Msgna. *J Surveill Secur Saf* 2022;3:150-73
DOI: 10.20517/jsss.2022.07

Journal of Surveillance, Security and Safety

Original Article

Open Access

Check for updates

Anatomy of attacks on IoT systems: review of attacks, impacts and countermeasures

Mehari Msgna

Critical Infrastructure Security and Resilience Group (CISaR), Norwegian University of Science and Technology (NTNU), Gjøvik 2815, Norway.

Correspondence to: Dr. Mehari Msgna, Critical Infrastructure Security and Resilience Group (CISaR), Norwegian University of Science and Technology (NTNU), Teknologivegen 22, Gjøvik 2815, Norway. E-mail: mehari.g.msgna@ntnu.no; ORCID: 0000-0002-4207-8717

How to cite this article: Msgna M. Anatomy of attacks on IoT systems: review of attacks, impacts and countermeasures. *J Surveill Secur Saf* 2022;3:150-73. http://dx.doi.org/10.20517/jsss.2022.07

Received: 3 Mar 2022 **First Decision:** 11 Apr 2022 **Revised:** 6 Jun 2022 **Accepted:** 9 Nov 2022 **Published:** 5 Dec 2022

Academic Editor: Kshirasagar S. Naik **Copy Editor:** Ying Han **Production Editor:** Ying Han

Abstract

Aim: The Internet of Things is a disruptive technology that converts physical objects into a constant source of information. Internet-connected devices bridge the gap between the physical and virtual worlds through their data-generating set of sensors. Due to the large-scale proliferation of Internet-of-Things systems into practically every sector of modern life, they have also become the centre of growing cybersecurity threats and attacks. This is exacerbated by the connectivity between different kinds of devices and the lack of standardisation to govern them. The majority of papers on the security of the Internet of Things discuss one attack or threat at a time, which could lead to a fragmented understanding of their overall security posture. The aim of this paper is to provide a concise review of attacks on an Internet-of-Things system, their impacts on IoT assets and possible countermeasures.

Methods: We review the available layered representation and functional components of the Internet of Things. We then identify the system's assets and review the literature on IoT attacks. We categorise these attacks into groups using common classification criteria and map them against the assets they target. We also identify the possible impacts that these attacks could have on an IoT system. We explore a number of security controls that could be deployed to detect or prevent the attacks. Finally, we evaluate these countermeasures against the assets they protect and the impacts they intend to prevent.

Results: To clearly show the security of IoT systems, we identify assets, categorise the different attacks and map them to the different components of an IoT system. Further, we identify the different countermeasures and evaluate

their effectiveness against IoT assets and attacks.

Conclusion: The paper provides a clear and concise description of IoT functional components and computational models. It also presents an anatomy of attacks on such a system. In addition, the main assets of a typical IoT system are identified and elaborated. The different types of attacks that can be launched in an IoT environment are categorised and mapped against the different functional components. Further, the different assets are identified and countermeasures are evaluated on their effectiveness to protect them.

Keywords: Internet of Things, attacks, impacts, countermeasures

1. INTRODUCTION

The Internet of Things (IoT) is rapidly closing the gap between physical objects and the cyber-space. It converts any physical object into a constant source of information. For instance, a smart hospital bed equipped with sensors can constantly monitor a patient's condition (up to 35 data points including body temperature, weight, heartbeat, etc.) and update their medical record. This is only one of the potentially endless IoT use cases that can be deployed to improve the efficiency of existing processes. As more and more information is processed through this expanding network of devices, the security of IoT systems becomes increasingly important. However, securing an IoT system is a complex endeavour. The heterogeneous nature of the connected devices, as well as the lack of standardisation to govern them, exposes an IoT system to a variety of security threats, in some cases to attacks that no longer pose a threat to traditional networked systems.

In an IoT environment, security vulnerabilities may occur in one or more of its components including services, software, physical devices, communication protocols, and sometimes even the people who use them. According to the *Unit 42* 2020 IoT threat report, 83% of medical IoT devices run on an unsupported operating system due to the decision of Microsoft to declare that the Windows 7 operating system has reached its end of life[1]. The consequences of such decisions on the overall security posture of these devices are severe as it opens them to new attacks, such as Cryptojacking[2], and brings back long-forgotten attacks, such as Conficker[3]. According to the same report, 57% of all IoT devices are vulnerable to medium- or high-severity attacks, making them primary targets to attackers. According to another 2021 IoT threat survey, 51% of organisations indicated that IoT devices are segmented on a separate network from the one they use for primary business devices and applications (e.g., HR system, email server or finance system), and another 26% of respondents in the same group said that IoT devices are microsegmented within security zones, an industry best practice to create tightly controlled security zones on their networks to isolate IoT devices and keep them separated from IT devices to prevent hackers from moving laterally on a network[4]. According to a 2021 security report by PSA Certified, only 47% of respondents said they carry out threat analysis in the design of new products[5]. All these surveys indicate that IoT systems are vulnerable to a host of threats, and this is further exacerbated by organisations' weak security approach in their design and operation.

Security attacks on an IoT system and their corresponding countermeasures have been covered extensively in the literature but mostly in a fragmented approach. Most of the published works discuss attacks and their countermeasures in isolation, which could lead to a partial view of the overall security posture of an IoT system. As with any other system, the IoT is only as strong as its weakest link, and to truly secure it, we have to look at all of its components and subsystems. The aim of this paper is to provide a holistic view of the security of an IoT system. To do this, the different functional blocks and system assets need to be identified. The various attacks must be classified into groups of similar nature based on the skills required, assets they target and impacts they may have. Similarly, the different countermeasures must also be classified and evaluated against the assets they protect and attack impacts they intend to prevent or minimise.

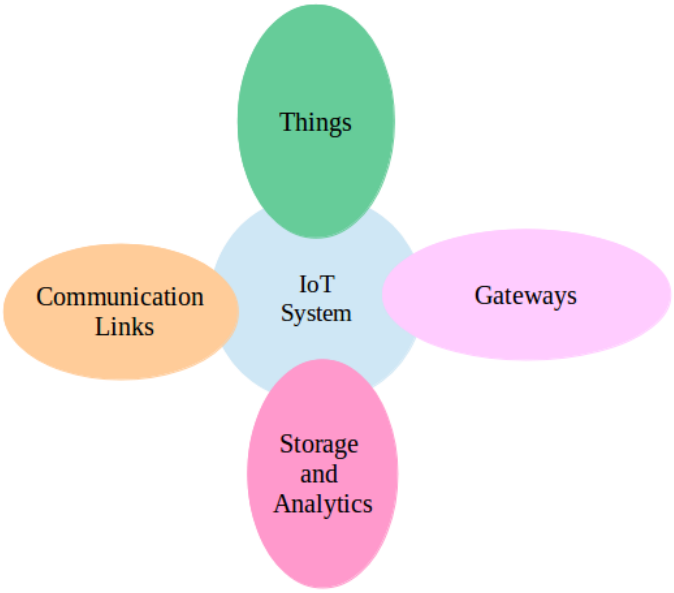

Figure 1. Functional building blocks of an IoT system.

The rest of the paper is organised as follows. Section 2 provides a brief description of the layered architecture, different functional components and computational models of a typical IoT system. Section 3 presents a detailed discussion of IoT system assets, attacks and their impacts. In this section, we identify the primary IoT system assets and classify the attacks into groups of similar nature using three-point categorisation criteria. We map these attacks against the primary assets they target. We also classify the various impacts such attacks can have on the system. Section 4 discusses different countermeasures that can be used to protect the primary assets. These countermeasures are evaluated against the assets they protect and the impacts they intend to prevent. Section 5 highlights the open research issues. Finally, Section 6 provides concluding remarks.

2. IOT ARCHITECTURE, COMPONENTS AND COMPUTATION MODELS

An IoT system is a combination of a variety of technologies. Each application use case combines the core components of an IoT system in a suitable architecture and computational model that enhances its efficiency. In this section, we provide a brief discussion of the different technological components that make up the basic building blocks of an IoT system, how these components are integrated, their roles and the different computational models that determine where and how data are stored and analysed.

2.1. Components

An IoT system comprises multiple functional building blocks. These building blocks can generally be grouped into four components: **things**, **communication link**, **gateways** and **storage and analytics** [Figure 1]. Each of these components represents different domains of technological innovation, and when they are put together, they form a complex network of connected devices known as the Internet of Things.

2.1.1. Things

An IoT system is based on things (devices) that provide the sensing, actuation and monitoring activities. These devices collect data through their connected sensors, exchange data with other nearby devices (via wired or short-range wireless communication technologies), process data locally or send them to the gateway. In some cases, these devices may also act as local gateways. Generally, an IoT device/thing may consist of the following

parts:

- I/O interface to sensors;
- interface for network connectivity including the Internet;
- interface to external memory (RAM) and storage;
- audio/video I/O interface;
- I/O interface to signalling systems (such as LEDs).

2.1.2. Communication link

This component performs the data transmission and reception activities between the sensors and their control devices. Sensor devices may exchange data with other sensors, collect other sensors' data or simply send only their measurements to the nearest control device. The primary means of communication between the sensors and their control devices is wireless communication technology, but wired communication could also be used. These communication protocols generally work in the data link layer, network layer, transport layer or even application layer.

2.1.3. Gateway

The primary purpose of a gateway is to bridge between different types of communication technologies. It creates a bridge between the devices and the Internet. An IoT gateway aggregates data from different devices, translates protocols or ever processes data before sending them. In addition to the above-mentioned features, an IoT gateway may also have the following key features.

- Serve as a data cache, buffer and streaming device.
- Provide offline services and real-time control of the devices.
- Additional intelligence can be loaded to the gateways for some IoT devices and systems.
- Provide additional security measures such as local device authentication or authorisation.
- Perform device configuration and change management roles.

2.1.4. Storage and analytics infrastructure

The storage and analytics infrastructure is a set of servers where the IoT generated data are stored and analysed to provide application-specific services to the users. These servers can be located in different places depending on the storage and computational requirements of the systems. For instance, some IoT systems might choose a cloud infrastructure. Others may choose to shift some of the storage and processing needs closer to the field devices, e.g. a server connected within the gateway or even into the gateway itself. Some may even go further by placing lightweight data processing capabilities in the control devices themselves.

2.2. Architecture

The IoT provides a fascinating solution to a lot of challenges. The solutions depend on how the aforementioned components are integrated, including how the devices communicate and how the data are processed to provide actionable insights. To illustrate how such integration works, several layered architectural representations of an IoT system are proposed. These layered architectural representations are known as the *three-layer*, *five-layer* and *seven-layer* IoT architectures. Table 1 illustrates the *five-layer* architecture of an IoT system and the role of each layer in the overall system.

2.3. Computation models

The proliferation of IoT into every sector generates a massive volume of data at a rapid pace. An unprecedented volume and variety of data are generated by connected devices. This demands an efficient data storage and computation model. Since the IoT is being deployed almost everywhere, such as in industrial automation, agriculture, healthcare, intelligent transportation, etc., it is somewhat impractical to develop one general computational model and design architecture that suits all systems. Therefore, the computational model varies

Table 1. The roles and features of layers in a five-layer IoT architecture

Architecture layers	Roles and functional features
Perception/sensing	The functional feature of this layer is sensing capability; it collects and gathers information about the environments in which the IoT devices are located
Transport	The transport layer's main feature is transferring the information collected by the perception layer to the processing layer and vice versa through different available networks such as WiFi, 3G, 4G, etc
Processing	The main feature of this layer is to store and process the collected data to allow various kinds of services
Application	This layer delivers application-specific services to the users based on the processed data received from the processing layer
Business	This layer manages the entire IoT system. All applications of the IoT systems will be managed from this layer

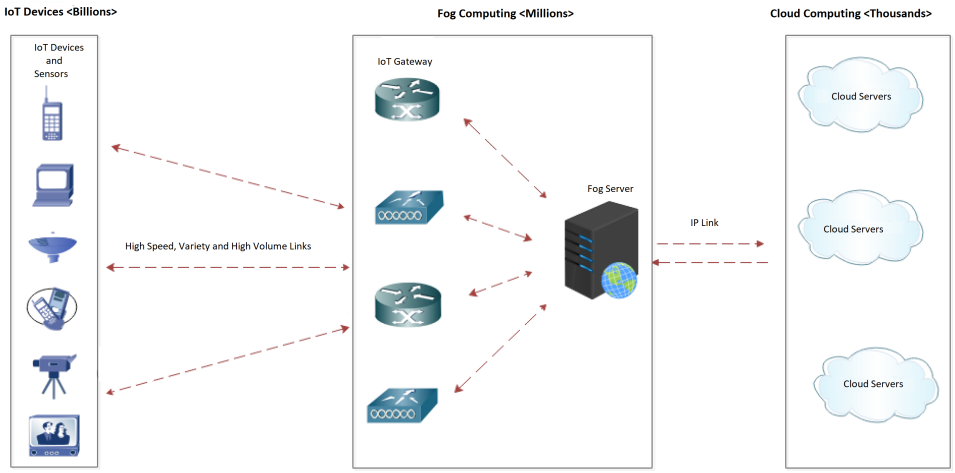

Figure 2. IoT computational models.

from application to application. Thus far, three different computational models are used by different IoT use cases: cloud [Section 2.3.1], fog [Section 2.3.2] and edge [Section 2.3.3] computing. Figure 2 illustrates the different stages of an IoT system where data may be stored and processed depending on the data computation model.

2.3.1. Cloud computing

Could computing, at its basic level, is a way for business to use the Internet to connect to an off-premise storage and compute infrastructure. In the context of IoT, the cloud is used to store and process the generated data. This provides a central platform for storing, processing and managing IoT devices and their data that can be accessed anywhere, anytime, from any device. Cloud computing allows IoT systems to scale easily without the need for significant investment in their back-end infrastructure. However, it also comes at its own cost. Sometimes by the time the data makes its way to the cloud for analysis and decision making, the opportunity to act on the data might be gone. This could end up causing a disaster, given IoT systems are increasingly being deployed in systems that can affect the safety of citizens and infrastructures. Cloud computing for IoT systems is suitable in the following situations:

- When there are only small independent systems that may have few connected devices, cloud offers an economically viable option to quickly set up and operate an IoT system.
- There are many devices distributed over a large geographic area and it is impractical to connect them to a local network, e.g. roadside sensors of an urban intelligent transportation network.
- The IoT system does not require time-sensitive and real-time decision making.

- The connected devices are embedded in third-party OEM devices and it is either impractical or uneconomical to create another local network to connect them.
- when secondary monitoring of devices that are already connected to other local networks and systems is needed, the cloud can be used to connect these devices and perform analysis on the aggregated data from all other sub/systems.

2.3.2. Fog computing

As data generation explodes, central cloud systems are being constrained for storage, computation and management of connected devices. In addition, the cloud servers might take time to act upon the data received from the IoT devices as they have to work as a centralised infrastructure to store, process and communicate the data, and often they are located far from the IoT devices. This creates the demand for better and faster technologies. This demand has contributed to the emergence and popularity of fog computing to take on some of the burden of cloud servers. In fog computing, the collected data are processed closer to their source, mainly within the IoT gateway or fog nodes connected to the LAN network. Compared to cloud, fog computing offers the following advantages[6]:

- Fog analyses the most time-sensitive data locally (closer to the source), enabling a real-time response for the connected device to act upon quickly.
- Fog optimises the utilization of available bandwidth by sending only selective data to the cloud for archiving and historical analysis purposes.
- Fog minimises network latency by shortening the path between the data source (IoT devices) to the processing location.

According to IDC, in 2015, the amount of data analysed on Fog nodes was estimated to be approaching 40% of all data generated by IoT systems[7]. This number was expected to increase as the trend of shifting data processing away from cloud servers was on a sharp rise.

2.3.3. Edge computing

Both edge computing and fog computing offer similar functionalities in terms of pushing both intelligence and data to nearby analytic platforms that are located either on or near the source of data, be it cars, motors or ICU beds. Edge computing places the computing resources at the edge of the network closer to the edge devices than fog computing. In edge computing, the data are processed by the sensors or devices to which the sensors are directly connected. According to[8], edge computing is built on the back of *cloudlets*, which were originally proposed in 2009. The idea of *cloudlets* is to introduce computing "hotspots" similar to WiFi "hotspots" to provide selected cloud services without an Internet connection.

It is worth mentioning that these computational models are not necessarily deployed separately. Usually, a large IoT network utilises all three models together. For instance, highly time-sensitive data are processed by the sensors or devices that are directly connected to the sensors. Less time-sensitive data are then forwarded to nearby fog nodes for extra processing. Finally, selected data are sent to the cloud for storage and historical analysis. Another criterion for deciding which data need to be processed locally and which need to be sent to the centralised servers is the security requirements. Edge devices are highly secure from cybersecurity threats as data are kept locally. Fog nodes are regarded as more secure than cloud servers as they keep the data distributed across a number of nodes.

3. OVERVIEW OF IOT ASSETS, ATTACKS AND THEIR IMPACTS

Having discussed the functional components, architecture and computational models of an IoT system, this section provides a detailed discussion of IoT system's assets, the attacks they face and their impacts.

Table 2. Internet-of-Things system assets

Asset	Description
Physical Objects	Physical objects refer to the hardware assets of an IoT system. Such assets include sensors, gateways, routers and servers
Data	The perception/sensing layer of an IoT system converts an object into a constant source of information. Such information is stored in edge, fog or cloud servers. Data asset refers to the information stored in these servers across the IoT network
Link/Protocols	The link/protocol asset category refers to a set of communication technologies used to establish connectivity between the sensors, networking devices, services and applications of a given IoT system
Software	A typical IoT system consists of various software ranging from firmware, operating system and applications. These software are critical to the successful operation of IoT networks

3.1. Assets

An asset is any hardware, software, data or other components of an organisation's system that is valuable. Understanding and proper classification of assets are paramount to the identification, deployment and operation of efficient security controls. Asset classification is the process of assigning assets into groups based on common characteristics. The popular ISO[1] standard *ISO27001* recommends all information assets to be classified[9]. Before we delve into the details of attacks on IoT systems, it is important to classify and define their assets. Depending on the nature and role, we classify IoT system assets into four categories. These categories are presented in Table 2.

3.2. Attacks

IoT system attacks are multifaceted. Imran *et al.*[10] discussed the anatomy of the various malware attacks on IoT systems. However, malware attacks are only one vector of attack on IoT systems. They come in different forms and have varying consequences. To understand the overall anatomy of attacks on IoT systems, it is important to classify them into groups of similar nature and discuss each group. For this purpose, we classify the attacks into six groups using the following criteria:

- The component of an IoT system the attack targets: An IoT system has four main components and an attack exploits a specific weakness in one or more of these components. For instance, a given attack may take advantage of a flaw in the communication protocol, a vulnerability in software or a design flaw in the underlying hardware.
- The knowledge required to carry out the attack: Another criterion in our classification is the knowledge required to carry out a successful attack. Different attacks require different sets of skills; for example, the skills needed to launch a successful attack on a hardware platform, a radio frequency communication protocol or social engineering attack are vastly different.
- The severity of attack impact: Every attack has some impact on the targeted organisation. However, the severity of an attack may vary from one organisation to another, ranging from minor service disruptions to permanent data and service destruction. The severity of the impact depends on the system architecture, the security posture of individual components and the capabilities of an attacker.

3.2.1. Physical attacks

Physical attacks are attacks that require the target device to be exposed and directly attacked through physical means[11]. Physical attacks, at least in theory, can compromise the security of any secure processor chip. However, they require expensive equipment and large investment in terms of time to produce meaningful results.

[1]International Organization for Standardization, https://www.iso.org/home.html

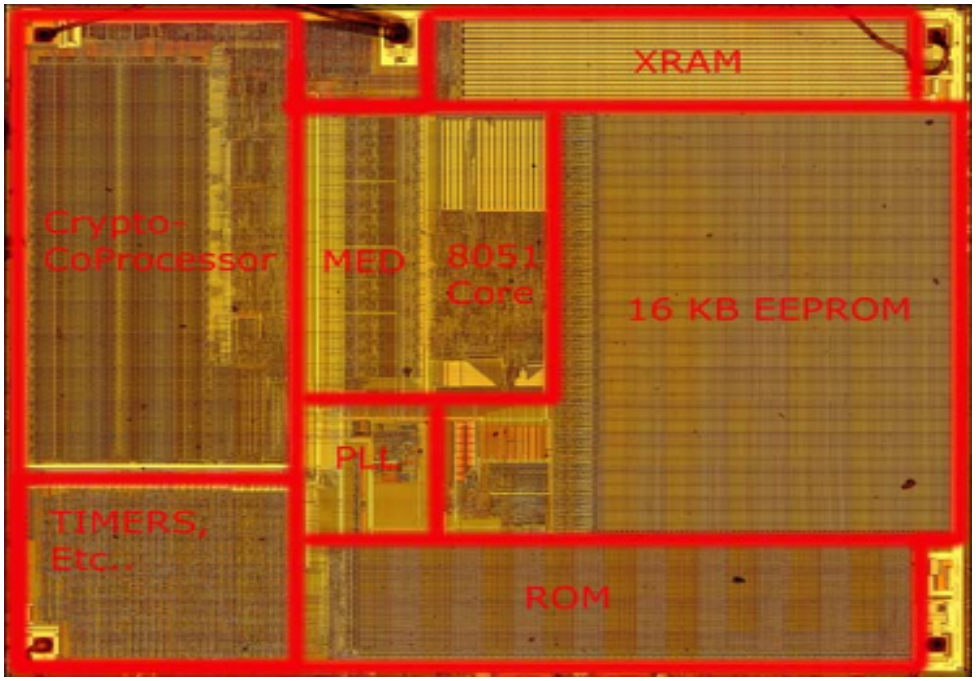

Figure 3. Physical attacks on integrated circuit (IC) chips *(images taken from [15])*.

By nature, physical attacks are invasive, and they tend to modify some of the physical properties of the target device irreversibly [12]. Successful physical attacks involve performing a number of tasks precisely at different stages during the attack [13]. The goal of a physical attack could be revealing secret information kept inside the processor (such as cryptographic keys), modifying the original circuit design and/or reverse engineering the semiconductor chip itself [14]. Common tasks performed during a physical attack are delayering, microprobing and circuit tracing. The target blocks of such tasks are illustrated in Figure 3.

1. **Delayering:** An integrated circuit (IC) chip is made up of multiple layers of metal and silicon oxide covered by a global top layer of epoxy resin. The layer below is made of silicon oxide, which protects the chip from environmental hazards and ionic contaminations. Delayering is the process of removing each layer, one at a time, without damaging the chip to gain access to the underlying circuit. It usually involves dry and wet etching and polishing; for instance, the epoxy resin is removed using a fuming nitric acid [16]. However, before removing the layers, several scanning methods, such as scanning electron microscopes (SEM) [17] and transmission electron microscopes (TEM) [18], are applied to determine the composition and thickness of each layer [19].
2. **Microprobing:** Probing is the process of attaching microscopic needles onto the internal wiring of a captured and delayered device to observe the information flow between the different internal components of the device, such as the processor, memory, IO, etc. Once the probes are attached, the attacker can read out internally communicated data that may be used to launch other attacks. A microprobing attack requires tools such as microscope, micro-positioners, probing needles and amplifiers [20,21].
3. **Circuit tracing:** Once external layers of a target IC are successfully removed, an attacker proceeds to trace the internal circuitry to understand the structure, circuit and functions of the delayered IC. This task requires photographing the exposed chip followed by an analysis of the pictures to identify the components (such as transistors, coils, resistors and capacitors) and their interconnections. At the end of the analysis, a

standard *netlist*[2] file may also be created that can then be used to create an identical device to the target. A practical example of circuit tracing and extraction is described in.

IoT devices are often deployed in hostile environments where an adversary can physically access and tamper with them. Physical attacks are often used to identify new vulnerabilities in an IoT system by: (1) unsoldering the target device and reading out contents of flash memory; and (2) attacking the device's microcontroller chip using the aforementioned techniques. Physical attacks can be applied to launch a number of attacks on IoT nodes. The most common ones are the following:

1. **Node capture:** This involved taking control of an IoT node through physical attacks, e.g. node circuit modification, reading the contents of internal components via microprobing, etc. Node capture is a serious attack through which an attacker can perform various operations, such as making arbitrary queries on behalf of the intruder or providing false data to legitimate users, and can compromise the entire network[22].
2. **Node outage:** In this attack, an IoT sensor node or parent node is completely stopped. The main purpose of node outage is to disrupt communication between nodes and/or different clusters of nodes[23].
3. **Node malfunction:** It is the erroneous operation of a sensor node. This can be due to many different factors: faulty sensors, energy depletion, circuit modification, etc.
4. **Node destruction:** This refers to the physical destruction of nodes by any means, such as electrical surge, physical force and/or ammunition. The aim of this attack is to permanently damage nodes.
5. **Node replication:** Node replication refers to one or more node(s) illegitimately claiming the identity of legitimate node(s). Due to the physical exposure of IoT nodes to malicious actors, they can be replaced with replicated nodes that can communicate with the network[24]. Data collected from captured nodes can be used to re-program a node or create a replica node.

3.2.2. Side channel analysis

A side channel is immutable information released by a computer device while it performs tasks. The information is immutable because it cannot be suppressed. Such information reveals important details about the internal state of the computer device, such as instructions executed, data processed, states of registers, etc. Previously, side channel information has been widely utilised to extract cryptographic keys from algorithms such as AES[25–27], DES[28,29] and RSA[30]. However, there have also been attempts to use the side channel information of a computer device to reverse engineer embedded applications[31], verify embedded application runtime integrity[32,33], identify a device[34], detect malware[35,36], etc. Immutable information can also be used to understand the internal workings of IoT sensors, control devices or any other hardware security components. For instance, Jung-Change *et al.*[37] demonstrated that an attacker can identify Z-wave devices inside a household by simply analysing the side-channel signal. Another example is the interceptive side-channel attack, where the electromagnetic signal emitted by an IoT device can be correlated with an intercepted wireless signal to extract a secret 128-bit encryption key[38].

3.2.3. Cryptanalysis

Cryptanalysis is the study of ciphertexts [3], cryptographic algorithms [4] and cryptosystems [5] with the aim of understanding how they work and finding and improving techniques for defeating or weakening them. The main objective of cryptanalysis is to find weaknesses in or otherwise defeat cryptographic algorithms. However, the result of cryptanalysis can also be used to strengthen or replace flawed algorithms. In IoT, cryptanalysis attacks

[2]A netlist is a description of the circuit connectivity of an integrated circuit.

[3]Ciphertexts are the result of an encryption algorithm on plaintext data.

[4]Cryptographic algorithms are a set of mathematical functions designed to transform data from readable form to protected form and back.

[5]Cryptosystems are a suit of cryptographic algorithms designed to implement a particular security service.

target encrypted communications among sensors, control devices and gateways with the objective of extracting the private information they exchange without the knowledge of the encryption keys. More information on the cryptanalysis attacks on IoT systems can be found in [39–41].

3.2.4. Network and protocol attacks

Network and protocol attacks are attacks that primarily target the connectivity of an IoT system. They aim to either disrupt the communication medium or exploit weaknesses in the communication protocol. Below, we discuss some of the most common network and protocol attacks.

1. **Jamming:** The communication link is a critical component of IoT systems, and jamming is the deliberate act of blocking or interfering with authorised wireless communications. In most cases, jamming works by transmitting radio signals that disrupt communications by decreasing the signal-to-noise ratio. This concept can be used in wireless data networks to disrupt information flow. In IoT networks, jamming can be used to disrupt the communications between the sensors and their control devices or the gateway. Many networks, including wireless sensor networks (WSNs), use shared control channels for sending system control information. For example, in GSM, broadcast channels (BCH) carry the network identity, the structure of the current control channel and synchronisation information. Without such information, no user can communicate. However, this also makes the system vulnerable to jamming attacks. An attacker can launch a denial of service (DoS) attack on the network by jamming BCH instead of jamming the whole frequency band to stop the communication [42].
2. **Monitoring and eavesdropping:** According to the Unit 42 2020 IoT threat report, 98% of all IoT device traffic is unencrypted, exposing personal and confidential data on the network [1]. An adversary can simply gain access to private information by monitoring communications between IoT devices. A few wireless receivers placed strategically may be able to monitor the communications between IoT sensors and reveal detailed information about their communications [43]. Encrypting communications partly solves the eavesdropping problem but requires a robust key exchange and distribution scheme. However, such a scheme must be simple enough for the network to execute it and feasible for the resource-constrained sensors to implement it. The sheer number of communicating things in an IoT network makes end-to-end encryption impractical, since they cannot store a large number of encryption keys. Instead, most networks deploy hop-by-hop encryption, in which devices only establish a shared key with their neighbors. In such a configuration, capturing an IoT device simply renders encryption useless for any communication through the compromised device.
3. **Traffic analysis:** The traffic pattern of a network may be as valuable as the content of data packets for adversaries. Important information about the networking topology can be derived by analysing traffic patterns. The nodes closer to the base station, i.e. the sink, make more transmissions than the other nodes because they relay more packets than the nodes farther from the base station. Similarly, clustering is an important tool for scalability, and cluster heads are busier than the other nodes in the network. Detection of the base station, the nodes close to it or cluster heads may be very useful for adversaries because a denial-of-service attack against these nodes or eavesdropping on the packets destined for them may have a greater impact. By analysing the traffic, this kind of valuable information can be derived. Moreover, traffic patterns can pertain to other confidential information such as actions and intentions. In tactical communications, silence may indicate preparation for an attack, a tactical move or infiltration. Similarly, a sudden increase in the traffic rate may indicate the start of a deliberate attack or raid.
4. **Routing attacks:** Routing attacks are attacks that target the network layer of the IoT infrastructure. Routing attacks manipulate the flow of information to disturb, deny or compromise the services. The following are some of the common routing attacks.
    - **Selective forwarding:** In this attack, a corrupted/compromised node within the IoT network drops data packets selectively or randomly, trying to corrupt the network with respect to packet loss rate. The node may get compromised through any physical or logical attack technique or methodology. In

a selective forwarding attack, compromised nodes may be used for different purposes: (a) prohibiting the flow of information from other authorised devices/nodes, thus denying services; (b) neglecting to forward information from other devices/nodes and instead only sending its own packets; and/or (c) delaying the messages flowing through them to mislead the routing of data between other nodes. More information on selective forwarding attacks can be found in [44–48].

- **Bogus routing information:** Here, an attacker forwards ongoing messages to the wrong routing path intentionally. This can be achieved by fabricating bogus routing information advertisements and causing routing tables of other devices to update with this false information [49,50].
- **DNS tunnelling:** DNS tunnelling is an attack that encodes data of other programs into DNS queries and responses. DNS tunnelling provides an always-available back channel using DNS protocol as a covert communication channel [51–53]. DNS tunnelling can be used for: (1) sneaking data, such as malware, into a network; (2) sneaking out stolen information, such as customer data, from a private network; (3) remotely controlling compromised internal hosts; and/or (4) bypassing captive portals to avoid paying WiFi.

5. **Denial of service (DoS):** DoS attacks are the most common and easiest to implement on IoT systems [54]. They are defined as attacks that undermine the network of systems' capacity to perform expected functions and can come in different forms. They can be launched at different components of the system, including the physical, data link, transport and application layers. As mentioned above, the main purpose of DoS is to disable a service or slow it down by consuming its resources such as network bandwidth, memory, TCP sessions, etc. Traditionally, DoS attacks are used against networks and backend services and systems. However, according to the authors of [55], DoS is one of the most prominent threats to the security of an IoT system. DoS attack can be realised here by flooding the network with traffic or sending specially formatted information to trigger a crash [56]. More on DoS attacks on IoT systems can be found in [57,58]. Another DoS threat regarding the IoT systems is the use of connected devices as an attack botnet[6] to launch a distributed DoS attacks on other systems [59,60]. A high-profile example of such an incident is the Mirai attack that almost brought the Internet down [61].

3.2.5. Software attacks

Software attacks are cybersecurity attacks that target the software components of an IoT system. Mixed deployment of IoT and IT assets without proper network segmentation within an organisation increases the possibilities of threats in spaces that had never posed cybersecurity risks before. This can have an effect on critical systems, such as an intranet and database servers. IoT software threats are depicted in Figure 4.

According to Palo Alto's Unit 42 2020 IoT threat report [1], software threats are categorised into three groups: **exploits**, **malware** and **user practice**.

1. **Exploits:** According to the report, 41% of all threats fall under the category of *exploits*. Network, IP, port and vulnerability scans attempting to identify other systems are becoming increasingly frequent [62–64]. *IoT-Seeker* is one such scanning tool available as an open source program [65]. Some of the most common exploits are remote code execution, command injection, buffer overflow, SQL injection and zero-day vulnerabilities.
2. **Malware:** By definition, *malware* is maliciously designed software to infiltrate, gain access and/or damage a computer system. By the end of 2019, malware makes up 33% of all IoT software threats. According to CNNMoney [66], in 2014, nearly 1 million malware threats were released every day. More recent research indicates that *rootkits*, *ransomware*, *bots*, *viruses*, *worms* and *trojan horses* rank as the most common malware threats [67,68]. More details of the most common IoT security issues and important findings in recent years

[6] A botnet consists of several compromised devices controlled as a group without the consent of the owner.

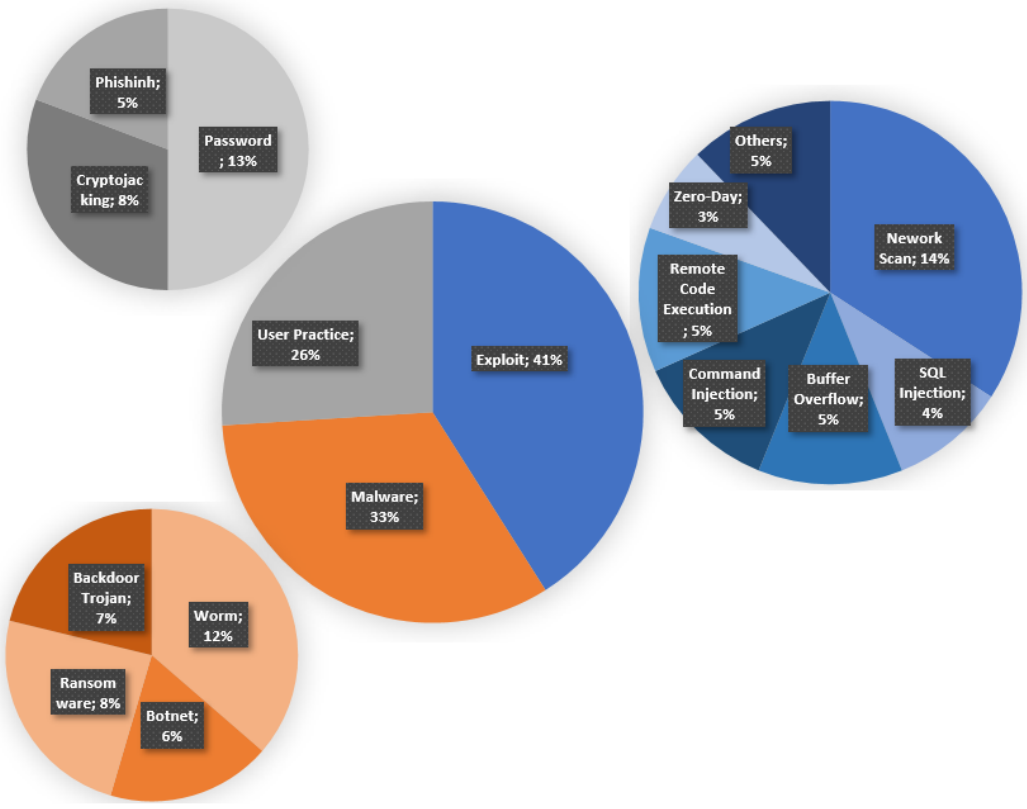

Figure 4. Breakdown of IoT software attacks. *Data taken from the Unit 42 2020 IoT threat report* [1].

are provided in [69,70].

3. **User practice:** The third category of IoT software threat is user practice, which accounted for about 26% of all threats in 2019. User practice threats refer to threats caused by the misuse of software features and policies. Default passwords and poor password security practices are the most common user practice threats. These threats continue to fuel password-related attacks on IoT devices. One high-profile example of an attack caused by password-related weakness is the **Mirai botnet** [61], which exploited CCTV cameras with default passwords to launch a large-scale DDoS attack. However, with California's SB-327 IoT law [71] now prohibiting the use of default credentials, we can expect this trend to change direction.

3.2.6. Social engineering

Social engineering is broadly used to describe a range of malicious activities accomplished through human interactions. It uses psychological manipulation to exploit users into making mistakes that weaken the security of a system or directly give out sensitive information. Attackers use social engineering techniques to collect login credentials for computer systems, which they can then use to get access to them. Malicious entities that use social engineering attacks typically use emails, phone calls or text messages to interact with their targets. The growing use of Internet-connected devices coupled with their dubious security makes IoT an attractive prospect for social engineering attacks. For instance, from December 2013 to January 2014, security provider Proofpoint detected a large number of malicious emails, three times a day in bursts of 100,000 emails, originating from IoT devices targeting businesses and individuals [72]. Successful social engineering attacks through IoT systems could lead to the perception of being surrounded by hostile devices and greatly retard development, making the consequences of social engineering attacks in the IoT very significant [73]. IoT represents a whole new and fertile territory for social engineering attacks, which blend some of the most effective attacks from the contemporary Internet with attacks more commonly found in the industrial-control world. Namely,

they mix attacks that seek to combine attacks intended to capture information with intrinsic value (passwords, account details and access to vulnerable systems) with those that seek to trick users into executing complex sequences of commands on the basis of misinformation.

3.3. Impacts

Attacks aim to disable, disrupt, destroy or control computer systems or to alter, block, delete, manipulate or steal the data held within these systems. An attack can be carried out to achieve one or more of these aims and their impacts vary from one attack to another as well as from one system to another. In this section, we categorise the possible impacts of an attack on an IoT system into six categories.

3.3.1. Data theft

Data theft is the act of illegal transfer of digital information stored in computer systems with the intent of compromising the victim's privacy or obtaining confidential information, such as personal, medical and/or financial information. IoT devices collect vast amounts of information about individuals and organisations. This information is used by organisations to get insights into their customers, deliver targeted advertising and improve sales. Similarly, such information can also be used by governments to deliver efficient services such as equitable resource distribution and service provisioning. Unfortunately, such information can also be stolen by criminals, and the dubious security around IoT systems means the information may easily fall into the wrong hands.

3.3.2. False data injection

False data injection refers to the undetected introduction of erroneous data into a system, which can influence future decisions that are based on analysing the collected datasets. False data injection was first introduced in the smart grid domain but has since attracted interest in other domains such as healthcare, finance, defence, etc.[74,75]. In an IoT environment, false data can be injected into the system through captured and physically altered devices, gaining access to the backend data storage infrastructure or taking advantage of insecure communication protocols. For instance, the McAfee research team took advantage of vulnerabilities in the RWHAT[7] protocol to inject faked patient data into the central monitoring station[76]. More information on false data injection in the healthcare domain is available at[77].

3.3.3. Data destruction

Data destruction is the process of deleting data stored on electronic media so that it cannot be accessed. Adversaries may destroy data and files on specific systems or in large numbers on a network to interrupt the availability of systems, services and network resources. Data destruction is likely to render stored data irrecoverable by forensic techniques through overwriting files of data on local and remote drives[78–80].

3.3.4. Data manipulation/modification

Beyond the simple theft or destruction of data, an adversary may also try to use the data in a more clever way. For instance, the attacker may manipulate and/or modify the acquired data or a subset of them.

- **Data manipulation** is the act of rearranging data or intentional omission of a subset of data to produce a predetermined result. For example, in 2016, Russian hackers breached and released data from the world anti-doping agency, revealing many famous athletes' medical records. However, by manipulating the data before their release, they targeted athletes participating in the 2016 games[81].
- **Data modification** is a fraudulent activity wherein an attacker alters, tweaks or modifies valuable digital assets and critical data. Data modification attacks take a great deal of time to recover from, with outcomes

[7]RWHAT is a proprietary communication protocol used by patient vital monitoring devices that uses unencrypted UDP packets.

Table 3. Attacks on IoT *vs.* their potential impacts

Attacks (Attack categories presented in *Section 3.2*)	Impacts (Potential impacts presented in *Section 3.3*)
Physical attacks	Device Malfunction/Destruction
Side channel attacks	Data Theft
Cryptanalysis	Data Theft False Data Injection Data Manipulation/Modification
Network & Protocol	Data Theft False Data Injection Data Destruction Data Manipulation/Modification Service Disruption/Destruction
Software attacks	Data Theft False Data Injection Data Destruction Data Manipulation/Modification Service Disruption/Destruction
Social engineering	Data Theft Service Disruption/Destruction Device Malfunction/Destruction

potentially including data theft and data destruction. The main reason for this is the difficulty in determining the extent of data manipulation and exactly how the data were manipulated. In the age of big data, if an attacker made under-the-radar data modifications, the victim would have to sift through all the data to check and double-check that every piece of information is accurate.

3.3.5. Service disruption/destruction

Denial and/or disruption of a service is a result of an attack that intends to permanently or temporarily shut down a machine, service or network, making it inaccessible to its intended users.

- **Disruption** is shutting down critical components of a service to make it inaccessible or unreliable, usually temporarily.
- **Destruction** is making the targeted service inaccessible to the intended users temporarily.

3.3.6. Device malfunction/destruction

Another impact of an attack on an IoT system is the malfunction or destruction of a device. Device malfunction refers to a device producing erroneous results, while device destruction is permanent damage to the target device. Device malfunction or destruction can be caused by a variety of attacks (e.g., physical attacks that modify the device circuit, fault injection attacks, etc.) or accidents (e.g., dropping the device, electric surge, etc.).

Attacks on an IoT system can have different impacts depending on the objectives of the attacker. Table 3 presents the connection between the different categories of attacks (see Section 3.2) and their potential impacts (see Section 3.3).

As presented above, physical attacks and side channel attacks lead to device malfunction/destruction and data theft impacts, respectively. Table 3 also shows that attacks that fall under the same category can have completely different impacts. For instance, a cryptanalysis attack on an encryption algorithm may lead to data theft (e.g., a successful cryptanalysis attack on an encryption attack may enable an attacker to successfully extract encrypted contents without the keys) or a similar attack on a hash function may lead to false data injection and/or data manipulation/modification (e.g., if the attacker successfully breaks the hash algorithm, it means they can inject new data or change existing data and recalculate the hash-based integrity value). Similarly, attacks on networks and communication protocols may lead to one or more of the impacts listed in Table 3. For instance, a jamming

attack can be launched to disrupt services, a DNS tunnelling attack to sneak data into or out of a network covertly and selective forwarding to drop (data destruction) information. Similar to network and protocol attacks, software attacks can also be launched to achieve a wide range of objectives, thus leading to different impacts. Social engineering attacks on IoT may also be used to get hold of user/corporate data, disrupt services and/or exploit consumer devices for illicit activity (such as sending out spam emails), thereby leading to device malfunctioning.

4. COUNTERMEASURES

Security is a multi-layer approach in any system or network, and IoT is no different. In any large system, security controls are applied at software, operating system, network and hardware levels. The specifics of each security control are dependent on the security need, system architecture, underlying infrastructure and threat landscape. To illustrate the threat posed to IoT systems, we map the different attacks against the assets they primarily target in Figure 5. With this principle, an IoT system is expected to have different security controls at different levels of the system; for instance, at the network level, it is expected to have firewalls and segmented network configuration, as well as intrusion detection and prevention at the back-end infrastructure. However, what makes security in IoT more challenging than in traditional networks is the nature and deployment of the sensors. Several security countermeasures are proposed addressing specific challenges in the IoT environment[82–86]. Discussing every countermeasure separately is practically an impossible task as there are hundreds, if not thousands, of them. In this paper, we categorise them into groups and discuss the common techniques that countermeasures employ to protect the underlying asset and infrastructure.

4.1. Encryption

Overall, 98% of all IoT device traffic is unencrypted, exposing personal and confidential data on the network[1]. Attackers who have successfully bypassed the first line of defense, most frequently via phishing attacks to get access to the network, are able to listen to unencrypted network traffic and collect personal or confidential information, and they can exploit that data for profit on the dark web. One of the security controls that can be deployed to protect data confidentiality is encryption. Encryption needs a shared key among the communicating nodes. Several encryption algorithms can be applied to protect the data at different points of the system. For instance, a stream encryption algorithm can be applied to protect the Bluetooth link between the sensors and the gateway and standard HTTPS to protect the data between the gateway and the back-end services. The main challenge of encryption in the IoT environment is the security and efficiency of the key sharing algorithm employed. To address this challenge, several sensor key sharing schemes have been proposed (see, e.g., [87–90]).

4.2. Redundant architecture

Redundant architecture refers to the deployment of identical systems to enhance the availability of services and systems. In large networked systems, such as the IoT, there are various levels of high availability that can be deployed; hence, organisations need to identify the level they want to achieve. The acquisition and operational cost of high-availability redundant architectures are other important factors that determine what level of availability is needed by organisations. In general, there are four redundancy architectures organisations can choose from:

- ***Data redundancy***: Data redundancy refers to architectures that keep the same data in multiple locations across the network. This is to preserve important data from attacks that intend to manipulate or destroy them. However, data redundancy may also lead to data inconsistency if all instances are not synchronised when a change is applied to one of them.
- ***Service redundancy***: The IoT system comprises a number of back-end services that consume the data generated by sensors. One way of ensuring the availability of these services from attacks is to deploy multiple instances of critical service in order to provide high service availability. This is achieved by deploying backup services to take over critical services in the event of an attack on one or more of them.

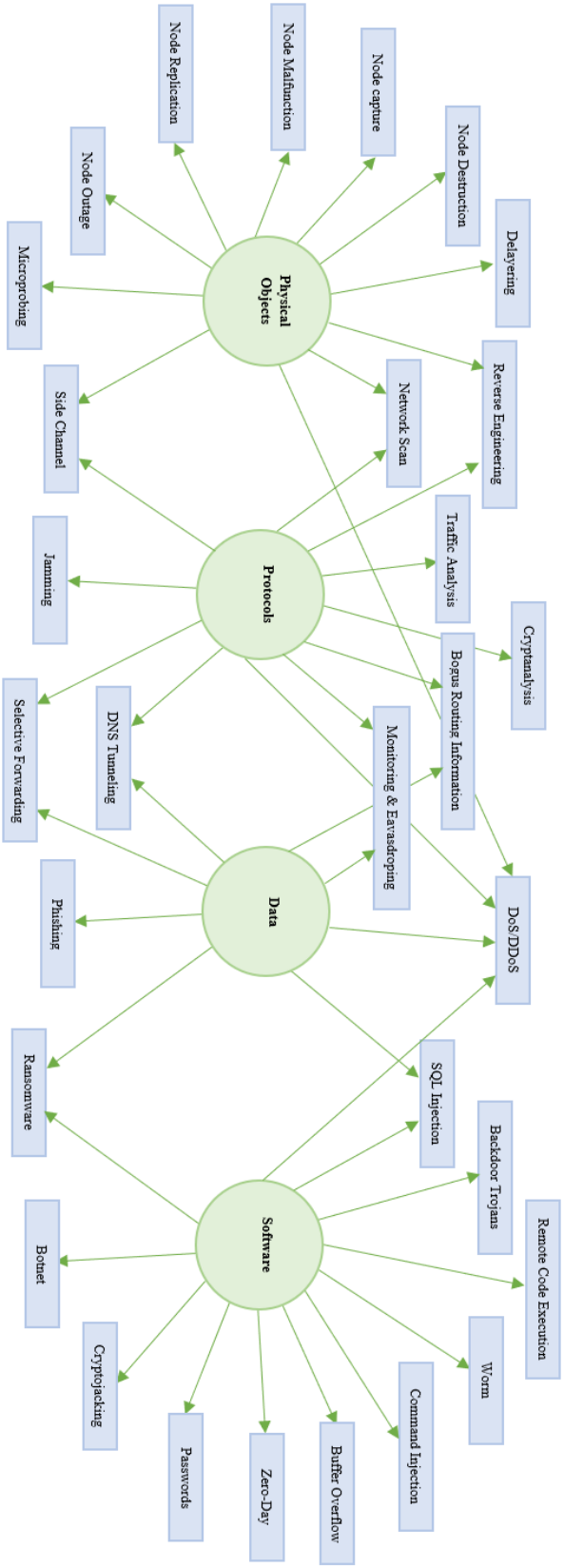

Figure 5. Classification of IoT attacks *vs.* assets.

- ***Link (communication) redundancy***: This is the process of adding additional instances of network devices and lines of communication to help ensure link availability and decrease the risk of failure along the critical data path. Redundant links in IoT must also support multiple communication protocols between devices and gateway/control devices, for example, Bluetooth, WiFi or NFC.
- ***Device redundancy***: A high level of device availability effectively means doubling up on devices to ensure that a backup device can take over in the event of a failed device. With this strategy, no single device failure should result in a loss of data generated by the sensor devices. Typically, device redundancy and link redundancy are coupled, which means that if one fails, the other follows.

4.3. Anti-Tamper mechanisms

Anti-tamper mechanisms are a set of controls put in place to detect and respond to physical tampering attacks, such as delayering and micro-probing, on the security of the target device. The main purpose of anti-tampering is to prevent unauthorised physical or electronic tampering against the product and is most effective when used in layers. There are two types of anti-tamper mechanisms: tamper resistance and tamper detection. Tamper resistance refers to the use of specialised materials to make tampering difficult, such as one-way screws and epoxy encapsulations. Tamper detection enables the hardware device to be aware of tampering in progress. Common examples of tamper detection are switches (detects the opening of a device or breach of security boundary), sensors (detect an operational and environmental change such as temperature, voltage and radiation) and circuitry (detect a puncture, break or attempted modification of a defined security envelope). Another important aspect of the anti-tamper mechanism is the response to tampering. Tampering response refers to countermeasures taken upon the detection of physical attacks, for instance, memory erasure, shutting down/disabling the device and/or enabling logging. Along with responding to tampering, the anti-tamper mechanism is also often designed to leave evidence behind, which can then later be used to check for deformity. Examples of tamper evidence are passive detectors (seals, tapes and glues) and special enclosure finishes. The details of various physical attacks and anti-tamper mechanisms are available in [91].

4.4. Network security controls

Network security controls protect the network infrastructure and data from breaches, intrusions and other threats. Securing a network requires a layered security approach. Various controls must work seamlessly to secure the network in its entirety. Some of the most common network security controls that can be deployed are as follows:

- *Firewall:* A firewall is a network security system that monitors and controls the incoming and outgoing network traffic based on predetermined security rules. A firewall typically establishes a barrier between a trusted network, such as the organisation's internal network, and an untrusted network, such as the Internet. The main purpose of a firewall is to protect the IoT systems, services and data from unauthorised malicious or unnecessary network traffic.
- *Network segmentation:* Network segmentation is an architectural approach that divides a network into multiple subnets, each acting as its own small network. Segmented networks often reflect the organisational structure and roles of organisational units. Overall, 72% of healthcare VLANs mix IoT and other IT assets, allowing malware to spread from users' computers to vulnerable IoT devices on the same network [1].
- *Intrusion detection:* Intrusion detection is a network device/system that monitors for malicious activities or policy violations. Any malicious activity or policy violation is then reported to an administrator or is collected and further processed by security information and management systems.
- *Virtual private network (VPN):* A VPN is a network device that extends the private network across an untrusted public network by applying strong encryption and integrity measures to the data while in transit.

4.5. Access control

Access control is a critical security component that enforces a policy on who is allowed to access and utilise an organisation's data and services. A typical access control system will have two separate modules:

Table 4. Effectiveness of countermeasures against Internet-of-Things assets

Countermeasures	Internet-of-Things assets			
	Physical objects	Data	Link/protocols	Software
Encryption	✗	✓	✗	✗
Data Redundancy	✗	✓	✗	✗
Service Redundancy	✗	✗	✗	✓
Link Redundancy	✗	✗	✓	✗
Device Redundancy	✓	✗	✗	✗
Anti-Tamper Mechanisms	✓	✗	✗	✗
Network Security Controls	✗	✓	✓	✓
Access Control	✓	✓	✗	✓
Secure Configuration	✓	✗	✗	✓

- *Authentication:* Authentication is the process of recognising an entity's identity. In an IoT environment, the authentication module verifies the identity of the connected devices and users. A strong IoT device authentication is required to ensure connected devices on the IoT can be trusted to be what they purport to be.
- *Authorisation:* Authorisation is the process of restricting authenticated entities' access to a resource, such as hardware, software, data or the communication link.

4.6. Secure configuration

System misconfigurations are one of the most common security gaps that attackers look to exploit. According to a report by Rapid 7, internal penetration tests encounter a network or service misconfiguration more than 96% of the time[92]. Product manufacturers often set the default configurations of new software and devices to be as open and multi-functional as possible. In the case of an IoT device, for example, this could be a predefined password, or in the case of an operating system, it could be the applications that come preinstalled. It is easier and more convenient to start using new devices or software with their default settings, but it is not the most secure. Accepting the default settings without reviewing them can create serious security issues and can allow attackers to gain easy, unauthorised access to your data. Failure to properly configure systems and services can lead to a wide variety of security problems.

Each of the aforementioned countermeasures protects certain assets against certain attacks. Often a single countermeasure provides different levels of protection to one or more assets. Table 4 illustrates the effectiveness of the countermeasures against the assets they are deployed to protect. In Table 4, we can observe that certain countermeasures protect a single asset while others protect multiple assets. For instance, access control protects the data, link and software assets but often does so by employing different techniques on each asset. We can also observe different countermeasures to protect the same asset by employing different techniques. For example, device redundancy, anti-tampering, access control and security configuration protect the physical objects, but they all provide protection against different threats. Device redundancy ensures services continue uninterrupted even in the event of device failure by maintaining a backup, while anti-tampering protects sensitive and secret intellectual property (IP) by detecting and often destroying the device in the event of physical attacks. Access control and security configuration restrict access and eliminate the threats of default credentials, respectively.

Similarly, more often than not, multiple countermeasures mitigate the same impact and a single countermea-

Table 5. Effectiveness of countermeasures against impacts of IoT attacks

	Attack Impacts					
Countermeasures	Data Theft	Data Destruction	Data Manipulation/ Modification	False Data Injection	Service Disruption/ Destruction	Device Malfunction/ Destruction
Encryption	✓	✗	✓	✓	✗	✗
Data Redundancy	✗	✓	✓	✓	✓	✗
Service Redundancy	✗	✗	✗	✗	✓	✗
Link Redundancy	✗	✗	✗	✗	✓	✗
Device Redundancy	✗	✗	✗	✗	✗	✓
Anti-Tamper Mechanisms	✗	✗	✗	✗	✗	✓
Network Security Controls	✓	✓	✓	✓	✓	✗
Access Control	✓	✓	✓	✓	✓	✗
Secure Configuration	✓	✗	✗	✗	✓	✓

sure mitigats multiple impacts. Table 5 shows the effectiveness of the countermeasures against the possible impacts of an attack on an IoT system. In this evaluation, data refer to information both in transit (on the network) and at rest (in storage platforms). For instance, *encryption* is an adequate countermeasure against *data theft*, *data manipulation/modification* and *false data injection* when applied to protect data in transit. Applying the same technique to protect data at rest comes at the cost of degrading the overall performance. This, however, could be mitigated by applying it only to highly sensitive data that is selected at rest. The correct application of network security and access control countermeasures can mitigate the impacts of attacks such as data theft, data destruction, data manipulation/modification, false data injection and service disruption/destruction. Similarly, device redundancy and anti-tampering mitigate the impacts of attacks that intend to destroy or damage IoT devices.

5. OPEN RESEARCH ISSUES

IoT is growing in both application areas and complexity. This, however, also means such systems are exposed to different actors in different environments. The growing complexity also means IoT systems will likely have more vulnerable points, and it is less likely that security measures will be effective. In the long term, we believe the following topics will continue to be important research challenges in realising the full potential of IoT systems.

5.1. Tamper resistance and detection

In many application use case scenarios, IoT sensors and devices are deployed in the field or areas that are easily accessible by anyone and are exposed to an array of physical tampering attacks. Several physical tamper resistance/detection mechanisms have previously been used to protect embedded devices such as dual-execution rails, mesh circuit and crystals under the integrated circuit epoxy resins. We believe that cost-effective and miniaturised tamper resistance/detection countermeasures are critical to the security of IoT networks and systems.

5.2. Side channel protection

Another critical research area is the side channel protection mechanism. IoT devices store sensitive information such as cryptographic keys, and thus can be targeted with side channel analysis attacks. To protect such sensitive information, it is paramount that IoT devices are equipped with side channel protection mechanisms.

5.3. Device authentication

One of the most challenging tasks in IoT is verifying the identity of connected devices. Data gathered through IoT sensors are increasingly being used in making important decisions, and sometimes those decisions are the difference between life and death. Such decisions can only be trusted if the source of data can also be trusted. That can only be achieved through a strong device authentication mechanism.

5.4. Re-Imagining the purpose of IoT gateway

IoT gateways are used to link IoT sensors/devices with back-end services such as the cloud. They also act as a communication protocol translator for Bluetooth, WiFI, SigFox, LoRa, NEC, etc. This is important as it enables them to support a host of heterogeneous IoT devices. However, a gateway can also have a broader purpose in securing IoT networks. Gateways are ideal for performing certain security functions, such as authenticating connected devices and authorising access to back-end services, as they are physically closer to the sensors and are much more computationally capable of performing heavy-duty security functions.

6. CONCLUSION

The IoT, attacks on it and countermeasures have been covered extensively in the literature. However, they are often discussed in isolation, which leads to a fragmented understanding of the IoT threat landscape. Understanding the security of an IoT system requires an understanding of the system in its entirety. This paper provides a holistic view of the security of IoT. A detailed discussion on the anatomy of attacks on Internet of Things is presented. The different layered architectural representations of an IoT are described along with the four functional components, which also represent the different technological advancements within the IoT ecosystem. The various computational models that can be deployed to store and analyse the collected data are also discussed.

The main assets of a typical IoT system are identified and briefly described. Attacks are categorised into groups of similar nature. The similarity is based on the components these attacks target, the skills required for a successful launch and the severity of their impact. The attacks are then mapped against the assets they primarily target to illustrate the threat landscape of IoT. Finally, possible countermeasures are discussed. Discussing individual countermeasures is almost impossible as there are too many of them. Hence, similar to the attacks, the countermeasures are also classified into groups based on the protection they offer and the general technique they employ. The effectiveness of these countermeasures is then evaluated for the assets they are deployed to protect and against the attack impacts they intend to prevent or minimise.

DECLARATIONS

Authors' contributions

Made substantial contributions to the conception of the study, systematic review and analysis of related literature materials, and state-of-the-art security of Internet of Things systems: Msgna M

Availability of data and materials

Not applicable.

Financial support and sponsorship

Not applicable.

Conflicts of interest

The author declared that there are no conflicts of interest.

Ethical approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Copyright

REFERENCES

1. Palo Alto Networks. 2020 unit 42 IoT threat report. Available from: https://unit42.paloaltonetworks.com/iot-threat-report-2020/. [Last accessed on 22 Nov 2022]
2. Kshetri N, Voas JM. Cryptojacking. *Computer* 2022;55:18-19. DOI
3. Porras PA, Saïdi H. A foray into conficker's logic and rendezvous points. In: Lee W, editor. 2nd USENIX Workshop on Large-Scale Exploits and Emergent Threats, LEET '09. USENIX Association; 2009. DOI
4. Networks P. The connected enterprise: IoT security report 2021. Available from: https://www.paloaltonetworks.com/apps/pan/public/downloadResource?pagePath=/content/pan/en_US/resources/research/connected-enterprise-iot-security-report-2021. [Last accessed on 22 Nov 2022]
5. Certified P. Security report 2021: bridging the gap; 2021. (Last accessed) January 2022.
6. Cisco. Fog computing and the internet of things: extend the cloud to where the things are. Available from: https://www.cisco.com/c/dam/en_us/solutions/trends/iot/docs/computing-overview.pdf. [Last accessed on 22 Nov 2022]
7. IDC. IDC futureScape: worldwide internet of things 2015 predictions. Available from: https://www.businesswire.com/news/home/20141203006197/en/IDC-Reveals-Worldwide-Internet-of-Things-Predictions-for-2015. [Last accessed on 22 Nov 2022]
8. Mach P, Becvar Z. Mobile edge computing: a survey on architecture and computation offloading. *IEEE Commun Surv* 2017;19:1628-56 DOI
9. ISO. ISO/IEC 27001 - information security management. Available from: https://www.iso.org/isoiec-27001-information-security.html. [Last accessed on 22 Nov 2022]
10. Makhdoom I, Abolhasan M, Lipman J, Liu RP, Ni W. Anatomy of threats to the internet of things. *IEEE Commun Surv Tutor* 2019;21:1636-75. DOI
11. Mayes K, Markantonakis K. Smart cards, tokens, security and applications. 1st ed. Springer; 2007. DOI
12. Anderson RJ. In: Security engineering - a guide to building dependable distributed systems. second edition ed. Wiley; 2008. Available from: https://www.amazon.com/Security-Engineering-Building-Dependable-Distributed/dp/0470068523. [Last accessed on 22 Nov 2022]
13. Tria A, Choukri H. Invasive attacks. In: van Tilborg HCA, Jajodia S, editors. Encyclopedia of cryptography and security (2nd Ed.). Springer; 2011. pp. 623-29. DOI
14. Helfmeier C, Nedospasov D, Tarnovsky C, et al. Breaking and entering through the silicon. In: Sadeghi A, Gligor VD, Yung M, editors. ACM SIGSAC Conference on Computer and Communications Security, CCS'13, Berlin, Germany, November 4-8, 2013. ACM; 2013. pp. 733-44. DOI
15. Filipovic B, Schimmel O. Protecting embedded systems against product piracy: technological background and preventive measures. Available from: https://www.aisec.fraunhofer.de/content/dam/aisec/Dokumente/Publikationen/Studien_TechReports/englisch/Whitepaper_ProductProtection.pdf. [Last accessed on 22 Nov 2022]
16. Hutle M, Kammerstetter M. Resilience against physical attacks. In: Skopik F, Smith P, editors. Smart Grid Security: innovative solutions for modernized grid. Syness 2015. pp.79-112. DOI
17. Swapp S, University of Wyoming. Scanning eletron microscopy(SEM). Available from: http://serc.carleton.edu/research_education/geochemsheets/techniques/SEM.html. [Last accessed on 22 Nov 2022]
18. The university of Iowa. Transmission electron microscopy. Available from: http://cmrf.research.uiowa.edu/transmission-electron-microscopy. [Last accessed on 22 Nov 2022]
19. Torrance R, James D. The state-of-the-art in IC reverse engineering. In: Clavier C, Gaj K, editors. 11th International Workshop Cryptographic Hardware and Embedded Systems - CHES. vol. 5747 of Lecture Notes in Computer Science. Springer; 2009. pp. 363-81. DOI
20. Kömmerling O, Kuhn MG. Design principles for tamper-resistant smartcard processors. Available from: https://www.usenix.org/legacy/publications/library/proceedings/smartcard99/full_papers/kommerling/kommerling.pdf. [Last accessed on 22 Nov 2022]
21. Bar-El H, Bar H. Discretix Technologies Ltd. Known attacks against smartcards. Available from: https://www.siliconinvestigations.com/REF/sftsec.pdf. [Last accessed on 22 Nov 2022]
22. Tague P, Poovendran R. Modeling Node Capture Attacks in Wireless Sensor Networks. Available from: https://mews.sv.cmu.edu/papers/allerton-08.pdf. [Last accessed on 22 Nov 2022]
23. Butun I, Österberg P, Song H. Security of the internet of things: vulnerabilities, attacks, and countermeasures. *IEEE Commun Surv Tutor* 2020;22:616-44. DOI
24. Xie H, Yan Z, Yao Z, Atiquzzaman M. Data collection for security measurement in wireless sensor networks: a survey. *IEEE Internet*

Things J 2019;6:2205-24. DOI
25. Lo O, Buchanan WJ, Carson D. Power analysis attacks on the AES-128 S-box using differential power analysis (DPA) and correlation power analysis (CPA). *J Cyber Secur Technol* 2017;1:88-107. DOI
26. C A, Roy B, Mandarapu BSV, Menezes B. "S-Box" implementation of AES is not side channel resistant. *J Hardw Syst Secur* 2020;4:86-97. DOI
27. Brisfors M, Forsmark S. Deep-learning side-channel attacks on AES.
28. Heyszl J, Miller K, Unterstein F, et al. Investigating profiled side-channel attacks against the DES key schedule. *TCHES* ;2020:22-72. DOI
29. Zhou Y, Feng D. Side-channel attacks: ten years after its publication and the impacts on cryptographic module security testing. Available from: https://eprint.iacr.org/2005/388.pdf. [Last accessed on 22 Nov 2022]
30. Finke T, Gebhardt M, Schindler W. A new side-channel attack on RSA prime generation. In: Clavier C, Gaj K, editors. Cryptographic Hardware and Embedded Systems - CHES 2009. CHES 2009. Lecture Notes in Computer Science, vol 5747. Springer, Berlin, Heidelberg. DOI
31. Msgna M, Markantonakis K, Mayes K. Precise instruction-level side channel profiling of embedded processors. In: Information Security Practice and Experience - 10th International Conference, ISPEC 2014, Fuzhou, China, May 5-8, 2014. Proceedings. vol. 8434 of Lecture Notes in Computer Science. Springer; 2014. pp. 129-43. DOI
32. Msgna M, Markantonakis K, Naccache D, Mayes K. Verifying software integrity in embedded systems: A side channel approach. In: Constructive Side-Channel Analysis and Secure Design - 5th International Workshop, COSADE 2014, Paris, France, April 13-15, 2014. Revised Selected Papers. vol. 8622 of Lecture Notes in Computer Science. Springer; 2014. pp. 261-80. DOI
33. Msgna M, Markantonakis K, Mayes K. The B-side of side channel leakage: control flow security in embedded systems. vol. 127 of Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering. Springer; 2013. pp.288–304. DOI
34. Sayakkara A, Le-Khac NA, Scanlon M. Leveraging electromagnetic side-channel analysis for the investigation of IoT devices. *Digit Investig* 2019;29:94-103. DOI
35. Kune DF, Ransford BA, Holcomb DE. Anomaly and malware detection using side channel analysis. Available from: https://patentimages.storage.googleapis.com/a0/c2/5d/d6df3332818d5c/WO2016115280A1.pdf. [Last accessed on 22 Nov 2022]
36. McDonald T. Side-Channel based detection of malicious software. In: 7th Software Security, Protection and Reverse Engineering Workshop (SSPREW); 2017.
37. Liou J, Jain S, Singh SR, Taksinwarajan D, Seneviratne S. Side-channel information leaks of Z-wave smart home IoT devices: demo abstract. In: Nakazawa J, Huang P, editors. SenSys '20: The 18th ACM Conference on Embedded Networked Sensor Systems, Virtual Event, Japan, November 16-19, 2020. ACM; 2020. pp. 637-8. DOI
38. Pammu AA, Chong K, Ho W, Gwee B. Interceptive side channel attack on AES-128 wireless communications for IoT applications. In: 2016 IEEE Asia Pacific Conference on Circuits and Systems, APCCAS; 2016. pp. 650-3. DOI
39. Mirtskhulava L, Globa L, Meshveliani N, Gulua N. Cryptanalysis of internet of things (IoT) wireless technology. In: 2019 International Conference on Information and Telecommunication Technologies and Radio Electronics (UkrMiCo); 2019. pp. 1-4. DOI
40. Tewari A, Gupta BB. Cryptanalysis of a novel ultra-lightweight mutual authentication protocol for IoT devices using RFID tags. *J Supercomput* 2017;73:1085-102. DOI
41. Dwivedi AD. Security analysis of lightweight IoT cipher: chaskey. *Cryptography* 2020;4:22. DOI
42. Chan A, Liu X, Noubir G, Thapa B. Broadcast control channel jamming: resilience and identification of traitors. *Int Symp Inf Theory* 2007. pp. 2496-500. DOI
43. Chan H, Perrig A. Security and privacy in sensor networks. *Computer* 2003;36:103-5. DOI
44. Khan W, Xiang Y, Aalsalem M, Arshad Q. The selective forwarding attack in sensor networks: detections and countermeasures. *Int J Microw Wirel Technol* 2012;2:33-44. DOI
45. Bysani LK, Turuk AK. A survey on selective forwarding attack in wireless sensor networks. In: 2011 International Conference on Devices and Communications (ICDeCom); 2011. pp. 1-5. DOI
46. Zhang Y, Minier M. Selective forwarding attacks against data and ACK flows in network coding and countermeasures. *J Comput Netw Commun* ;2012. DOI
47. Zhang Q, Zhang W. Accurate detection of selective forwarding attack in wireless sensor networks. *Int J Distrib Sens Netw* 2019;15. DOI
48. Yu B, Xiao B. Detecting selective forwarding attacks in wireless sensor networks. In: Proceedings 20th IEEE International Parallel Distributed Processing Symposium; 2006. DOI
49. Wood AD, Stankovic JA. Denial of service in sensor networks. *Computer* 2002;35:54-62. DOI
50. Karlof C, Wagner D. Secure routing in wireless sensor networks: attacks and countermeasures. In: Proceedings of the First IEEE International Workshop on Sensor Network Protocols and Applications; 2003. pp. 113-27. DOI
51. Zimba A, Chishimba M. Exploitation of DNS tunneling for optimization of data exfiltration in malware-free APT intrusions. Available from: https://ictjournal.icict.org.zm/index.php/zictjournal/article/view/26/13. [Last accessed on 22 Nov 2022]
52. Sammour M, Hussin B, Othman MFI, Doheir M, et al. DNS tunneling: a review on features. Available from: https://www.researchgate.net/profile/Mohammed-Talib/publication/327097730_DNS_Tunneling_a_Review_on_Features/links/5b77aa5c299bf1d5a711cb93/DNS-Tunneling-a-Review-on-Features.pdf. [Last accessed on 22 Nov 2022]
53. Do V, Engelstad PE, Feng B, Thanh D. Detection of DNS tunneling in mobile networks using machine learning; 2017. pp. 221-30. DOI
54. ScienceDirect. Denial-of-service attack. Available from: https://www.sciencedirect.com/topics/engineering/denial-of-service-attack. [Last accessed on 22 Nov 2022]

55. Zhao K, Ge L. A survey on the internet of things security. In: International Conference on Computational Intelligence & Security;. pp. 663-7. DOI
56. Alanazi S, Al-Muhtadi J, Derhab A, et al. On resilience of wireless mesh routing protocol against DoS attacks in IoT-based ambient assisted living applications. In: Application & Services in International Conference on EHealth Networking. DOI
57. Džaferović E, Sokol A, Almisreb A, Mohd Norzeli S. DoS and DDoS vulnerability of IoT: a review. Available from: https://pdfs.semanticscholar.org/de0e/93281cd7005cb17e2426614985766a41a4dd.pdf. [Last accessed on 22 Nov 2022]
58. Liang L, Zheng K, Sheng Q, Huang X. A denial of service attack method for an IoT system. In: 2016 8th International Conference on Information Technology in Medicine and Education (ITME); 2016. pp. 360-64. DOI
59. Olshansky S, Wilton R. Internet of things devices as a DDoS vector. Available from: https://www.internetsociety.org/blog/2019/04/internet-of-things-devices-as-a-ddos-vector/. [Last accessed on 22 Nov 2022]
60. Vlajic N, Zhou D. IoT as a land of opportunity for DDoS hackers. *Computer* 2018;51:26-34. DOI
61. Cybersecurity TNJ, (NJCCIC) CIC. Mirai Botnet. Available from: https://web.archive.org/web/20161212084605/https://www.cyber.nj.gov/threat-profiles/botnet-variants/mirai-botnet. [Last accessed on 22 Nov 2022]
62. Kumar D, Shen K, Case B, et al. All things considered: an analysis of IoT devices on home networks. Available from: https://www.usenix.org/system/files/sec19-kumar-deepak_0.pdf. [Last accessed on 22 Nov 2022]
63. Markowsky L, Markowsky G. Scanning for vulnerable devices in the internet of things. DOI
64. Agarwal S, Oser P, Lueders S. Detecting IoT devices and how they put large heterogeneous networks at security risk. *Sensors* 2019;19:4107. DOI
65. GitHub Inc. IoTSeeker. Available from: https://github.com/rapid7/IoTSeeker. [Last accessed on 22 Nov 2022]
66. Harrison V, Pagliery J. Nearly 1 million new malware threats released every day. Available from: https://money.cnn.com/2015/04/14/technology/security/cyber-attack-hacks-security/. [Last accessed on 22 Nov 2022]
67. Antonopoulos A, Kapatsori C, Makris Y. Hardware trojans in analog, mixed-signal, and RF ICs; 2018. pp. 101-23. DOI
68. Yang K, Hicks M, Dong Q, Austin T, Sylvester D. A2: analog malicious hardware. In: 2016 IEEE symposium on security and privacy (SP); 2016. pp. 18-37. DOI
69. Milosevic J, Sklavos N, Koutsikou K. Malware in IoT software and hardware. Available from: https://upcommons.upc.edu/bitstream/handle/2117/99318/FCTRU_2016_paper_29.pdf. [Last accessed on 22 Nov 2022]
70. Ngo QD, Nguyen HT, Van-HoangLe, Nguyen DH. A survey of IoT malware and detection methods based on static features. *ICT Express* 2020;6:280-6. DOI
71. Senate Bill No 327. Available from: https://openstates.org/ca/bills/20172018/SB327/. [Last accessed on 22 Nov 2022]
72. Heartfield R, Gan D. Social engineering in the internet of everything. Available from: https://gala.gre.ac.uk/id/eprint/16718/7/16718%20GAN_Social_Engineering_in_the_Internet_of_Everything_2016.pdf. [Last accessed on 22 Nov 2022]
73. McAfee. Social engineering in the internet of things (IoT). Available from: https://www.mcafee.com/blogs/other-blogs/executive-perspectives/social-engineering-internet-things-iot/. [Last accessed on 22 Nov 2022]
74. Ahmed M, Pathan AK. False data injection attack (FDIA): an overview and new metrics for fair evaluation of its countermeasure. *Complex Adapt Syst Model* 2020;8:4. DOI
75. Deng R, Xiao G, Lu R, Liang H, Vasilakos A. False data injection on state estimation in power systems - attacks, impacts, and defense: a survey. *IEEE Trans Industr Inform* 2017;13:411-23. DOI
76. Leyden J. McAfee: patient monitoring systems open to hack attacks. Available from: https://www.theregister.com/2018/08/14/patient_monitor_hack/. [Last accessed on 22 Nov 2022]
77. Ahmed M, Ullah ASSMB. False data injection attacks in healthcare. In australasian conference on data mining. Springer;2007. pp.192-202. DOI
78. Symantec. The shamoon attacks. Available from: https://community.broadcom.com/symantecenterprise/communities/community-home/librarydocuments/viewdocument?DocumentKey=281521ea-2d18-4bf9-9e88-8b1dc41cfdb6&CommunityKey=1ecf5f55-9545-44d6-b0f4-4e4a7f5f5e68&tab=librarydocuments. [Last accessed on 22 Nov 2022]
79. Mercer W, Rascagneres P. Olympic destroyer takes aim at winter olympics. Available from: https://blog.talosintelligence.com/2018/02/olympic-destroyer.html. [Last accessed on 22 Nov 2022]
80. Falcone R. Shamoon 3 Targets Oil and Gas Organization. Available from: https://unit42.paloaltonetworks.com/shamoon-3-targets-oil-gas-organization/. [Last accessed on 22 Nov 2022]
81. Perlroth N, Panja T. Microsoft says Russians hacked antidoping agency computers. Available from: https://www.nytimes.com/2019/10/28/sports/olympics/russia-doping-wada-hacked.html. [Last accessed on 22 Nov 2022]
82. Wazid M, Das AK, Khan MK, et al. Secure authentication scheme for medicine anti-counterfeiting system in IoT environment. *IEEE Internet Things J* 2017;4:1634-46. DOI
83. Zhou J, Cao Z, Dong X, Lin X, Vasilakos AV. Securing m-healthcare social networks: challenges, countermeasures and future directions. *IEEE Wirel Commun* 2013;20:12-21. DOI
84. Lin C, He D, Huang X, Choo KKR, Vasilakos AV. BSeIn: a blockchain-based secure mutual authentication with fine-grained access control system for industry 4.0. *J Netw Comput Appl* 2018;116:42-52. DOI
85. Yang H, Yuan J, Li C, et al. BrainIoT: brain-like productive services provisioning with federated learning in industrial IoT. *IEEE Internet Things J* 2022;9:2014-24. DOI
86. Srinivas J, Das AK, Wazid M, Vasilakos AV. Designing secure user authentication protocol for big data collection in IoT-based intelligent transportation system. *IEEE Internet Things J* 2021;8:7727-44. DOI

87. Chan H, Perrig A, Song D. Random key predistribution schemes for sensor networks. In: 2003 Symposium on Security and Privacy, 2003.; 2003. pp. 197- 213. DOI
88. Xiao Y, Rayi VK, Sun B, Du X, Hu F, et al. A survey of key management schemes in wireless sensor networks. *Computer Communications* 2007;30:2314-41. DOI
89. Banimelhem O, Al-Haija QA, Al-Badawi A. Performance Evaluation of Probabilistic Key Management Approaches for Wireless Sensor Networks. Available from: https://www.researchgate.net/profile/Qasem-Abu-Al-Haija/publication/259293090_Performance_Evaluation_of_Probabilistic_Key_Management_Approaches_for_Wireless_Sensor_Networks/links/02e7e52acf93db31f0000000/Performance-Evaluation-of-Probabilistic-Key-Management-Approaches-for-Wireless-Sensor-Networks.pdf. [Last accessed on 22 Nov 2022]
90. Du W, Deng J, Han Y, et al. A pairwise key predistribution scheme for wireless sensor networks. *ACM Trans Inf Syst Secur* 2005;8:228-58. DOI
91. Weingart SH. Physical security devices for computer subsystems: a survey of attacks and defenses. In: Koç ÇK, Paar C, editors. Cryptographic hardware and embedded systems - CHES 2000. Berlin, Heidelberg: Springer Berlin Heidelberg; 2000. pp. 302-17. DOI
92. Research R. Under the hoodie: lessons from a Season of Penetration Testing. Available from: https://www.rapid7.com/globalassets/_pdfs/research/rapid7-under-the-hoodie-2018-research-report.pdf. [Last accessed on 22 Nov 2022]