinted and scanned for vulnerabilities and registered in a *network inventory* virtual network function. By default, the wireless traffic is configured to run in client isolation mode and a *micro-segmenter* reprograms the smart gateway via a protocol such as OpenFlow. Wasicek, (2020) highlights the importance of applying microsegmentation in 5G enabled-smart home IoT environments, so as to reduce the attack surface and protect such environments from internal compromise involving lateral movements. The system realised in the context of home networks is described in Osman et al. (2020). The authors implemented microsegmentation in an emulated network topology, which included both IoT and non-IoT devices belonging to six functional groups (energy, management, controller/hubs, cameras, appliances, health monitors and non-IoT). The adopted microsegmentation approach identified and quarantined infected devices from accessing the LAN and WAN, whilst the non-malicious devices were automatically classified based on functionality and assigned to confined network micro-segments accordingly. The work found that microsegmentation reduces the attack surface exposed to a webcam infected with Mirai botnet by 65.85% compared to the baseline configuration, which was at the expense of preventing 2.16% valid network flows between devices. This deviation resulted from flows that would cross the functional micro-segments.

Mämmelä et al. Mämmelä et al. (2016) also discuss the concept of microsegmentation and its integration in 5G, and compare it with the concept of network slicing. The authors implemented a virtual microsegmentation, and a network slicing approach in an experimental test bed, which includes personal IoT health applications, and discuss the adminsitration of the micro-segments. This work also discusses the different trust models that can be adopted, such as the now popular *Zero Trust* model. Inherently, *Zero Trust* adopts a least privilege approach, and is best implemented in IoT applications with critical services, such as e-health. Different authentication and verification levels can be adopted in micro-segmented networks, and the authors suggest that micro-segments could have different security levels depending on the application and the service provided. For example, micro-segments for critical applications such as e-health could operate with different security requirements compared to non critical IoT micro-segments.

Network Segmentation and *Micro-Segmentation* are complementary approaches as they operate at different levels. Both rely on the principle of mediating network access and inter-connectivity and increasingly rely on the dynamic programmability brought about by techniques such as Software Defined Networking (SDN) and Network Function Virtualisation (NFV).

5.4 Software defined networks (SDN)

The application of SDNs techniques in IoT scenarios has attracted a significant interest in recent years. Baldini et al. (2020), present a light-weight policy-based approach designed as part of the *SerIoT* Research and Innovation Project, funded by the European Commission. The proposed system called *Autopolicy*, proposes an architecture for enforcing different traffic profiles according to the intended communications of an IoT system with other devices or systems in order to mitigate potential security risks. The *Autopolicy* system integrates a distributed machine learning approach based on deep learning (DL) and graph networks for risk monitoring, which analyses the network traffic in real-time. This SDN and AI integration realises a *Self-Awareness* that can achieve secure and QoS-based routing of traffic flows. Such efforts are aligned with the recent MUD standard, mentioned earlier.

(García et al., 2019) propose an architecture for managing, obtaining and enforcing MUD restrictions on SDN switches. The work analyses the applicability and advantages of using MUD in industrial environments, and provides a comprehensive performance evaluation of the processes required. Zarca et al. (2019) design an architecture that captures the security and privacy requirements in cyber-physical systems and IoT-CIs and make autonomous security decisions and re-configures the network using SDN and NFV. The reconfiguration in response to IoT threats is done through special purpose IoT agents, SDN and IoT controllers as well as NFV equipment, which enforce security countermeasures and dynamically adapt the system according to the context analysed by the integrated monitoring tools. This architecture has been implemented and evaluated in smart buildings, as part of the *ANASTACIA* H2020 EU research project. For more information about SDN and NFV, I refer the reader to (Alonso et al., 2019), which discusses extensively the characteristics of SDN and NFV and their potential to secure IoT environments.

Whilst SDN and Micro-segmentation leverage mediation and programability as the main characteristics to segregate and ensure security in dynamic IoT environments, this comes at a significant cost in complexity and management of this programmable network environment. Two aspects remain, in my view, as yet less well understood. Firstly, is it possible to adapt such dynamic configurations to the actual network and device usage in large environments? Device function may not fit neatly into segmented boundaries and may also change over time. Secondly, what are the vulnerabilities and potential for new threats introduced by such techniques? Their potential for being abused is not investigated as thoroughly as their use in support of security functions.

6 System availability, redundancy and resilience

As we increasingly rely on IoT for sensing the physical environment and adapting to it, we rely on the *availability* of the system even when it is subject to threats and compromise. Ensuring the *resilience* of the system entails minimising its loss of function over time and thus maximising its availability including when under threat, whilst at the same time ensuring the integrity of the functions delivered. Such resilience must be ensured for the system itself as well as for the data it delivers and on which decisions are based. To ensure resilience, especially in an IoT system where devices can be trusted to various extents, requires redundancy to absorb failures and compromises. This is particularity important if the system incorporate sense sensitive data.

6.1 Redundancy vs diversity

Redundancy is commonly applied in *fault-tolerant systems*. Applying redundancy means deploying additional system components identical to other systems' components. This method guarantees the system availability even if fault occurs in the original component. However, having identical redundant components is not helpful if the system is exposed to malicious interventions as identical components will also share the same vulnerabilities. If an attacker manages to exploit a vulnerability in one component, they can easily exploit the same vulnerability in its identical copies for negligible additional cost. Redundancy must therefore be coupled with ensuring *diversity* in the redundant resources, seeking to minimise the shared vulnerabilities. Note that a different form of diversity, in the network configuration rather than in the component provision, is ensured by *MTD* approaches discussed earlier in Section 3.4.

Redundancy, diversity and other key enablers for system resilience are discussed in Dobson et al. (2019), which focuses on the design principles in the area of self-organisation and resilience of networked systems. The authors have also previously published a survey, which discusses redundancy, besides diversity and connectivity as key enablers to achieve network resilience in terms of fault-tolerance, survivability, and performability (Sterbenz et al., 2014). The survey highlights that diversity is essential to provide survivability in addition to using redundancy for fault tolerance and connectivity which is recommended for disruption tolerance. Diversity provides alternatives, which can run simultaneously or as needed, so that even when a particular alternative is compromised, other alternatives can provide some degree of functionality. The survey also introduces a cross layering model that leverages these techniques to achieve a resilient network system.

Similarly, Laszka et al. (2018) propose a framework that leverages redundancy, diversity, and hardening techniques to improve the security and resilience in IIoT. The work evaluated the applicability of the framework in a water-distribution system, and their results revealed that integrating redundancy, diversity, and hardening helps reduced security risk whilst maintaining the same cost. Venkatakrishnan and Vouk, (2016) also investigate the use of redundancy with diversity, but focus on detecting "difficult-to-detect" attacks on web servers deployed as part of IoT applications. They conclude that, although the redundancy-based detection technique maybe costly, it is worth considering in IoT environment, especially when the system needs to operate for long periods of time without direct human intervention. The work also recommends masking design differences when adopting a redundancy-based detection approach.

Despite the benefits of having additional diverse or redundant resources in improving the resilience of the IoT system, applying

those methods can be costly, especially in applications where IoT networks are connected to critical infrastructures. Applying diversity also significantly increases system complexity (Linkov and Kott, 2019), as it introduces additional heterogeneity and makes it harder to manage the system in a uniform way. The following subsections discuss alternative methods to account for data loss, that have potential in-providing a form of data backup without increasing the cost.

6.2 Exploiting data correlation

Data resilience seeks to leverage data redundancy in order to ensure data integrity and availability. Redundancy in this context can manifest through monitoring the same measurements through different sensors (e.g., in Wireless Sensor Networks) or through correlations in the measurements made. Such correlations can be in time, in space, or across the different attributes measured by the sensors. For example, in Wireless Body Area Networks (WBAN), both the ECG and hemodynamic signals, such as blood pressure, have information mutually correlated due to the physiological inter-relation of the mechanical and electrical functions of the heart (Salayma et al., 2017). Such correlations enable the detection of measurement manipulation, an area known under slightly different terms *data spoofing*, *malicious data injection* or *false data injection* in different research communities. For a survey of such approaches in Wireless Sensor Networks see (Illiano and Lupu, 2015).

Whilst a number of techniques have been developed for correlations in time series and further techniques have been developed for exploiting spatial correlations (Illiano et al., 2017), the topic remains less investigated in IoT applications where few sensors can be deployed and they measure heterogeneous information. Information fusion, and the ability to exploit at the same time temporal, spatial and attribute correlations remains to be investigated further.

Exploiting data redundancy has a cost, more specifically that of measuring, transmitting and aggregating correlated, and thus redundant information. There is therefore a tension between the resilience requirement in IoT applications and the requirement to minimise function and energy costs. How to characterise this trade-off in a generic way is something that needs to be further investigated. Another consideration arises from the necessary contextualisation of the measurements and the information acquired. The correlations between the measurements vary with the physical phenomena being measured and with the deployment context of the WSN. For example, in healthcare applications the correlations vary according to the activities undertaken by the patient. In fixed, wireless sensor networks, e.g., such as those monitoring temperature or volcanic eruptions, the correlations vary when an eruption happens (Illiano et al., 2017). Establishing a base-line in a network guaranteed to be uncompromised, remains a significant challenge.

6.3 Exploiting data overhearing

Overhearing is a phenomenon usually occurring in wireless sensor networks (WSN), where each active, or idle sensor node overhears data not designated to itself but sent by neighbouring nodes. Overhearing is, in essence, a side effect of the broadcast nature of the wireless medium. Although this phenomenon has often been considered as drawback in WSN, as it causes extra energy consumption and therefore leads to fast energy depletion, data overhearing can be exploited to overcome data loss as it can provide a form of redundancy. This allows two or more network devices to help each other in transmitting the information to a common destination. Such approaches are commonly referred to as *cooperative network*. This cooperation between network nodes can be useful especially in critical IoT applications where data has to be delivered reliably and with high priority. If an attack causes data loss in such applications, overhearing can be exploited to account for the lost data. Despite its potential to ensure the reliability and resilience of data transmission this area remains under investigated, especially in the context of cyber attacks.

(Rosenberg and Reinhardt, 2021) investigate the potential of exploiting redundant information using collaborative methods to mitigate data loss in LoRa network. In this work, a centralised gateway encodes data frames sent from nodes, which combine their neighbours overheard data into their own transmissions. Two collaborative approaches are proposed: i) Neighbour Data Re-transmission (NDR) in which a node appends its most recently overheard frame to its subsequent frame to be transmitted, and the ii) Combined Data Re-transmissions (CDR), in which a node combines its most recently overheard frame and its subsequent frame using a bitwise exclusive-OR (XOR) i. e., it applies Network Coding (NC). The work evaluated its potential to mitigate data loss through both simulations and empirical studies.

Overhearing is also exploited to detect malicious data injections. In essence, each node overhears the transmissions of their neighbours and compares the measurements reported with the ones they have measured themselves. Trust-based algorithms have then been proposed to allow sensor nodes to vote on the trustworthiness of their neighbours based on the results of this comparison. Such approaches have been often proposed in the literature but suffer from an intrinsic limitation: a compromised node can lie about the measurements or about the trustworthiness of its neighbours or both. Distinguishing attacks on the basis of data analysis alone becomes then very difficult.

Overhearing also introduces a tension between the data availability and confidentiality objectives. Overhearing allows data to be replicated and re-transmitted but this offers attackers more opportunities to access the data through sensor or network compromise. Finally, overhearing introduces additional energy costs both to listen to the neighbours' transmissions and to re-transmit the data.

6.4 Network coding (NC)

Network Coding is commonly covered in information theory, and has many practical applications in networking systems. It is often used to increase the throughput and robustness of a network by using diverse paths for different packet combinations (Ahlswede et al., 2000). Using NC, the nodes recombine input packets into one or more output packets. This allows intermediate nodes to transmit packets that are linear combinations of the previously received

packets instead of mere forwarding packet by packet to the same destination. Hence, NC provides a form of data redundancy that not only can be used to mitigate data loss in lossy links, but also to recover lost data whilst reducing the energy costs required by the transmission. In particular, it avoids unnecessary packets re-transmission and acknowledgements. As a result, the application of network coding in Internet of things can also contribute to "green IoT networking", whilst providing a degree of resilience (Li et al., 2017).

(Ar-Reyouchi et al., 2020) proposes a network coding-based protocol (NCBP) to detect and recover lost packets and correct errors in an IoT networks. NCBP is based on a random network coding (RNC) scheme that allows nodes to generate linear combination of input packets into coded packets separately over a finite field. The work compares the performance of NCBP with that of legacy algorithms such Forward Error Correction (FEC) and Automatic Repeat Request (ARQ). Their results show that NCBP can effectively recover lost data, increases throughput with less re-transmissions and minimising delay, bandwidth and energy consumption, thus emphasising its suitability in IoT environments.

(Liao et al., 2019) investigates the use of NC for data recovery in IoT networks focusing on the detection and storage requirements. The work proposes an eavesdropping prevention technique that combines NC and Recurrent Neural Network (RNN), and models an optimisation problem for minimising device storage subject to meeting a set of security requirements. The work proposes two allocation algorithms for IoT device storage failure prediction; the Failure-Aware Greedy Allocation (FAGA) and the Failure-and-Load-Aware Greedy Allocation (FLAGA). It evaluates the performance of the proposed techniques using a real dataset, and shows that the proposed technique can meet strong security requirements.

The main drawback of NC-based techniques is that they are based on trust among nodes. It is easy for any malicious node to join the network and act as an intermediate node that could forge encoded packets. The receiver may not be able to detect attacks, and hence may not be able to recover the original data packet correctly under pollution attacks, but will attempt to reconstruct from the wrong data. What makes it even more difficult to distinguish the valid encoded packets from malicious ones, is that the packets received by the receiver are combined with several other packets originating from multiple sources. In an attempt to address this issue, Lee and Lee, (2020) propose a method to detect compromised packets among the packets received at the receiver in an IoT environment that adopts NC for data recovery. Their detection method enables the receiver to identify the valid packet amongst the "look-like-valid" packets without requiring re-transmissions. The proposed scheme shows that a receiver can recover a valid packet with a high probability when there is sufficient redundancy.

7 Conclusion and future trends

The security and resilience of IoT environments remains a complex topic that spans across the entirety of their life-cycle from design and realisation to their deployment operation and decommissioning. The challenges are being addressed in the design of individual devices and in the design and operation of deployments. Like in the case of enterprise or more traditional computing environments, new techniques are being developed to make devices more trustworthy and new techniques are being developed to make systems more resilient and trustworthy by detecting, mitigating and responding to threats at run-time.

However, the IoT presents unique challenges that do not encumber the development of solutions in traditional systems to the same extent. The heterogeneity of devices and contexts of use make it difficult to develop and apply common standards, e.g., for design, deployment or certification and to develop general solutions. The increased dynamicity of the environment makes statically planned architectures and frameworks more difficult to employ. Dynamic adaptation and continuous risk management are required. The impossibility of guaranteeing security and robustness in light of the ever-growing attack surface due to IoT adoption slightly shifts the focus from security to resilience; from trying to ensure that a system is robust to trying to ensure that it can continue to operate, even when it has been partially compromised. This trend is further emphasised by the connection of the IoT to the physical world. System availability becomes more important as inter-dependencies across systems mean that disrupted operation can have significant cascading effects. The connection to the physical world also makes contextualisation a much more difficult challenge as systems must accommodate a variety of contexts of use, themselves evolving over time.

This survey has collated current research into risk and threat mitigations in the IoT environment, providing an overview and presenting these uniquely throughout the life cycle of the IoT device, starting from its design, to the moment when a device joins a network, while the IoT device operates in the network, and until it leaves (or is removed) from the network and is eventually decommissioned, hence adopting a "defence-in-depth" approach. A summary of the discussed mitigation techniques is presented in Table 1. The survey has discussed threat mitigation techniques applied across different IoT application contexts, and has elaborated on how to apply each mitigation technique, its benefits and limitations and the extent to which progress has been reported in the literature.

The research landscape is heavily dominated by technologies and more importantly their adoption and acceptance. From trusted platforms such IntelSGX, to blockchain, AI, SDN and virtualisation, the application of new technologies forms the focus of many research investigations. Such technologies make it possible to implement old principles that have withstood the test of time such as mediation, isolation, detection, remediation, on which defences will continue to be based. Their adoption and use make it possible sometimes to attempt to address some of the challenges specific to the IoT.

However, this is not enough. Security and resilience are emergent properties and more so in dynamic IoT systems than in enterprise systems and traditional networked environments. But these emergent properties need to arise from addressing inherently conflicting goals and requirements that are more stringent in IoT environments. Confidentiality and privacy can conflict with availability. Distribution and redundancy conflict with resource usage particularly in end devices, which often are very constrained in their energy consumption. Cost pressures and low

margins make the deployment of hardware security solutions more difficult. Due to the pervasiveness of the IoT and its application across an infinite spectrum of applications and usages, it is difficult to characterise well the trade-offs underpinning these conflicts and develop more generic solutions, or at least patterns for such solutions.

The usage of new technologies also has a side-effect: their vulnerability; they themselves are vulnerable to malicious attacks. This has been amply demonstrated with AI, with smart-contracts on the block-chain, with Intel SGX. The robustness of SDN to adversarial attacks remains under-researched. Ensuring the robustness of new technologies to adversarial attacks is sometimes far from trivial, the case of AI/Machine Learning standing as a particularly prominent example. The attack surface of Machine Learning algorithms and how to make them more robust remain fundamental research challenges. Yet, there is frequently more enthusiasm for the adoption of new technologies and their use towards ensuring security and resilience, than there is in ensuring their robustness. This is, at least in part, driven by the market pressures to innovate.

However, progress is being made across all areas: in trusted platforms and secure hardware, in formal verification, in architectures that offer better mediation at a finer level of granularity (e.g., microsegmentation), in leveraging redundancy for resilience. Many of the solutions developed for more traditional computing settings are being extended and transferred to the IoT domain. But in doing so the dynamcity and contextualisation of IoT systems prove to be a major challenge.

Data availability statement

The original contributions presented in the study are included in the article/Supplementary material, further inquiries can be directed to the corresponding author.

Author contributions

MS: Funding acquisition, Formal Analysis, Writing–original draft.

Funding

The author declares financial support was received for the research and authorship of this article. This work was supported by PETRAS National Centre of Excellence for IoT Systems Cybersecurity (PETRAS 2), Grant number is EP/S035362/1.

Acknowledgments

I thank PETRAS National Centre of Excellence for IoT Systems Cybersecurity for funding this work *https://petras-iot.org/*.

Conflict of interest

The author declares that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

References

Abdel-Basset, M., Moustafa, N., Hawash, H., and Ding, W. (2022). "Internet of things security requirements, threats, attacks, and countermeasures," in *Deep learning techniques for IoT security and privacy* (Springer), 67–112.

Abdul-Ghani, H. A., Konstantas, D., and Mahyoub, M. (2018). A comprehensive iot attacks survey based on a building-blocked reference model. *Int. J. Adv. Comput. Sci. Appl.* 9, 355–373. doi:10.14569/ijacsa.2018.090349

Ahlswede, R., Cai, N., Li, S.-Y., and Yeung, R. W. (2000). Network information flow. *IEEE Trans. Inf. theory* 46, 1204–1216. doi:10.1109/18.850663

Alladi, T., Chamola, V., Sikdar, B., and Choo, K.-K. R. (2020). Consumer iot: security vulnerability case studies and solutions. *IEEE Consum. Electron. Mag.* 9, 17–25. doi:10.1109/mce.2019.2953740

Alonso, R. S., Sittón-Candanedo, I., Rodríguez-González, S., García, Ó., and Prieto, J. (2019). "A survey on software-defined networks and edge computing over iot," in International Conference on Practical Applications of Agents and Multi-Agent Systems (Springer), 289–301.

Ankergård, S. F. J. J., Dushku, E., and Dragoni, N. (2021). State-of-the-art software-based remote attestation: opportunities and open issues for internet of things. *Sensors* 21, 1598. doi:10.3390/s21051598

Ar-Reyouchi, E. M., Lamrani, Y., Benchaib, I., Rattal, S., and Ghoumid, K. (2020). "NCBP: network coding based protocol for recovering lost packets in the internet of things," in *Advanced communication systems and information security*, 38–49. doi:10.1007/978-3-030-61143-9_4

Baldini, G., Fröhlich, P., Gelenbe, E., Hernandez-Ramos, J. L., Nowak, M., Nowak, S., et al. (2020). Iot network risk assessment and mitigation: the seriot approach. *Secur. Risk Manag.* 88. doi:10.1561/9781680836837.ch5

Bogdanov, A., Knudsen, L. R., Leander, G., Paar, C., Poschmann, A., Robshaw, M. J., et al. (2007). "Present: an ultra-lightweight block cipher," in International workshop on cryptographic hardware and embedded systems (Springer), 450–466.

Butun, I., Österberg, P., and Song, H. (2019). Security of the internet of things: vulnerabilities, attacks, and countermeasures. *IEEE Commun. Surv. Tutorials* 22, 616–644. doi:10.1109/comst.2019.2953364

Castiglione, L. M., and Lupu, E. C. (2020). "Hazard driven threat modelling for cyber physical systems," in CPSIOTSEC'20 Proceedings of the 2020 Joint Workshop on CPS&IoT Security and Privacy, New York, NY, USA (London, England: Association for Computing Machinery), 13–24. doi:10.1145/3411498.3419967

Castrucci, M., Neri, A., Caldeira, F., Aubert, J., Khadraoui, D., Aubigny, M., et al. (2012). Design and implementation of a mediation system enabling secure communication among critical infrastructures. *Int. J. Crit. Infrastructure Prot.* 5, 86–97. doi:10.1016/j.ijcip.2012.04.001

Cedric Levy-Bencheton, E. A. (2015). Security and resilience of smart home environments: good practices and recommendations.

Chaabouni, N., Mosbah, M., Zemmari, A., Sauvignac, C., and Faruki, P. (2019). Network intrusion detection for iot security based on learning techniques. *IEEE Commun. Surv. Tutorials* 21, 2671–2701. doi:10.1109/comst.2019.2896380

Chio, A., Bouloukakis, G., Hsu, C.-H., Mehrotra, S., and Venkatasubramanian, N. (2019). "Adaptive mediation for data exchange in iot systems," in ARM '19 Proceedings of the 18th Workshop on Adaptive and Reflexive Middleware, New York, NY, USA (New York, NY: Association for Computing Machinery), 1–6. doi:10.1145/3366612.3368122

Choi, S., Yang, C.-H., and Kwak, J. (2018). System hardening and security monitoring for iot devices to mitigate iot security vulnerabilities and threats. *KSII Trans. Internet Inf. Syst.* 12, 906–918. doi:10.3837/tiis.2018.02.022

Choo, K.-K. R., Gai, K., Chiaraviglio, L., and Yang, Q. (2021). A multidisciplinary approach to internet of things (iot) cybersecurity and risk management. *Comput. Secur.* 102, 102136. doi:10.1016/j.cose.2020.102136

Christina Skouloudi, e. a. (2020). Guidelines for securing the internet of things: secure supply chain for iot.

CNSSI (2015). Committee on national security systems (cnss) glossary.

Davies, N., Taft, N., Satyanarayanan, M., Clinch, S., and Amos, B. (2016). "Privacy mediators: helping iot cross the chasm," in HotMobile '16 Proceedings of the 17th International Workshop on Mobile Computing Systems and Applications (New York, NY, USA: Association for Computing Machinery), 39–44. doi:10.1145/2873587.2873600

Debar, H., Dacier, M., and Wespi, A. (1999). Towards a taxonomy of intrusion-detection systems. *Comput. Netw.* 31, 805–822. doi:10.1016/s1389-1286(98)00017-6

Demme, J., Maycock, M., Schmitz, J., Tang, A., Waksman, A., Sethumadhavan, S., et al. (2013). "On the feasibility of online malware detection with performance counters," in ISCA '13 Proceedings of the 40th Annual International Symposium on Computer Architecture, New York, NY, USA (New York, NY: Association for Computing Machinery), 559–570. doi:10.1145/2485922.2485970

Dhavlle, A., Shukla, S., Rafatirad, S., Homayoun, H., and Pudukotai Dinakarrao, S. M. (2021). "Hmd-hardener: adversarially robust and efficient hardware-assisted runtime malware detection," in 2021 Design, Automation Test in Europe Conference Exhibition (DATE), 1769–1774. doi:10.23919/DATE51398.2021.9474036

Dobson, S., Hutchison, D., Mauthe, A., Schaeffer-Filho, A., Smith, P., and Sterbenz, J. P. (2019). Self-organization and resilience for networked systems: design principles and open research issues. *Proc. IEEE* 107, 819–834. doi:10.1109/jproc.2019.2894512

Eldefrawy, K., Tsudik, G., Francillon, A., and Perito, D. (2012). Smart: secure and minimal architecture for (establishing dynamic) root of trust. *Ndss* 12, 1–15.

ENISA (2017). Baseline security recommendations for iot: in the context of critical information infrastructures.

ENISA (2019a). Enisa good practices for iot and smart infrastructures tool.

ENISA (2019b). Good practices for security of iot: secure software development lifecycle.

ETSI (2015). Methods for testing and specification; risk-based security assessment and test-ing methodologies.

Fortino, G., Messina, F., Rosaci, D., Sarné, G. M., and Savaglio, C. (2020). A trust-based team formation framework for mobile intelligence in smart factories. *IEEE Trans. Industrial Inf.* 16, 6133–6142. doi:10.1109/tii.2020.2963910

Franco, J., Aris, A., Canberk, B., and Uluagac, A. S. (2021). A survey of honeypots and honeynets for internet of things, industrial internet of things, and cyber-physical systems. *IEEE Commun. Surv. Tutorials* 1, 2351–2383. doi:10.1109/COMST.2021.3106669

Frank, C., Jarocki, S., Nance, C., Pauli, W. E., and Madison, S. (2018). Protecting iot devices from the mirai botnet. *J. Inf. Syst. Appl. Res.*

García, S. N. M., Molina Zarca, A., Hernández-Ramos, J. L., Bernabé, J. B., and Gómez, A. S. (2019). Enforcing behavioral profiles through software-defined networks in the industrial internet of things. *Appl. Sci.* 9, 4576. doi:10.3390/app9214576

Gerodimos, A., Maglaras, L., Ferrag, M. A., Ayres, N., and Kantzavelou, I. (2023). Iot: communication protocols and security threats. *Internet Things Cyber-Physical Syst.* 3, 1–13. doi:10.1016/j.iotcps.2022.12.003

Hamadeh, H., Chaudhuri, S., and Tyagi, A. (2017). Area, energy, and time assessment for a distributed tpm for distributed trust in iot clusters. *Integration* 58, 267–273. doi:10.1016/j.vlsi.2016.12.005

Hamza, A., Gharakheili, H. H., and Sivaraman, V. (2020). *Iot network security: requirements, threats, and countermeasures. arXiv preprint arXiv:2008.09339.*

Mohamad Noor, M. B., Hassan, W. H., et al. (2019). Current research on internet of things (iot) security: a survey. *Comput. Netw.* 148, 283–294. doi:10.1016/j.comnet.2018.11.025

Hassija, V., Chamola, V., Saxena, V., Jain, D., Goyal, P., and Sikdar, B. (2019). A survey on iot security: application areas, security threats, and solution architectures. *IEEE Access* 7, 82721–82743. doi:10.1109/ACCESS.2019.2924045

Hemberg, E., Zipkin, J. R., Skowyra, R. W., Wagner, N., and O'Reilly, U.-M. (2018). "Adversarial co-evolution of attack and defence in a segmented computer network environment," in GECCO '18 Proceedings of the Genetic and Evolutionary Computation Conference Companion (New York, NY, USA: Association for Computing Machinery), 1648–1655. doi:10.1145/3205651.3208287

Hromada, D., Costa, R. L. d. C., Santos, L., and Rabadão, C. (2023). "Security aspects of the internet of things," in *Research anthology on convergence of blockchain, internet of things, and security* (IGI Global), 67–87.

Illiano, V. P., and Lupu, E. (2015). Detecting malicious data injections in wireless sensor networks: a survey. *ACM Comput. Surv.* 48, 1–24. doi:10.1145/2818184

Illiano, V. P., Muñoz-González, L., and Lupu, E. C. (2017). Don't fool me!: detection, characterisation and diagnosis of spoofed and masked events in wireless sensor networks. *IEEE Trans. Dependable Secure Comput.* 14, 279–293. doi:10.1109/tdsc.2016.2614505

Kamalov, F., Pourghebleh, B., Gheisari, M., Liu, Y., and Moussa, S. (2023). Internet of medical things privacy and security: challenges, solutions, and future trends from a new perspective. Sustain. *IoT device Hardw. Prot. Def. approaches* 15, 3317. doi:10.3390/su15043317

Koeberl, P., Schulz, S., Sadeghi, A.-R., and Varadharajan, V. (2014). "Trustlite: a security architecture for tiny embedded devices," in Proceedings of the Ninth European Conference on Computer Systems, 1–14.

Dinesh Kumar, K., Venkata Rathnam, T., and Venkata Ramana, R. M. (2023). "Towards the integration of blockchain and iot for security challenges in iot: a review," in *Research anthology on convergence of blockchain, internet of things, and security*, 193–209.

Kumar, V., Das, A. K., and Sinha, D. (2021). Uids: a unified intrusion detection system for iot environment. *Evol. Intell.* 14, 47–59. doi:10.1007/s12065-019-00291-w

Kuzlu, M., Fair, C., and Guler, O. (2021). Role of artificial intelligence in the internet of things (iot) cybersecurity. *Discov. Internet Things* 1, 7–14. doi:10.1007/s43926-020-00001-4

Laszka, A., Abbas, W., Vorobeychik, Y., and Koutsoukos, X. (2018). "Synergistic security for the industrial internet of things: integrating redundancy, diversity, and hardening," in 2018 IEEE International Conference on Industrial Internet (ICII), 153–158. doi:10.1109/ICII.2018.00025

Lee, Y., and Lee, G. (2020). Attack detection using network coding in iot environment. *Sensors (Basel, Switz.* 20, 1180. doi:10.3390/s20041180

Leo, M., Battisti, F., Carli, M., and Neri, A. (2014). "A federated architecture approach for internet of things security," in 2014 Euro Med Telco Conference (EMTC), 1–5. doi:10.1109/EMTC.2014.6996632

Li, J., Liu, Y., Zhang, Z., Ren, J., and Zhao, N. (2017). Towards green iot networking: performance optimization of network coding based communication and reliable storage. *IEEE Access* 5, 8780–8791. doi:10.1109/access.2017.2706328

Liao, C.-H., Shuai, H.-H., and Wang, L.-C. (2019). Rnn-assisted network coding for secure heterogeneous internet of things with unreliable storage. *IEEE Internet Things J.* 6, 7608–7622. doi:10.1109/JIOT.2019.2902376

Linkov, I., and Kott, A. (2019). "Fundamental concepts of cyber resilience: introduction and overview," in *Cyber resilience of systems and networks* (Springer), 1–25.

Little, R. G. (2003). "Toward more robust infrastructure: observations on improving the resilience and reliability of critical systems," in 36th Annual Hawaii International Conference on System Sciences, 2003. Proceedings of the (IEEE), 9.

Liu, Y., Wang, J., Yan, Z., Wan, Z., and Jäntti, R. (2023a). A survey on blockchain-based trust management for internet of things. *IEEE Internet Things J.* 10, 5898–5922. doi:10.1109/jiot.2023.3237893

Liu, Y., Wang, J., Yan, Z., Wan, Z., and Jäntti, R. (2023b). A survey on blockchain-based trust management for internet of things. *IEEE Internet Things J.* 10, 5898–5922. doi:10.1109/JIOT.2023.3237893

Lu, D., Han, R., Shen, Y., Dong, X., Ma, J., Du, X., et al. (2020). xtseh: a trusted platform module sharing scheme towards smart iot-ehealth devices. *IEEE J. Sel. Areas Commun.* 39, 370–383. doi:10.1109/jsac.2020.3020658

Lu, Y., and Xu, L. D. (2019). Internet of things (iot) cybersecurity research: a review of current research topics. *IEEE Internet Things J.* 6, 2103–2115. doi:10.1109/JIOT.2018.2869847

Luiijf, E., and Klaver, M. (2021). Analysis and lessons identified on critical infrastructures and dependencies from an empirical data set. *Int. J. Crit. Infrastructure Prot.* 35, 100471. doi:10.1016/j.ijcip.2021.100471

Mahmoud, R., Yousuf, T., Aloul, F., and Zualkernan, I. (2015). "Internet of things (iot) security: current status, challenges and prospective measures," in 2015 10th International Conference for Internet Technology and Secured Transactions. London, UK: ICITST), 336–341. doi:10.1109/ICITST.2015.7412116

Mämmelä, O., Hiltunen, J., Suomalainen, J., Ahola, K., Mannersalo, P., and Vehkaperä, J. (2016). Towards microsegmentation in 5g network security.

Martin, A., Rashid, A., Chivers, H., Schneider, S., Lupu, E., and Danezis, G. (2021). *Introduction to cybok knowledge area version*.

Matheu, S. N., Hernández-Ramos, J. L., Skarmeta, A. F., and Baldini, G. (2020a). A survey of cybersecurity certification for the internet of things. *ACM Comput. Surv. (CSUR)* 53, 1–36. doi:10.1145/3410160

Matheu, S. N., Hernández-Ramos, J. L., Skarmeta, A. F., and Baldini, G. (2020b). A survey of cybersecurity certification for the internet of things. *ACM Comput. Surv.* 53, 1–36. doi:10.1145/3410160

Meidan, Y., Bohadana, M., Mathov, Y., Mirsky, Y., Shabtai, A., Breitenbacher, D., et al. (2018). N-baiot—network-based detection of iot botnet attacks using deep autoencoders. *IEEE Pervasive Comput.* 17, 12–22. doi:10.1109/mprv.2018.03367731

Meneghello, F., Calore, M., Zucchetto, D., Polese, M., and Zanella, A. (2019). Iot: internet of threats? a survey of practical security vulnerabilities in real iot devices. *IEEE Internet Things J.* 6, 8182–8201. doi:10.1109/jiot.2019.2935189

Mercado-Velázquez, A. A., Escamilla-Ambrosio, P. J., and Ortiz-Rodríguez, F. (2021). A moving target defence strategy for internet of things cybersecurity. *IEEE Access* 9, 118406–118418. doi:10.1109/ACCESS.2021.3107403

Mohamad Noor, M. B., and Hassan, W. H. (2019). Current research on internet of things (iot) security: a survey. *Comput. Netw.* 148, 283–294. doi:10.1016/j.comnet.2018.11.025

Mohan, S., Asplund, M., Bloom, G., Sadeghi, A.-R., Ibrahim, A., Salajageh, N., et al. (2018). "Special session: the future of iot security," in 2018 International Conference on Embedded Software (EMSOFT) (IEEE), 1–7.

Molina Zarca, A., Bernabe, J. B., Trapero, R., Rivera, D., Villalobos, J., Skarmeta, A., et al. (2019). Security management architecture for nfv/sdn-aware iot systems. *IEEE Internet Things J.* 6, 8005–8020. doi:10.1109/JIOT.2019.2904123

Najmi, K. Y., Alzain, M. A., Masud, M., Jhanjhi, N. Z., Al-Amri, J. F., and Baz, M. (2021). A survey on security threats and countermeasures in iot to achieve users confidentiality and reliability. *Mater. Today Proc.* 81, 377–382. doi:10.1016/j.matpr.2021.03.417

Navas, R. E., Cuppens, F., Boulahia Cuppens, N., Toutain, L., and Papadopoulos, G. Z. (2021). Mtd, where art thou? a systematic review of moving target defence techniques for iot. *IEEE Internet Things J.* 8, 7818–7832. doi:10.1109/JIOT.2020.3040358

Navas, R. E., Sandaker, H., Cuppens, F., Cuppens, N., Toutain, L., and Papadopoulos, G. (2020). "IANVS: a moving target defence framework for a resilient internet of things," in The 25th IEEE Symposium on Computers and Communications (ISCC) (Rennes, France: IEEE), The 25th IEEE Symposium on Computers and Communications (ISCC), 1–6. doi:10.1109/ISCC50000.2020.9219728

Osman, A., Wasicek, A., Köpsell, S., and Strufe, T. (2020). "Transparent microsegmentation in smart home IoT networks," in 3rd USENIX Workshop on Hot Topics in Edge Computing (HotEdge 20) (USENIX Association).

Pa, Y. M. P., Suzuki, S., Yoshioka, K., Matsumoto, T., Kasama, T., and Rossow, C. (2015). "Iotpot: analysing the rise of iot compromises," in 9th USENIX Workshop on Offensive Technologies (WOOT 15) (Washington, D.C.: USENIX Association).

Pacheco, J., Benitez, V. H., and Pan, Z. (2019). Security framework for iot end nodes with neural networks. *Int. J. Mach. Learn. Comput.* 9, 381–386. doi:10.18178/ijmlc.2019.9.4.814

Paulsen, C., and Byers, R. (2019). "Glossary of Key Information Security Terms," in *NIST Interagency/Internal Report (NISTIR)*. Gaithersburg, MD: National Institute of Standards and Technology, [online]. Available at: https://doi.org/10.6028/NIST.IR.7298r3 (Accessed January 13, 2024)

Pauna, A., Bica, I., Pop, F., and Castiglione, A. (2019). On the rewards of self-adaptive iot honeypots. *Ann. Telecommun.* 74, 501–515. doi:10.1007/s12243-018-0695-7

Ravi, S., Raghunathan, A., and Chakradhar, S. (2004). "Tamper resistance mechanisms for secure embedded systems," in 17th International Conference on VLSI Design. Proceedings. 605–611. doi:10.1109/ICVD.2004.1260985

Rayes, A., and Salam, S. (2022). "Internet of things security and privacy," in *Internet of things from hype to reality* (Springer), 213–246.

Rosenberg, H., and Reinhardt, A. (2021). "Wip: collaborative approaches to mitigate links of variable quality in lora networks," in 2021 IEEE 22nd International Symposium on a World of Wireless, Mobile and Multimedia Networks (Pisa, Italy: WoWMoM), 244–247. doi:10.1109/WoWMoM51794.2021.00044

Salayma, M., Al-Dubai, A., Romdhani, I., and Nasser, Y. (2017). Wireless body area network (wban) a survey on reliability, fault tolerance, and technologies coexistence. *ACM Comput. Surv. (CSUR)* 50, 1–38. doi:10.1145/3041956

Shrobe, H., Shrier, D. L., and Pentland, A. (2018). CHAPTER 9 Moving Target Techniques: Cyber Resilience through Randomization, Diversity, and Dynamism, MIT Press, 293–311.

Sidhu, S., Mohd, B. J., and Hayajneh, T. (2019). Hardware security in iot devices with emphasis on hardware trojans. *J. Sens. Actuator Netw.* 8, 42. doi:10.3390/jsan8030042

Sinha, S. (2021). Devices growing 9% to 12.3 billion globally, cellular iot now surpassing 2 billion.

Soikkeli, J., Muñoz-González, L., and Lupu, E. C. (2019). Efficient attack countermeasure selection accounting for recovery and action costs.

Spitzner, L. (2003). Endpoint protection: dynamic honeypots.

Stellios, I., Kotzanikolaou, P., Psarakis, M., Alcaraz, C., and Lopez, J. (2018). A survey of iot-enabled cyberattacks: assessing attack paths to critical infrastructures and services. *IEEE Commun. Surv. Tutorials* 20, 3453–3495. doi:10.1109/comst.2018.2855563

Sterbenz, J. P., Hutchison, D., Çetinkaya, E. K., Jabbar, A., Rohrer, J. P., Schöller, M., et al. (2014). Redundancy, diversity, and connectivity to achieve multilevel network resilience, survivability, and disruption tolerance invited paper. *Telecommun. Syst.* 56, 17–31. doi:10.1007/s11235-013-9816-9

Stergiopoulos, G., Dedousis, P., and Gritzalis, D. (2020). Automatic network restructuring and risk mitigation through business process asset dependency analysis. *Comput. Secur.* 96, 101869. doi:10.1016/j.cose.2020.101869

Stergiou, C. L., Bompoli, E., and Psannis, K. E. (2023). Security and privacy issues in iot-based big data cloud systems in a digital twin scenario. *Appl. Sci.* 13, 758. doi:10.3390/app13020758

Swessi, D., and Idoudi, H. (2022). A survey on internet-of-things security: threats and emerging countermeasures. *Wirel. Personal. Commun.* 124, 1557–1592. doi:10.1007/s11277-021-09420-0

Vacca, J. R. (2012). Computer and information security handbook (Newnes).

Venkatakrishnan, R., and Vouk, M. A. (2016). Using redundancy to detect security anomalies: Towards iot security attack detectors: The internet of things (ubiquity symposium). Ubiquity 2016, 1–19.

Vetterl, A. (2020). Honeypots in the age of universal attacks and the internet of things

Voas, J., and Laplante, P. A. (2018). Iot's certification quagmire. *Computer* 51, 86–89. doi:10.1109/MC.2018.2141036

Wagner, N., Şahin, C. Ş., Pena, J., Riordan, J., and Neumayer, S. (2017a). "Capturing the security effects of network segmentation via a continuous-time Markov chain model," in Proceedings of the 50th Annual Simulation Symposium, 1–12.

Wagner, N., Sahin, C. S., Pena, J., and Streilein, W. W. (2017b). "A nature-inspired decision system for secure cyber network architecture," in 2017 IEEE Symposium Series on Computational Intelligence (SSCI) (IEEE), 1–8.

Wagner, N., Şahin, C. Ş., Winterrose, M., Riordan, J., Hanson, D., Peña, J., et al. (2017c). Quantifying the mission impact of network-level cyber defensive mitigations. *J. Def. Model. Simul.* 14, 201–216. doi:10.1177/1548512916662924

Wagner, N., Şahin, C. Ş., Winterrose, M., Riordan, J., Pena, J., Hanson, D., et al. (2016). "Towards automated cyber decision support: a case study on network segmentation for security," in 2016 IEEE Symposium Series on Computational Intelligence (SSCI) (IEEE), 1–10.

Wasicek, A. (2020). The future of 5g smart home network security is micro-segmentation. *Netw. Secur.* 2020, 11–13. doi:10.1016/s1353-4858(20)30129-x

Xing, L. (2021). Cascading failures in internet of things: review and perspectives on reliability and resilience. *IEEE Internet Things J.* 8, 44–64. doi:10.1109/JIOT.2020.3018687

Xu, C., Liu, H., Li, P., and Wang, P. (2018). A remote attestation security model based on privacy-preserving blockchain for v2x. *Ieee Access* 6, 67809–67818. doi:10.1109/access.2018.2878995

Zarca, A. M., Bernabe, J. B., Trapero, R., Rivera, D., Villalobos, J., Skarmeta, A., et al. (2019). Security management architecture for nfv/sdn-aware iot systems. *IEEE Internet Things J.* 6, 8005–8020. doi:10.1109/jiot.2019.2904123

Zarpelão, B. B., Miani, R. S., Kawakani, C. T., and de Alvarenga, S. C. (2017). A survey of intrusion detection in internet of things. *J. Netw. Comput. Appl.* 84, 25–37. doi:10.1016/j.jnca.2017.02.009

Zavalyshyn, I., Given-Wilson, T., Legay, A., and Sadre, R. (2020). "Brief announcement: effectiveness of code hardening for fault-tolerant iot software," in *Stabilization, safety, and security of distributed systems*. Editors S. Devismes, and N. Mittal (Cham: Springer International Publishing), 317–322.