Journal of Cyber Security Technology

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/tsec20

Securing IoT Devices Against Emerging Security Threats: Challenges and Mitigation Techniques

Mohammed Aziz Al Kabir, Wael Elmedany & Mhd Saeed Sharif

JOURNAL OF CYBER SECURITY TECHNOLOGY
2023, VOL. 7, NO. 4, 199–223
https://doi.org/10.1080/23742917.2023.2228053

REVIEW ARTICLE

OPEN ACCESS

Securing IoT Devices Against Emerging Security Threats: Challenges and Mitigation Techniques

Mohammed Aziz Al Kabir[a], Wael Elmedany[a] and Mhd Saeed Sharif[b]

[a]College of Information Technology, University of Bahrain, Sakhir, Kingdom of Bahrain; [b]School of Architecture, Computing and Engineering, UEL, London, UK

ABSTRACT

The increasing prevalence of IoT devices has brought about numerous security challenges due to their relatively simple internal architecture and low-powered hardware warranted by their small footprint requirement. As there are billions of IoT devices in use today, the sheer number of such devices pose a great security challenge as they are often constrained by a number of hardware and software limitations in addition to being designed with a focus on convenience, ease of use, mass production, and low cost, rather than security. The seemingly exponentially increasing number of such devices make it harder to keep track of – and patch – insecure IoT devices. This paper explores the common security threats, attacks, and vulnerabilities relating to IoT devices and highlights the challenges associated with securing them against emerging security threats and cyberattacks. Due to their role as gateways to connected devices and susceptibility to forming botnets or facilitating man-in-the-middle attacks, IoT devices are a lucrative target for cybercriminals. The paper discusses various remediation and mitigation techniques that can be implemented to better secure IoT devices, including access control mechanisms, secure communication protocols, and regular updates and patches. By better understanding the security challenges associated with IoT devices and implementing effective mitigation techniques, individuals and businesses can ensure the safety, security, and privacy of their connected devices and networks.

ARTICLE HISTORY
Received 6 March 2023
Accepted 13 June 2023

KEYWORDS
IoT security; Cybersecurity; Internet of things; Security threats; Mitigation techniques

RESEARCH HIGHLIGHTS

- IoT devices pose significant security challenges due to their simple and low-footprint nature, which makes them incompatible with advanced cryptographic techniques and existing security solutions.

CONTACT Mhd Saeed Sharif s.sharif@uel.ac.uk School of Architecture, Computing and Engineering, UEL, Dockland Campus, London E16 2RD, UK

- Cybercriminals often target IoT devices as they are essentially gateways to other connected devices and can be used to form botnets or facilitate man-in-the-middle attacks.
- The paper discusses common security threats, attacks, and vulnerabilities associated with IoT devices and highlights the challenges associated with securing them against emerging security threats and cyberattacks.
- Various remediation and mitigation techniques are covered, including access control mechanisms, secure communication protocols, and regular updates and patches.
- By implementing effective mitigation techniques, individuals and businesses can ensure the safety, security, and privacy of their connected devices and networks.

1. Introduction

IoT devices work by sending, receiving, and analysing raw data from the real world, and are then used, either in part or wholly, to perform pre-programmed or user-defined actions. It is estimated that by the year 2030, there will be 25.44 billion active IoT devices worldwide, or in other words, three IoT devices for every single person on this planet [1]. An overwhelmingly large number like this certainly adds a great deal of credibility to their sheer pervasiveness, and it is safe to assume that the number of IoT devices will only continue to grow every year as we continue to find more practical applications for their use in numerous different fields such as, but certainly not limited to, healthcare, wearables, home entertainment, security (ironically), agriculture, shipping and tracking, transportation, city infrastructures, power generation, and retail as well as manufacturing industries [2–4].

As the underlying technologies in IoT devices become more sophisticated day by day, so do their use in our daily lives. This makes them a lucrative target for cybercriminals because most IoT devices are inherently insecure due to their small or limited sizes that are only capable of housing low-powered embedded microcontrollers, simple sensors, actuators, power supply units, and other tiny electronic components such as memory and storage. This size constraint coupled with low-power consumption requirements, simple or basic operating systems, and limited computation power often represses the adoption of advanced or even modern cryptographic techniques, let alone entire security solutions. Moreover, much to the dismay of cybersecurity specialists and researchers, the vast majority of IoT devices are still only designed with minimal built-in security, and this is especially true for cheap, off-brand devices that are mass-manufactured for both consumer and commercial markets to perform a set of basic functions [5].

Due to inadequate security measures, IoT devices are typically more vulnerable to a range of security threats such as using default passwords that can be easily compromised by attackers – which in turn will then allow them to use the compromised device to launch attacks on other connected devices or networks, being stuck with outdated firmware that may be susceptible to known vulnerabilities, lacking secure boot mechanisms – which would allow attackers to modify the device's firmware and gain persistent access, and lacking encryption.

Moreover, it has been widely reported that a staggering 98% of all the traffic consisting of user data, commands, and sensor readouts are transmitted in open channels over the Internet without even being encrypted, which easily makes them vulnerable to even the most basic forms of man-in-the-middle attacks that will allow attackers to intercept and read or even modify sensitive plaintext data without the knowledge of the sender or the recipient [1,6,7]. Moreover, it has also been reported that up to 57% of all connected IoT devices today are still in fact vulnerable to most moderate-to-high severity attacks that is covered in the upcoming sections of this paper [8]. It also does not help that most users tend to leave their devices unprotected by using default user credentials and factory-configured or 'out-of-the-box' settings [9].

It is therefore crucial for manufacturers to prioritise the inclusion of proper security measures in their IoT devices such as using strong but lightweight encryption, providing over-the-air firmware updates and security patches, and implementing strong authentication mechanisms where applicable. However, it is often the case that manufacturers simply prioritise cost savings over security, and this is particularly true for inexpensive and under-powered IoT devices. Moreover, most IoT devices also have limited processing power, which makes it difficult to implement robust security features.

This paper attempts to list a fair amount of common and notable threats, attacks, and vulnerabilities that concern IoT devices and their widespread use. The paper also analyses the key challenges posed by the very nature of IoT devices in addition to hinting towards promising remediation techniques to make those devices more secure. The paper should therefore serve as a good point for any individual at a beginner or intermediate level who is interested in pursuing the trending and ever-important field of securing pervasive computing devices such as IoT devices as it is absolutely vital to ensure that any technology stack is built with sufficient security by design from the very start.

2. Methodology & structure

This paper assumes that the reader is already well-versed in IoT devices and their underlying technologies. The qualitative and quantitative findings of this endeavour consisted of performing a systematic literature review of 70 recently published works in the field of cybersecurity, particularly IoT devices. The remainder of this paper are as follows: Section III provides readers with background information on the layers of a typical IoT device, Section IV enumerates and highlights some of the most common threats, vulnerabilities, and attacks to each layer in detail, Section V is split into two subsections – one that discusses the various challenges of addressing the listed threats given the nature of IoT devices and another that draws attention to what is currently being done in addition to introducing what security solutions and countermeasures may be adopted, and finally, Section VI concludes the paper with a digest of the topics discussed in previous sections, and hints at future works that may be performed to enhance the safety, security, and privacy of IoT devices, their users, the network they are a part of, and the sensitive data they collect as well process as a part of providing a specific service to users.

3. Background

Despite the fact that IoT devices are utilised for providing diverse solutions that cater to vastly different markets, user-bases, and industries, the underlying architecture of a typical IoT device can still be broadly classified into three or four layers [3,10–14]. It may be mentioned that the four-layer architecture is basically the end result of inserting an additional data processing layer between application and network layers of the three-layer architecture of a typical IoT device as depicted in Figure 1.

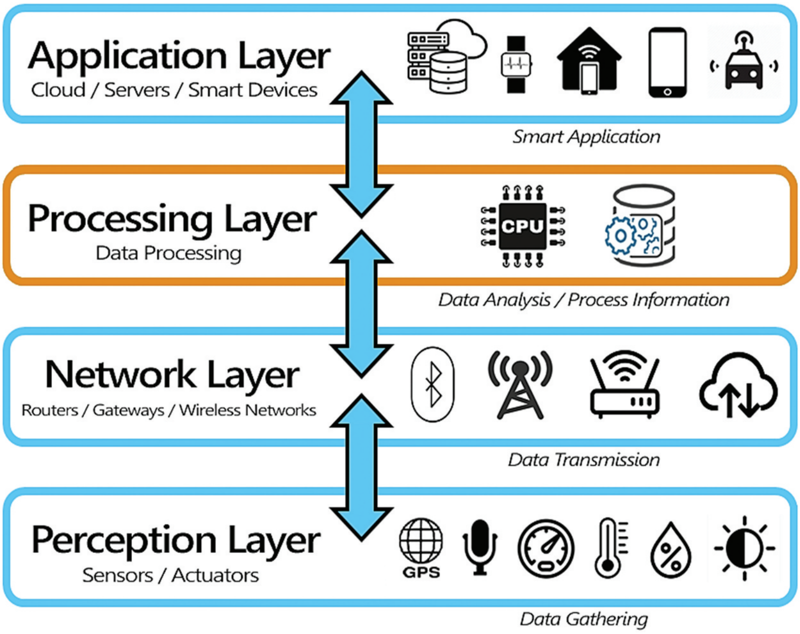

Figure 1. An overview of the four most common layers of the architecture of a typical IoT device.

However, depending on the level of granularity or segregation required, the architecture of an IoT device can be comprised of five or even up to seven layers as the actual number of layers does not strictly have to conform to a universal standard due to the numerous types of IoT devices and their seemingly innumerable use cases with varying requirements, specifications, and designs [2,15]. The overall complexity and number of architectural layers therefore depends on one use case to another.

There are numerous other business or service-specific additional layers which either augment or entirely redefine the standard three or four-layer architectural layers and covering them is beyond the scope of this paper at the time of writing. As a result, this section of the paper briefly goes through the perception, network, processing, and application layers in order to give an idea as to why protecting each of those layers is absolutely essential.

3.1 Perception layer

Also known as the physical or 'sensing' layer, it is the first and the lowest layer that is comprised of sensors, actuators, and several other tiny electrical components that gather and then relay readings, changes in physical or environmental parameters, and other data from the real world over to the digital world by passing the accumulated data over to the next layer via machine-to-machine (M2M) communication, so that certain predefined actions can then be performed based on the received data [13].

The sensors periodically detect, monitor, and record physical properties such as temperature, humidity, water level, light intensity, GPS, RFID tag readings, captured audio and video, electric and magnetic measurements, acceleration and deceleration, altitude, and atmospheric pressure, which are then converted from analogue to digital data prior to their transmission to subsequent layers where a more in-depth analysis of the accumulated data may be performed for autonomously or programmatically deciding the best course of action to take or leave it up to the end user [1,10,13].

3.2 Network layer

Also known as the data transport or transmission layer, this layer acts as a bridge between the perception layer and data processing or application layer for four and three-layer architectures, respectively. It allows communication to and from the perception layer over a communication channel in a wired or wireless setting with the help of protocols such as IPv6 Low Power Personal Area Network (6LoWPAN), Routing Protocol for Low-Power and Lossy Networks (RPL), Stream Control Transmission Protocol (SCTP), TLS, and IPsec in addition to the utilisation of regular TCP (Transmission Control Protocol) and UDP (User Datagram Protocol) transport layer protocols over IPv6 and IPv4 [16,17].

The actual data or command is relayed using networking technologies like Wi-Fi, Bluetooth Low Energy (LE), NFC, Low Powered-WAN (LP-WAN), WiMAX, ZigBee, Z-Wave, Ethernet, infrared, and cellular networks such as 5G, LTE (Machine), and 4G [17].

3.3 Processing layer

The data or event processing layer can be thought of as a `middleware' that sits between the network and application layers in the four-layer architecture of a typical IoT device with the role of aggregating and processing the raw, unfettered, and unstructured data that is received and accumulated from sensing devices in the perception layer to allow data abstraction, validation and in-depth analysis to take place prior to the transmission of relevant data to the application layer [13,18,19]. This layer also plays a key role in certain IoT applications with a varying number of additional layers such as fog or edge computing layers and operational or business logic layers, in addition to aiding services like data storage and cloud computing operations [12,20,21].

3.4 Application layer

In addition to being the top-most layer of the architecture of a typical IoT device, it is also a business or service-oriented layer that an end-user is most likely to directly interact with as it is what allows interfacing between them and the rest of the layers with the aid of messaging protocols such as Data Distribution Service (DDS), Message Queue Telemetry Transport (MQTT), Advanced Message Queuing Protocol (AMQP), Extensible Messaging and Presence Protocol (XMPP), Constrained Application Protocol (CoAP), and other more standard web protocols like WebSocket, SOAP, and REST – all of which rely on TCP and UDP protocols for communication [16,17,22,23].

The application layer allows smart applications to work as intended by relaying messages and commands to and from an IoT device with the help of underlying layers. It receives heterogeneous data from adjacent layers and then uses them for context-aware decision-making to allow for ubiquitous computing applications to be feasible in various different fields of the industry that are actively employing IoT devices [2,11].

Securing each layer against potential threats, exploitation, and threat actors is therefore imperative to ensure confidentiality, integrity, availability, authenticity, and non-repudiation of the services an IoT device provides, but also the constant stream of data it receives, relays, processes, or stores [14].

4. Common threats, vulnerabilities, and attacks

There is a humorous saying that the 'S' in IoT stands for 'Security'. The underlying truth here is that much like the modern-day Internet and its predecessor, the ARPANET, most commercial IoT devices are also typically not built with security in mind, and this is especially true for cheap, consumer-oriented, mass-produced IoT devices where various cost-cutting measures are taken to keep the prices low and profits high, which leads to devices being shipped with default passwords and other old or known vulnerabilities that can be exploited by attackers with relative ease [24]. Most IoT devices, particularly the ones that are connected to the Internet, are inherently more vulnerable because they are often designed with limited security features and do not receive regular security updates. These devices are often designed to be inexpensive and have limited processing power, which makes it more difficult to include robust or modern security features. Additionally, as mentioned previously, many device manufacturers prioritise cost savings over security due to a lack of awareness, reluctance to invest in security, resource constraints, lack of sufficient funding, meeting set quotas and worrying about the added time and complexity of implementing 'extra' security features, and having a higher risk tolerance (willingness to accept a certain level of risk in order to save money or to better utilise it someplace else).

In fact, much of the underlying protocols such as IP, ICMP, TCP, UDP, HTTP, SMTP, and FTP that are utilised by just about every IoT device for communicating over the Internet do not have any security or encryption built in, so data such as sensor readings, messages, and commands are transmitted in plaintext [25,26]. These insecure yet standard protocols are spread out across multiple architectural layers of any given IoT device, and they are still used today because of their relatively simplistic nature that causes less processing overhead and guarantees interoperability.

The sheer pervasiveness of IoT devices therefore increase the attack surface for bad actors to find and exploit vulnerabilities that either targets the device itself or uses it as a vector to launch an even larger attack on other vulnerable devices that are in the same network or compromise the entire network itself. This section of the paper attempts to list some of the more common threats, vulnerabilities, and attacks on each layer of a typical IoT device.

4.1 Perception layer

4.1.1 Eavesdropping

This is a form of Man-in-the-Middle (MitM) attack that, as the name implies, allows a third person (adversary) to tap into a communication stream between an IoT device and its authorised user, application, server, or another device for intercepting message or data transmissions innocuously for the theft of potentially sensitive information or to perform reconnaissance (traffic analysis) with the goal of identifying other assets on the same network and plan large-scale attacks on vulnerable devices or even cripple the entire network [1,26,27].

4.1.2 Jamming

This is a type of Denial of Service (DoS) attack that overwhelms wireless sensor networks to effectively render them incapable of transceiving data and signals by transmitting radio frequency signals that interfere with the communication channel in use, thereby leading to unavailability of certain resources or services for a period of time [3,13,26].

4.1.3 Node capturing

This involves compromising an IoT device and taking full control to capture and possibly disclose potentially sensitive information that is being sent or received by the device, therefore compromising its confidentiality [3]. Integrity of a piece of information may be brought into question as well if it is inconspicuously altered before reaching its intended recipient. An attacker could also disable various security parameters or replace the device's firmware to better serve their malicious agenda [1].

4.1.4 Resource exhaustion or depletion

Also known as sleep deprivation attack, this involves tricking an IoT device into depleting its resources and having it go offline. The aforementioned jamming as well as other forms of DoS attacks can lead to not just wasting network bandwidth but also shut down battery-powered or energy-constrained IoT devices due to a much greater energy consumption brought on by repeated retransmission attempts and signal collisions that prevent the device from entering into sleep or low-power state [28].

This increased processing overhead for prolonged periods of time can also potentially cause permanent damage to the hardware and cause premature device failure due to the excess heat generated during an attack.

4.1.5 Side-channel attack

Side-channel information such as a device's power consumption, processing time, acoustic and electromagnetic output, cache, and fault analysis can be used to perform cryptanalysis, differential attacks, and other reverse-engineering techniques to determine what operations are being executed, if any errors are being encountered, and other potentially vital information in relation to encryption, decryption, and key generation procedures [1,29].

4.1.6 Feeding false data

This form of low-level node tampering attack entails the insertion of false, deliberately manipulated sensor or telemetry data by either establishing a wireless connection to the target device via Bluetooth, Wi-Fi or some other wireless networking medium, or by simply attaining physical access to the device [1].

4.2 Network layer

Besides also being susceptible to signal jamming, collision and desynchronization attacks, eavesdropping and various other MitM attacks that typically involve Address Resolution Protocol (ARP) spoofing or poisoning, this layer is also vulnerable to other notable security attacks such as:

4.2.1 Forming botnets

It is widely believed that millions of IoT devices with default passwords and open Telnet, Internet Relay Chat (IRC), and peer-to-peer (P2P) ports fall victim to becoming part of large botnets in recent years for carrying out large-scale Distributed Denial of Service (DDoS) attacks when self-propagating worms like BASHLITE, Mirai, Hajime, Remaiten, BrickerBot, and Persirai manage to infect a single vulnerable IoT device in a network and then attempt to find and infect other vulnerable devices in the network with the goal of adding as many nodes to a botnet as possible [30–32].

In addition to rendering infected devices inoperable during botnet attacks, it also leads to higher bandwidth consumption and potentially blacklisting otherwise innocuous residential IP addresses by linking them to DDoS attacks.

4.2.2 Sybil attack

This occurs when a malicious node can successfully fake its own identity by stealing or spoofing the identity of a genuine node in a wireless sensor network and impersonating that node for sending false information to the receiving end of an IoT application or to receive something that is meant for the node being impersonated [33].

In an ad-hoc network, a single malicious node can have several forged identities at different instances or even concurrently by transmitting a false address or location of each faux (virtual) Sybil node within a network during protocol handshakes as illustrated in Figure 2 [33,34].

Sybil attacks hurt the reputation, integrity, and privacy of IoT applications by invading a network whilst impersonating as a node and generating false reports, spamming legitimate nodes in the network, bogus packet insertion, disseminating malware, and launching phishing attacks to trick users into sharing their credentials [1,3].

4.2.3 Fragmentation replay attack

Numerous resource constrained IoT devices utilise 6LoWPAN for connecting to the Internet via IPv6 and with adherence to the IEEE 802.15.4 standard, and this makes them vulnerable to various fragmentation replay attacks [35].

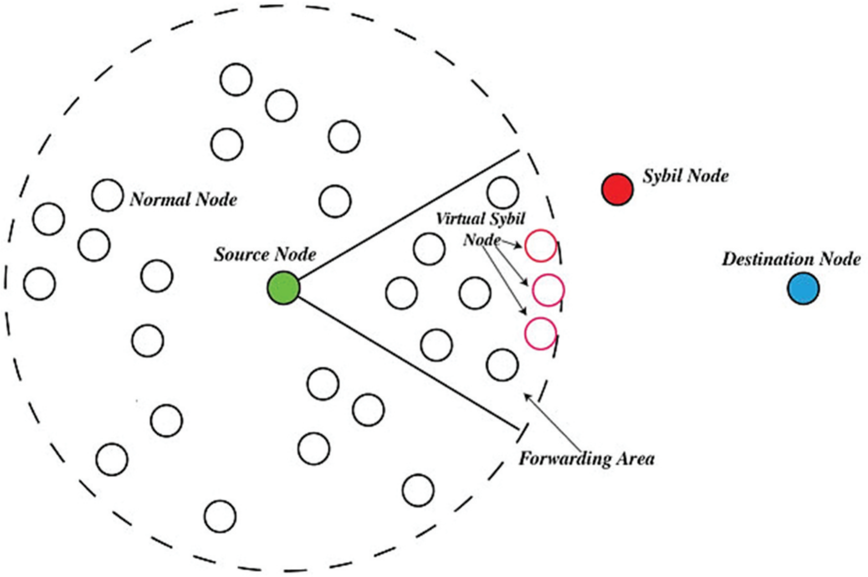

Figure 2. A Sybil node meticulously reporting its fake identities and locations in a wireless sensor network.

This is because each IPv6 packet must be split or 'fragmented' during transmission from one sub-layer to the next (within the network layer) and then reassembled since the size of a single packet exceeds the maximum frame size defined by IEEE 802.15.4 [35,36]. However, for the sake of simplicity and keeping processing overhead to a minimum, the reassembly process does not involve any verification of a fragment's origin or run any integrity checks, and this allows attackers to inject spoofed, forged, or duplicate (replay) captured fragments to overload a wireless sensor network by filling or overflowing the buffer space [36]. A 'ping of death' (D)DoS attack occurs when a targeted node is unable to handle the reassembled oversized packet, thereby causing it to become unresponsive (freeze) or crash.

Using IPsec is also not a viable option for low-powered IoT devices due to the additional resources (such as bits, memory, processing, and energy) required to perform security functionalities like encrypting the payload as well as ensuring its integrity and verifying the sender's origin [37].

Furthermore, the replaying of captured, 'incomplete', or forged packets lead to what is known as a buffer reservation attack, which essentially involves filling up the limited buffer space of an IoT device to subsequently deny legitimate packets from passing through [3,20,38].

4.2.4 Flooding

It is almost inevitable for any networking layer to become a target to various forms of (D)DoS attacks that involve the flooding of messages, packets, requests, or commands to a target network or a node in the network, and IoT devices are certainly not an exception. In fact, there is no shortage of flooding attacks, as enumerated:

- **TCP SYN floods** occur when attackers direct an abnormally high volume of initial connection requests in the form of SYN packets, typically with spoofed source IP addresses, to a target node in order to render the device unable to respond to legitimate connection requests in a timely manner or even at all as it struggles to respond to each malicious connection request by opening a port and sending a corresponding SYN/ACK packet to acknowledge the communication request and then waits for the final ACK packet which never arrives, therefore leaving the connection attempt in a 'half-open' state and causing an exhaustion of all available ports [39,40].
- **TCP ACK floods** are a variation of the aforementioned SYN floods in the sense that it too abuses three-way TCP handshake sessions to severely limit a node's ability to respond to legitimate requests, but it involves sending a large number of forged ACK packets to the target node with the aim of exhausting its resources by overwhelming it with far too many packets to process individually, leading to significantly higher CPU usage and memory consumption [41].
- The way **UDP floods** work is also quite similar to the previously mentioned flooding attacks in terms of how it is executed (using UDP instead of TCP) and its ramifications on a target node as it involves overwhelming it by sending a large number of UDP packets with spoofed source IP addresses to one of its randomly selected ports and having the device process each packet and responding with an ICMP (ping) packet to every request it received, thus severely affecting (limiting) its ability to serve legitimate requests [42,43].
- **ICMP (ping) floods**, on the other hand, involve flooding the target node with ICMP echo requests (pings) and then having it send back ICMP echo replies to each request, thereby slowing down the network traffic and the targeted node [26,39].
- **Smurf attacks** involve exploiting a flaw in the ICMP protocol wherein an attacker attempts to flood a victim machine with an overwhelming amount of ICMP echo replies by sending spoofed ICMP echo requests with the victim machine's public IP address to a relatively large pools of innocuous networked machines or to broadcast IP addresses, causing them to flood the victim machine with their responses and hampering its ability to respond to legitimate requests and responses [26].
- And finally, **hello flooding** targets the routing protocols used by most IoT devices for reporting their existence and availability to their neighbouring nodes [44]. It involves a malicious or Sybil node to falsely identify itself as a legitimate local node in the network and tricking other connected nodes into thinking that it is (the closest) in their communication range by

periodically sending ‘hello’ packets using high-power transmission signals to each of them, thereby establishing itself as a neighbouring node that is open to communication [27].

‘Hello’ packets typically play a role in determining how close one node is from another node in the same network, and nodes use this information to determine the shortest route for communicating with the base station as well as other nodes. [45]

If the malicious node is masquerading as the parent node or a base station, then every other connected node will attempt to communicate with it via multi-hop routing, especially in a large network with several connected devices spanning across multiple rooms, which can cause noticeable delays in responses and can even lead to extra bandwidth and power consumption by the nodes as they continue assuming that their messages are being communicated using the ‘shortest path’ despite being physically out of radio range in reality. [1,45,46]

4.2.5 Selective forwarding attack

Also known as SFA, this MitM routing attack consists of an attacker haphazardly dropping intercepted network packets while selectively allowing others to be forwarded to their respective destinations by exploiting vulnerabilities in lightweight network routing protocols such as RPL (short for Routing Protocol for Low Power and Lossy Networks), which do not perform any redundancy checks, in addition to taking advantage of the lossy nature of the communication medium [27,45,47]. This compromises the confidentiality, integrity, and availability of the data or information being transmitted.

4.2.6 Wormhole attack

This is a form of a replay attack that also happens to be an SFA of sort as it consists of intercepting network traffic (packets) in one node and replaying, redirecting, or ‘tunnelling’ them to another node in the network, causing a state of confusion and congestion in the network [1].

4.2.7 Black-hole attack

As the name would imply, this is a high-impact variation of SFA wherein a single or multiple Sybil (malicious) nodes trick routing protocols into thinking that they offer the shortest path to the destination nodes, and instead of forwarding the received packets to neighbouring or the destination node, they are all dropped [1,48].

4.2.8 Sinkhole attack

This kind of an attack shares a similarity with black-hole attacks in the sense that it too results in all network packets being dropped instead of reaching their intended destinations, but it involves compromising a network's central node and then overriding or overwhelming it for rendering it unavailable [1]. A Sybil or malicious node can then take over its place and then be used for SFA, black-hole attacks, or even packet tampering.

4.3 Processing layer

Given the heterogeneous nature of this layer that interlinks with other architectural layers, it is mainly susceptible to threats, attacks, and vulnerabilities that have already been covered thus far, most notably, active and passive eavesdropping, (D)DoS attacks, fragmentation replay attacks, collision attacks, exhaustion (resource depletion), various side channel attacks, and even malware being injected or embedded into incoming data streams [12,21,49].

4.4 Application Layer

Being the topmost architectural layer that allows end users to communicate with connected devices and hardware in the perception layer as well as receive crucial information from them, the application layer makes itself a lucrative target for cybercriminals by having a broader attack surface as well as having a much higher risk of compromising the confidentiality, integrity, and availability of data and services by being the closest to users and making itself one of the most vulnerable or exposed to all kinds of attacks [50–52].

The majority of the various cyberattacks illustrated in Figure 3 often exploits vulnerabilities in the application layer or depends on the human element to first

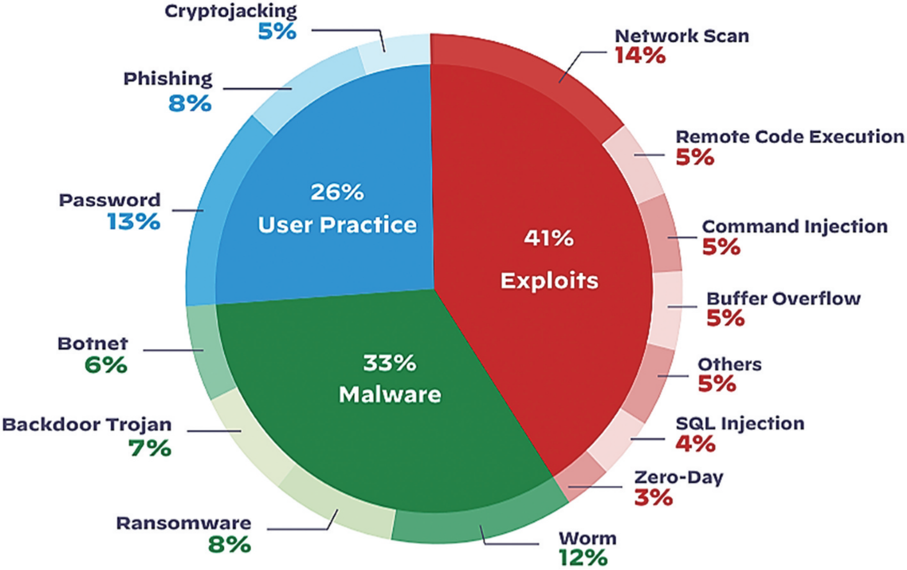

Figure 3. A pie-chart of the most common threats and attacks on IoT devices.

compromise an IoT device and then move on to compromising other connected devices in the same network [53,54].

Some of the more notable threats and attacks are as follows:

4.4.1 Malware infection

Due to inadequate security measures from both the manufacturers' side as well the users' side, IoT devices are more susceptible to becoming infected with malware, particularly worms that scan for open ports in a local network or utilise various other means of infecting other connected devices in the same network. Certain types of malwares exist that can cause permanent damage to devices by rendering them completely unusable (often referred to as 'bricking' the device). This is achieved by corrupting the device's firmware, exploiting a known or previously unknown vulnerability, or even causing physical damage, such as overheating critical components. There have also been instances of infected IoT devices being used to form botnets for launching DDoS attacks in addition to being used as Trojan horses to infect other connected devices with malware or to launch widespread ransomware and crypto-jacking attacks [15,45,55,56].

4.4.2 Unauthorised firmware modification

Given their nature, most IoT devices come with a relatively simple firmware with basic functionalities for which security often comes as an afterthought. This means they often offer little to no protection against device takeover or hijacking. An IoT device can be accessed and compromised by an attacker either physically or by establishing a remote connection, and then subsequently modify the device's system software or firmware to carry out unauthorised actions or to change its security configurations to make it even more vulnerable to attacks. The exploitation can be performed via binary patching, remote code execution, code substitution and code extension among other various techniques [45].

4.4.3 Security misconfiguration

The implementation of misconfigured security settings as well as untouched factory default settings in an IoT device often makes it an easy target for attackers to compromise the device and the data it might be storing or processing in addition to using the device to try and gain access to other connected nodes such as web servers, cloud instances, web applications, firewalls, and other end-point devices [57].

4.4.4 Insufficient logging

Given their low-powered hardware coupled with simple or basic firmware, most IoT devices lack the ability to periodically keep logs of all ongoing events and activities such as failed login or authentication attempts, making it easier for attackers to avoid being detected by a user or an intrusion detection system by concealing their malicious activities [58]. Moreover, the relatively small amount of data that is logged is often not encrypted at all. It can also be argued that this also applies to the perception layer.

4.4.5 Password-guessing & cracking

It is widely known that users often do not take the extra step of changing the default username and password of their IoT device, thus leaving it completely vulnerable to password-guessing and cracking attacks like credential stuffing, brute-forcing, dictionary-based guessing, masked or rule-based guessing, and even using the help of trained neural networks to generate thousands of highly plausible guesses [59].

4.4.6 Abusing web or cloud interfaces

Dubious activities like phishing and social engineering, malicious code injection, SQL and XSS injection attacks, cross-site forgery, session hijacking, and exploiting zero-day vulnerabilities all allow a potential attacker from accessing a target device via its web or cloud interface [10].

4.4.7 Abusing physical interfaces

If an attacker has physical access to an IoT device, then they can perform malicious acts like resetting the device to its default insecure state, extract the device's firmware and stored information, tamper with the sensing equipment or hardware, tamper with the data prior to its transmission to another layer, flash modified or unsigned firmware, tamper with the device boot sequence, and disable automatic over-the-air updates [11].

4.4.8 Abusing the MQTT protocol

Short for Message Queuing Telemetry Transport, MQTT is a simple, lightweight, efficient, and scalable messaging protocol that is designed for IoT devices with limited resources such as processing power and bandwidth. It facilitates communication and data transfer between IoT devices and servers through a publish-subscribe model. This model allows devices to publish messages to specific topics, and other devices can then subscribe to those topics to receive the published messages [60].

The simple and lightweight nature of MQTT brokers make them vulnerable to various threats listed hereinafter which revolve around exploiting the role of a broker in receiving and forwarding various messages between clients and other connected devices [61].

- Many of these vulnerabilities stem from improper message validation. In particular, maliciously crafted MQTT packets could easily render brokers unresponsive by causing a stack overflow attack. Similarly, an intentionally malformed request packet can also be used to cause a (D)DoS attack against the broker.

- MiTM attacks like eavesdropping or message spoofing can easily be performed on messages being exchanged due to the fact that they are being transmitted in plaintext.
- MQTT brokers cannot block repeated failed authentication attempts, so a determined attacker could either keep trying until they are able to get access to the MQTT device or overload the broker and eventually incapacitate with the sheer number of unending failed attempts.
- Attackers can gain control over data and functions of MQTT devices by taking advance of improperly set message publishing/subscribing permissions.
- Other security issues refer to compromising user authentication and data integrity by bypassing access control mechanisms altogether and accessing (subscribing to) all MQTT topics coming from all publishers, including sensitive data which can easily compromise the integrity and confidentiality of said data.

It has also been reported that MQTT brokers are vulnerable to SQL and XSS injection attacks, and can even be used to perform remote code executions as well as push malicious codes by misusing the firmware patch or update functionalities that utilise the MQTT communication protocol [42].

4.4.9 Retrieval of plaintext information

Due to lack of encryption, adversaries can easily capture and retrieve user credentials, encryption and decryption keys, security certificates, device information, and various other potentially sensitive user data and information pertaining to the device from the memory without having to worry about deciphering any of the captured information [5]. This issue is often aggravated by poor and insecure coding practices by the people responsible for coding the firmware for these devices. It may be argued that the distinct lack of data encryption can just as easily affect other architectural layers of IoT devices.

5. Classification of common attacks & remediation/mitigation techniques

This section attempts to analyse what has been established in previous sections and subsequently does not feature any new insights from other related works. Security should never be an afterthought. It must be a top priority that is factored in every stage of an application, system, or a device's development, including its design phase. As such, regardless of how many abstract or architectural layers an IoT device might have, proper security measures must be taken to ensure that each layer is secure from threats, vulnerabilities, and attacks

Table 1. Mapping of all the threats, vulnerabilities, and attacks mentioned in this paper to their respective architectural layers of a typical IoT device. In this context, a tick indicates the presence of a threat, vulnerability, or attack in the corresponding layer, while a cross marks its absence.

Layers:	Perception	Network	Processing	Application
Abusing MQTT Protocol	×	×	×	√
Abusing Physical Interface	×	×	×	√
Abusing Web/Cloud Interface	×	×	×	√
Black-hole	×	√	×	×
Collision	×	√	√	×
Desynchronisation	×	√	×	×
Eavesdropping/MiTM	√	√	√	×
Feeding False Data	√	×	×	×
Flooding/DoS/DDoS	×	√	√	×
Forming Botnets	×	√	×	×
Fragmentation Replay	×	√	√	×
Insufficient Logging	√	×	×	√
Jamming	√	√	×	×
Lack of Encryption	√	√	√	√
Malware	×	×	√	√
Node Capturing & Cloning	√	×	×	×
Password Guessing & Cracking	×	×	×	√
Resource Exhaustion	√	×	√	×
SQL & XSS Injection	×	×	×	√
Security Misconfiguration	×	×	×	√
Selective Forwarding	×	√	×	×
Side-Channel Attacks	√	×	√	×
Sinkhole	×	√	×	×
Sybil Nodes	×	√	×	×
Unauthorised Firmware Modification	×	×	×	√
Wormhole	×	√	×	×

as each layer presents a different attack surface that require different security measures to mitigate such threats and vulnerabilities summarised in Table 1.

Given the fact that the application and perception layers are the most exposed layers in the sense that they are tangible and can directly be interacted by people, proper thought must be given to securing both layers as they might be subjected to more advanced persistent threats (APTs), in addition to securing all the other layers in between, of course, to minimise exposed attack surfaces by as much as possible. Addressing security threats at every layer is the only way to ensure better confidentiality, integrity, availability, non-repudiation, and authenticity of data in a networked environment consisting of various IoT devices that communicate with one another as well as the concerned users.

5.1 Security challenges

Proper (sufficient) security controls must be implemented at the web, software application, and OS or firmware level to ensure the best possible protection of security and privacy at each layer of an IoT device, but given the fact that these devices are often characterised by their restricted memory capacity, low energy, and limited processing power, it is often quite challenging to properly implement sufficient security measures in IoT devices as their hardware and subsequent software limitations pose a huge barrier for software developers while implementing core functionalities of the device.

However, it may be argued that most of these limitations stem from their two most basic requirements – small footprint size (to occupy as little physical space as possible whilst consuming as little power as possible) and the necessity for a light or stripped down version of a full operating system with just the bare minimum essential functions and features to run without requiring additional cooling or even memory and processing overhead.

Due to their prevalent and amorphous nature, IoT devices suffer tremendously from 'platform fragmentation' as well as lack of interoperability and common technical standards which result in vastly different internal hardware that require different variations of a lightweight operating system, which makes achieving consistency, pushing over-the-air updates or patches, and developing applications or functions very difficult and time-consuming for vendors. This could be a reason as to why (older) IoT devices do not normally receive frequent software updates and have shorter vendor support periods. This presents huge security risks as the number of unsupported or devices with outdated (and therefore insecure) operating systems continue to rise.

Weak, poor, and insecure coding practices used in the firmware mass-produced cheap IoT devices have a high risk of not just compromising the device itself, but also various other devices that are connected to the same network. Most IoT devices still rely on simple username and password combinations as their only way of authenticating users, and subsequently relies on insecure password recovery techniques due to their inability to support two or multi-factor authentication schemes.

Furthermore, as IoT devices continue to see more adoption in various different applications in our day-to-day lives, their ubiquity also gives rise to privacy concerns among people, especially with the recent trend of incorporating data mining, big data infrastructures, and machine learning that brings a level of discomfort to security and privacy-conscious consumers.

5.2 Possible Remediation and Mitigative Solutions

This mere subsection of the paper was originally intended to be a full-fledged section spanning over several pages, but it was later decided to be hold off for an extended version of the paper instead. It therefore features a few related works to lay down the groundwork for future work.

- The need for real-time computing power and low-latency from a growing number of certain IoT applications such as smart cities, grids, and healthcare system continue to strengthen the demand for **edge computing platforms** to augment these low-powered devices with the additional resources required to comfortably meet those needs by reducing latency, response times, and bandwidth usage in addition to allowing pooling of computational power [50,62].
- Similarly, the integration of **cloud computing solutions** also plays a great role in increasing the efficiency of each connected IoT device by allowing them to pool and share their resources from the cloud server they are connected to for overcoming various limitations in processing, storage, and communication resources [4].
- Studies are currently being performed to assess the viability of developing secure IoT architecture using **software-defined networking (SDN)** and SDN controllers in addition to looking into the feasibility of using technologies like **Communications Platform as a Service (CPaaS)** and **Secure Access Service Edge (SASE)** to defend IoT devices against various threats [24,37].
- As **machine learning** and **deep learning** continue to become more sophisticated for their use in various applications, it comes as no surprise that numerous academic as well as practical efforts are now being poured into determining the feasibility of applying artificial intelligence into real-time, intelligent threat and various other anomaly detection by training neural networks and models using better datasets from live environments for developing next generation intrusion detection and prevention systems as well as firewalls and complete AI-based security suites to safeguard IoT devices and their networks from just about every threats, attacks, and vulnerabilities listed in this paper [6,9,13,15,21,39,63–65]. Machine learning techniques such as Dyna-Q, Q-Learning, Multivariate Correlation Analysis, Naive Bayes, Random Forest, Support Vector Machine, k-Nearest Neighbors, X-Mean, and many others have shown great potential in this regard [10].
- Various tools such as **Shodan**, **Dojo**, and **Nessus** may also be used to find vulnerable IoT devices as well as list their vulnerabilities, so that proper or timely preventative and mitigative actions can be taken before an actual attacker discovers and exploits them [11].
- There is a definite need for the creation as well as standardisation of **lightweight cryptographic protocols** such as PRESENT and CLEFIA, but for each and every architectural layer of an IoT device to allow for proper (sufficient) encryption techniques to exist for keeping information secure (indecipherable) should it fall into the wrong hands [5,17]. These protocols must be built with IoT devices in mind, particularly their hardware and

software limitations, for them to work as expected without impacting the device's performance or lifespan.

- Strong incentives exist for the widespread implementation of an improved version of the traditional three or four-layer architectural layer that incorporates **security as an all-encompassing layer** to establish a protective shield or bubble to protect every other layer. For instance, the six-layered architecture proposed by Burhan et al. [21] consists of perception, observer, processing, security, network, and application layers. The observer layer consists of data and user authentication whereas the aptly named security layer consists of data encryption, decryption, and hashing by using TLS among other cryptographic protocols for end-to-end data protection.
- Recently there have also been numerous instances of **Zero Trust Architectures (ZTAs)** being incorporated into IoT networks in mostly industrial and commercial settings, but consumer IoT devices and networks can also benefit from this technique of essentially adding a protective layer of abstraction to every connected node to keep them secure and isolated from outside threats [66,67].It is a security framework that assumes that no device or user can be trusted by default, and it works by essentially implementing a number of security controls such as micro-segmentation, principle of least privilege (PoLP) access control, and continuous monitoring of network traffic as well as device behaviour to make it difficult for potential attacks from taking place by identifying any suspicious activities in real time and taking mitigative actions. [68,69]A ZTA allows businesses and individuals a practical and affordable way to identify, classify, and secure all of their IoT devices in a given network by forming a fortified buffer or bubble that isolates a node from the rest of the public Internet by routing all its IP traffic through a secure partner (server), similar to how Tor's onion network works in the sense that an outsider would not be able to determine the device's network, physical location, or company association [68–70]. However, it is not a one-time implementation, and it must be continuously monitored and kept up-to-date to stay ahead of evolving cyberthreats.

- All IoT devices should come with **proper security configurations** and **secure booting** out-of-the-box for preventing unauthorised or malicious code from running when the device boots up, in addition to having embedded **Trusted Platform Modules (or TPMs)** with cryptographic keys for the authentication and protection of end-point devices.
- Other recommended practices include mandating secure and proper coding practices, performing periodic code reviews, hardening the application against common vulnerabilities and exploits, having basic firewall and DDoS protection, using HTTPS/TLS, using intrusion detection systems,

installing (critical) security patches and software updates as soon as they are made available, using better authentication controls, and having regular security awareness workshops and stringent company-defined measures and policies to deter the effects of common user errors.

A recurring theme all throughout this paper is the existence of various threats, vulnerabilities, and attacks against IoT devices that cannot be dealt with in an efficient and effective manner due to the resource-constrained nature and poor (default) security mechanisms that are typically associated with such devices, but progress is being made to make even the most low-powered embedded SoCs and microprocessors more capable than ever before by industry pioneers like Intel, Samsung, ARM, and Qualcomm. So, hope is on the horizon for finally having proper security on IoT devices.

Conclusion

IoT devices have been a hot topic in the fields of academia, business, and IT for a long time since their inception because of their limitless applications and seemingly ubiquitous nature that ultimately made the issue of securing them from threat actors and exploitation a matter of great concern for cybersecurity specialists. Their 24/7 availability coupled with easy remote-access has both positive and negative impacts, but it can be argued that the scale tips towards the 'negative' side as securing these devices are often hindered by what makes them suitable for so many applications, namely low-powered hardware paired with custom tailored lightweight firmware to take full advance of the low-end hardware for providing a particular service.

In conclusion, an analysis of the more prevalent cyberattacks on IoT devices was performed in this paper. The paper also looked at the common challenges that IoT devices typically pose when it comes to addressing those threats, vulnerabilities, and attacks. As for future work, there is scope for extending the paper by looking at promising remediation techniques, including but certainly not limited to, Zero Trust Architectures, security-augmented architectural layers, utilisation of well-trained machine and deep learning models, cloud and edge computing solutions, and by just having security as a forethought in general.

Disclosure statement

No potential conflict of interest was reported by the authors.

ORCID

Mhd Saeed Sharif http://orcid.org/0000-0002-4008-8049

References

[1] Krishna RR, Priyadarshini A, Jha AV, et al. State-of-the-art review on iot threats and attacks: taxonomy, challenges and solutions. Sustainability. 2021;13(16). doi: 10.3390/su13169463

[2] Gupta M, Jain S, Patel RB. 2021. Security issues in internet of things: principles, challenges, taxonomy. In: Singh PK, Singh Y, Kolekar MH, Kar AK, Chhabra JK, and Sen A, editors. Recent Innovations in Computingpp. 651–667. Singapore. doi: 10.1007/978-981-15-8297-4_52

[3] Mohindru V, Garg A. 2021. Security attacks in internet of things: a review. In: Singh PK, Singh Y, Kolekar MH, Kar AK, Chhabra JK, and Sen A, editors. Recent Innovations in Computing. p. 679–693. Singapore. doi: 10.1007/978-981-15-8297-4_54

[4] Singh S, Singh A, Goyal V. 2021. Cloud of things: a systematic review on issues and challenges in integration of cloud computing and internet of things. In: Singh PK, Singh Y, Kolekar MH, Kar AK, Chhabra JK Sen A, editors. Recent Innovations in Computing. p. 573–587. Singapore. doi: 10.1007/978-981-15-8297-4_46

[5] Anand P, Singh Y, Selwal A. 2021. Internet of things (iot): vulnerabilities and remediation strategies. In: Singh PK, Singh Y, Kolekar MH, Kar AK, Chhabra JK, and Sen A, editors. Recent Innovations in Computing. p. 265–273. Singapore. doi: 10.1007/978-981-15-8297-4_22

[6] Alani MM. Detection of Reconnaissance Attacks on IoT Devices Using Deep Neural Networks. Cham, Switzerland: Springer International Publishing; 2022. p. 9–27. doi: 10.1007/978-3-030-90708-2_2.

[7] Giess M. Cpaas and sase: the best defences against iot threats. Network Secur. 2021;2021(9):9–12. doi: 10.1016/S1353-4858(21)00103-3

[8] Ekoramaradhya M, Thorpe C. Novel DevSecOps model for robust security in an MQTT internet of things. Int Conf Cyber Warfare Sec. 2022;17(1):63–71. doi: 10.34190/iccws.17.1.31

[9] Ahanger TA, Aljumah A, Atiquzzaman M. State-of- the-art survey of artificial intelligent techniques for iot security. Comput Netw. 2022;206:108771. doi: 10.1016/j.comnet.2022.108771

[10] Anand P, Singh Y, Selwal A, et al. Iot vulnerability assessment for sustainable computing: threats, current solutions, and open challenges. IEEE Access. 2020;8:168825–168853. doi: 10.1109/ACCESS.2020.3022842

[11] Anand P, Singh Y, Selwal A, et al. Iovt: internet of vulnerable things? threat architecture, attack surfaces, and vulnerabilities in internet of things and its applications towards smart grids. Energies. 2020;13(18):4813. doi: 10.3390/en13184813

[12] Aydos M, Vural Y, Tekerek A. Assessing risks and threats with layered approach to Internet of Things security. Meas Con- Trol. 2019;52(5–6):338–353. doi: 10.1177/0020294019837991

[13] Malhotra P, Singh Y, Anand P, et al. Internet of things: evolution, concerns and security chal- lenges. Sensors. 2021;21:1809. doi: 10.3390/s21051809

[14] Srivastava A, Gupta S, Quamara M, et al. Future iot-enabled threats and vulnerabilities: state of the art, challenges and future prospects. Int J Commun Syst. 2020;33:e4443. doi: 10.1002/dac.4443

[15] Pal S, Jadidi Z. Analysis of security issues and counter- measures for the industrial internet of things. Appl Sci. 2021;11(20):9393. doi: 10.3390/app11209393

[16] Bayılmı¸s C, Ebleme MA, Ku¨¸cu¨k K, et al. A survey on communication protocols and performance evaluations for Internet of Things. Digital Communications And Networks. 2022;8(6):1094–1104. doi: 10.1016/j.dcan.2022.03.013

[17] Sowmya KV, Teju V, Pavan Kumar T (2021). An Extensive Survey on IOT Protocols and Applications. In *International Conference on Intelligent and Smart Computing in Data Analytics*, pages 131–138. Singapore: Springer.

[18] Jabraeil Jamali MA, Bahrami B, Heidari A, et al. 2019. IoT Architecture. Towards the Internet of Thingspp. 9–31. Cham, Switzerland:Springer. doi: 10.1007/978-3-030-18468-1_2.

[19] Kakkar L, Gupta D, Saxena S, et al. (2021). IoT Archi- tectures and Its Security: a Review. In *Proceedings of the Second Interna- tional Conference on Information Management and Machine Intelligence*, pages 87–94. Singapore: Springer.

[20] Bhale P, Prakash S, Biswas S, et al. 2019. BRAIN: buffer Reservation Attack PreventIoN Using Legitimacy Score in 6LoWPAN Network. Innovations for Community Servicespp. 208–223. Switzerland, Cham: Springer doi: 10.1007/978-3-030-37484-6_12.

[21] Burhan M, Rehman RA, Khan B, et al. IoT Elements, Layered Architectures and Security Issues: a Comprehensive Survey. Sensors. 2018;18(9). DOI:10.3390/s18092796

[22] Agyemang JO, Kponyo JJ, Gadze JD, et al. A Lightweight Messaging Protocol for Internet of Things Devices. Technol. 2022;10(1):21. doi: 10.3390/technologies10010021

[23] Jienan D, Xiangning C, Shuai C (2021). Overview of Application Layer Protocol of Internet of Things. In *2021 IEEE 6th International Conference on Computer and Communication Systems (ICCCS)*, Chengdu, China, p. 922–926. IEEE.

[24] Bhardwaj S, Harit S. 2022. Sdn-enabled secure iot architecture development: a review. In: Ranganathan G, Fernando X Shi F, editors. Inventive Communication and Computational Technologies. p. 599–619. Singapore. doi: 10.1007/978-981-16-5529-6_47

[25] Kayas G, Hossain M, Payton J, et al. (2020). An overview of upnp-based iot security: threats, vulnerabilities, and prospective solutions. In *2020 11th IEEE Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, Vancouver, British Columbia, Canada, p. 452–460.

[26] Sinha P, Jha VK, Rai AK, et al. (2017). Security vulnerabilities, attacks and countermeasures in wireless sensor networks at various layers of osi reference model: a survey. In *2017 International Conference on Signal Processing and Communication (ICSPC)*, Coimbatore, India, p. 288–293.

[27] Khanam S, Ahmedy IB, Idris MYI, et al. A Survey of Security Challenges, Attacks Taxonomy and Advanced Countermeasures in the Internet of Things. IEEE Access. 2020;8:219709–219743. doi: 10.1109/ACCESS.2020.3037359

[28] Kepceoglu B, Murzaeva A, Demirci S. 2019. Performing energy consuming attacks on iot devices. In *27th Telecommunications Forum (TELFOR)*, Belgrade, Serbia, pages 1607–1614.

[29] Devi M, Majumder A. 2020. Side-Channel Attack in Internet of Things: a Survey. Applications of Internet of Things. p. 213–222. Singapore: Springer. doi: 10.1007/978-981-15-6198-6_20.

[30] Bertino E, Islam N. Botnets and Internet of Things Secu- rity. Compu. 2017;50(2):76–79. doi: 10.1109/MC.2017.62

[31] Dange S, Chatterjee M. 2019. IoT Botnet: the Largest Threat to the IoT Network. Data Communication and Networkspp. 137–157. Singapore: Springer. doi: 10.1007/978-981-15-0132-6_10.

[32] Weam Saadi Hamza HMI. IoT Botnet Detection: challenges and Issues. Test Eng Man. 2020;83:15092–15097.

[33] Arshad A, Hanapi ZM, Subramaniam S, et al. A survey of Sybil attack countermeasures in IoT-based wireless sensor networks. Peer J Comput Sci. 2021;7:e673. doi: 10.7717/peerj-cs.673

[34] Zhang K, Liang X, Lu R, et al. Sybil Attacks and Their Defenses in the Internet of Things. Internet Of Things Journal, IEEE. 2014;1(5):372–383. doi: 10.1109/JIOT.2014.2344013

[35] Alyami S, Alharbi R, Azzedin F. Fragmentation Attacks and Countermeasures on 6LoWPAN Internet of Things Networks: survey and Simulation. Sensors. 2022;22(24). doi: 10.3390/s22249825

[36] Hossain M, Karim Y, Hasan R (2018). SecuPAN: a Secu- rity Scheme to Mitigate Fragmentation-Based Network Attacks in 6LoW- PAN. In *CODASPY '18: Proceedings of the Eighth ACM Conference on Data and Application Security and Privacy*, pages 307–318. Association for Computing Machinery, New York, NY, USA.

[37] Glissa G, Meddeb A. 6lowpsec: an end-to-end security protocol for 6LoWPAN. Ad Hoc Networks. 2019;82:100–112. doi: 10.1016/j.adhoc.2018.01.013

[38] Ray D, Bhale P, Biswas S, et al. (2020). ArsPAN: attacker Revelation Scheme using Discrete Event System in 6LoWPAN based Buffer Reservation Attack. In *2020 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*, New Delhi, India, p. 1–6. IEEE.

[39] Gupta BB, Chaudhary P, Chang X, et al. Smart defense against distributed Denial of service attack in IoT networks using supervised learning classifiers. Comput Electr Eng. 2022;98:107726. doi: 10.1016/j.compeleceng.2022.107726

[40] Abbas SG, Hashmat F, Shah GA, et al. Generic signature development for IoT Botnet families. Forensic Sci Int. 2021;38:301224. doi: 10.1016/j.fsidi.2021.301224

[41] Sudar KM, Deepalakshmi P, Singh A, et al. TFAD: tCP flooding attack detection in software-defined networking using proxy-based and machine learning-based mechanisms. Cluster Comput. 2022;26:1461–1477. doi: 10.1007/s10586-022-03666-4

[42] Birleanu S, Glavan D, Racuciu C, et al. At- tacks on IoT devices for power consumption. Scientific Bulletin Of Naval Academy. 2021;24(1):111–116. doi: 10.21279/1454-864X-21-I1-013

[43] Safar NZM, Abdullah N, Kamaludin H, et al. Characterising and detection of botnet in P2P network for UDP protocol. Indonesian J Electrical Eng Computer Sci. 2020;18 (3):1584–1595. doi: 10.11591/ijeecs.v18.i3.pp1584-1595

[44] Aditya Sai Srinivas T, Manivannan SS. Prevention of Hello Flood Attack in IoT using combination of Deep Learning with Improved Rider Optimization Algorithm. Comput Commun. 2020;163:162–175. doi: 10.1016/j.comcom.2020.03.031

[45] Makhdoom I, Abolhasan M, Lipman J, et al. Anatomy of Threats to the Internet of Things. IEEE Commun Surv Tutorials. 2018;21(2):1636–1675. doi: 10.1109/COMST.2018.2874978

[46] Dubey A, Meena D, Gaur S. A Survey in Hello Flood Attack in Wireless Sensor Networks. Int J Eng Res Technol. 2014;3(1):1882–1887.

[47] Airehrour D, Gutierrez JA, Ray SK. SecTrust-RPL: a secure trust-aware RPL routing protocol for Internet of Things. Future Gener Comput Syst. 2019;93:860–876. doi: 10.1016/j.future.2018.03.021

[48] Tseng F-H, Chou L-D, Chao H-C. A survey of black hole attacks in wireless mobile ad hoc networks. Hum Cent Comput Inf Sci. 2011;1(1):1–16. doi: 10.1186/2192-1962-1-4

[49] Najmi KY, AlZain MA, Masud M, et al. (2021). A survey on security threats and countermeasures in IoT to achieve users confidentiality and reliability. *Mater. Today: Proc.* Barcelona, Spain.

[50] Jurcut A, Niculcea T, Ranaweera P, et al. Security Considerations for Internet of Things: a Survey. SN Comput Sci. 2020;1(4):1–19. doi: 10.1007/s42979-020-00201-3

[51] Obaidat MA, Obeidat S, Holst J, et al. A Comprehensive and Systematic Survey on the Internet of Things: security and Privacy Challenges, Security Frameworks, Enabling Technologies, Threats, Vulnerabilities and Countermeasures. Computers. 2020;9(2):44. doi: 10.3390/computers9020044

[52] Ogonji MM, Okeyo G, Wafula JM. A survey on privacy and security of Internet of Things. Comput Sci Rev. 2020;38:100312. doi: 10.1016/j.cosrev.2020.100312

[53] Rodriguez E, Verstegen S, Noroozian A, et al. User compliance and remediation success after IoT malware notifications. J Cyber Secur. 2021;7(1). doi: 10.1093/cybsec/tyab015

[54] Unit 42 (2020). 2020 Unit 42 IoT Threat Report. *Unit 42*.

[55] Borys A, Kamruzzaman A, Thakur HN, et al. (2022). An Evaluation of IoT DDoS Cryptojacking Malware and Mirai Botnet. *2022 IEEE World AI IoT Congress (AIIoT)*, Online, p. 725–729.

[56] Varga P, Plosz S, Soos G, et al. (2017). Security threats and issues in automation IoT. *2017 IEEE 13th International Workshop on Factory Communication Systems (WFCS)*, Trondheim, Norway, p. 1–6.

[57] Loureiro S. Security misconfigurations and how to prevent them. Network Secur. 2021;2021(5):13–16. doi: 10.1016/S1353-4858(21)00053-2

[58] Neshenko N, Bou-Harb E, Crichigno J, et al. Demystifying iot security: an exhaustive survey on iot vulnera- bilities and a first empirical look on internet-scale iot exploitations. IEEE Commun Surv Tutorials. 2019;21(3):2702–2733. doi: 10.1109/COMST.2019.2910750

[59] Al Kabir MA, Elmedany W (2022). An Overview of the Present and Future of User Authentication. In *2022 4th IEEE Middle East and North Africa COMMunications Conference (MENACOMM)*, Amman, Jordan, p. 10–17. IEEE.

[60] Dinculeanˇa D, Cheng X. Vulnerabilities and limitations of mqtt protocol used between iot devices. Appl Sci. 2019;9(5):848. doi: 10.3390/app9050848

[61] Nebbione G, Calzarossa MC. Security of IoT Application Layer Protocols: challenges and Findings. Future Internet. 2020;12(3):55. doi: 10.3390/fi12030055

[62] Zhang WE, Sheng QZ, Mahmood A, et al. (2020). The 10 research topics in the internet of things. In *2020 IEEE 6th International Conference on Collaboration and Internet Computing (CIC)*, Atlanta, GA, USA, p. 34–43.

[63] Ahmad R, Alsmadi I. Machine learning approaches to IoT security: a systematic literature review. Internet Things. 2021;14:100365. doi: 10.1016/j.iot.2021.100365

[64] Latif S, Zou Z, Idrees Z, et al. A Novel Attack Detection Scheme for the Industrial Internet of Things Using a Lightweight Random Neural Network. IEEE Access. 2020;8:89337–89350. doi: 10.1109/ACCESS.2020.2994079

[65] Almrezeq N, Almadhoor L, Alrasheed T, et al. Design a secure IoT Architecture using Smart Wireless Networks. Int J Commun Net Inf Secur. 2020;12(3). doi: 10.17762/ijcnis.v12i3.4877

[66] Wylde A (2021). Zero trust: never trust, always verify. In *2021 International Conference on Cyber Situational Awareness, Data Analytics and Assessment (CyberSA)*, Dublin, Ireland, pages 1–4. IEEE.

[67] Dimitrakos T, Dilshener T, Kravtsov A, et al. (2021). Trust Aware Continuous Authorization for Zero Trust in Consumer Internet of Things. In *2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, Guangzhou, China, p. 1801–1812. IEEE.

[68] Shah SW, Syed NF, Shaghaghi A, et al. LCDA: lightweight Continuous Device-to-Device Au- thentication for a Zero Trust Architecture (ZTA). Computers & Security. 2021;108:102351. doi: 10.1016/j.cose.2021.102351

[69] Teerakanok S, Uehara T, Inomata A. Migrating to Zero Trust Architecture: reviews and Challenges. Secur Commun Net. 2021;2021. DOI:10.1155/2021/9947347

[70] Bertino E. Zero Trust Architecture: does It Help? IEEE Secur Privacy. 2021;19(5):95–96. doi: 10.1109/MSEC.2021.3091195