International Journal of Information Security (2025) 24:13
https://doi.org/10.1007/s10207-024-00930-z

SURVEY

IoT device security audit tools: a comprehensive analysis and a layered architecture approach for addressing expanded security requirements

Ashutosh Kumar[1] · L. Kavisankar[1] · S. Venkatesan[1] · Manish Kumar[1] · Suneel Yadav[2] · Sandeep Kumar Shukla[3] · Rahamatullah Khondoker[4]

Published online: 5 November 2024

Abstract

The Internet of Things (IoT) has the potential to bring unprecedented accessibility and efficiency to a wide range of critical applications and access control services. With the advent of IoT technology, there is a surge in potential threats and challenges that engender the risk of IoT devices interconnected over the Internet infrastructure. The mitigation of potential threats and risks needs a comprehensive analysis of security threats and relevant attack vectors in IoT networks, especially in IoT devices. Auditing is crucial to ensure that all IoT devices in the ecosystem are operating accurately and securely. This research has examined several physical and remote IoT security auditing tools to identify their drawbacks. This paper has also explored possible security threats, audited these threats to prevent them proactively by using the proposed novel seven-layer architecture, and presented expanded security requirements for IoT devices. Even more, we have examined the existing audit tools using an IoT device (IP camera). The analysis has shown that audit features concerning security requirements are missing from the existing audit tools. Our proposed seven-layer IoT device architecture with expanded security requirements has the potential to be a security audit benchmark for all IoT devices at the manufacturing and end-user levels.

Keywords Internet of Things (IoT) · IoT device · Auditing · Security requirements · Reconnaissance · Penetration testing

1 Introduction

IoT terminology can be broadly defined as the collective network of connected sensors, actuators, and other computing devices that enables communication among a variety of devices. These devices can sense their contextual surroundings and communicate data through a network with little or no human intervention at all. This network communication enables users to monitor and further optimize the IoT ecosystem for reduced latency, increased throughput, minimum power consumption, and enhanced security.

Currently, 62.12 billion IoT devices are connected to the worldwide network. In the year 2015, around 15.41 billion were connected, and it is expected that by the end of 2025, the number of IoT-based devices in action will increase by at least five times, estimated to be 75.44 billion [1, 2]. The IoT devices are highly susceptible to cyberattacks. This increases

✉ Ashutosh Kumar
rsi2023002@iiita.ac.in

L. Kavisankar
prf.lkavisankar@iiita.ac.in

S. Venkatesan
venkat@iiita.ac.in

Manish Kumar
manish@iiita.ac.in

Suneel Yadav
suneel@iiita.ac.in

Sandeep Kumar Shukla
sandeeps@cse.iitk.ac.in

Rahamatullah Khondoker
rahamatullah.khondoker@mnd.thm.de

[1] Department of Information Technology, Indian Institute of Information Technology, Allahabad, Prayagraj, Uttar Pradesh, India

[2] Department of Electronics and Communication Engineering, Indian Institute of Information Technology, Allahabad, Prayagraj, Uttar Pradesh, India

[3] Department of Computer Science and Engineering, Indian Institute of Technology Kanpur, Kanpur, Uttar Pradesh, India

[4] Department of Business Informatics, THM, Giessen, Germany

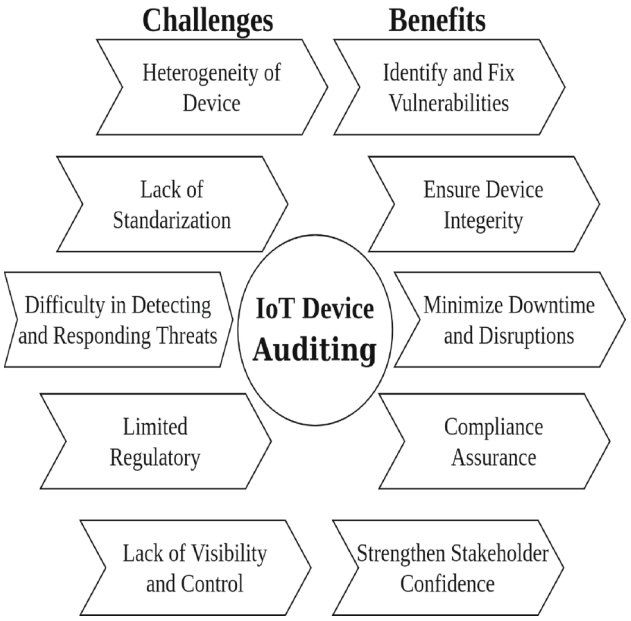

Fig. 1 Challenges and benefits of IoT device auditing

the need for auditing the security of IoT devices. Manufacturers often overlook security auditing because of supply or cost consistency during device manufacturing, thereby potentially jeopardizing the security measures of IoT devices. However, an explicit security audit of IoT devices could lower the risk of adversarial attacks by identifying weaknesses in connected devices. Although security audits mitigate risk, the following reasons pose a challenge for audits:

- No mandate: There is no mandate for the manufacturers or the administrators to perform the security audit. Also, administrators and end users are not aware of the impact of security exploitation.
- Service disruption: The organization that possesses a massive number of IoT devices needs more time for security audits and is likely to stop the service during auditing.
- Cost saving: The organizations focus on cost-saving and thus do not perform security audits with the assumption that every device and network is safe.

Figure 1 shows the challenges and corresponding benefits of performing IoT device auditing. In fact, since IoT is a network of a wide variety of devices, it challenges the diverse devices of the IoT ecosystem to communicate with each other, especially in a secure manner. The usage of security requirements and their standardization can help in the smooth integration of efforts and in the formulation of novel IoT usage scenarios.

Attacks that are automated or deliberately produced are quite likely to occur and evolve quickly, making them more challenging to prevent or stop. Traditional security audit tools are insufficient to fully exploit the vulnerability of different IoT devices, and in addition, organizations are unable to see or manage the IoT devices connected to their network. The precedence of performing the security audit on the IoT devices will not only help in identifying vulnerabilities and fixing gaps but will also ensure device integrity, minimize downtime and disruptions, assure compliance, and strengthen stakeholder confidence. Some of the most recent attacks have had major impact on IoT ecosystem due to lack of regular security audit. For instance, Mirai bot demonstrates the security loophole in IoT devices. It exploited the default passwords of the various IoT devices and performed DDoS attacks on the Dyn DNS servers [3]. Verkada [4], a cloud-based video surveillance provider, was hacked in 2021. The attackers were able to access the credentials of the clients, thereby gaining access to live feeds of various cameras across different institutions.

IoT is a multilayered architecture, and each layer may have one or more security issues if not implemented correctly. Since each layer of the IoT architecture has some security issues, this paper focuses on the security audit of IoT devices to resolve the security issues that may affect the operation of the entire IoT ecosystem. Exploiting the vulnerabilities at the first level, i.e., the IoT device level, which is the source of many security issues, can eventually help to strengthen the IoT ecosystem from a broader perspective.

1.1 Motivation and contributions

During the design and manufacturing of IoT devices, if manufacturers do not follow IoT security requirements or standards, then as a consequence, attackers can exploit the entire IoT ecosystem through the vulnerabilities in the IoT devices. The IoT device includes automation devices such as smart bulb, smart camera, smart meter, smart voice device, smart ECG device, and Programmable Logic Controller besides the network devices such as gateway, Layer 2 and Layer 3 switches. The vulnerabilities in the network devices cause more damage to the network. To assess whether the currently deployed devices in the IoT ecosystem are functioning securely or not, it is highly essential to perform the security audit. Even though different audit tools are available, the limited coverage of testing and complexity is an issue. This motivated us to analyze the existing IoT security audit tools, identify the gaps, and propose the necessary enhancements to improve the tools' capabilities. The primary contributions of this paper are:

- Firstly, we analyze different existing auditing tools with respect to the reconnaissance and the penetrating testing. The analysis presents the missing features in each tool.

Table 1 Existing physical access auditing tools for IoT devices

Author(s)	Audit approach	Audited layer	Auditing	Limitation
Syed et al. [5]	Physical	Session, network, firmware and hardware	The audit covers hardware, firmware, security, connectivity, and data privacy for IoT devices	This audit did not take into account the different IoT devices with their own protocols
Ibrahim et al. [6]	Remote, physical	Session, operating system, network, firmware and hardware	This work assesses IoT device security using an open-source methodology that addresses firmware, hardware, and communication issues	It lacks in identifying existing CVEs. It is open source with less coverage of identification in audited layers
Dong et al. [7]	Physical	Operating system and hardware	Audits system resources and also detects anomalous system behaviour	It focuses on Contiki OS and not other OS used by IoT device
Ursprung et al. [8]	Remote, physical	Session, network, firmware and hardware	The audit is performed by evaluating IoT device information using the UART port and presents the different attack scenarios of web applications and mobile applications	This audit relies heavily on open source frameworks and tools
Arshad et al. [9]	Remote, physical	Session and network	The authors have discussed the gradient approach for the security audit of IoT devices	This is a theoretical audit framework that lacks real-time implementation

- We propose a novel seven-layer architecture for IoT device security auditing. We classify the IoT device into layers and map its services and components.
  Each layer of the device has certain tasks, services, or components that should fulfill the desired security requirements.
- We present the expanded security requirements for the IoT device. Security requirements encompass a proactive and strategic approach to developing a strong defense against potential security threats.
  The manufacturer, the user, or both should be able to satisfy each security requirement.
- We test a wireless IP camera with the existing audit tools and present the audit results.
- We perform the gap analysis between the security requirements and the audit tools that exist and are in use. Additionally, we identify the features that may improve the existing audit tools and would be used by the scientific researchers to perform an advanced security audit.

Table 2 List of abbreviations

Acronyms	Description
IoT	Internet of things
DDoS	Distributed denial of service
DNS	Domain name server
SSH	Secure shell
BASH	Bourne again shell
TELNET	Teletype network
MQTT	Message queuing telemetry transport
CoAP	Constrained application protocol
OS	Operating system
RAM	Random access memory
SSD	Solid state drives
I2C	Inter-integrated circuit
SPI	Serial peripheral interface
UART	Universal asynchronous receiver-transmitter
MitM	Man-in-the-middle
JTAG	Joint test action group

The rest of the paper is organized as follows: Sect. 2 presents the related works culled from the available literature, Sect. 3 investigates the publicly available device auditing tools and their approaches, Sect. 4 proposes a novel seven-layer IoT device architecture for security auditing. In Sect. 5, the possible attack at various layers of IoT devices has been described. Section 6 discusses the analysis of IoT applications followed by Sect. 7, which presents the expanded security requirements of IoT devices. Section 8 discusses the audit results of different existing tools. Section 9 discusses the gaps and highlights the open research challenges in the existing audit tools. Section 10 concludes the paper. Further, Table 2 gives the list of abbreviations used in this paper.

Table 3 Existing remote access auditing tools for IoT devices

Author(s)	Audit approaches	Audited layer	Auditing	Limitation
Saeed et al. [10]	Remote	Application, session and hardware	ARM TrustZone-based hybrid security monitor this tool allows third-party or device owner audits. It is used to audit the sensors	Power consumption is higher than usual and not ideal for constrained IoT devices
Vasaka et al. [11]	Remote	Session, operating system and network	PENTOS IoT penetration tests are performed on IoT devices. It audits the Bluetooth and WiFi information of the target device	Has less reconnaissance coverage and does not work for constrained devices
Yiwen Xu et al. [12]	Remote	Firmware	An adaptive security architecture with real-time behaviour auditing, helps protect IoT devices from malware. It uses a behavioural model to audit suspicious activity and detect similar processes to identify malware	It only supports Linux-based IoT devices for malware auditing. This audit is performed on the available dataset inside the real-time IoT devices
Ashutosh et al. [13]	Remote	Application, session, network and operating system	This tool performs the security audit of IP-based devices and uses MAC and Port, and machine learning to predict device type and model	Focuses on IP-based devices only, and if no port is open, device type prediction fails

Table 4 Comparison of existing work based on audit tool features

Audit tool features	[6]	[5]	[7]	[8]	[14]	[15]	[16]	[17]	[18]	[19]	[20]	[21]	[13]	Ours
Audit approach- Remote[R]/ Physical[P]	R,P	P	P	R,P	R	R	R	R,P	R	R	R	R	R	R,P
Hardware auditing	✓	✓	x	✓	x	x	x	✓	x	x	x	x	x	✓
Software auditing	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Target medium	1,2,3	1	–	1,2	1,2	1,2	1	1,2,3	1,2,3	1	1	1	1	1
Usage of existing tools for auditing	✓	✓	x	✓	✓	x	✓	✓	✓	x	–	–	x	✓
Outside network[O]/ Part of network[PN]	PN	–	PN	PN	PN	PN	PN	PN	–	PN	O	O	PN	PN
Reconnaissance	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Penetration testing	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	x	x	✓	✓
Layer-wise auditing	x	x	x	x	x	x	x	x	x	x	x	x	x	✓
Security requirements	x	x	x	x	x	x	x	x	x	x	x	x	x	✓

1- WiFi, 2- Bluetooth, 3- Zigbee

2 Related work

There are various IoT security audit tools in existence and they can perform the physical access based and remote based auditing of devices. We present the existing physical access based audit tools in the following Table 1, with their approaches, layers at which they work, features and limitations. Ibrahim et al. [6] have used different techniques to evaluate the security of IoT devices, including software, hardware, and connectivity. However, there is the limitation of audit features and no provision to map with existing Common Vulnerabilities and Exposures (CVEs). Syed et al. [5] have discussed the modular approach to evaluate only the corporate network that contains a pool of auditing questions and also justified the need for auditing IoT devices. Dong et al. [7] have audited the devices that are embedded with only contiki-running IoT devices and kernel resource-sensitive events are collected. Ursprung et al. [8] have extracted the Operating System (OS) name and version of the IoT device (Philips Hue) using the UART port, performed the Nmap command to obtain details of the opened port, and scanned mobile and web apps to identify vulnerabilities in various scenarios. Arshad et al. [9] have discussed detailed conceptual steps to

carry out a security audit, compared the security standards for constrained devices, and discussed the framework.

Apart from the physical audit, there are remote audit tools as given in Table 3. Saeed et al. [10] have tracked sensor activity and used virtualization hardware and ARM to store activity logs for auditors. This strategy is very power-consuming for limited IoT devices. Vasaka et al. [11] have employed PENTOS IoT penetration testing that automatically gathers Bluetooth and Wi-Fi data from the target device. This system is limited in its capabilities for reconnaissance and attacks and is not suitable for constrained device protocol. Yiwen Xu et al. [12] have developed an adaptive security auditing for Linux-based IoT devices, even those with heterogeneous architectures, through real-time behaviour auditing. In addition to auditing the IoT network in real time, they have also audited the dataset. Ashutosh et al. [13] have presented a security audit testbed for IP-based devices covering multiple attack surfaces. The Fing [19] audit tool audits the device concerning application, session, and network layers. The open source intelligence tools such as Shodan [20] and Censys [21] audit at the application and session layer. The other security audit tools, such as Automated Security Audit Testbed (ASAT) [13] and Asheem et al. [15], have more coverage on reconnaissance and perform penetration testing. These can audit the application, session, network, and operating system layers, while Siboni et al. [17] have extended the audit process to the firmware layer.

Analyses are performed on the existing audit tools; however, these tools are not developed and analyzed considering the device layer architecture and the security requirements. Table 4 lists the audit features that have to be considered in the audit tool. The table shows that the existing tools do not fulfill all the audit features and thus need to follow our layered approach and security requirements to improve the auditing of the IoT devices.

3 Device security audit

Device auditing, in the realm of security is a methodical inspection and evaluation of individual devices within a network or system. The main goal for conducting security focused device auditing is to identify vulnerabilities, analysis of the overall security stance, and detection of any indications of unauthorized access or harmful activity. The IoT device audit methodology aims to conduct a comprehensive security assessment of the IoT device, encompassing the hardware, firmware, and software. The IoT device audit can be performed either by physically or remotely accessing the device. We analyze both the methods along with their advantages and disadvantages.

3.1 Physical access

Physical access to the device is necessary to guarantee the security auditing of the IoT device. The physical interaction and manipulation with the hardware is referred to as physical access to the device. Hence, providing physical access to an unauthorized person can tamper with the components of the devices and obtain confidential data, or can compromise its security.

IoT devices are typically placed in remote areas or at secure places, gaining physical access to the device would be difficult. If one can get physical access to the device, then they might exploit the devices by accessing it through a Universal Asynchronous Receiver / Transmitter (UART), Serial Peripheral Interface (SPI), or Joint Test Action Group (JTAG) interface. These interfaces are provided by the manufacturer for debugging purposes. An attacker accessing the device through this interface could get the sensitive information, i.e., user credentials, etc. Also, it will be possible to extract the firmware of the device that could expose sensitive information. The extracted information could be used to exploit other devices in case credentials remain the same. The following physical audit can be performed to identify the vulnerabilities, thus ensuring the security of the device:

3.1.1 Bootlog analysis

The bootlog provides important information regarding the boot-up sequence of the device. The boot-up sequence exposes potential security vulnerabilities such as hard-coded credentials and unpatched version usage. Inspecting the bootlog of an IoT device is an essential aspect of conducting a security assessment. The bootlog offers a comprehensive description of the device's initialization procedure, enabling security experts to thoroughly examine it for indications of weaknesses, unauthorized entry, or manipulation [22]. In case the bootlog includes any sensitive information, then it has to be fixed before supplying the product. The manufacturer should ensure the applicability of this requirement. Shivam et al. [23] have proposed a bootlog extraction and analysis tool using a low-cost external device.

3.1.2 Firmware analysis

Firmware security audits are essential to protect IoT devices from vulnerabilities and threats [24]. The Low-level firmware controls the hardware of devices and must be evaluated. Cryptographic hashing verifies the integrity of firmware during delivery to prevent tampering. The manufacturer should ensure that the firmware services cannot be accessed by third parties without knowing the credentials. The tools available for firmware analysis are listed below and Table 5 shows the comparison between the two tools.

Table 5 Comparison of firmware analysis tools

Features	Firmwalker [25]	Firmadyne [26]
Purpose	Firmware information extraction and analysis	Firmware emulation for vulnerability analysis
Language	Bash script	Python
Type of analysis	Static	Dynamic (Emulation)
Supported emulation	Not Applicable	Controlled environment
Automation	Basic automation for information extraction	Automated emulation for dynamic analysis
Dependency	Requires standard Linux utilities	QEMU and additional dependencies
Regular maintenance	Not maintained. Last commit was made in september of 2020	Maintained. The last commit was made in october of 2023

- Firmwalker: It is used to examine the firmware extracted from the IoT device. It is implemented as a bash script and inspects uncommon strings to do a static analysis of the firmware. Firmware images can include sensitive data such as hard-coded credentials and that could be extracted with this tool. Attackers performing this task can control the respective device as well as the similar devices. This tool searches the firmware for specific patterns that are known to exist and could point to the existence of this kind of data [25].
- Firmadyne: It is a dynamic analysis of Linux-based embedded firmware. The software-based whole-system emulation can automate large-scale dynamic analysis of embedded binaries extracted from the images [26].

3.1.3 Device physical interfaces

The Universal Serial Bus (USB) is the dominant standard for establishing wired connections between peripheral devices, such as USB devices, and a host to transmit the information. The majority of IoT devices on the market have a USB port that the manufacturers left open for charging or device access. These interfaces can be used to audit these devices by simply plugging an external audit device to the USB port, check for any vulnerabilities associated with the device, and also check the compatibility of whether one can update the firmware or not.

Tool available for physical interface analysis is USBlyzer, a software protocol analyzer that helps Windows users to see USB traffic from the host perspective [27]. Information with respect to USB devices and how they interact with the system can be recorded, analyzed, and displayed.

Auditing the devices based on physical access has several advantages as well as disadvantages. The advantages are given below:

- We can get more information about the device, such as the software installed, the services running on non-reserved ports and secured storage of data.
- The UART, SPI, and JTAG interfaces can be accessed only through physical access.
- There are devices that may not run any service to provide the information, if the audit device initializes the auditing remotely.
- The information that are collected is accurate for instance, the OS version, firmware version, etc.

The disadvantages or the limitations of physical access-based auditing are given below:

- It is hard to get physical access to the devices, as most of the time these IoT devices are deployed in remote or restricted areas.
- There are devices that do not have USB-based access to them, i.e., the open ports left by the manufacturers are kept for power supply only and not for data transfer.
- Tampering may damage the device.

Generally, physical access-based audit is best suited to the security audit, as it allows a thorough audit of the hardware, memory, and firmware. However, it requires significant resources and a skill set. In some cases, it is difficult to physically access the device as it may be installed in hazardous places that need proper safety measures and may also disrupt the normal operation of the devices.

3.2 Remote access

The ability to connect and control devices from a location other than physically accessing it is known as remote access to devices. The IT administrators, support teams, and other authorized users require this functionality to manage, diagnose, and configure devices remotely. To avoid unwanted

Table 6 Comparison of existing tools/testbed with respect to reconnaissance of IoT device

Test parameter	[14]	[15]	[16]	[17]	[18]	[19]	[20]	[21]	[28]	[13]
MAC address	✓	✓	✓	✓	x	✓	x	x	x	✓
Ports open	✓	✓	✓	✓	x	✓	✓	✓	✓	✓
Traffic analysis	✓	x	✓	✓	x	x	x	x	x	✓
OS fingerprint	x	✓	x	x	x	✓	x	✓	x	✓
SSL certificate	✓	✓	✓	x	x	x	✓	✓	✓	✓
Time synchronization	✓	x	x	✓	x	x	x	x	x	✓
Firewall	x	x	x	x	x	x	x	x	x	✓
Device fingerprinting	✓	x	x	✓	x	✓	x	x	x	✓
SSL certificate weakness	x	x	✓	x	x	✓	x	x	x	✓
Battery percentage	x	x	x	x	x	x	x	x	x	x
Local storage data	x	x	x	x	x	x	x	x	x	x
Hardware (Firmware)	x	x	x	x	x	x	x	x	x	x
Radio frequency identification	x	x	x	x	x	x	x	x	x	x

Table 7 Comparison of existing tools/testbed with respect to exploitation of IoT device

Test parameter	[14]	[15]	[16]	[17]	[18]	[19]	[13]
SSH attempts	✓	✓	✓	✓	x	x	✓
Password brute-force	✓	✓	✓	✓	x	x	✓
Weak passwords	x	✓	x	x	x	x	✓
Downgrading attack (SSL)	✓	x	x	✓	x	x	✓
Downgrading attack (SSH)	x	x	x	x	x	x	✓
TELNET	✓	x	x	x	x	x	✓
FTP	✓	x	x	x	x	x	✓
Deauthentication	x	x	x	x	✓	✓	✓
MitM	✓	x	x	✓	x	x	✓
MQTT password brute-force	x	✓	x	x	x	x	✓
Unauthenticated MQTT	x	✓	x	x	x	x	✓
CoAP	x	x	x	x	x	x	✓

access and possible security breaches, it is essential to guarantee the security of remote access. If one can enter an IoT network, then s/he might easily get the information (IP address) of all the connected devices. With the help of an IP address, one can perform reconnaissance of the IoT device and can exploit the vulnerabilities, if any, with the help of available or customized tools. A similar procedure can also be used to audit the IoT device. While auditing the IoT device, we can collect the information (reconnaissance) of the device and perform penetration testing, that is, exploit its vulnerabilities to verify whether the exploit is possible or not. There are different strategies and tools that exist to perform the reconnaissance and the exploit. In this section, we will present and analyze the tools with respect to reconnaissance and the penetration testing.

3.2.1 Reconnaissance

The reconnaissance is the process of actively probing the target device to get various information in order to understand its weakness. This is done by using various open source and proprietary tools. Table 6 shows the parameters to perform the reconnaissance auditing of the IoT device. Such tools are discussed here.

- Omnia et al. [14] have developed a testbed that automates the security auditing of IoT devices. To mitigate malicious activity on the target device, the testbed detects and reports vulnerabilities besides device communication details.
- Asheem et al. [15] have presented an EXPLIoT framework for assessing and leveraging the security of IoT products and infrastructure. It comes with a set of plugins (test cases) that are used to do the evaluation and can be readily expanded with new ones.

– Rohit et al. [16] have proposed a framework to improve smart home security by auditing the devices. The framework presents the Common Vulnerability Scoring System (CVSS) base score of each vulnerability for the user to prioritize the fix.
– Siboni et al. [17] have developed a security testbed that conducts a security testing of the IoT devices with varying software and hardware configurations. The testbed uses powerful machine-learning techniques to monitor the functionality of IoT devices. Several testing scenarios have demonstrated the operation of testbed on various IoT devices. The results of the testbed have revealed that it is capable of detecting vulnerabilities and hacked IoT devices.
– Romain et al. [18] have developed a framework called Mirage that performs security auditing of IoT devices. This framework is for IoT wireless communications analysis and provides a generic, modular, unified, and low-level audit environment which can be easily adaptable to new protocols.
– Fing [19] is a network monitoring and device-blocking tool that lists out the connected devices. It shows the open ports, MAC address, running operating system, certificate details, and the services that are running on the ports. It also shows the currently active devices and has the provision to disconnect them.
– Shodan [20] is an online tool that searches the internet for devices that are available to the general public. When a device is scanned using its IP address, the open ports, banners associated with them and the details of Transport Layer Security (TLS), such as the TLS version, certificate details, and expiry date are identified.
– Censys [21] is an online tool and users may use IP addresses to find devices and information such as open ports, services that are using those ports, and the details of TLS, such as its version, certificate details, and its expiry date.
– IVRE [28] is an open-source network reconfiguration framework. It collects the device information using well-known open-source tools such as Nmap, Masscan, ZG-rab2, ZDNS, and Zeek (Bro), saves it in a database for further analysis.
– Mitre Caldera [29] is a scalable and Automated Adversary Emulation Platform that assesses the security of devices. This is a remote-based assessment platform; however, the testing agent should be deployed on the target machine. The agent can be deployed by having the authentic user access to the machine or by exploiting the vulnerability such as a weak password or no password. This platform performs various assessments on the target machine, and we may also consider this assessment platform as the physical access-based platform since the target machine should run the platform's agent code.
– ASAT [13] analyzes IP-based IoT devices by mounting different attacks, perform reconnaissance, and generate test reports not only for a single device but for multiple devices, and all IP-based IoT devices connected to a router. Further, it stores the acquired data of the test module into a database for further analysis.

3.2.2 Exploitation

In remote access based auditing, we can use the available information and perform the attack on the device to identify whether it is vulnerable to the exploit or not. Table 7 shows exploit parameters and the existing models or tools that are available to perform the exploit-based auditing of the IoT device. The tools such as ASAT [13], Omnia et al. [14], Asheem et al. [15], Rohit et al. [16], Siboni et al. [17], and Fing [19] perform the reconnaissance as well as the exploitation called penetration testing.

Like physical access-based auditing, the remote-based auditing also has several advantages and disadvantages. The advantages are as follows:

- It does not need physical access. One can get access to the network and perform remote auditing of the connected devices.
- Any device connected to the network can be audited even if the device does not have USB-based access.

The disadvantages of the remote-based security audit are as follows:

- Attacker can collect the information and perform the exploit without the knowledge of the administrator or the users.
- The remote-based auditing could only carry out limited information collection and exploitation.
- The information collected may not be accurate, for instance, about the operating system and its version.

Remote auditing relies primarily on network connectivity to intercept packets and perform the audit. However, it can also be hindered by network connectivity issues, which can affect the accuracy and effectiveness of reconnaissance and penetration testing audit results. This audit process is also limited by device heterogeneity, where a proprietary protocol may not have adequate support for the generic audit tools. Figure 2 shows both remote and physical auditing processes of a device.

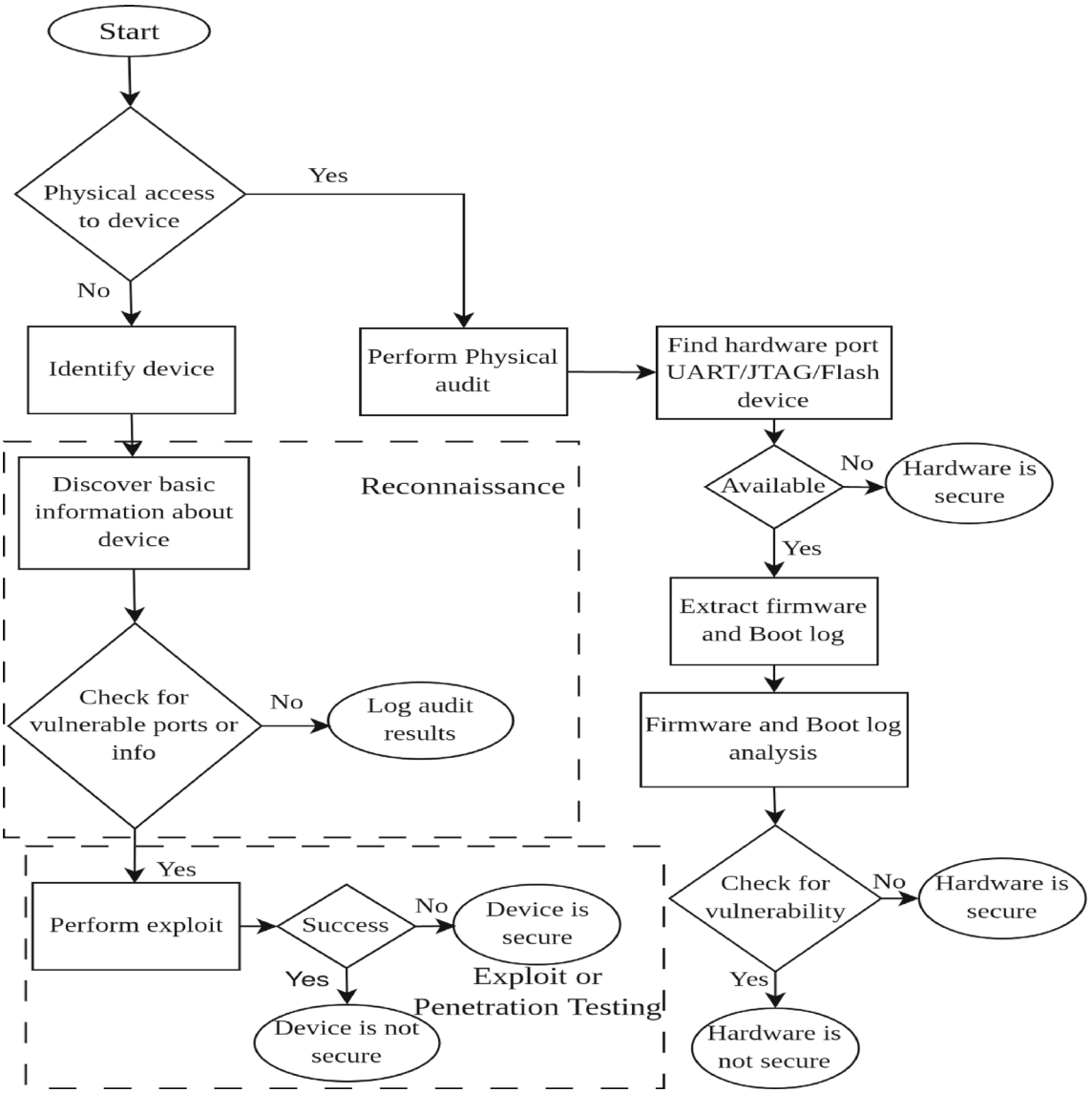

Fig. 2 Flow diagram of audit process

3.3 Ethical implications and requirement of device auditing

The Auditing of devices includes evaluating their security, privacy, and performance to meet the defined security standards. There are ethical implications in achieving this practice. Hence, the auditor should ensure that the following requirements are similar to the proposal by Ricardo et al. [30].

- Auditors might gain access to personal or sensitive data stored on these devices. Auditor should ensure that no data get misused.
- The auditor should report the vulnerability, if any, immediately to the respective individual or team and ensure that it cannot be exploited or leaked to malicious persons.
- Auditors should ensure that the auditing process does not have an adverse effect on the performance of audited devices.
- Auditors should always ensure that the organization abide by the compliance and regulations.

Every audit tool except the physical access based one needs time to perform the auditing without interrupting the devices' expected task. The time requirement is based on the reconnaissance and the penetration testing coverage as well as the test cases.

4 IoT device layers

Every IoT device has various physical components, software, firmware, etc., to provide the desired services. Each component or service implemented on the device may have vulnerabilities if implemented weakly. The attackers can exploit these vulnerabilities and harm the device and applications. We can categorize the device components and its services in to layers. MITRE [31] has classified embedded device into four layers and accordingly security audit can be performed. The four layers are application software, system software, hardware, and networking. We have classified the IoT device components and services into seven layers and mapped the services and components for effective security auditing. The seven layers of the IoT devices and their services, possible attacks, impacts, auditing methods, and audit priority are presented in Fig. 3. The proposed layering is fully or partially applicable to all IoT devices, irrespective of their manufacturer. The details of each layer are described below:

- Application layer: This layer provides a user interface to interact with the IoT devices. It includes various applications software platforms, i.e. the web service, the runtime environment, and programming editors. These would be implemented with role-specific functionality and services.
- Session layer: This layer manages the communication between the devices and responsible for the transmission of data in the IoT ecosystem. The Session layer includes different application protocols such as MQTT and CoAP and their implementations.
- Network layer: This layer of the device ensures the connection between the IoT devices or with the server directly or via different interconnecting devices. Due to the constrained nature of the IoT devices, the protocol at this layer must meet several specific requirements. These include concurrent communication, efficient data transfer, and communication security, as well as scalability in the case of network topology changes.
- Operating system layer: This layer includes the kernel and the device drivers for smooth and seamless interaction with the lower layer. The operating system must be designed in accordance with the constrained devices. The operating system provides the support for the network, session, and application layers.
- Firmware layer: It is software built on an embedded system and generally stored in non-volatile memory. To enable the functionality of the device, firmware is an entire operating system image that combines the file system and kernel with several binaries and libraries.
- Memory layer: IoT devices can have different storages such as Solid State Drive (SSD), Random Access Memory (RAM), Read Only Memory (ROM), etc. Its primary role is to store code for booting IoT devices, firmware, and embedded operating systems. It also stores device data, credentials, user information, sensor data, etc. This layer supports the above layers.
- Hardware layer: This is the base layer and includes processor, sensor, actuator, and transceiver that collects data from the environment and provides the necessary services to the upper layers of the device.

The discussed device layer will be helpful for the auditor in precisely detecting vulnerabilities or malicious activity, thus reducing security breaches. It also aids the organization in enhancing its proactive measures to tackle potential threats in advance. The components and servers of these layers are vulnerable to various attacks, and one can exploit them.

5 Possible threats to IoT devices layer

Kaspersky reported a total of 1.5 billion cyberattacks on IoT devices in the first half of 2021, upto 639 million from the previous year. Despite technological advances, attackers are able to adapt to security patches and constantly introduce new vulnerabilities [32]. According to a report published in 2020, 98% of IoT device traffic is unencrypted, hence anyone with access to the network can see it. This clearly indicates the security failure in the IoT devices. Around 57% of devices are vulnerable to medium or high severity attacks because they have insecure software or are insecurely deployed. As the report describes [33], this is an easy target for anyone to poke and break into, unlike an IT environment where things are more frequently patched, configured and protected.

IoT devices have malfunctions and security flaws since their manufacturing is done with limited resources and have a limited number of objectives to minimize costs and complexities [34]. This provides a larger attack surface for attackers to use the devices. Another popular threat is a compromise of one IoT device allowing attackers to carry out fraudulent actions against other network devices. Understanding these potential security risks layerwise is important to effectively audit the IoT device. The possible attacks [35] that can be performed on the IoT devices are categorized according to the proposed layer and are shown in Fig. 3.

5.1 Application layer

Application layer feature includes terminals, browsers, and other user interfaces. IoT devices may come with certain security vulnerabilities, depending on the features and functionalities of their application layer. Some examples of potential application layer attacks are listed below:

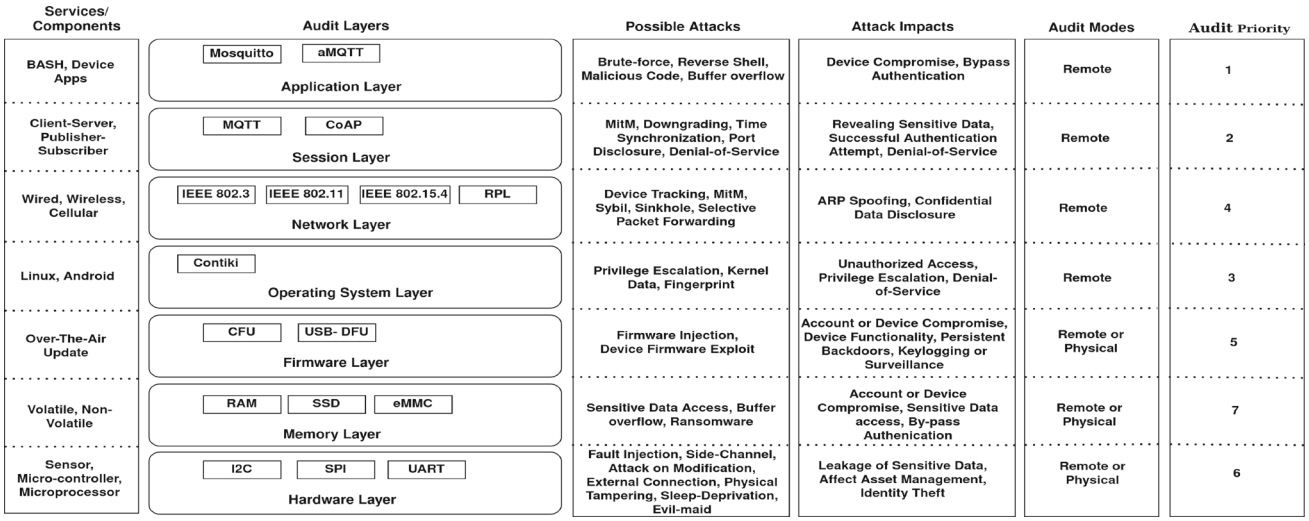

Fig. 3 A novel seven-layer architecture for security auditing of an IoT device

- *Brute-force attack*: It has the potential to find passwords intended to be used to compromise the account of users or, in the worst case, the entire IoT device [36].
- *Reverse shell*: Attackers use reverse shell to gain unauthorized access and exfiltrate data from a target machine. This attack has the potential to control the entire IoT device. Reverse shell attacks could be identified by checking for untrusted code, checking if any listener is set up encrypted or obfuscated to avoid detection, checking the shell session on the intended target machine, checking if it is issuing commands, and potentially escalating privileges to gain further access to the system or network [37].
- *Malicious code*: The input validation flaw could allow malicious code to be inserted into an IoT device. The malicious code injection can change the way the application does its task, and this can have a significant impact on the entire IoT device [38].
- *Buffer overflow attack*: Consuming the memory beyond the allotted space is referred to as the buffer overflow attack. Using this attack, the attacker can extract sensitive data or bypass authentication on IoT devices [39]. The applications not validating the memory access are vulnerable to the buffer overflow attack.

5.2 Session layer

This layer services and components include client–server, and publish-subscribe model. Depending on the service and components vulnerabilities could exist in IoT devices. The session layer attacks that might be carried out by the attackers are outlined below:

- *Man-in-the-middle attack (MitM)*: It occurs when an attacker uses a technique such as ARP poisoning to gain access to privately monitor in and out conversations between two IoT devices, or modify the traffic exchanged between them.
- *Downgrading attack (SSH/SSL/TLS)*: Attackers always try to downgrade the protocol version while establishing the connection. The downgraded protocol will have the vulnerability and attackers exploit it. On successful downgrading attack, attackers may be able to analyze traffic and obtain sensitive data from an IoT device if protocols such as SSH/TLS are downgraded.
- *Time synchronization attack*: The IoT application may force the devices to have synchronized time for running the service. The devices will regularly communicate with other devices to synchronize the time. In case of non-synchronization of time, an IoT device cannot actively participate in the application. Attackers make use of synchronization process to isolate an IoT device by performing the time synchronization attack.
- *Port disclosure attack*: The IoT device runs various services on specified ports and this can be identified by any other nodes by sending a request. The response discloses sensitive information such as ports that are open, services running on the port, and version number.
- *Denial of service attack*: Attackers make use of the protocol to perform the denial of service. For example, initiating a large number of connections with the MQTT broker at the same time. This makes the MQTT broker unavailable for the genuine nodes. The severity of the attack depends on how long the MQTT broker is kept busy. The Message Queuing Telemetry Transport (MQTT) is Publisher-Subscriber protocol used for the message exchange.

5.3 Network layer

The network layer includes wired, wireless, cellular, and other services. Depending on the services, IoT device may have specific vulnerabilities. Various routing attacks target this layer due to its multi-hop environment. Below is a summary of potential network layer attacks:

- *Device tracking*: The disclosure of device identity such as MAC address could allow device tracking. Device tracking can be the starting point for other attacks such as spoofing, which could significantly impact the security of IoT devices [40].
- *Sybil attack*: It occurs when an attacker sets up multiple fictitious identities or nodes in the network to deceive and compromise the application. The attacker controls each false identity to influence others, obstruct communication, or violate network integrity.
- *Selective forwarding attack*: The attack node in the routing path will selectively forward the packets to the next node. This allows the attacker to perform a DDoS attack by dropping all data packets and forwarding only control packets. This can cause massive disruptions in the routing.
- *Sinkhole attack*: An attacker convinces the target IoT device stating that the most optimal route for all other nodes is through it. This causes all traffic from the neighboring node to be routed through the attacked node. This allows the attacker to intercept data, resulting in a data breach or disruption to network services.

5.4 Operating system

The IoT devices use different operating system including Linux and Android. The services of the operating system may have vulnerabilities and lead to exploit. Various attacks at this layer is discussed here:

- *Privilege escalation attack*: The attacker with a low privilege can get the higher privilege access if the operating system access control has the vulnerability. This attack leads to disclosure of sensitive information and compromise of the device.
- *Kernel data attack*: Altering the kernel data is called the kernel data attack and this can be achieved through the buffer overflow exploit. The buffer overflow replaces kernel memory to execute arbitrary code and kernel-level rootkits that covertly modify kernel data to hide malicious activity. Effects of this attack are gaining unauthorized access, escalating their privileges, or compromising the system [41].
- *Fingerprint attack*: The device never discloses its operating system type and version because the disclosure may bring various exploits if there is a vulnerability. Attackers will try to get the OS fingerprint (OS type and version) of the IoT device using different methods including the uniqueness in the network packet structure, Time To Live (TTL), etc.

5.5 Firmware layer

The firmware layer comprises of several components and services for the device. Also allows for over-the-air updates. However, firmware layer components and services may have vulnerabilities and exploits. The potential firmware layer attacks are discussed below:

- *Device firmware exploitation*: Attackers having the firmware, can extract the hard coded password if any, and identify if the outdated firmware with known vulnerabilities in use. This will allow attackers to bypass security measures, extract private data, or even take complete control of the device [42].
- *Firmware injection*: Attackers can exploit firmware flaws to execute arbitrary code. This attack allows unauthorized access and changes the behaviour of the IoT device [24].

5.6 Memory layer

The memory layer comprises of volatile and non-volatile storages. The possible memory layer attacks are shown below:

- *Buffer overflow attack*: Attackers could overwrite the restricted memory of an IoT device to get privileged access or corrupt the running service. Inaccurate assumptions about the size or composition of data are cause for the attack [39].
- *Sensitive data access*: The sensitive data stored in the memory may be accessed by the unauthorized users by exploiting various vulnerabilities including the buffer overflow attack and privilege escalation attack.
- *Ransomware*: The ransomware encrypts the device data without the knowledge of the users. A ransom is demanded in exchange for the normal operation of the device. This attack injects malicious code and affects the data in the memory of the IoT device [43].

5.7 Hardware layer

IoT devices may have vulnerabilities in hardware layer components and services. The components include sensors, microcontrollers, microprocessors, ROM, RAM and debug

ports. A summary of potential hardware layer attacks that could be possible is given below:

- *Fault injection*: A defect is intentionally injected into the system to alter its intended function. This attack reveals sensitive information or provides illegal access [44].
- *Side-channel attack*: This attack targets the implementation instead of the protocol or algorithm. Using this attack, the attacker might acquire insights into the internal processes of the device and extract the sensitive or confidential information [45]. The sensitive information such as cryptography keys are extracted by analyzing the device power consumption, electromagnetic emissions, etc.
- *Attack on modification*: Attackers may physically modify hardware to fulfill their malicious goals. These modifications run the risk of undermining the security of the device. This attack would create an open door for illegal access, data tampering, etc [46].
- *Physical tampering*: Attackers may get physical access to the device and manipulate its connections, hardware, or external components to obtain sensitive information, gain unauthorized access, or change the behavior of the device. [47].
- *External connection*: The interfaces on the IoT devices for debugging can be used by the attacker to get connected with the device and gather sensitive information or inject malicious scripts and data. [48].
- *Sleep deprivation attack*: IoT devices use low-power or idle sleep mode to save power and extend battery life. An attacker might drive the device into an extended sleep state to cause a denial of service [49].
- *Evil maid attack*: An attacker can gain temporary physical access to a device and during this access window, the attacker could modify the firmware, install malicious software, or take sensitive data from the IoT device [50].

6 Analysis of IoT applications

Researchers have discussed various home automation applications for IoT devices. Cristina et al. [51] have developed a smart home application based on ESP8266 and Raspberry Pi designed for home and building automation, including device communication, access, and security. OpenHAB [52] is an open-source web application developed in JAVA and uses REST API. It helps to simplify home automation by supporting different protocols and plugins. An open-source web application ioBroker [53] supports all hardware platforms that help integrate various home automation systems as a single unit. It has a modular structure and can be scaled according to the needs of individuals. Amma et al. [54] have discussed an Android application enabling a user to control home appliances remotely. It uses the MQTT protocol to communicate with the IoT devices.

The IoT-based smart home intrusion detection system has been proposed by Kesswani et al. [55] and it can help mitigate impersonation and manipulation. Tiago et al. [56] have proposed an innovative DDoS detection scheme in soft-ware-defined networks to help mitigate DDoS in the IoT ecosystem. Secure communication between sensors and actuators provided by the cloud as a service is essential for IoT applications [57]. Therefore, device manufacturers or users can use these audit results to prevent threats before they occur.

7 Security requirements

Security requirements encompass a proactive and strategic approach to develop a strong defense against security threats. The security requirements for the IoT device is similar to the information technology device; however, public placing of devices and resource constraints of device brings difference. An IoT ecosystem is considered secure only when the devices fulfill the following expanded security requirements. The manufacturer, the user, or both should fulfill the security requirements. Here, the user refers to both the administrator and the end users.

7.1 Data security

The sensitive data stored in the devices should not be accessible by unauthorized users and to achieve it, the following should be ensured:

- Securely store credentials (A1): The credentials of the device should be hashed with a salt and stored. [Manufacturer and User]
- Securely store sensitive data (A2): The sensitive data of the user should be encrypted and stored. The key should be derived from the Trusted Platform Module (TPM). [Manufacturer and User]
- Non-disclosure of device's sensitive data (A3): The device should not disclose any sensitive data such as battery percentage and baud rate to unauthorized users.

7.2 Network security

The network traffic carrying sensitive data should be made confidential, either using a secure protocol or non-secured protocol with encryption.

- Secure protocols (B1): It is mandatory to establish a secure tunnel between the device and the receiver/transmitter before transmitting or receiving the data. [Manufacturer and User in case of own application installation]
- Necessary network interfaces and services only (B2): The device should only run essential network services and interfaces such as wireless, wired, and Bluetooth. [Manufacturer and User]
- Secure remote access (B3): The device should only be accessed through the secure mode and the user can be authenticated using the password or key. [Manufacturer and User]

7.3 Hardware security

The devices can have the TPM or similar controllers to prevent the boot virus, etc. The hardware components of the device should not leak any sensitive information or allow users to get the unauthorized access.

- Secure boot (C1): The secure boot of the system can be ensured using TPM or other modules. [Manufacturer]
- Side channel attack (C2): The hardware should be resistant to various side channel attacks such as power analysis, timing analysis and electromagnetic attack. [Manufacturer]
- No sensitive data leakage in boot log (C3): Sensitive data such as passwords and keys should not be leaked in the booting process. [Manufacturer]
- No access of hardware (C4): The device should have the sensing to overcome data leakage for external access. For instance, attackers can access the ROM using an SPI, JTAG, and I2C. [Manufacturer]

7.4 Software security

The updated and secure software or firmware has to be used in the IoT devices. No vulnerable applications, operating systems, firmware, drivers, and interfaces should be used.

- Privilege control (D1): The operating system should have the proper privilege control to access the services. [Manufacturer and User]
- Secure default settings (D2): All secure settings should be enabled by default. For example, the UDP echo and chargen should be disabled. [Manufacturer]
- Necessary software services only (D3): The device should run only the necessary software services by default. [Manufacturer and User]

7.5 Management security

Security hardening of device is essential for mitigating various attacks. Following requirements should be fulfilled to ensure the management security of the device.

- Default or weak passwords (E1): Users of the devices should not be able to use the default or weak password. Manufacturers should ensure that the default password is modified during the first access as well as a strong password and password lifetime policy applied. [Manufacturer and User]
- Unique password (E2): The administrator should ensure the unique password for all devices in the network. [Administrator]
- Multi-factor authentication (E3): If needed, the device should support multi-factor authentication.
- Need only services (E4): The device should run only desired services (applications). [Manufacturer and User]
- Unique identification of devices (E5): The devices in the network should be uniquely identified without any spoofing possibilities. [Manufacturer and User].

The possible layer-wise security requirements and security challenges are presented in the Table 8.

8 Audit results of tools

To analyze all the audit tools, we have used the IP based camera since it is extensively deployed in different environments, and the security of these devices has become a significant concern. The camera used is a wireless IP camera equipped with a 4 mm fixed lens that allows 360° panning and 114° tilting. It also has a built-in speaker and microphone for two-way voice communication. In our experiments, we have considered a wireless IP camera, an Android phone that was used to set up the camera for streaming, and a personal computer with the configuration (Processor: Core i7, RAM: 16 GB) to run the audit tools (UI-based). The audit tools interact with the target or test device to perform audits based on network packets.

Tables 9, 10, 11, 12, and 13 show the security audit results of wireless IP camera using the existing tools Asheem et al. [15], Fing [19], Shodan [20], Censys [21], and ASAT [13] respectively. Using these tools, we perform audits on many test cases. One of them is the extensive port scanning; the audit results showed that ports 443, 554, and 2020 were open in the case of the Fing and ASAT tool, and 443 and 554 were open in the case of other tools; it is observed that it has a low severity, and it belongs to the session layer of device layers. It could be inferred that it is not as highly vulnerable.

Table 8 Security requirements at IoT device layers

Layer	Security requirement	Security challenge
Application layer	Data security, software security, management security	One can get remote access to the device or take control or can log in to the application by performing a brute-force attack
Session layer	Network security, data security, software security, management security	Uses of MQTT or CoAP, are meant for constrained devices and lack of authentication may have adverse effects when implemented for bulky networks
Network layer	Network security	This layer suffers from these attacks, i.e., device tracking, and MitM attacks
Operating system layer	Data security, software security, management security	This requires an authentication protocol, and a secure directory traversal mechanism to safeguard against unauthorized access, tampering, and data breaches
Firmware layer	Software security	One can extract the firmware and may use this to exploit the devices
Memory layer	Data security	It poses a substantial threat that leads to manipulation. Memory compromise leads to device integrity, sensitive data leakage, memory leaks or device crashes
Hardware layer	Hardware security	It deals with the physical components of devices that may be exploited by physical tampering or also suffer from side-channel attacks

Table 9 Audit results of IP camera with Asheem et al. [15]

Test case	Vulnerable	Results discussion	Audited layer	Severity
Extensive port scanning	**x**	Port 443 (HTTPS), 554 (RTSP streaming), 8800 (sunwebadmin) and 2020 (xinupageserver) were open	Session	Low
Operating system	✓	It was guessing the different version of Linux OS and gave accuracy in percentage	Operating system	Medium
SSL version	✓	It was using an outdated version of SSL/TLS, which can be utilized to attack the device	Session	Medium
SSL certificate	✓	It showed the detailed information of certificate (issuer's name, expiry date)	Session	Medium
MQTT password brute-force	**x**	Not applicable (Not using MQTT protocol)	Application	null
Un-authenticated MQTT	**x**	Not applicable (Not using MQTT protocol)	Application	null
CoAP	**x**	Not applicable (Not using CoAP protocol)	Application	null

Table 10 Audit results of IP camera with Fing (v12.5.3) [19]

Test case	Vulnerable	Results discussion	Audited layer	Severity
Extensive port scanning	x	Port 443 (HTTPS), 554 (RTSP streaming) and 2020 (xinupageserver) were open	Session	Low
Deauthentication	✓	Able to deauthenticate the device from the access point	Network	High
MAC address	✓	It discloses the MAC address	Network	Medium

Table 11 Audit results of IP camera with Shodan [20]

Test case	Vulnerable	Results discussion	Audited layer	Severity
Extensive port scanning	x	Port 443 (HTTPS) was opened	Session	Low
SSL certificate	✓	It was showing certificate issuer and expiry details	Session	Medium
SSL version	✓	It was showing the version of SSL/TLS used for communication	Session	Medium

Table 12 Audit results of IP camera with Censys [21]

Test case	Vulnerable	Results discussion	Audited layer	Severity
Extensive port scanning	x	Port 443 (HTTPS) was open	Session	Low
SSL certificate	✓	It was showing the detailed information of certificate (issuer's name, expiry date)	Session	Medium
SSL version	✓	It was showing the version of SSL/TLS used for communication	Session	Medium
Operating system	✓	It was guessing the name of operating system	Operating system	Medium

However, using the same Fing audit tool, it is observed that a deauthentication attack can be performed on the camera and it has a high-severity vulnerability. It causes the device to disconnect from the access point intermittently.

Censys performs an additional security check by guessing the name of the operating system. Asheem et al. [15], Shodan, and Censys provide certificate issuer and expiration details, the version of SSL/TLS used for communication for test cases such as SSL certificate and SSL version.

In the case of Asheem et al. [15], MQTT password brute-force, unauthenticated MQTT, and CoAP are the additional provisions for auditing devices compared to other security audit tools. However, in the case of the IP camera, these test cases were not helpful because they did not use the MQTT protocol. Asheem et al. [15] also performed a security audit of guessing the different versions of the Linux operating system and gave the accuracy in percentage. All the test cases discussed by Asheem et al. [15], except extensive port scanning, had a medium severity level.

ASAT in Table 13 identifies the vulnerabilities in the IP camera for the test cases such as checking firewall availability, SSL version, MitM, operating system type, device fingerprinting, DNS reconnaissance, and deauthentication. Firstly, by checking the firewall, it has been found that no firewall protection is provided; it is highly severe and vulnerable to attack. Secondly, the MitM attack was found to be possible using the testbed. It was also found to have high severity and is vulnerable to attack. Next, the SSL version is checked, and an outdated version of SSL/TLS is found, which can be used to attack the device. Similarly, the operating system and device fingerprinting information have remotely been audited. It has been found that it is able to guess the operating system name and identify the device information, including the vendor name and device type. It also checks for deauthentication, and it is found that it performs a deauthentication attack on the IP camera. The SSL version, operating system, device fingerprinting, and deauthentication test cases discussed have a medium severity. Test cases such as MQTT password brute-force, unauthenticated MQTT, and CoAP information are negligible as neither the MQTT nor the CoAP protocol is used. It also checks whether a fake port is running or not. An SSH attempt is not possible as the IP camera does not open port 22 (SSH). The audit result of ASAT along with details is shown in Fig. 4.

Table 13 Audit results of IP camera with ASAT [13]

Test case	Vulnerable	Results discussion	Audited layer	Severity
Extensive port scanning	x	Port 443 (HTTPS), Port 554 (RTSP streaming) and 2020 (xinupageserver) were open	Session	Low
Check firewall	✓	No firewall protection	Network	High
SSL version	✓	It was using an outdated version of SSL/TLS, which can be utilized to attack the device	Session	Medium
SSL certificate	✓	It was showing the detailed information of certificate (issuer's name, expiry date)	Session	Medium
SSH attempt	x	Port 22 (SSH) was not open on this device	Session	null
MitM	✓	Able to launch a MitM attack using the testbed	Session	High
Operating system	✓	It was guessing the OS type	Operating system	Medium
Device fingerprinting	✓	Able to identify the device information, including the vendor name and device type	Hardware	Medium
DNS reconnaissance (Hostname)	x	Not able to get the hostname of the device	Session	null
Deauthentication	✓	Able to perform this attack on this device	Network	Medium
MQTT password brute-force	x	Not applicable (Not using MQTT protocol)	Application	null
Un-authenticated MQTT	x	Not applicable (Not using MQTT protocol)	Application	null
Fake port number	x	No fake port number found on this device	Session	null
CoAP	x	Not applicable (Not using CoAP protocol)	Application	null

8.1 Comparative analysis of audit tool

We considered five web-based audit tools that are freely available to audit IP-based devices. While performing the security audit on IoT devices by considering different test cases, these test cases varied from reconnaissance to penetration testing. During the audit, disclosing any vulnerability can cause an unauthorized person to compromise with the device. The comparison of audit tools output for the IP-based camera is presented in Fig. 5. The blue bar in the graph indicates the total number of test cases each audit tool has completed. It can identify vulnerabilities in the device with medium and high severity, shown in orange and red. Out of these five audit tools, Fing and ASAT were able to find the vulnerability having a severity as high during the audit of the IP camera.

8.2 Recommended mitigation

It has been found while auditing that the tested IP based camera is vulnerable to MitM and deauthentication attacks. Also, the obtained information of reconnaissance can be used to exploit the device. The deauthentication attack can be defended using Wireless Protected Access 3 (WPA3) [58].The sensitive information disclosure such as open ports can be hidden by running the services on non-default ports or implementing the port knocking technique. The ARP poisoning based MitM can be mitigated through the IP and MAC address binding. The device should run firewall to protect the device from malicious users. The operating system fingerprint disclosure can be mitigated by modifying the default values for the attributes for example modifying the default value of the TTL. The device should never use vulnerable versions of the SSL and SSH protocols. The strong credentials should be used in the device for authentication to avoid the password guessing attack. Similarly, for all identified vulnerabilities, a fix available that can be applied to mitigate actual attacks by the attackers. The significant requirement is to identify the vulnerability and that can be achieved by performing the security audit using the audit tools.

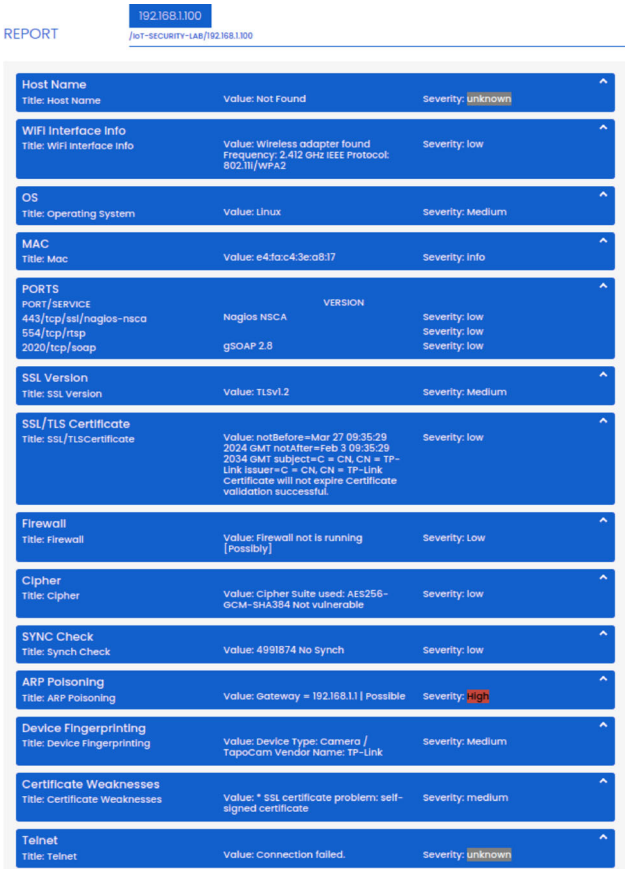

Fig. 4 Audit results of ASAT tool

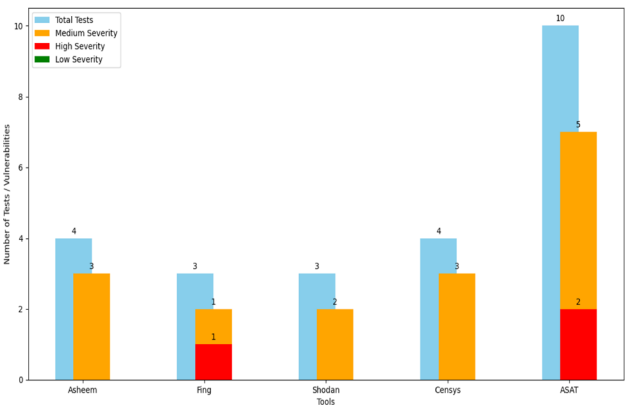

Fig. 5 Comparative analysis of audit tools results

9 Open research challenges and directions

Different IoT ecosystems help the life of people to automate their daily tasks. For example, smart home, smart city automation, and industry control systems are the popular ecosystem. Because of the extensive use of the various IoT devices in smart home automation and other ecosystems, there is a need for the security audit tools to audit the connected devices and mitigate the cyberattacks [51].

Although a wide range of audit tools are available for IoT cybersecurity, there are still many gaps in these tools. Further, the existing open source audit tools are covers less reconnaissance and penetration testing. Following are the test features that can be added to the existing audit tools to render them capable of auditing a large pool of modern threats or attacks.

- Side-channel attack: In today's world, attackers can collect sensitive data even without privileges or running malicious code on mobile and IoT devices. Attackers can use Side-channel Analysis (SCA) to obtain confidential data from the embedded devices [45]. The audit tool should perform Side-channel Analysis to identify and prevent unauthorized information leakage from IoT devices.
- Rogue device detection: A rogue node is an IoT device controlled by malicious or unauthorized users. It tricks other devices into connecting to it by pretending to be a legitimate device within the IoT network. It introduces false data into the IoT network. Thus, a rogue IoT node can be considered as a malicious entity within an IoT network [59]. The existence of rogue or counterfeit IoT devices poses a significant risk to the privacy and security of user data within the IoT network.
- Reconnaissance: Most existing audit tools rely on traditional methods for fingerprinting the device name, operating system, and version, with the help of Nmap and other open-source tools [60]. But nowadays, these traditional techniques do not give accurate results, and most of the time, they fail to fingerprint the names of devices, operating systems, and their versions in different scenarios. Machine learning approaches could help to get trained with the existing dataset eventually, classify the results with higher accuracy, and perform the audit process effectively [61].
- Secure boot: The system boot process loads various services including Basic Input/Output System (BIOS), operating system and system utilities. It is possible that the attackers may insert the malware during the boot process and it cannot be detected by the anti-virus or other security solutions. Hence, we need secure boot to verify the integrity before loading any services. The audit tool should test whether the secure boot is implemented in the IoT device or not.
- Device resilient to outages: Resilience audits are essential for ensuring that IoT devices continue to function in case of any failure such as network connectivity. The resilience required especially in applications where downtime can have serious consequences. The IoT devices are highly heterogeneous because of their different manufacturers and intended uses, which makes it

Table 14 Open research challenges with security requirements

Features	Security requirements	Challenges
Side-channel attack	A2, A3, C2	It is hard to perform side channel audit because we need a specialized device to measure the accurate power consumption and electromagnetic emission
Rouge device detection	E3, E5	Because of the massive volume of data generated by devices, diversity in device types, different communication protocols, and mobility of devices make it difficult to identify or audit rouge devices
Reconnaissance	B1, B3, D3	A standardization protocol may prevent the audit device from being tailored according to the manufacturers, or this effort may be hindered because of segmented networks or encrypted communication
Secure boot	C1, C3	Custom firmware used by different manufacturers and lack of complete visibility into the firmware or boot process may hinder the secure boot audit process of the device
Device resilient to outages	D1	In case of power failure and network disruptions, evaluation of data loss, or recovery process, may be problematic as different devices handle data differently. Also, the rapid evolving nature of threats may hinder the audit process
Need only data sharing	A3, D3	Some ports may exhibit intermittent or conditional behavior depending on the connected device or current configuration, making it difficult to determine their capabilities consistently
Reset To Default Settings	D2	A remote audit may fail to do this, as it requires physical access to the device to check the presence of any hard reset button on the device or any means of access to the device menu for the hard reset options
Secure storage of sensitive data	A2, A3	In the case of proprietary or non-standard encryption methods result in detailed logs and data integrity being exposed by devices
Firmware Fingerprint	A2, B3	A remote audit process may not fetch the firmware details in case shared through secure channel

more challenging to handle resilience issues on all of the devices [62]. The auditing tool must identify whether the device is resilient to device outage or not.

- Need only data sharing: It is required to identify whether the data is shared with the other devices based on need or not. It is possible that an injected malware may share the data with the other devices without authentic need. This leads to the sensitive data disclosure. Also, the audit tool should verify whether the data is shared securely or not with the authorized devices or users.
- Reset to default settings: The IoT device resetting to factory defaults is a crucial requirement in IoT device security [63]. If an attacker causes any exploitation, the devices must reset to their factory defaults to avoid further damages.
- Secure storage of sensitive data: The audit tool must be so designed as to assess secure storage methods on IoT devices [64].
- Firmware fingerprint: The audit tool should be able to identify the firmware name and version when remotely auditing the IP-based device, helping to determine whether or not the device is running the latest version. These features are essential when deploying IoT devices in remote or insecure locations [24].

Table 15 Analysis of security requirements with respect to existing tools

Security requirements category	Sub-category	Attempted
Data security	A1	No
	A2	No
	A3	Yes
Network security	B1	Yes
	B2	Yes
	B3	Yes
Hardware security	C1	No
	C2	No
	C3	Yes
	C4	Yes
Software security	D1	Partial
	D2	Partial
Management security	E1	Yes
	E2	No
	E3	No
	E4	Partial
	E5	No

Table 14 shows the gap in the existing auditing tools with respect to the expanded security requirements and discusses the challenges in implementing it. Table 15 presents the analysis of the existing audit tools with respect to the security requirements' category and sub-category.

10 Conclusion and future scope

The Internet of Things gains massive importance these days because of its effectiveness and efficiency. Also, IoT applications are an easy target for attackers, since the devices are lightweight, provide easy access, etc. We have explored the possible attacks in the IoT devices and presented the security requirements to protect them. Each IoT device including the network device has different components and services and may be vulnerable to the attacks. It is required to identify the vulnerabilities, if any, present in the device by performing a security audit. This paper has categorized the components and services of the IoT devices into seven layers to perform the security audit effectively. There are different IoT security audit tools; however, they do not cover the security requirements. Hence, this paper has analyzed various physical and remote audit tools and approaches and presented the gaps for future development. We have also presented the audit tool results of a test device, that is, an IP based camera, and discussed the results, showing the existing tools' limited coverage. As a result, we can use our seven-layer IoT device architecture as a benchmark for IoT device component classification and for security audit, both in the manufacturing process and at their deployment within the IoT ecosystem. The future work will be to find an effective solution to overcome the challenges in implementing missing features in the audit tool.

Acknowledgements This work is part of the Project titled 'Development of Security Audit Framework for secure IoT network' funded by C3iHub, Indian Institute of Technology, Kanpur under the National Mission on Interdisciplinary Cyber-Physical Systems (NM-ICPS) of the Department of Science and Technology, Government of India. The authors would like to thank the manufacturers of the devices used in this study.

Declarations

Conflict of interest The authors declare that they have no Conflict of interest.

References

1. Clark M'Kaila, J., Rajabion, L.: A strategic approach to IoT security by working towards a secure IoT future. Int. J. Hyperconnect. Internet Things (IJHIoT) **7**(1), 1–18 (2023)
2. Talal, H., Zagrouba, R.: Mads based on DL techniques on the internet of things (IoT) survey. Electronics **10**, 2598 (2021)
3. GÜVEN, E.Y., et al.: Mirai botnet attack detection in low-scale network traffic. Intell. Autom. Soft Comput. **37**(1), 1–19 (2023)
4. Pawlicka, A., Puchalski, D., Pawlicki, M., Kozik, R., Choraś, M.: How to secure the IoT-based surveillance systems in an elegant way. In: 2023 IEEE International Conference on Cyber Security and Resilience (CSR), pages 636–640 (2023)
5. Rizvi, S., Zwerling, T., Thompson, B., Faiola, S., Campbell, S., Fisanick, S., Hutnick, C.: A modular framework for auditing IoT devices and networks. Comput. Secur. **132**, 103327 (2023)
6. Nadir, I., Ahmad, Z., Mahmood, H., Shah, G.A., Shahzad, F., Umair, M., Khan, H., Gulzar, U.: An auditing framework for vulnerability analysis of IoT system. In: 2019 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW), pages 39–47. IEEE (2019)
7. Li, D., Zhang, Z., Liao, W., Xu, Z.: KLRA: A Kernel level resource auditing tool for IoT operating system security. In 2018 IEEE/ACM Symposium on Edge Computing (SEC), pages 427–432. IEEE (2018)
8. Ursprung, L.: Analyse der sicherheit von IoT-geräten und methoden zur durchführung von penetrationstests für iot-geräte (2024)
9. Dar, A.A., Reegu, F.A., Ahmed, S., Hussain, G.: Strategic security audit protocol: Safeguarding smart home iot devices against vulnerabilities. In: 2024 11th International Conference on Computing for Sustainable Global Development (INDIACom), pages 1386–1391 (2024)
10. Mirzamohammadi, S., Chen, J.A., Sani, A.A., Mehrotra, S., Tsudik, G.: Ditio: trustworthy auditing of sensor activities in mobile & IoT devices. In: Proceedings of the 15th ACM conference on embedded network sensor systems, pages 1–14 (2017)
11. Visoottiviseth, V., Akarasiriwong, P., Chaiyasart, S., Chotivatunyu, S.: Pentos: penetration testing tool for internet of thing devices. In: TENCON 2017 - 2017 IEEE Region 10 Conference, pages 2279–2284 (2017)
12. Yiwen, X., Yin, Z., Hou, Y., Liu, J., Jiang, Yu.: Midas: safeguarding IoT devices against malware via real-time behavior auditing. IEEE

Trans. Comput. Aided Des. Integr. Circuits Syst. **41**(11), 4373–4384 (2022)
13. Kumar, A., Peshvani, B., Venkatesan, S., Kumar, M., Yadav, S., Shukla, S.K.: Automated security audit testbed for IP-based IoT devices without physical access. In: 2023 10th International Conference on Internet of hings: Systems, Management and Security (IOTSMS), pages 96–103 (2023)
14. Waraga, O.A., Bettayeb, M., Nasir, Q., Talib, M.A.: Design and implementation of automated IoT security testbed. Comput. Secur. **88**, 101648 (2020)
15. Nordnes, K.: Iotective: automated penetration testing for smart home environments. Master's thesis, NTNU (2023)
16. Akhilesh, R., Bills, O., Chilamkurti, N., Chowdhury, M.J.M.: Automated penetration testing framework for smart-home-based IoT devices. Fut. Internet **14**(10), 276 (2022)
17. Siboni, S., Sachidananda, V., Meidan, Y., Bohadana, M., Mathov, Y., Bhairav, S., Shabtai, A., Elovici, Y.: Security testbed for internet-of-things devices. IEEE Trans. Reliab. **68**(1), 23–44 (2018)
18. Cayre, R., Nicomette, V., Auriol, G., Alata, E., Kaaniche, M., Marconato, G.: Mirage: towards a metasploit-like framework for IoT. In: 2019 IEEE 30th International Symposium on Software Reliability Engineering (ISSRE), pages 261–270. IEEE (2019)
19. Andrews, A., Oikonomou, G., Armour, S., Thomas, P., Cattermole, T.: Reliable identification of IoT devices from passive network traffic analysis: Requirements and recommendations. In: 2023 IEEE 9th World Forum on Internet of Things (WF-IoT), pages 1–6. IEEE (2023)
20. Mulero-Palencia, S., Monzon Baeza, V.: Detection of vulnerabilities in smart buildings using the Shodan tool. Electronics **12**(23), 4815 (2023)
21. Jian, Q., Ma, X., Liu, W., Sang, H., Li, J., Xue, L., Luo, X., Li, Z., Feng, L., Guan, X.: On smartly scanning of the internet of things. IEEE/ACM Trans. Netw. **32**(2), 1019–1034 (2024)
22. Broström, T., Zhu, J., Robucci, R., Younis, M.: IoT boot integrity measuring and reporting. ACM SIGBED Rev. **15**(5), 14–21 (2018)
23. Mishra, S., Ray, A., Singh, M., Venkatesan, S., Anand, A.S.: Automated hardware auditing testbed for uart and spi based iot devices. In: 2023 10th International Conference on Internet of Things: Systems, Management and Security (IOTSMS), pages 75–82 (2023)
24. Bettayeb, M., Nasir, Q., Talib, M.A.: Firmware update attacks and security for IoT devices: Survey. In: Proceedings of the ArabWIC 6th Annual International Conference Research Track, pages 1–6 (2019)
25. Visoottiviseth, V., Jutadhammakorn, P., Pongchanchai, N., Kosolyudhthasarn, P.: Firmaster: analysis tool for home router firmware. In: 2018 15th International Joint Conference on Computer Science and Software Engineering (JCSSE), pages 1–6 (2018)
26. Jang, D., Kim, T., Kim, D.: Dynamic analysis tool for IoT device. In: 2020 International Conference on Information and Communication Technology Convergence (ICTC), pages 1864–1867 (2020)
27. Ticu, M.: USB traffic analyzer-digusb. In: 2021 12th International Symposium on Advanced Topics in Electrical Engineering (ATEE), pages 1–5. IEEE (2021)
28. Aarseth, H.: Identifying vulnerable services using non-intrusive techniques. Master's thesis (2023)
29. Landauer, M., Mayer, K., Skopik, F., Wurzenberger, M., Kern, M.: Red team redemption: a structured comparison of open-source tools for adversary emulation. *arXiv preprint* arXiv:2408.15645 (2024)
30. Silva, R., Iqbal, R.: Ethical implications of social internet of vehicles systems. IEEE Internet Things J. **6**(1), 517–531 (2019)
31. MITRE. EMB3D: Mitigating embedded system threats. https://emb3d.mitre.org/. Accessed: 2024-10-03
32. Montasari, R.: Internet of things and artificial intelligence in national security: Applications and issues. In: Countering Cyberterrorism: The Confluence of Artificial Intelligence, Cyber Forensics and Digital Policing in US and UK National Cybersecurity, pages 27–56. Springer (2023)
33. Hammi, B., Zeadally, S., Khatoun, R., Nebhen, J.: Survey on smart homes: vulnerabilities, risks, and countermeasures. Comput. Secur. **117**, 102677 (2022)
34. Yu, T., Sekar, V., Seshan, S., Agarwal, Y., Xu, C.: Handling a trillion (unfixable) flaws on a billion devices: Rethinking network security for the internet-of-things. In: Proceedings of the 14th ACM Workshop on Hot Topics in Networks, pages 1–7 (2015)
35. Ahemd, M.M., Shah, M.A., Wahid, A.: IoT security: a layered approach for attacks & defenses. In: 2017 international conference on Communication Technologies (ComTech), pages 104–110. IEEE (2017)
36. Bošnjak, L., Sreš, J., Bosnjak, B.: Brute-force and dictionary attack on hashed real-world passwords. In: 2018 41st international convention on information and communication technology, electronics and microelectronics (mipro), pages 1161–1166. IEEE (2018)
37. Kaushik, K., Aggarwal, S., Mudgal, S., Saravgi, S., Mathur, V.: A novel approach to generate a reverse shell: Exploitation and prevention. International Journal of Intelligent Communication, Computing and Networks Open Access Journal, pages 2582–7707 (2021)
38. Dongdi, W., Xiaofeng, Q.: Status-based detection of malicious code in internet of things (IoT) devices. In: 2018 IEEE Conference on Communications and Network Security (CNS), pages 1–7. IEEE (2018)
39. Habibi, J., Panicker, A., Gupta, A., Bertino, E.: Disarm: mitigating buffer overflow attacks on embedded devices. In: Network and System Security: 9th International Conference, NSS 2015, New York, NY, USA, November 3-5, 2015, Proceedings 9, pages 112–129. Springer (2015)
40. DANG, M.T., NGUYEN, D.T.: Development of an IoT system for traffic analysis purposes from capturing mac address based data. J. Eastern Asia Soc. Transp. Stud. **13**, 60–69 (2019)
41. Nagy, R., Németh, K., Papp, D., Buttyán, L.: Rootkit detection on embedded IoT devices. Acta Cybernet. **25**(2), 369–400 (2021)
42. Dan, Yu., Zhang, L., Chen, Y., Ma, Y., Chen, J.: Large-scale IoT devices firmware identification based on weak password. IEEE Access **8**, 7981–7992 (2020)
43. Ahanger, T.A., Tariq, U., Dahan, F., Chaudhry, S.A., Malik, Y.: Securing IoT devices running Pureos from ransomware attacks: leveraging hybrid machine learning techniques. Mathematics **11**(11), 2481 (2023)
44. Gangolli, A., Mahmoud, Q.H., Azim, A.: A systematic review of fault injection attacks on IoT systems. Electronics **11**(13), 2023 (2022)
45. Lightbody, D., Ngo, D.-M., Temko, A., Murphy, C.C., Popovici, E.: Attacks on IoT: side-channel power acquisition framework for intrusion detection. Future Internet **15**(5), 187 (2023)
46. Li, C., Qin, Z., Novak, E., Li, Q.: Securing SDN infrastructure of IoT-fog networks from MITM attacks. IEEE Internet Things J. **4**(5), 1156–1164 (2017)
47. Pathak, A.K., Saguna, S., Mitra, K., Åhlund, C.: Anomaly detection using machine learning to discover sensor tampering in iot systems. In: ICC 2021-IEEE International Conference on Communications, pages 1–6. IEEE (2021)
48. Hosenkhan, M.R., Pattanayak, B.K.: A framework for secure communication on internet of things (IoT). In: Progress in Advanced Computing and Intelligent Engineering: Proceedings of ICACIE 2020, pages 599–605. Springer (2021)
49. Bada, M., von Solms, B.: A cybersecurity guide for using fitness devices. In: The Fifth International Conference on Safety and Security with IoT: SaSeIoT 2021, pages 35–45. Springer (2022)
50. Boursalian, A., Stamp, M.: Bootbandit: A macos bootloader attack. Eng. Rep. **1**(1), e12032 (2019)

51. Stolojescu-Crisan, C., Crisan, C., Butunoi, B.-P.: An IoT-based smart home automation system. Sensors **21**(11), 3784 (2021)
52. Tsakalidis, S., Tsoulos, G., Kontaxis, D., Athanasiadou, G.: Design and implementation of a versatile openhab iot testbed with a variety of wireless interfaces and sensors. In: Telecom, volume 4. MDPI (2023)
53. Triantafyllou, A., Sarigiannidis, P., Lagkas, T.D.: Network protocols, schemes, and mechanisms for internet of things (IoT): features, open challenges, and trends. Wirel. Commun. Mobile Comput. **2018**(1), 5349894 (2018)
54. Eleyan, A., Fallon, J.: IoT-based home automation using android application. In: 2020 International Symposium on Networks, Computers and Communications (ISNCC), pages 1–4 (2020)
55. Kesswani, N., Agarwal, B.: Smartguard: an IoT-based intrusion detection system for smart homes. Int. J. Intell. Inf. Database Syst. **13**(1), 61–71 (2020)
56. Linhares, T., Patel, A., Barros, A.L., Fernandez, M.: SDNTruth: innovative DDoS detection scheme for software-defined networks (SDN). J. Netw. Syst. Manage. **31**(3), 55 (2023)
57. Makda, T.J., Barros, A.L., Dilek, S.: A secure cloud-based infrastructure for virtual sensors in iot environments. In: 2023 Sixth International Conference of Women in Data Science at Prince Sultan University (WiDS PSU), pages 156–161. IEEE (2023)
58. Neal, Z., Sha, K.: Analysis of evil twin, deauthentication, and disassociation attacks on wi-fi cameras. In: 2023 32nd International Conference on Computer Communications and Networks (ICCCN), pages 1–7 (2023)
59. Bodhe, A., Dhanrao, P., Sangle, A., Narayana, J.: Design secure WSN with advancement in finding rouge access point with soft computing tools. 11 (2020)
60. Calderon, P.: NMAP Network Exploration and Security Auditing Cookbook: Network discovery and security scanning at your fingertips. Packt Publishing Ltd (2021)
61. González-Soto, M., Díaz-Redondo, R.P., Fernández-Veiga, M., Fernández-Castro, B., Fernández-Vilas, A.: Decentralized and collaborative machine learning framework for iot. *Computer Networks*, **239**, 110137 (2024)
62. Benson, K.: Enabling resilience in the internet of things. In: 2015 IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops), pages 230–232 (2015)
63. Baghsorkhi, S.S., Margiolas, C.: Automating efficient variable-grained resiliency for low-power IoT systems. In: Proceedings of the 2018 International Symposium on Code Generation and Optimization, pages 38–49 (2018)
64. Ayoade, G., El-Ghamry, A., Karande, V., Khan, L., Alrahmawy, M., Rashad, M.Z.: Secure data processing for IoT middleware systems. J. Supercomput. **75**, 4684–4709 (2019)