Kristu Jayanti College
AUTONOMOUS Bengaluru
Reaccredited A++ Grade by NAAC | Affiliated to Bengaluru North University

ANALYSIS OF SECURITY OF IOT DEVICES

Project Report

Master of Science Forensic Science

IV Semester

By

SHAHRUKH KHAN

23MFOR31

Under the Guidance of

Mr. DON CAEIRO

ASSISTANT PROFESSOR

DEPARTMENT OF FORENSIC SCIENCE

KRISTU JAYANTI COLLEGE (AUTONOMOUS)

Submitted to

DEPARTMENT OF FORENSIC SCIENCE

KRISTU JAYANTI COLLEGE (AUTONOMOUS)

K. NARAYANAPURA, KOTHANUR P.O.

BANGALORE -560077

MAY 2025

Kristu Jayanti College
AUTONOMOUS Bengaluru
Reaccredited A++ Grade by NAAC | Affiliated to Bengaluru North University

CERTIFICATE

*This is to certify that the report of the Project Work entitled **"ANALYSIS OF SECURITY OF IOT DEVICES "** submitted to The Department of Forensic Science, Kristu Jayanti College (Autonomous), in partial fulfillment of the requirements for the award of the degree of the Master of Science is a record of original work done by **Mr. Shahrukh Khan** during December 2024 and May 2025 of her study in the Department of Forensic Science, Kristu Jayanti College (Autonomous), under my guidance and it has not formed the basis for the award of any degree/ diploma / associateship / fellowship or other similar title to any candidate of any College / University.*

Countersigned
Head
Department of Forensic Science

Signature of the Guide
Mr. Don Caeiro
Assistant Professor
Department of Forensic Science

Kristu Jayanti College
AUTONOMOUS Bengaluru
Reaccredited A++ Grade by NAAC | Affiliated to Bengaluru North University

DECLARATION

*I, **Shahrukh Khan** hereby declare that the report of the Project entitled **"ANALYSIS OF SECURITY OF IOT DEVICES"** submitted to The Department of Forensic Science, Kristu Jayanti College (Autonomous), in partial fulfillment of the requirements for the award of the degree of the Master of Science is a record of original work done by me during December 2024 and May 2025 under the guidance of **Mr. DON CAIERO**, Department of Forensic Science, Kristu Jayanti College (Autonomous).*

Signature of the Student
SHAHRUKH KHAN
23MFOR31

ACKNOWLEDGEMENTS

First and foremost, I express my profound gratitude to God Almighty for His divine grace, wisdom, and unwavering guidance, which have been my strength throughout the course of this research.

I am deeply thankful to Rev. Fr. Dr. Augustine George, Principal of Kristu Jayanti College, for his constant support, encouragement, and for providing a conducive academic environment that greatly facilitated the completion of my research work.

I owe a deep sense of gratitude to Dr. Calistus Jude AL for his guidance during the research work.

My heartfelt thanks to Mr. Don Caeiro, Head, Department of Forensic Science, for his continuous assistance, constructive suggestions, and for ensuring that I remained focused and motivated throughout the project.

I am also grateful to all the teaching and non-teaching staff of the Department of Forensic Science, Kristu Jayanti College, for their kind cooperation, encouragement, and assistance during my academic journey.

I am equally thankful to my fellow group members, Arsha J B (23MFOR12), Riki Leadingson Siangshai (23MFOR26), and Sree Lakshmi.S (23MFOR34) for their collaboration, team spirit, and moral support.

I would like to express my deep appreciation to my family, friends, and well-wishers for their unconditional support, patience, and motivation, which have been instrumental in helping me reach this milestone.

Finally, I extend my sincere thanks to all individuals who, directly or indirectly, contributed to the successful completion of this dissertation. Your support has been invaluable and truly appreciated.

ABSTRACT

Internet of Things (IoT) is currently one of the most widespread technologies in today's world, integrated deeply in various aspects of human life such as home automation, self-driving cars, medicine, and manufacturing. The phenomenal development of IoT has transformed the way devices connect and communicate across environments. Now, IoT has become a way of life—the smart home, wearable tech, and even infrastructural systems. To this end, though, there are immense opportunities that accompany such wild adoption and thereby, earnest issues and security weaknesses. Convenience of use, cost-effectiveness, and omnipresence tend to sacrifice the quality of devices employed with IoT, as well as utilize insecure ones. More vulnerabilities have exposed IoT environments to every type of attack, resulting in more cybercrimes and operational interruptions. Most IoT devices do not have basic security capabilities, including robust authentication, encryption, and malware detection systems. Adding insult to injury, IoT devices are often under the constraints of power, communication, computational power, hardware heterogeneity, and storage.

This research is intended to explore the most neglected threats and vulnerabilities of IoT systems. These are poor authentication mechanisms, old firmware, unencrypted channels, and a lack of encryption, which all lead to unauthorized access, data leakage, and massive botnet attacks. In addition, this research intends to suggest necessary security requirements and countermeasures against these threats, thereby improving the dependability and security of Internet-connected devices for end users.

Keywords: Wi-Fi devices, Internet of Things, Vulnerabilities, Smart devices, Device security, Nmap, Wireshark, Air cracking, IoT security

TABLE OF CONTENTS

SLNO.	CONTENTS	PAGE NO.
1	Introduction	11
2	Review of Literature	24
3	Methodology	34
4	Result and Analysis	39
5	Discussion	102
6	Conclusion	106
7	References	110
8	Annexure	114

LIST OF FIGURES

FIGURE NUMBER	FIGURE DESCRIPTION	PAGE NO.
4.1.1	Scan for open ports	45
4.1.2	Nmap basic scan	45
4.1.3	Version detection scan	45
4.1.4	Scan for known vulnerabilities	46
4.1.5	Aggressive scan	46
4.1.6	Wireshark Packet capturing	46
4.1.7	Wireshark Analysis	47
4.1.8	Capturing SSID and Channel Number Using Wireshark	47
4.19	Deauthentication Attack Traffic Captured in Wireshark	47
4.1.10	Wireshark Capture Showing Unsuccessful WPA/WPA2 Handshake Crack Attempt	48
4.1.11	JADX GUI Analysis	48
4.1.12	Obfuscated or Unreadable Code in JADX-GUI	48
4.2.1	Basic port scan	53
4.2.2	Scan for common ports	53
4.2.3	Version Detection	53
4.2.4	Scan for known vulnerabilities	54
4.2.5	Aggressive scanning	54
4.2.6	Capturing Packets	54
4.2.7	Analyzing Packets	55
4.2.8	Capturing SSID and Channel Number Using Wireshark	55

4.2.9	Deauthentication Attack Traffic Captured in Wireshark	55
4.2.10	Wireshark Capture Showing Unsuccessful WPA/WPA2 Handshake Crack Attempt	56
4.3.1	Basic port scan	60
4.3.2	Scan for common ports and version detection	60
4.3.3	Scan for vulnerabilities	60
4.3.4	Aggressive scanning	61
4.3.5	Wireshark Analysis	61
4.3.6	Capturing SSID and Channel Number Using Wireshark	61
4.3.7	Deauthentication Attack Traffic Captured in Wireshark	62
4.3.8	Wireshark Capture Showing Unsuccessful WPA/WPA2 Handshake Crack Attempt	62
4.4.1	Aggressive scan	66
4.4.2	Basic port scan	66
4.4.3	Common service ports	67
4.4.4	Version scan detection	67
4.4.5	Capturing Packets	67
4.4.6	Analyzing Packets	68
4.5.1	Aggressive scan	72
4.5.2	Vulnerability scan	72
4.5.3	Service version detection	72
4.5.4	Stealth scan	73
4.5.5	Wireshark Analysis	73
4.6.1	Operating system scan	76
4.6.2	Vulnerability scan	77
4.6.3	Aggressive scan	77
4.6.4	Stealth scan	78

4.6.5	Capturing Packets using Wireshark	78
4.6.6	Analyzing Packets	79
4.7.1	Ping Scan	82
4.7.2	Port Scanning (Version and OS scan)	83
4.7.3	Netcat Scan	83
4.7.4	Wireshark Analysis	83
4.7.5	Wi-fi attack	84
4.7.6	Cracking Password	84
4.8.1	Port Scanning (version and OS scan)	86
4.8.2	Traffic Analysis- Wireshark	87
4.8.3	Wi-fi attack	87
4.8.4	Cracking Password	87
4.9.1	a. Version detection, b. Port scan on a specific port.	91
4.9.2	a. Disable host discovery, b. Aggressive scan with OS and version detection, skipping host discovery, using faster time.	91
4.9.3	a. Version detection, b. Perform a stealth TCP SYN scan on specified ports without pinging the host first.	92
4.9.4	a. Run vulnerability detection on script, b. Perform a stealth TCP SYN scan, skipping host discovery, and run vulnerability detection scripts, c. Perform a stealth TCP SYN scan on ports 1000 to 2000, skipping host discovery, and run vulnerability detection scripts, d. Perform a stealth TCP SYN scan on ports 2000 to 3000, skipping host discovery, and run vulnerability detection scripts	93
4.9.5	Wireshark ICMP Request Analysis	94

4.9.6	Wireshark TCP SYN Scan	94
4.10.1	a. Ping scan on target ip, b. Aggressive scan, c. Performs a TCP SYN (half-open) scan on specified ports to detect which ports are open without completing the full TCP handshake	99
4.10.2	2 a. Performs an aggressive scan with OS detection, version detection, script scanning, and traceroute using a faster timing template (T4) for quicker execution, b. Performs service version detection on port 10082 and runs vulnerability detection scripts against the identified service.	100
4.10.3	3 a. Performs a TCP SYN (half-open) scan on the specified common ports (80, 443, 21, 22, 23, 25) to detect which services are open on the target IP address without completing the full TCP handshake, b. Performs a TCP SYN (half-open) scan on all 65,535 ports and detects the service versions running on the open ports of the target IP address.	100
4.10.4	a. Performs a TCP SYN (half-open) scan on the specified common ports (80, 443, 21, 22, 23, 25) to detect which services are open on the target IP address without completing the full TCP handshake, b. Performs a TCP SYN (half-open) scan on all 65,535 ports and detects the service versions running on the open ports of the target IP address.	101
4.10.5	Ettercap 0.8.3.1 in use for local network reconnaissance	101

INTRODUCTION

1. INTRODUCTION

1.1 Emergence of Internet of Things (IoT)

The phrase Internet of Things (IoT) describes a wide network of connected physical devices with sensors, actuators, software, and other technologies, which allow these devices to communicate and exchange, and respond to data through the use of the internet. The idea of the IoT was initially put forward by Kevin Ashton in 1999 while working at Procter & Gamble, where he had envisioned the RFID tags to be used as part of the supply chain management to provide better results (Ashton, 2009). The vision has since expanded enormously to include a plethora of applications and transformed the physical world into a rich data-ped entity's interactive world. IoT has matured at a fast pace due to the interlinkage of various technical developments, including cloud computing, edge computing, machine learning, wireless sensor networks, and miniaturized HW components (Gubbi et al, 2013). This convergence has enabled the embedding capability of computing and communication into an extensive toolkit consisting of smart thermostats, smart lights, industrial robots, and self-driving vehicles. The outcome, thus, is a digitized real world in which real-time data can be collected, analysed, and acted upon, often with little human intervention. As Statista reports (2023), the population of IoT-connected devices around the world is predicted to increase to above 29 billion in the year 2030, being an outstanding rise from the 13.1 billion devices attested in 2022. So rapid expansion is enabled by the presence of growing demand from consumers for automation and smarter infrastructure in urban and rural settings. Availability of cheap sensors all over the world, higher connectivity (including 5G network), and better storage of data have accelerated the development and launch of IoT technologies in industries further.

In essence, IoT is a paradigm change in the way devices interact among themselves and their users. Whereas previously, traditional computing systems necessitated the direct instructions of a user, often in the form of API calls, IoT devices operate more autonomously, monitoring their surroundings and taking action based on predefined rules or real-time analytics (Atzori, Iera, & Morabito, 2010). This unseamly conjunction of physical and digital systems is crucial to the concept of ubiquitous computing, in which, following the futuristic vision mapped out by American researchers, technology seems to seamlessly become part of the ordinary order of things

and surroundings. Nevertheless, the tremendous rate and reach with which the IoT has been adopted show that significant problems are associated with this, especially security, privacy, and trust. With an increase in connected devices comes an increase in vulnerability, surface area for cyberattacks. With the integration of IoT in the infrastructure and personal settings, the repercussions from device failure and compromise become wider and disastrous. Therefore, even though the emergence of the IoT represents a significant technological progress, urgent and continuous focus on architectural and security weaknesses of the IoT is required.

1.2 Applications and Benefits of IoT

Internet of Things (IoT) has seen abundant applications in various sectors owing to its ability to offer intelligent automation, enhanced efficiency, and real-time data monitoring and decision making. The flexibility and expansiveness of IoT have embedded it as a central element of current digital infrastructure, with industries, governments, and citizens alike changing their interactions with their surroundings. In healthcare, the ability to enable remote patient monitoring as well as wearable health on individual tracking, and also smart diagnostics has been supported by IoT-enabled devices. For example, wearable biosensors can sense such vital signs as heart rate, blood pressure, and glucose level, and transfer them to the healthcare providers for early proactive care and intervention (Islam et al, 2015). These abilities are particularly important in the treatment of chronic illnesses and the enhancement of patient results with fewer hospital stays and cost of operationIn the agriculture sector, IoT is the focal point of precision farming that is made easier through monitoring of soil moisture, temperature, health of crops and state of equipment with the help of, IoT. Intelligent irrigation systems that depend on sensor data can maximize the utilization of water and increase crop yields. These applications help to support sustainable farming practices, especially in areas with climate change and resource scarcity conditions (Wolfert et al. 2017). The transportation and logistics industry is empowered by IoT to monitor in real time the vehicles, goods, and shipments. Fleet management systems use GPS and onboard diagnostics to track the status of operations, including driver conduct, fuel usage, and maintenance requirements. Smart traffic systems & autonomous vehicles are other examples of how IoT is playing a role in decreasing congestion, increasing safety, and lubricating urban mobility (Al-Fuqaha et al, 2015).

The implications of the IoT in industrial automation are huge, which is often referred to as the Industrial Internet of Things (IIoT). It entails the use of sensors and connected devices in

monitoring machinery, detecting faults, and predicting maintenance needs in manufacturing plants. Not only is this more efficient when operating, but this also reduces downtime and increases the lifetime of the equipment (Gilchrist 2016). In the smart homes environment, IoT-enabled devices like thermostats, lighting systems, voice assistants, as well as cameras for security purposes, provide personalized automation, energy saving, and improved security. Convenience and control of the environment are enhanced for consumers, which can frequently be achieved through the use of smartphone apps or by voice commands. In general, the main benefits of the IoT include greater operational efficiency, cost savings, safety improvement, and better customer experience, all of which are real-time responsive. Through the provision of the ability to collect and analyse huge amounts of data, IoT systems support the making of well-informed decisions and predictive analytics. Additionally, the IoT interoperability with emerging technologies such as artificial intelligence (AI), cloud computing, and edge computing expands the utility and use of these technologies (Miorandi et al, 2012). However, the increasing reliance on IoT systems increases the exposure of the system to disruption if security is breached. The dependent nature of devices carries with it the fact that if an issue arises from a single device component, this can impact entire systems, hence the need to have strong security frameworks and constant monitoring

1.3 Inherent limitations of IoT Devices

Even with their potential for transformation, IoT devices frequently experience severe technical and operational limitations, which leave them prone to breaches and performance issues. Such limitations are largely attributable to the limitations of power supply, processing power, memory, and network bandwidth – aspects that are often undermined at the device design stage in the interest of cost efficiency and mass market scalability (Sicari et al, 2015). Most of the IoT devices are designed using basic hardware resources to reduce production costs. As such, they are usually computationally weaker and do not possess the capability of supporting sophisticated encryption algorithms and higher-order security mechanisms. For instance, the use of public-key cryptography necessary for secure communication in a conventional computing environment is usually cost-prohibitive (Granjal, Monteiro, & Silva, 2015) in low-horsepower microcontrollers because the power consumption is prohibitive and the processing time is extensive. This limitation exposes devices to vulnerabilities like a man-in-the-middle attack, spoofing, and unauthorized access. Another inherent problem, however, is the limited memory and storage space of the IoT devices. Such limitations do not allow deploying complete operating systems or operating security

software that could monitor for anomalies, update firmware, or train access control. As a result, a large portion of devices come with out-of-date or unpatched firmware and cannot receive OTA updates, placing them forever at risk of known exploits (Roman, Zhou, & Lopez, 2013).

Security implementation is further complicated by power constraints. Most IoTs are battery-operated or energy-harvesting in a remote or mobile context. Integrating with powerful security mechanisms, including real-time encryption, regular authorization, or continuous spying, could reduce the battery capacity of the device, thereby decreasing the operational life span (Bandyopadhyay & Sen, 2011). To save energy, the manufacturers generally choose static lightweight security protocols or nothing at all. Besides, heterogeneity in IoT platforms contributes to an additional level of complexity. Devices of various vendors can function on various protocols, standards, or architectures, which can complicate the problem of interoperability and security enforcement (Suo et al., 2012). This diversity increasingly leads to fragmented ecosystems where devices cannot be centrally controlled or updated, eventually increasing the risk of a systemic vulnerability. In addition, IoT devices often use such radio communication channels as Wi-Fi, Zigbee, and cellular networks, which are naturally open to eavesdropping, interference, and denial-of-service (DoS) attacks. Data between IoT device transmittals, transmitted in an unsecured channel without proper encryption or a secure key exchange mechanism, is susceptible to being intercepted or even manipulated by the bad guys. The collective impact of these constraints is a wide, vulnerable attack surface over the IoT landscape. The absence of resilient inbuilt mechanisms and unfriendly configurations, combined with underwhelming design parameters, means that many IoT systems are woefully unprepared to cope with even relatively well-organized attacks. The implications of these constraints become more dire as the penetration of IoT devices into critical infrastructure increases, ranging from healthcare to energy grids.

1.4 Common IoT vulnerabilities

1.4.1 Default credentials : Most IoT devices have default usernames and passwords, such as admin: admin, and for the most part, users never even bother to change them. This makes the devices highly vulnerable to outside access and makes them vulnerable to attack (Krebs, 2016).

1.4.2 Lack of Encryption: Most IoT devices do not have proper encryption when transmitting data, making sensitive information open to modification or interception while in transit. These security

vulnerabilities normally come about as a result of manufacturers focusing on reducing costs at the expense of employing robust security measures (AmbiMate, 2023).

1.4.3 Poor Patch Management: IoT devices are also rarely updated with regular firmware, and where this occurs, it is intermittent. Without timely patches and updates, well-known vulnerabilities go unaddressed, and criminals can easily take advantage of them (Goodin, 2022).

1.5 Notable Incident: The Mirai Botnet

The 2016 Mirai botnet attack is a popular instance of IoT hacking.The malware brought about a massive-scale DDoS attack against the DNS hosting provider Dyn using compromised IoT devices with their original passwords still on. Consequently, major online platforms such as Twitter, Netflix, Reddit, and PayPal were found to be adversely affected (Krebs, 2016). Such an incident ended up affirming the sheer requirement of fundamental security requirements for IoT devices.

1.6 Recommendations

To combat these weaknesses, a couple of simple actions will make a big impact:

1.6.1 Credential Management: Default passwords must not be reused for all products. There must be a unique login for each product, and the user must be forced to update it upon their initial login.

1.6.2 Encryption: All information exchanged between networks and devices must be encrypted with existing encryption techniques such as TLS or SSL, to ensure that others cannot see it.

1.6.3 Patch Management: Periodic software updates must be done. IoT devices must be automatically updatable (over-the-air), and thus any security problems are fixed as they occur.

1.6.4 User Awareness: The majority of the threats can be prevented by simply informing the users. Instructing individuals on fundamental security practices, such as renewing passwords or reconfiguring default passwords, can do much to keep systems safe.

1.7 Consequences of IoT Vulnerabilities

If IoT devices are not protected, they can provide an entry point for cyber attacks that can result in huge financial loss, interfere with daily activities, and even destroy public trust. This is not only a big business problem—individual consumers stand to be affected as well. The impact can be widespread, reaching personal information, business resources, and even public safety.

1.7.1 Distributed Denial of Service (DDoS) Attacks: One of the finest examples in the real world is the Mirai botnet attack of 2016. The attack made headlines because hackers used default login

credentials to hijack millions of Internet of Things devices, including security cameras and routers. The hackers hijacked the devices to create a

huge botnet, which they used to launch huge distributed denial-of-service (DDoS) attacks. One of the highest-profile casualties was Dyn, a large DNS service provider, which disrupted services to popular services such as Netflix, Twitter, Reddit, and PayPal. The attack exposed a significant weakness: how simple it is for hackers to gain entry to devices that are left with default usernames and passwords. This attack demonstrated the need for IoT device security and not to underestimate simple security measures, such as default credentials. (Krebs, 2016)

1.7.2 Data Breaches: Unless IoT devices are secured, these devices are susceptible to attacks. The devices handle, transmit, and store a massive amount of corporate and personal information and are therefore an attractive target for hackers. Hackers can easily get sensitive information if there are not enough encryption and authentication controls.

For example, in 2018, IoT IoT-connected heart rate monitor in the gym was hacked, where thousands of members' health data were compromised with poor encryption practices (ZDNet, 2018). Smart home assistants with voice control and baby monitors have also been hacked on their end, and personal conversations and video streams have leaked(Wired, 2020). These failure modes are imperiling privacy, and can imperil user safety as well, and erode trust in the devices.

1.7.3 Manipulation of Smart Systems: If not secured, IoT devices are exploitable to control vital systems. A terrifying demonstration of this occurred in 2021 when a new water treatment facility in Oldsmar, Florida, was hacked. The intruder managed to get into the system and increase the concentration of sodium hydroxide (lye) in the water system to an extremely toxic high value that would have resulted in public health risks. Fortunately, the attack was found and stopped before it could do any damage(CISA, 2021). The attack is an example of how vulnerable the IoT systems in critical infrastructure are to the repercussions of public health and safety.

1.7.4 Financial and Reputational Losses: Apart from technical consequences, companies can suffer from severe financial and reputational losses because of IoT-related incursions. From the IBM Cost of a Data Breach Report (2023), the average cost of an IoT endpoint data breach is more than the cost of system breaches, largely due to the challenges of detection and response. The companies that market IoT solutions fail to protect their customers from potential risks, and while one-time payments and legal consequences are bad enough, the loss of consumer trust can be even more devastating in the long term.

1.8 Risks to Privacy, Public Safety, and Critical Infrastructure

1.8.1 Risks to Privacy: IoT devices present one of the most serious threats in terms of privacy protection of personal data. Numerous gadgets, ranging from devices used in smart homes, including wearables and voice assistants, are always recording users' location, behavior, and health. Under such conditions, without adequate security measures, users can become targets of breaches of data and surveillance. A study from 2019 brought to light that over half, over 60%, of smart home devices were vulnerable to existing security issues, which put the personal data of consumers in danger of unauthorized entry (Baker, 2019). Such flaws also enable cybercriminals to reap gain by selling user data to the dark web, stealing identities, or tracking individuals, causing public confidence in IoT devices to be undermined.

1.8.2 Risks to Public Safety: Flaws with Internet of Things technologies, therefore, could seriously harm public safety. Attackers can target core infrastructure elements such as self-driving cars, grid management systems, and healthcare machinery, with actual and real danger to these devices. By way of illustration, the cyberattack on Ukraine's power infrastructure in 2017 caused massive power disruptions, felt by hospitals, transport networks, emergency services, and electricity delivery in large regions (Scott, 2017). The IoT-based traffic or surveillance systems' compromise in smart cities can lead to disruption of everyday life and public safety threats.

1.8.3 Risks to Critical Infrastructure: As IoT devices are applied in core infrastructures, the exposure is dangerous for national security, and therefore, increased safeguarding is needed immediately. Integration of IoT devices elevates risks of cybersecurity to critical systems, such as transport, electricity supply, and water distribution. An event in 2021 brought the threat to light when hackers tried to poison the public water supply using the chemical treatment at the Oldsmar NPDES water treatment plant in Florida (CISA, 2021). It highlighted the ease with which hackers can seize control of crucial tools for safety when the required safety measures are not present. Critical infrastructure will remain susceptible to coordinated attacks that could have far-reaching effects if we keep using IoT devices without bolstering their security.

1.9 Gaps in Regulation and Standardization

1.9.1 Lack of Consistent Global Security Standards: Being behind in creating worldwide, legally binding guidelines poses a significant problem to IoT security. Some countries have indeed passed legislation, but the absence of a definitive architecture for all IoT devices to comply with a minimum level of security persists. Taking it as an example, the United States has passed the IoT Cybersecurity Improvement Act of 2020, which requires federal agencies to utilize specified IoT security controls when using the devices. On the other hand, the private sector receives an exemption from these provisions by the U. S. Congress (2020). Similarly, the EU Cybersecurity Act establishes standards for digital goods all across Europe, though the global strategy has yet to be written. In the absence of a standardized framework, manufacturers are unable to enforce a uniform security regime across the globe.

1.9.2 Existing Initiatives and Their Limitations: Although several efforts are going on to resolve the security vulnerabilities present in the Internet of Things, they have not succeeded in making a truly secure IoT ecosystem. To protect IoT devices, organizations such as the European Union Agency for Cybersecurity (ENISA) and the IoT Security Foundation (IoTSF) have or are in the process of producing best practices and recommendations. Although these frameworks are set to be best practices, they are not mandatory, constraining them to cause wide-scale sectoral transformation (IoTSF, 2020). Consumer Technology Association (CTA) has rolled out a few IoT security guidelines, which can differ from location to location and have limited enforcement. To yield meaningful results.

These guidelines need to flex and have better faith based on regulation as Internet of Things devices become an integral part of our lives.

1.9.3 Challenges in Enforcement: As a consequence of the dramatic growth of the market and the numerous manufacturers involved, the implementation of a common security standard for IoT devices has become much harder. Numerous smaller or mid-size manufacturers favour the minimisation of costs, but make systems inherently insecure in the process. The begets of the fragmented market and the rapid deployment of many untested devices have left the U.S. Federal Trade Commission (FTC) pointing to the challenges of regulatory control of IoT security (FTC,

2018). Lacking strong and enforceable security requirements, it is likely that inadequately secured devices will spill over onto the market.

The benefits of the advancement in IoT have not been all smooth, with increased security risks that can be exploited by cybercriminals. Leveraging the weaknesses of IoT devices and networks, the threats such devices and networks pose, and lastly suggesting ways to offset such weak points, this essay aims to improve IoT system security.

The paper looks into how common issues of IoT devices, such as weak password application, insufficient encryption, and lack of patch management, can compromise critical infrastructure, personal privacy, and shared security. However, in the paper, it is analyzed how the international standardization gaps and legislative obstacles make it harder to ensure the security of the systems of IoT. The research is intended to provide comprehensive recommendations that would help enhance IoT security and reduce the likelihood of falling victim to cyber threats by noting these weaknesses.

1.10 Devices Used for Experimentation

1.10.1 Wipro Smart Plug: The Wipro Smart Plug, with the help of a smartphone app, voice assistants such as Alexa and Google Assistant, enables users to gain control and awareness of their appliance use. The plug

provides features to monitor energy consumption, program schedules, and remotely control various typical appliances such as chargers and heaters, and lights. The more convenience, the potential for a risk of security. Because it can be compared to another smart device, it can become a target for hackers without the protection of secure communication protocols that are secured, common firmware updates, and correct authentication procedures. Studies have revealed that inadequately protected smart plugs may likely make hackers the owners of devices or provide hackers with access to a user's routines (Lee et al., 2020).

1.10.2 Philips Smart Bulb: A part of the popular Philips Hue range, the Philips Smart Bulb can be controlled from afar via voice assistants or smartphone apps because of its Wi-Fi or Zigbee capabilities. It makes it easy and saves energy at home because users can manually control colors, brightness, and even set routines. However, established companies have no walls against security

vulnerabilities. When smart bulbs are connected to home hubs, there is likely to be network vulnerabilities. Open firmware and Zigbee protocol vulnerabilities create the threat that the devices can induce unauthorized access to the primary home network (Ronen & Shamir, 2016).

1.10.3 AnyCast Wireless Casting Device: Using the economical and user-friendly AnyCast casting gear, users can wirelessly stream or reflect content from computers, tablets, or phones onto a TV or projector. AnyCast can also reach out to both Android and iOS devices as AnyCast can sync with major streaming standards like Miracast, DLNA, and AirPlay. AnyCast is not inclusive in using pricier devices such as Apple TV or Google Chromecast, as it has a cost-effective and user-friendly alternative.

AnyCast is susceptible to network attacks, including spoofing attacks, unauthorized screen sharing, and eavesdropping, because of its lax security measures. According to reports by specialists, these devices can indicate open services or enable their easy availability in local networks, especially in the default settings (Lee et al., 2020). Consequently, these devices can become the weak links in a home network to allow hackers the means to get access to other devices or sensitive network details.

1.10.4 The Amazon Echo Dot Speaker: Echo Dot is a smart speaker that works through the voice of Amazon's virtual assistant, Alexa. It acts as a hub of the smart home world, allowing users to control connected devices, play music, set reminders, and get live information using voice instructions. The Echo Dot is continually listening out for a wake word and transmitting recorded audio to Amazon's cloud-based processing services. Although useful, such functionality can be associated with the issue of continuous passive listening, privacy of information, and cloud-related vulnerabilities (Iqbal et al., 2020). The device can be networked through Wi-Fi, and it can interface with a myriad of third-party IoT platforms, with this enabling the expansion of the attack surface for the device.

1.10.5 Wipro Smart Bulb: Wipro Smart Bulb is a remote-controlled LED bulb that saves electricity, which can be operated using a mobile application or through voice commands in applications like Alexa and Google Assistant. It has a Wi-Fi connection and customizable brightness, colour temperature, and scheduling options. With these, Wipro bulb is a good example

of many of the challenges related to the security in the secure environment of smart homes – those being insecure APIs, lack of firmware updates, and unencrypted communication protocols (Abomhara & Koien, 2015). Although it may look like harmless stuff, compromised smart lighting systems can be used as a point of access into broader critical networks.

1.10.6 Fire-Boltt Oracle Smartwatch: Fire-Boltt Oracle is a rich-featured smartwatch that targets fitness freaks and tech-resident users. It generally provides a vibrant AMOLED screen, Bluetooth calling, and a package of health monitoring features, including heart rate monitoring, SpO_2 monitoring, and sleep analysis. The watch has various sports modes, smart notifications, music control, and comes with weather updates. Its sleek design and interface make it a fashionable piece of wear for everyday use.

1.10.7 Xiaomi Mi Wireless Home Security Camera 2i: Xiaomi Mi Wireless Home Security Camera 2i is a small, low-cost indoor security solution. Its characteristics include a 360-degree rotating lens that captures full coverage of the room, 1080p video, night vision using infrared light, and motion sensing based on artificial intelligence. The camera has two-way audio communication – possible to use in real-time, and possesses storage in the cloud and in local formats. It works perfectly with the Mi Home app to allow remote control.

1.10.8 Sony BRAVIA 4K UR3 Smart TV: The Sony BRAVIA 4K UR3 Smart TV model KD-55X8000H enhances in-home entertainment and is smart with Android TV OS version 10. With Android TV ecosystem connectivity and Ultra HD 4K quality, clients can browse hundreds of apps directly from the Google Play Store, and also use voice control via Google Assistant. The gadget integrates well with several streaming services, from Netflix to YouTube and Prime Video. It has a user-friendly interface that guarantees it offers a smooth multimedia experience. The version of the system kernel is kept at 4.19.75. It is the security patch level, dated 1 December 2024, that is the evidence that TV is regularly improved for some speed and for some security.

1.10.9 BPL ULTRA HD 4K Smart TV: Made to make the TV energy efficient and provide high definition in vision output, Ultra HD 4K Smart TV, model 55UL-E7300/2024, is a 133 cm smart TV (approximately 55”). It consumes 225 kWh annually according to its Bureau of Energy Efficiency (BEE) label, and its label shows that it uses moderate power for its capabilities, plus its

screen size. It has Dolby Audio support so it can provide a better multimedia experience with more engrossing sound quality. Since BPL smart TV is part of a new generation of smart TVs, smart connectivity features are likely to support apps, cast media, and control functions remotely.

1.10.10 HIKVISION 2MP COLORVU MD 2.0 Camera (IPC-T229HA(2.8MM)): The 2-megapixel surveillance device created for intensive use in low-light settings, Hikvision ColorVu MD 2.0 Fixed Turret Network Camera, model IPC-T229HA(2.8mm), has been designed to provide full-color video monitoring. The camera includes built-in ColorVu technology so that it can deliver bright imaging for the users even at night. This makes the camera good for 24/7 security operations. Camera supports network connection, and it also enables remote video retrieval via mobile apps or desktop applications. This aspect allows real-time surveillance, including playback.

REVIEW OF LITERATURE

2. REVIEW OF LITERATURE

2.1 A vulnerability detection method for IoT protocol based on a parallel fuzzy algorithm - Han et al

Han et al. (2024) develop an extended method for vulnerability detection in IoT communication protocols based on the parallel fuzzy testing algorithm optimized by genetic means. The research underpins power distribution IoT systems and formalizes communication protocols to increase vulnerability analysis. The method achieves high test accuracy and efficiency by preprocessing test samples and incorporating genetic algorithm techniques, e.g., Gray code for encoding, fitness functions involving attack features, as well as adaptive genetic operations. The authors show how this method cuts down on both missed and false detections, making room for greater coverage of protocol vulnerability.

Experimental results demonstrate performance of the proposed method is better than existing methods, with a missed detection rate of less than 4% and consistently low false detection rates. The technique was verified on the CIC IoTQ dataset and proved itself in a simulated IoT environment, which indicates its practical efficiency. Conclusively, the paper states that the fusion of fuzzy logic and genetic algorithms is a powerful enhancement to vulnerability detection in the distributed IoT systems, and provides a robust paradigm to improve security of the power networks' communication protocol (Han et al, 2024).

2.2 A Survey of Security Vulnerability Analysis, Discovery, Detection, and Mitigation on IoT Devices - Yu et al

Yu et al. (2020) describe the state-of-the-art systematic review on the security vulnerabilities of IoT devices, and state that the attack surface is increasingly growing because of problems with hardware, firmware, communication protocols, and user interfaces. Public vulnerabilities range from buffer overflows, injection flaws to weak authentication, and misconfigured services. The authors characterize threats according to users of architectural layers: hardware, firmware, and protocol interfaces, and, for example, highlight risks of insecure debugging ports, uncovered flash memory, and unsafe remote management interfaces.

In the course of the research, three stages of vulnerability assessment are described: the cohabitation influence. discovery, detection, and mitigation. It compares static tools (e.g., symbolic execution, taint analysis), and dynamic approaches (e.g, fuzzing, emulation), referring

to tools such as Firmadyne, IoTFuzzer, Avatar2. In mitigation, it mentions automated patching and access control systems that include: FlowFence, ContexIoT, and Tyche. The authors argue the need for AI-based analysis, platform testing, and trustworthy firmware analysis to control evolving IoT threats.

2.3 Security Vulnerability Analysis for IoT Devices Raspberry Pi using PENTEST - Arreaga et al

Arreaga et al. (2023) perform realistic testing of the IoT weaknesses in the form of a penetration test on a Raspberry Pi with the installed operating system Raspbian. The study shows that IoT devices are often devoid of strict security measures, hence being mostly accessed through attacks like Man-in-the-Middle (MITM), Backdoor, and Denial-of-Service (DoS). As compared to a Windows 10 virtual machine, the Raspberry Pi was relatively easy to compromise due to a lack of built-in antivirus and intrusion detection systems. Attacks of such kinds as MITM, Backdoor took a few seconds to attack, while delays by DoS attacks were severe in server responses.

The tools used for simulation of the attacks were open-source ones such as Ettercap, Metasploit, and Wireshark. Their results point to typical IoT vulnerabilities, including default credentials, unencrypted connections, and accessible services. To prevent these risks, they suggest securing devices from modification of the default settings, maintaining software up-to-date, and applying encryption and firewall protections. The research necessitates the more immediate need to secure the IoT devices, especially if used in sensitive or public-facing deployments.

2.4 A novel approach for detecting vulnerable IoT devices connected behind a home NAT - Meidan et al

Meidan et al. (2020) present a new machine learning architecture to detect vulnerable IoT devices that are behind a home network address translation (NAT). Realizing that viable Denial-of-Service attack devices in smart homes can be exploited for large-scale DDoS attacks, the authors present a flow-based detection technique that is nonintrusive and avoids deep packet inspection. Using NetFlow information and LightGBM classifiers, trained on features related to IoT traffic, such as flow duration and protocol usage, their approach finds concrete vulnerable IoT models without accessing packet-level information, which makes it privacy-preserving and at scale.

The work shows that the method is effective, even in an adverse environment such as encrypted or non-TCP traffic, and provides a significant advance compared to existing approaches, especially identification of the device models hiding behind NAT. The system allows telecommunication providers to perform preventive surveillance and tackle threats from compromised consumer IoT devices. A labelled set of traffic from commercial IoT devices was also created and distributed for continued research in the area (Meidan et al., 2020).

2.5 Ethical hacking for IoT: Security issues, challenges, solutions, and recommendations - Yaacoub et al

Yaacoub et al. (2023) examine the role played by ethical hacking in securing the Internet of Things (IoT), focusing on the rising number of IoT devices and increasing cyber threats. This paper introduces different types of attackers, their motives, and the type of vulnerabilities that exist in IoT systems, including those that exist in devices, firmware, networks, and applications. It is proposed that ethical hacking and penetration tests are effective methods of determining and reducing these vulnerabilities. According to the authors, securing resource-constrained IoT devices poses distinct challenges, which require regular testing at three levels: devices, infrastructure, and users. Catching security flaws ahead of the pack should be the priority.

The research gives an all-around view of cyberattacks on different domains of IoT, particularly the medical, industrial, military, agricultural, and financial domains. Tools, techniques, and case studies are discussed, along with revealing weak points in the IoT ecosystems via simulated attacks. Moreover, it establishes the need for a structured framework that will combine ethical hacking with vulnerability tests and security best practices. Practical recommendations and the importance of cybersecurity awareness and training among IT staff to create resilience against emerging, increasingly dangerous IoT-based threats close the paper (Yaacoub et al., 2023).

2.6 Security and Privacy Issues in the Internet of Things - Sicari et al

Sicari et al (2015) conducted a broad view of the security and privacy threats in the Internet of Things (IoT), exposing the threats due to interconnection and heterogeneity of IoT devices. The authors describe essential security requirements, including confidentiality, integrity, authentication, and availability, and point out the necessity of secure management of data and

communication protocols. They study typical threats along a variety of layers of an IoT architecture, such as physical attacks on devices, intrusions at the network level, and attacks on the application level, emphasizing that conventional security mechanisms typically fail to suffice in resource-restricted IoT environments.

The paper also surveys different privacy-preserving mechanisms, studying the boundaries of the existing cryptographic directions and the significance of trust and reputation models. The authors argue in favor of such an approach to security that implies the implementation of security and privacy as design attributes of IoT systems during the development phase. They provide a research agenda which includes lightweight encryption, context-aware access control and a decentralized trust framework to better protect user data and device integrity. On balance, the paper is an excellent starting point for an understanding and addressing of the complex security scenario of the IoT ecosystem (Sicari et al., 2015).

2.7 A Survey on Internet of Things: Architecture, Enabling Technologies, Security and Privacy - Alaba et al

A detailed survey of the Internet of Things (IoT) has been given by Alaba et al. (2017), describing its architecture, enabling technologies, and, in particular, its security and privacy challenges. The paper describes the multi-layered IoT architecture, composed of the perception, network, and application layers, and the technologies supporting IoT, namely in the form of RFID, wireless sensor networks, cloud computing, and big data analytics. The authors note that the variability and size of IoT systems bring much complexity to providing secure communication and authentication of devices, as well as the security of data.

Using the survey, important security issues are determined, including data confidentiality, integrity, and availability, as well as privacy threats involving unauthorized data collection and tracking. Alaba et al. explore different existing means to tackle issues like encryption, access control, and IDS systems, and point out their issues in IoT devices' tight-resource settings. The authors end the discussion by emphasizing the need for lightweight security frameworks as well as strong privacy policies and interdisciplinary research to build more resilient and safer IoT ecosystems (Alaba et al., 2017).

2.8 Understanding the Security Challenges in Smart Home IoT Devices - Acar et al

Acar et al. (2018) investigate the issues of security threats of smart home devices in the IoT context, and they are interested in real-world vulnerabilities of consumer-grade products. The authors investigate ways in which insecure firmware, insufficient encryption, incorrect authentication systems, and uneven software updates leave smart devices vulnerable to critical security threats. They point out that users many times do not have the technical knowledge to configure or monitor device security, which point of further increase risks. The study takes a look at a variety of commercially available devices and discovers conventional problems such as hardcoded credentials, unprotected communications, and lazy or nonexistent access control.

The paper also emphasizes privacy and network security implications of these vulnerabilities. According to Acar et al., vendors must embrace tighter security-by-design values and regular security updates. They promote regulatory standards on a baseline for protection across IoT devices as well as improvements in consumer transparency for the security posture of smart home products. The study concludes that robust security in smart homes will need manufacturers to join hands with researchers and policymakers (Acar et al., 2018).

2.9 Securing the Internet of Things: A Standardization Perspective - Weber, R. H

The security and privacy threats presented by the fast-growing Internet of Things (IoT) are explored in Weber (2010) from the perspective of legislation and standardization. The paper presents the heterogeneous and shattered topography of IoT technologies, which is a challenge for exerting efforts to ensure interoperability and coherent security measures. According to Weber, traditional IT security mechanisms cannot accommodate IoT because the devices are characterized by heterogeneity, limited resources, and on a large scale. The author names key threats, namely unauthorized access to data, misuse of personal information, and use of connected devices for malicious purposes.

The gist of the argument is the urgent necessity for the standardization work of international character, which should not only cover technical specifications but also legal frameworks and privacy matters. Weber suggests that efficient regulation should strike a balance between innovation and trust from the users by having privacy and security by design developed into the

IoT construction. The paper ends by supporting a collaborative process between industry, policy makers, and standards organizations to develop uniform rules and protocols, to make IoT systems secure, transparent, and legally compliant (Weber, 2010).

2.10 A Study on Vulnerabilities and Threats to Wearable Devices - Blow et al

Blow et al (2020) discuss security and privacy risks inherent in wearable devices that are rapidly becoming popular in health, fitness, and communication applications. The following critical vulnerabilities are identified in the paper: insecure wireless communication (e.g., Bluetooth and Wi-Fi), weak authentication protocols, and lack of encryption. Such vulnerabilities make users vulnerable to such dangers as eavesdropping, data modification, tracking location, and exposure of their personal information to unwanted persons. The authors highlight that limited processing power and battery life in wearables (among other resources) make it hard to deploy traditional security (among other measures) smoothly.

The study also measures the wider impact of wearable device insecurity in terms of trust of the user and data integrity. Blow et al. state that the absence of strong privacy controls and transparency on the part of data handling practices may deter consumers from using the solutions. To solve them, the paper suggests lightweight encryption algorithms, secure firmware updates, and user-centric access control mechanisms. It ends by emphasizing the requirement of industry-wide standards and regulatory systems focusing on device performance, as well as security (Blow et al., 2020).

2.11 Analysis on Security Vulnerabilities of Medical Wearable Devices (Fitness Trackers) - Saleh et al

Saleh et al (2020) study the vulnerabilities in medical wearable devices, especially fitness trackers, that are growing in their use for real-time health monitoring and collection of data. Although these devices provide convenience and better patient care, they come with exposures to risks of significant risks involving the exposure of sensitive healthcare data. The authors point out such problems as authentication vulnerabilities, sharing of user data with third parties, and poor storage methods of which might result in user data breaches. One of the serious issues is the absence of encryption and secure communication processes that make user information vulnerable to

interception when it comes to such attacks as man-in-the-middle (MitM). The paper suggests employing blockchain technology as a secure, decentralized solution for health data storage and transfer, with the decline in the threat of centralized data breaches.

The study further examines the wider aspects of user awareness and the duty of a manufacturer as commenting that most users are ignorant of the threats that are posed by wearable fitness gadgets. Applications that are used to track fitness may inadvertently cause data leaks resulting from lapses in security in the cloud or insecure Bluetooth connections. Some case studies that have been conducted on Fitbit devices indicate how it could be possible for attackers to exploit system vulnerabilities and extract or ransom personal data. The results from the surveys provided in the paper provide evidence of the unawareness of the public on risks, further showing the need to implement better standards of security. According to Saleh, Brennan, and Verma, the authors propose the incorporation of effective mechanisms of data protection, secure practices of communication, and improved regulation to protect wearable medical technologies (Saleh et al., 2020).

2.12 A Study of Threats, Vulnerabilities and Countermeasures: An IoT Perspective - Choudhary et al

Choudhary et al (2021) discuss the primary threats and vulnerabilities affecting Internet of Things (IoT) devices, and threats include weak authentication, insecure network services, outdated components, as well as poor update mechanisms, among others. The paper describes vulnerabilities throughout the IoT architecture, sensor, network, edge, and cloud, and the expansion of the attack surface owing to the permeation of devices. Practical examples of such real-world incidents as intrusions into Ring cameras and Tesla controls show what dangers of poor IoT security can look like.

To overcome these issues, the authors suggest the security-by-design approach based on industry standards like OWASP, GSMA, and the Cloud Security Alliance. Suggested countermeasures are the employment of strong passwords, encryption, secure firmware updates, and minimizing administrative rights. The paper also supports the usage of security tools such as AWS IoT Defender and Cisco IoT solutions. Choudhary et al. promote the fact that there should be constant improvement, awareness, and worldwide standards to enable safe and effective IoT deployment.

2.13 Unmasking vulnerabilities by a pioneering approach to securing smart IoT cameras through threat surface analysis and dynamic metrics - Bhardwaj et al

Bhardwaj et al. (2024) suggest a new security model for smart IoT cameras, paying attention to threat surface analysis and dynamic parameters that may prevent device-level vulnerabilities. The research defines the threat surface of the IoT as the overall exposure of devices to risks for hardware, OS, libraries, and network layers. In contrast with the traditional models that are concerned with external threats, the framework focuses on internal and interconnected threats in home networks. It entails device categories identification, exposure indicators determination, and calculation of threat pathways in designing proactive security.

The methodology employs ecosystem mapping and identifies threat vectors and formulas for calculating Threat Surface Areas (TSAs) and Threat Scores with the help of the CVSS system for severity measurement. Experiments on smart webcams are used to validate the framework and prove that vulnerabilities are reduced in the post-implementation phase. The paper also describes upcoming research priorities such as AI-threatened prediction, secure protocols, as well as blockchain storage. Overall, the work should help to direct stakeholders in securing IoT cameras through limiting the digital footprints and strengthening the proactive mitigation of risks (Bhardwaj et al., 2024).

2.14 Vulnerabilities and Challenges on Cybersecurity and Artificial Intelligence Based on Internet of Things - a Comprehensive Study - Alanisi, S., & Mohamed, A.

According to the study by Alanisi and Mohamed (2021), the analysis of the vulnerabilities and cybersecurity risks of the Internet of Things (IoT), as well as the role of Artificial Intelligence (AI) in strengthening or degrading system security provided herein, was comprehensive. Key IoT threats are classified by the authors as hardware-based attacks, network attacks, and software vulnerabilities, according to which there is a significant exposure of interconnected devices to a huge attack surface. They also discuss AI's two-sided effect – how machine learning models can find anomalies and improve intrusion detection systems, while also creating a new set of threats, including adversarial attacks, data poisoning, etc.

Otherwise, the research puts a lot of emphasis on the complexity of the security of the Internet of Things systems because of their heterogeneity and scalability problems, and computational limitations. It includes the absence of standard protocols for implementing security and the difficulty of ensuring the privacy of data in varied IoT settings. Alanisi and Mohamed recommend integrating the AI-based security systems with lightweight encryption, real-time detection of threats, and the collaborative sharing of intelligence between the IoT nodes. Their work lays Notice for balancing innovation in AI & IoT while implementing proactive cybersecurity measures to counter future threats.

2.15 oT Device Identification Using Unsupervised Machine Learning -Koball et al

Koball et al. (2023) address the crucial problem, which is the accurate determination of devices in the Internet of Things (IoT) systems, a prerequisite to guaranteeing the security of the network and effective control of its access. Old school identification practices, like the use of Media Access Control (MAC) addresses, are vulnerable to spoofing and therefore poor for stateful security. The authors put forward an unsupervised machine learning approach that utilizes dynamic actions of the devices, especially the patterns of the network traffic employed, to detect the IoT devices without the use of labeled datasets. Such an approach increases flexibility and scalability, particularly in dynamic IoT environments with regular joining and leaving of devices to and from the IoT.

The proposed framework has a multi-step procedure. Operations of data preprocessing, outlier removal, feature selection, and clustering building up to device prediction accordingly to the K-Nearest Neighbors (KNN) classifier. Through comparing both supervised and unsupervised machine learning techniques with the same dataset, the research shows that methods of an unsupervised nature can produce similar accuracy, absent the need for vast amounts of labeled data. This research highlights the capability of unsupervised learning for building scalable and adaptable IoT security solutions that do not require manual labeling of data and provide real-time detection of unauthorized or rogue devices.

METHODOLOGY

3. METHODOLOGY

This research aims to discover, analyze, and assess security threats and vulnerabilities in the most widely used Internet of Things (IoT) devices by performing practical examinations via network scanning, traffic analysis, wireless attack, and reverse engineering of relevant applications. The research aims to identify understated risks of IoT ecosystems and advance feasible mitigation measures that could make the security and reliability of the IoT deployment more pervasive.

Objectives:

- To identify and understand the working of IoT
- To scan and identify threats and vulnerabilities in IoT
- To list out the vulnerabilities identified in the devices
- To recommend secure practices for the devices

We took a practical approach to reveal the risks and weaknesses of the IoT devices. We used an organized approach with a total of 12 IoT devices. Device selection, information gathering, network analysis, traffic analysis, Wi-Fi attacks, app security, and firmware, amongst others, are the seven major steps of this process. The aim is to assess and determine the weaknesses and vulnerabilities of practical IoT settings.

3.1 Device Selection and Information Gathering

- Wipro smart plug
- Philips smart bulb
- Anycast casting device
- Hikvision Cctv camera
- BPL smart TV
- Sony smart TV
- The Amazon Smart Speaker- Echo Dot
- Wipro Smart Bulb
- FireBoltt Oracle smart watch
- Xiaomi 360 home security camera 1080p 2i

The manufacturer, model, version, ID, connection, features, communication protocols, and MAC address of every device were all collected by manual and online inspection. Understanding each device's functions and architecture was the primary goal.

3.2 Network Scanning (Using Nmap)

With the help of the Nmap program, a complete network scan was performed to locate hosts and determine whether any open ports or services were found.

- Basic port scan: To locate open TCP/UDP ports.
- Common Ports Scan: Search for well-known services and ports that were targeted.
- Service and Version Detection (-sV): Identified which services were active and the version they were.
- Vulnerability Scanning: In search of the existing vulnerabilities (CVE-based), Nmap scripts (–script vuln) were applied.
- IoT protocol detection: Detection of common protocols such as MQTT, UPnP, and CoAP.
- Firewall/IDS/IPS Checks: Attempted to detect active firewalls or intrusion prevention systems.
- Aggressive Scan (-A): A traceroute, script, OS detection, & Version detection combined.

3.3 Traffic Analysis (Using Wireshark)

Wireshark was employed to record and analyze in-situ traffic from every device so that: Occupy cloud servers or companion apps, as well as track communications.

- Recognise unencrypted data transmissions.
- Identify when insecure protocols (HTTP and Telnet) are being used.
- Track messages transmitted in broadcast and DNS requests.
- Look at the techniques of authentication, handshakes, and session creation.

3.4 Wireless and Protocol-Based Attacks (Using Aircrack-ng)

Aircrack-ng was used to evaluate the wireless security of IoT devices.

- Tests of the strength of the password consisted of dictionary attacks and recording of WPA/WPA2/WPA3 handshakes.
- Attempting deauthentication attacks revealed the kind of response received from forced disconnections.
- Identified weak wireless settings reader the resilience of the encryption.
- Contributed to the assessment of actual vulnerability to prevalent Wi-Fi-based attack methods.

3.5 Reverse engineering (using JADX GUI tool) - Exclusive for smart plug

- Android APK files were decompiled into readable Java source code using JAD-GUI.
- Accessed using officially downloaded APKs obtained from official sources or device access from IoT companion apps.
- checked source code for insecure logic, unencrypted communication, and hardcoded credentials.
- Two such vulnerabilities were discovered: incorrect authentication and data leakage.
- Security vulernabilities in IoT ecosystems at the application layer were assessed.
- adds to a genuine understanding of the apps' interaction with their respective networks/ devices at a deeper level.

3.6 Firmware Acquisition

- Attempted to obtain IoT device firmware to examine embedded vulnerabilities in greater detail.
- Investigated techniques such as extracting through device interfaces, obtaining from official manufacturer websites, and keeping an eye on OTA updates.
- Used tools and methods to dump firmware, but ran with limitations due to restricted access and proprietary or encrypted formats.
- Major drawbacks were the inability to conduct a thorough firmware analysis because of these limitations.

3.7 Documentation and Mitigation Reporting

- Identified and documented existing threats and vulnerabilities without making configuration changes to the devices.
- Findings were written based solely on observation, analysis, and testing methods.
- Noted potential risks and suggested mitigations hypothetically, based on detected issues and industry best practices.

RESULT AND ANALYSIS

4. RESULT AND ANALYSIS

4.1 Wipro Smart Plug

A) Device information:

- Device name: Wipro Smart Plug
- Device model: DSP1160
- Manufacturer: Wipro
- Voltage: 240 VAC, 50Hz
- MAC address: 40:f5:20:f7:b8:ca
- Connectivity: 16Amps , Wi-fi 2.4GHz
- Firmware: Proprietary, OTA updates supported via Tuya cloud
- Mobile app: Wipro Smart Home App
- Chipset: ESP8266

B) Network Scanning (Using Nmap):

Nmap scanning for open ports and services was conducted.

a) Basic port scan

Command: sudo nmap -Pn -sS -p 1-100 192.168.1.23

The scan mapped out that all 100 ports were ignored or blocked, meaning that incoming traffic to these ports was being limited, most likely through a firewall/Intrusion Prevention system.

b) Common service ports

Command: sudo nmap -Pn -sS -p 80,443,21,22,23,25 192.168.1.23

When the ports such as 80 (HTTP), 443 (HTTPS), 21 (FTP), 22 (SSH), 23 (Telnet), and 25 (SMTP) were scanned, a filtered state was logged. Remarkably, the device seemed to be doing what it was supposed to do, but was also filtering/blocking incoming connections actively, the purpose probably being the improvement of security against unwanted access.

c) Version scan detection

Command: sudo nmap -Pn -sS -sV -p 1-65,535 192.168.1.23

The only notable detection was on Avahi (local network service), but all other TCP port results concluded that the ports were filtered and had no major vulnerabilities.

Running in-depth scans in the port covering between ranges 6000–65000 and 500–1000 also indicated that the Avahi service may This finding suggested the probability that the mDNS discovery process could be susceptible to an exploitable vulnerability under some conditions. However, practical exploitation proved hard due to constraints in the ports, even in the attempt at exploitation.

d) Scan for Known Vulnerabilities

Command: sudo nmap -Pn -sS –script=vuln -p 1-65535 192.168.1.23

After the initial scan, Avahi-dos showed that hosts were running but were not currently at a high risk. We did not discover any vulnerabilities among the first 100 filtered TCP ports.

A further, detailed scan of TCP ports between 6000 and 65535, as well as 1 and 500, exposed two particular vulnerabilities in the Avahi service. The results recommended that Avahi, a discovery service for local networks, might be subject to a DoS attack. This means that under specified conditions, multicast DNS traffic created intentionally could disrupt the device locally.

Despite our analysis, the previously noticed ports were either disabled or blocked upon repeat scans, which made the process of exploitation much harder or even unattainable. Changes in service profiles, short-term openings during initialization, or quick responses by the device to plug-detected vulnerabilities may all explain this outcome.

e) Checking the firewall or security systems

Command: sudo nmap -Pn -sA -p 1-65000,535 192.168.1.23

The fact that filtering of each probe port is the response proved that the deployed device used active packet filtering, which is usually offered by an integrated firewall or access control

mechanisms. The non-responses from these scans significantly reduce the device's detectability, making it much harder to be subjected to automated attacks, which itself is evidence that security safeguards are being put in place.

f) Common IoT protocol scan

Command: sudo nmap -Pn -p 1883,5353,8080,10000 --script=mqtt-subscribe,upnp-info 192.168.1.23

To see if IoT devices are using standard protocols, such as MQTT, mDNS, HTTP-ALT, and Web-based administration ports. The data indicated, all of these TCP ports were filtered drastically limiting attack surface at the protocol level and meaning that these channels either were not enabled or blocked by firewalls or the devices themselves All these TCP ports were found blocked hence greatly reducing the risk at protocol level of a TCP connection attack and hence these services were not in use or purposely cut by firewalls or underlying devices.

g) Aggressive scan

Command: sudo nmap -Pn -A -T4 192.168.1.23

It was found that AMANDAIDX was running on the 10082/TCP through the scan. The device could be either a Nokia IPSO 4.x firewall or a Netgear CM500 cable modem, common in broadband/ enterprise setups that came across as a part of the Nmap scan.

3 Network hops were identified by the traceroute function of the scan. The transfer of communications from the probe machine was likely to go through the gateway, which was a D-Link router connected at 192.168.0.1, the first hop in the traceroute. And although the majority of the services were not exposed, the successful hop path verified that the target device was reachable externally.

Running sudo nmap -Pn -sV -p 10082 --script vuln192.168.1.23, it was verified that the identified service was safe, as there were no known vulnerabilities associated with it.

C) Traffic Analysis using Wireshark

- Interface: Wireshark was set in WLAN0 (home Wi-Fi) to watch traffic from the smart plug.
- Filters applied: To monitor HTTP and HTTPS connections, frequently related to IoT devices, filters were configured on IP address 192.168.1.23 and TCP ports 80 and 44.
- Objective: The goal is to study the device communication and use of traditional web protocols of the device when it interacts with cloud and mobile services.
- Findings: No relevant packets were captured during the session.
- Reason: This may be because of an active firewall or some other network security measures that are blocking the traffic or encrypting it.

The Smart Plug is built with security aspects to avoid passive eavesdropping, but more experiments under lab conditions might better explain its capabilities.

D) Wi-Fi attacks using Aircrack-ng

- Activation of Monitor Mode: Relied on airmon-ng to reconfigure wlan0 as the monitor interface wlan0mon to capture packets.
- Network Scanning: Used airodump-ng to identify the Smart Plug's Wi-Fi network, pick out its BSSID and the channel (Channel 10, BSSID: a0:ab:1b:e7:27:6e).
- Handshake Capture: The WPA2 handshake was captured right after using Aireplay-ng with a deauthentication attack on the Smart Plug's network.
- Cracking Attempt: Attempted to crack the password with a wordlist called Rockyou.txt, but was unable to crack the password.

The Smart Plug's Wi-Fi network has a safe password that does not belong in common wordlists, which are easily used by dictionary-based attacks.

E) Reverse engineering using JADX GUI tool (Exclusive)

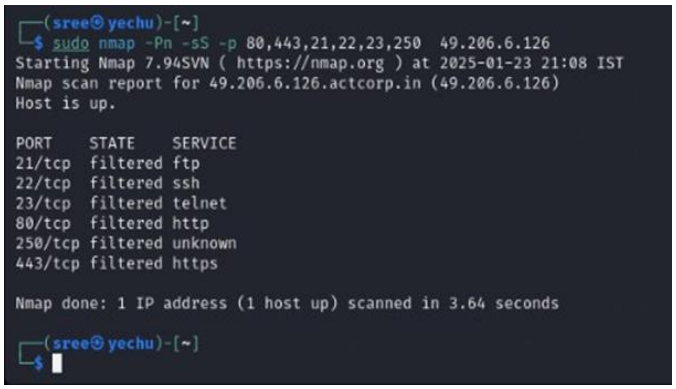
- At the outset, the APK version of the Wipro Smart Home app was downloaded.
- After that, the system had the JADX GUI tool launched for use.
- By navigating to "File → Open File" and picking the downloaded APK, the file was loaded into JADX GUI.

- Following the loading process, JADX attempted to convert the APK into clear Java source code.
- To evaluate security-related behaviors, functions, API usage, and package structures were explored in the app code.
- Most aspects of the code, however, were significantly obfuscated.
- Due to obfuscation of the code, an attempt to decompile the application was unsuccessful. Thus, analyzing the application for potential vulnerabilities became very difficult. Code obfuscation is an accepted security measure that aims to block reverse engineering attempts.

F) Firmware Acquisition

- Looked for downloadable firmware images for the Wipro Smart Plug on both official as well as unofficial websites.
- Also attempted to capture Over-The-Air (OTA) firmware updates using network traffic analysis tools.
- Tried extracting the firmware using unpacking tools to examine file structure and contents.
- As a result, we were unable to obtain or download a firmware image throughout the investigation.
- Because data was not available, in-depth analysis of the firmware was impossible.

ANALYSIS



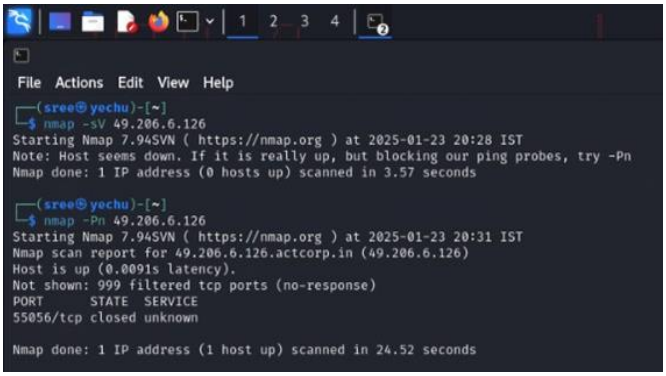
```
┌──(sree㉿yechu)-[~]
└─$ sudo nmap -Pn -sS -p 80,443,21,22,23,250  49.206.6.126
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-01-23 21:08 IST
Nmap scan report for 49.206.6.126.actcorp.in (49.206.6.126)
Host is up.

PORT    STATE    SERVICE
21/tcp  filtered ftp
22/tcp  filtered ssh
23/tcp  filtered telnet
80/tcp  filtered http
250/tcp filtered unknown
443/tcp filtered https

Nmap done: 1 IP address (1 host up) scanned in 3.64 seconds

┌──(sree㉿yechu)-[~]
└─$
```

Fig 4.1.1 Scan for open ports

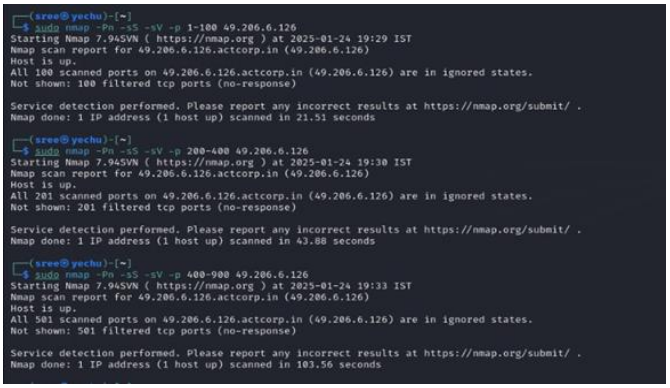


```
File  Actions  Edit  View  Help
┌──(sree㉿yechu)-[~]
└─$ nmap -sV 49.206.6.126
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-01-23 20:28 IST
Note: Host seems down. If it is really up, but blocking our ping probes, try -Pn
Nmap done: 1 IP address (0 hosts up) scanned in 3.57 seconds

┌──(sree㉿yechu)-[~]
└─$ nmap -Pn 49.206.6.126
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-01-23 20:31 IST
Nmap scan report for 49.206.6.126.actcorp.in (49.206.6.126)
Host is up (0.0091s latency).
Not shown: 999 filtered tcp ports (no-response)
PORT      STATE  SERVICE
55056/tcp closed unknown

Nmap done: 1 IP address (1 host up) scanned in 24.52 seconds
```

Fig 4.1.2 Nmap basic scan



```
┌──(sree㉿yechu)-[~]
└─$ sudo nmap -Pn -sS -sV -p 1-100 49.206.6.126
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-01-24 19:29 IST
Nmap scan report for 49.206.6.126.actcorp.in (49.206.6.126)
Host is up.
All 100 scanned ports on 49.206.6.126.actcorp.in (49.206.6.126) are in ignored states.
Not shown: 100 filtered tcp ports (no-response)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 21.51 seconds

┌──(sree㉿yechu)-[~]
└─$ sudo nmap -Pn -sS -sV -p 200-400 49.206.6.126
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-01-24 19:30 IST
Nmap scan report for 49.206.6.126.actcorp.in (49.206.6.126)
Host is up.
All 201 scanned ports on 49.206.6.126.actcorp.in (49.206.6.126) are in ignored states.
Not shown: 201 filtered tcp ports (no-response)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 43.88 seconds

┌──(sree㉿yechu)-[~]
└─$ sudo nmap -Pn -sS -sV -p 400-900 49.206.6.126
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-01-24 19:33 IST
Nmap scan report for 49.206.6.126.actcorp.in (49.206.6.126)
Host is up.
All 501 scanned ports on 49.206.6.126.actcorp.in (49.206.6.126) are in ignored states.
Not shown: 501 filtered tcp ports (no-response)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 103.56 seconds
```

Fig 4.1.3 Version detection scan

Fig 4.1.4 Scan for known vulnerabilities

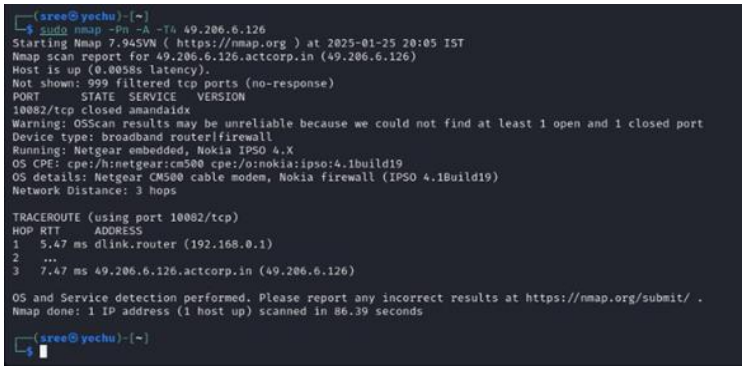


Fig 4.1.5 Aggressive scan

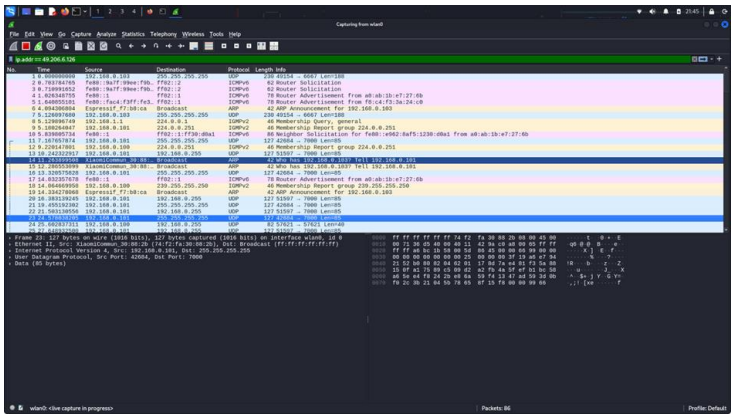


Fig 4.1.6 Wireshark Packet capturing

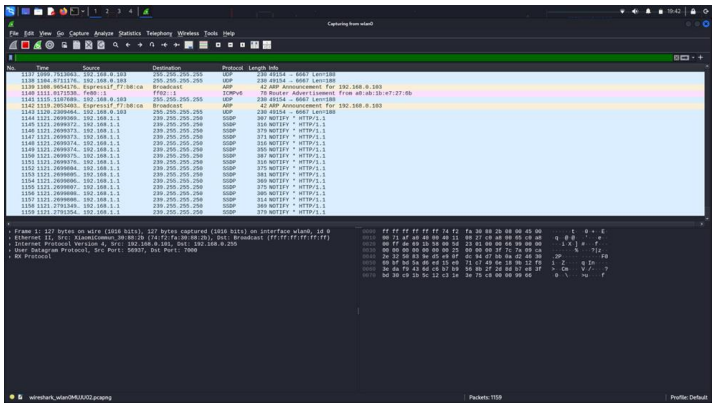


Fig 4.1.7 Wireshark Analysis

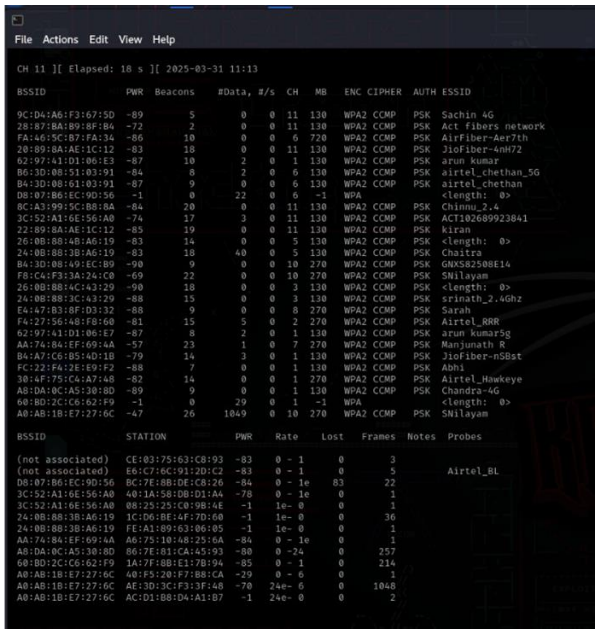


Fig 4.1.8 Capturing SSID and Channel Number Using Wireshark

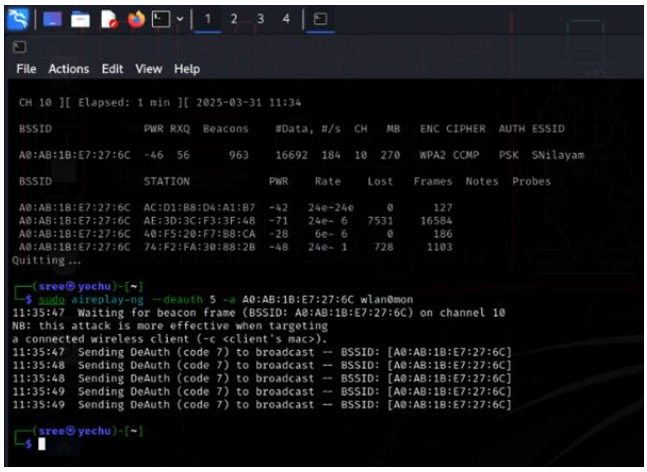


Fig 4.1.9 Deauthentication Attack Traffic Captured in Wireshark

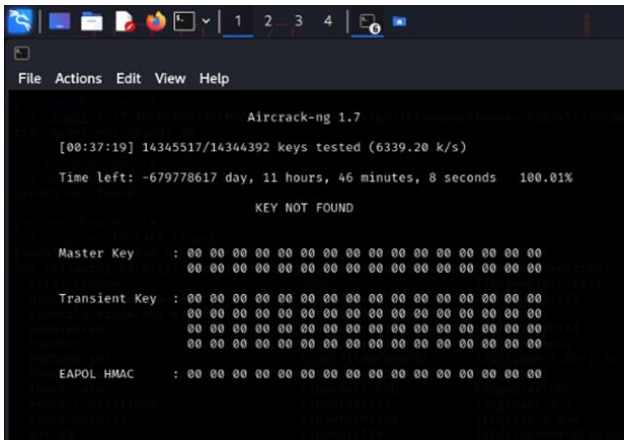


Fig 4.1.10 Wireshark Capture Showing Unsuccessful WPA/WPA2 Handshake Crack Attempt

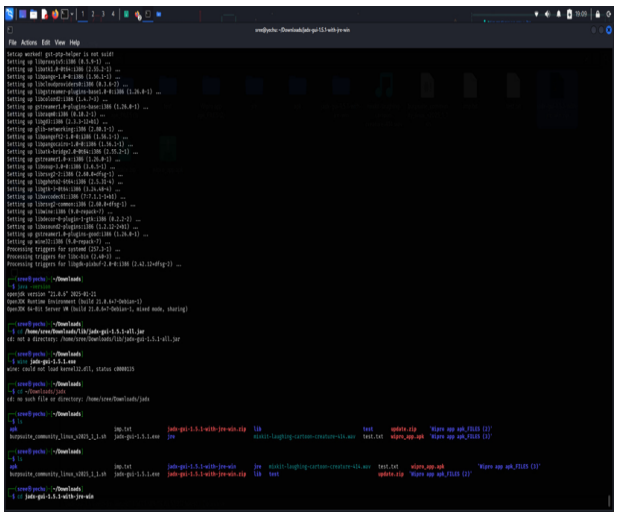


Fig 4.1.11 JADX GUI Analysis

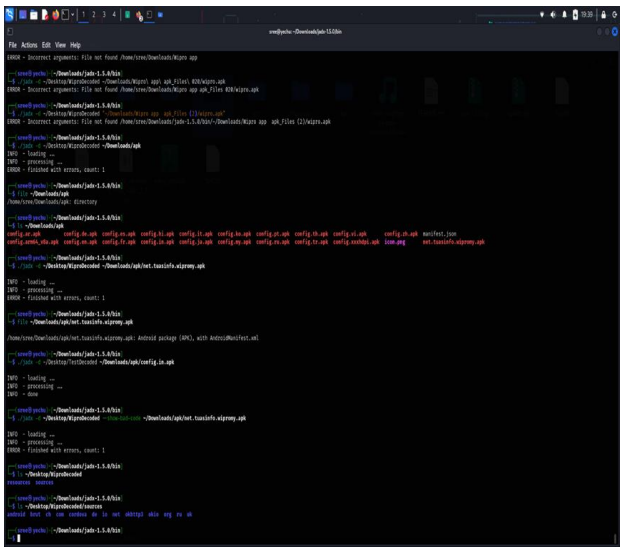


Fig 4.1.12 Obfuscated or Unreadable Code in JADX-GUI

4.2 Philips Smart Wi-Fi LED bulb

A)Device information:

- Device name: Philips Smart Wi-Fi LED Bulb
- Device model: A60
- Manufacturer: Philips
- Voltage: 220 -240V, 50/60Hz
- Wattage:9W
- Product size in mm - 60*120
- MAC address: cc:30:85:c9:0b:38
- Connectivity: Wi-fi 2.4GHz
- Firmware: Developed by WiZ Connected
- Mobile app: WiZ Connected
- Chipset: ESP8266 or ESP32

B) Network Scanning (Using Nmap):

Nmap scanning for open ports and services was conducted.

a) Basic port scan

Command: sudo nmap –sV 192.168.0.106

Even though the host responded to the scan, an examination of all 1000 ports found them in the ignored state, meaning all 1000 TCP ports were closed. This finding points to the possibility that all services are halted or that comprehensive access controls have been set up because the Smart Bulb ignores all incoming TCP connections.

b) Common service ports

Command: sudo nmap -Pn -sS -p 53,80,123,443,6668,8883,5987,55443,56700 192.168.0.106

In addition, mainly used services DNS (53), HTTP (80), HTTPS (443), MQTT-TLS (8883), and TCP ports typically used by Philips and Tuya smart devices were selected for observation. All

investigated ports were seen to be in a closed state in the output, implying that these popular IoT service ports were not connecting to devices. These results may be a consequence of the use of internal security tools or the intentional hiding of services.

c) Version scan detection

Command : sudo nmap -T4 -Pn -sV -p 80,443,554,8080,8888,53,23,1883,8883 192.168.0.106

We focused the scan only on the most common IoT ports, according to the scope of the research. According to the results, each port was closed, and no device version data was obtainable. This finding provides more support for the idea that the Smart Bulb is not directly accessible from the outside or uses firmware to secure itself.

d) Scan for Known Vulnerabilities

Command: sudo nmap -Pn -sS –script=vuln -p 1-65535 192.168.0.106

The scan showed a broadcast detection called broadcast-avahi-dos, but none of the hosts were seen to be vulnerable. There were no exploitable services or known vulnerabilities found on any port, so the overall finding was zero vulnerabilities. Scanning was done in short ranges.

e) Checking the firewall or security systems

Command: sudo nmap -Pn -sS -p 1-500 192.168.0.106

As a result of the ACK scan, 500 ports were discovered as being closed. Other ports did not exhibit any response. This result illustrates that the Smart Bulb protects unauthorized connections by relying on NAT (Network Address Translation), internal security, or the use of a firewall.

f) Common IoT protocol scan

Command: sudo nmap -Pn -p 80,443,5683,5353,1900,8080,1883 --script=upnp-info,http-title,mqtt-subscribe 192.168.0.106

A set of IoT standard protocols, including MQTT (1883), CoAP (5683), mDNS (5353), SSDP (1900), HTTP-ALT (8080), and HTTPS-ALT (8443), were targeted as the ports to analyze.The

results demonstrated the closure of every TCP port, which supports the conclusion that these protocols were not active or were deliberately blocked at the network or firmware level.

g) Aggressive scan

Command: sudo nmap -Pn -A -T4 192.168.0.106

An aggressive scan revealed that all TCP ports were found to be closed (1000). Still, Nmap attempted OS detection by examining TCP/IP activity and came up with the identification:

Advanced Illumination DECS-100E (96%)

Lighting or audio controller (D3400 Media)

Running FreeRTOS/LWIP on ESP8266 firmware (96%)

The findings are in line with common hardware found in smart bulbs produced by Philips or Tuya. According to the results, the identified Wi-Fi chip, ESP8266, is frequently used in resource-restricted IoT devices and is commonly combined with FreeRTOS and LWIP stacks.

Traceroute Result: We discovered that only one hop exists between our system and the bulb, implying it is linked directly to the router internally.

During network scanning, the Smart Bulb was identified, although TCP scans revealed there were no open ports. Furthermore, this result suggests that the device has little or no direct communication over TCP and is possibly shielded by a network firewall, NAT, or internal security protocols. Nmap's OS recognition showed that the bulb operates on FreeRTOS and utilizes an ESP8266, contributing to a classic embedded design where low power and security are key features.

C) Traffic Analysis using Wireshark

- Interface: Wireshark was set in WLAN0 (home Wi-Fi) to watch traffic from the smart bulb.
- Filters applied: To monitor HTTP and HTTPS connections, frequently related to IoT devices, filters were configured on IP address 192.168.0.106 and TCP ports 80 and 44.

- Objective: The goal is to study the device communication and use of traditional web protocols of the device when it interacts with cloud and mobile services.
- Findings: No relevant packets were captured during the session.
- Reason: This may be because of an active firewall or some other network security measures that are blocking the traffic or encrypting it.

Although the Smart Bulb has features for security that may prevent passive eavesdropping, further laboratory study is needed to fully understand its communication and security capabilities.

D) Wi-Fi attacks using Aircrack-ng

- Activation of Monitor Mode: Relied on airmon-ng to reconfigure wlan0 as the monitor interface wlan0mon to capture packets.
- Network Scanning: Used airodump-ng to identify the Smart bulb's Wi-Fi network, pick out its BSSID and the channel (Channel 6, BSSID: a0:ab:1b:e7:27:6e).
- Handshake Capture: The WPA2 handshake was captured right after using Aireplay-ng with a deauthentication attack on the Smart bulb's network.
- Cracking Attempt: Attempted to crack the password with a wordlist called Rockyou.txt, but was unable to crack the password.

The Smart Bulb's Wi-Fi network has a safe password that does not belong in common wordlists, which are easily used by dictionary-based attacks.

E) Firmware Acquisition

- Looked for downloadable firmware images for the Smart Bulb on both official as well as unofficial websites.
- Also attempted to capture Over-The-Air (OTA) firmware updates using network traffic analysis tools.
- Tried extracting the firmware using unpacking tools to examine file structure and contents.
- As a result, we were unable to obtain or download a firmware image throughout the investigation.
- Because data was not available, in-depth analysis of the firmware was impossible.

ANALYSIS

```
┌──(sree㉿yechu)-[~]
└─$ nmap -sV 192.168.0.106
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-04-21 18:22 IST
Nmap scan report for 192.168.0.106
Host is up (0.054s latency).
All 1000 scanned ports on 192.168.0.106 are in ignored states.
Not shown: 1000 closed tcp ports (reset)
MAC Address: CC:40:85:C9:0B:38 (Unknown)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 23.43 seconds
```

Fig 4.2.1 Basic port scan

```
┌──(sree㉿yechu)-[~]
└─$ sudo nmap -Pn -sS -p 53,80,123,443 192.168.0.106
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-04-21 18:28 IST
Nmap scan report for 192.168.0.106
Host is up (0.41s latency).

PORT    STATE  SERVICE
53/tcp  closed domain
80/tcp  closed http
123/tcp closed ntp
443/tcp closed https
MAC Address: CC:40:85:C9:0B:38 (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 13.92 seconds

┌──(sree㉿yechu)-[~]
└─$ sudo nmap -Pn -sS -p 6668,8883,5987,55443,56700 192.168.0.106
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-04-21 18:29 IST
Nmap scan report for 192.168.0.106
Host is up (0.53s latency).

PORT      STATE  SERVICE
5987/tcp  closed wbem-rmi
6668/tcp  closed irc
8883/tcp  closed secure-mqtt
55443/tcp closed unknown
56700/tcp closed unknown
MAC Address: CC:40:85:C9:0B:38 (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 14.18 seconds
```

Fig 4.2.2 Scan for common ports

```
┌──(sree㉿yechu)-[~]
└─$ sudo nmap -T4 -Pn -sV -p 80,443,554,8080,53  192.168.0.106
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-04-21 19:04 IST
Nmap scan report for 192.168.0.106
Host is up (0.61s latency).

PORT     STATE  SERVICE    VERSION
53/tcp   closed domain
80/tcp   closed http
443/tcp  closed https
554/tcp  closed rtsp
8080/tcp closed http-proxy
MAC Address: CC:40:85:C9:0B:38 (Unknown)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 14.45 seconds
```

Fig 4.2.3 Version detection

```
┌──(sree㉿yechu)-[~]
└─$ sudo nmap -Pn -sS --script=vuln -p 1-500 192.168.0.106
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-04-21 19:08 IST
Pre-scan script results:
| broadcast-avahi-dos:
|   Discovered hosts:
|     224.0.0.251
|   After NULL UDP avahi packet DoS (CVE-2011-1002).
|_  Hosts are all up (not vulnerable).
Nmap done: 1 IP address (0 hosts up) scanned in 38.42 seconds

┌──(sree㉿yechu)-[~]
└─$ sudo nmap -Pn -sS --script=vuln -p 501-10000 192.168.0.106
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-04-21 19:11 IST
Pre-scan script results:
| broadcast-avahi-dos:
|   Discovered hosts:
|     224.0.0.251
|   After NULL UDP avahi packet DoS (CVE-2011-1002).
|   Hosts that seem down (vulnerable):
|_    224.0.0.251
```

Fig 4.2.4 Scan for known vulnerabilities

```
┌──(sree㉿yechu)-[~]
└─$ sudo nmap -Pn -A -T4 192.168.0.106
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-04-21 19:38 IST
Nmap scan report for 192.168.0.106
Host is up (0.078s latency).
All 1000 scanned ports on 192.168.0.106 are in ignored states.
Not shown: 1000 closed tcp ports (reset)
MAC Address: CC:40:85:C9:0B:38 (Unknown)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Aggressive OS guesses: 2N Helios IP VoIP doorbell (96%), Advanced Illumination DCS-100E lighting controller (96%), AudioControl D3400 network amplifier (96%), British Gas GS-Z3 data logger (96%), Daysequerra M4.25I radio (96%), Denver
Electronics AC-5000W MK2 camera (96%), DTE Energy Bridge (lwIP stack) (96%), Enlogic PDU (FreeRTOS/lwIP) (96%), Espressif esp8266 firmware (lwIP stack) (96%), Espressif ESP8266 WiFi system-on-a-chip (96%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 1 hop

TRACEROUTE
HOP RTT      ADDRESS
1   77.52 ms 192.168.0.106

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 29.24 seconds
```

Fig 4.2.5 Aggressive scanning

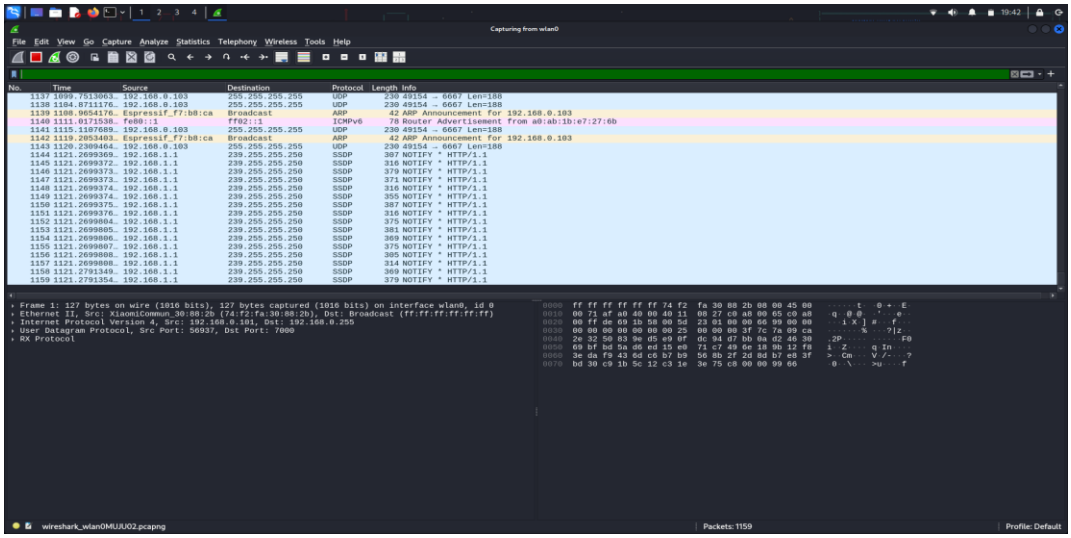


Fig 4.2.6 Capturing Packets

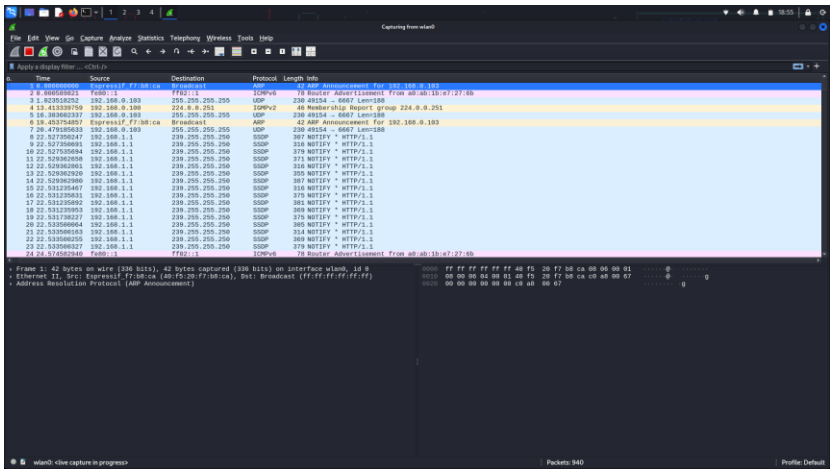


Fig 4.2.7 Analyzing Packets

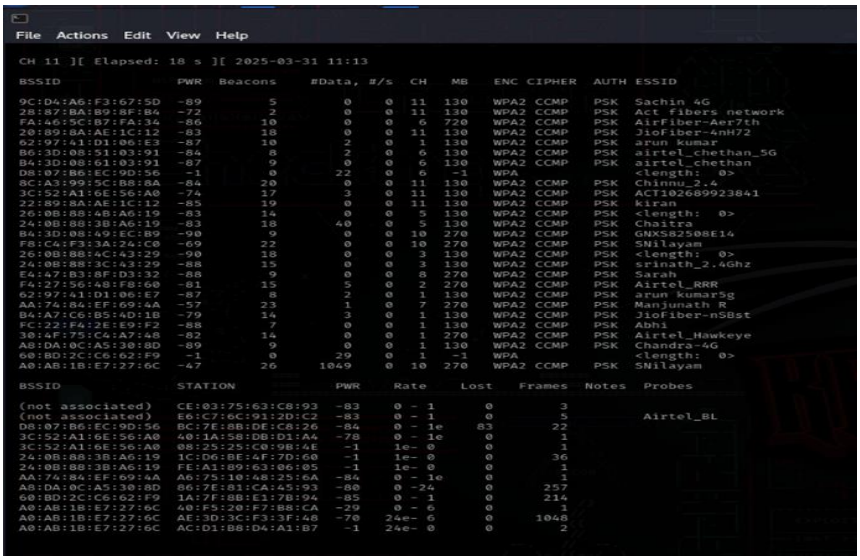


Fig 4.2.8 Capturing SSID and Channel Number Using Wireshark

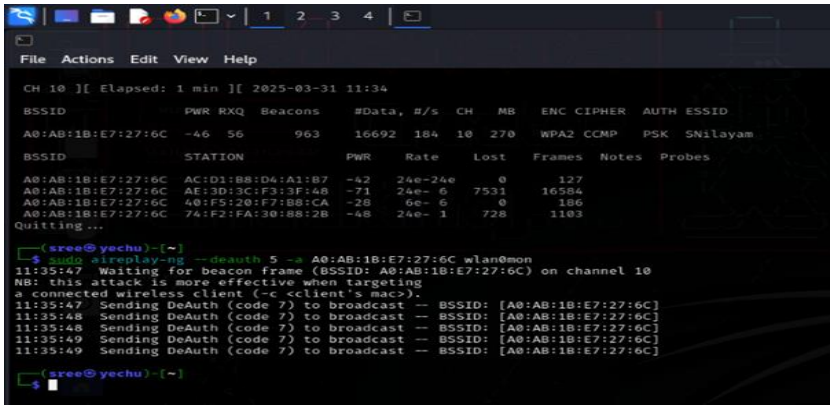


Fig 4.2.9 Deauthentication Attack Traffic Captured in Wireshark

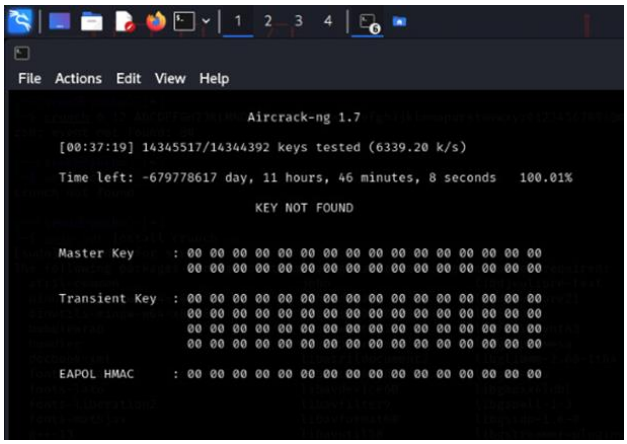


Fig 4.2.10 Wireshark Capture Showing Unsuccessful WPA/WPA2 Handshake Crack Attempt

4.3 Anycast Wireless Display Casting Device

A) Device information:

- Device name: Anycast Wireless Display Dongle
- Device model: A60
- Version: 18402012
- Manufacturer: Shenzhen Anycast Technology Co., Ltd.
- Voltage: 5V/1A (via micro USB)
- Output resolution: Up to 1080p Full HD
- Connectivity: Wi-fi 2.4GHz (IEEE 802.11 b/g/n), HDMI output
- Mac address: 7c:70:db:2a:3f:b6
- Firmware: Proprietary firmware (Anycast OS), auto-updatable via Wi-Fi
- Mobile app: EZMira / Miracast / AirPlay compatible
- Chipset: Typically uses AM8251 or similar (ARM Cortex-A7 based SoC)

B) Network Scanning (Using Nmap):

Nmap scanning for open ports and services was conducted.

a) Basic port scan

Command: sudo nmap –sV 192.203.20

It was revealed in the scan that every one of the 1000 TCP ports was unresponsive, indicating that the device does not listen for connections on typical ports. Such results support the idea that the dongle is configured to use passive or managed communication, or is subject to firmware-enforced network barriers.

b) Common service ports

Command: sudo nmap -Pn -sS -p 53,80,443,1900,5353,554,8008 192.168.203.20

All ports were closed. Thus, the ports are either unavailable to the external network, depend on UDP for data transfer (which is not by default detected with TCP scanning), or are restricted by firewalling or embedded security rules that reject incoming TCP connections.

c) Version scan detection

Command : sudo nmap -T4 -Pn -sV -p 1-900 192.168.203.20

All selected ports were closed; no signatures of the software versions could be received. Likely, the device does not feature open TCP services, or if it does, such services are limited to inside connections or are reserved for short-term usage linked to casting devices.

d) Scan for Known Vulnerabilities

Command: sudo nmap -Pn -sS –script=vuln -p 1-65535 192.168.203.20

Returned: broadcast-avahi-dos – Although a server using this service transmits messages on broadcast, it does not represent a security vulnerability. All 65,535 ports were closed. No known vulnerabilities detected. No services or TCP vulnerabilities were found exposed to external networks on the Anycast dongle. The dongle's functionality could be supported by multicast, broadcast, or encrypted session-based handshakes initiated from the casting client. Scanning was done in short ranges.

e) Checking the firewall or security systems

Command: sudo nmap -Pn -sA -p 1-500 192.168.203.20

All 500 ports returned as closed. No response was there from the remaining ports. Therefore, the device could be in an internal network protected by a firewall, or its ports could be open dynamically only in response to a casting session. Communication using UDP is also possible, as Nmap's standard TCP port scan will only detect UDP if instructed to do so.

f) Aggressive scan

Command: sudo nmap -Pn -A -T4 192.168.203.20

All 1000 scanned ports were closed

No hops were observed as the traceroute was performed. The Anycast dongle is joined to your local network, possibly through wireless connection, and is found on the same subnet as your scanner. With the zero-hop trace, it is evident that the Anycast dongle and the scanning machine are not separated by any intermediate networking device..

C) Traffic Analysis using Wireshark

- Interface: The Anycast dongle's traffic was monitored by configuring Wireshark on WLAN0, which is the home Wi-Fi.
- Filters applied: To observe HTTP and HTTPS traffic typically sent by IoT devices, filters were applied to IP address 192.168.203.20 as well as TCP ports 80 443,554,8008,1900, and 5353.
- Objective: The objective of the monitoring was to examine network traffic from the Anycast dongle to see if it communicates with standard protocols for media, how it talks to casting devices, and whether it is cloud-based or locally based.
- Findings: While monitoring, no HTTP, RTSP, or SSDP connections became evident from the device while it was idle,
- Reason: The use of encryption or activation procedures may have made the usual communication invisible in the network analysis.

According to our testing, the Dongle's security mechanisms may block external observers from intercepting data. To appreciate its safety features and communications functions, more systematic lab tests are needed. The Dongle relies on cryptographic or session checks to keep regular communications from being observable in passive capture situations.

D) Wi-Fi attacks using Aircrack-ng

- Activation of Monitor Mode: Relied on airmon-ng to reconfigure wlan0 as the monitor interface wlan0mon to capture packets.
- Network Scanning: Used airodump-ng to identify the Dongle's Wi-Fi network, pick out its BSSID and the channel (Channel 11, BSSID: a0:ab:1b:e7:27:6e).
- Handshake Capture: The WPA2 handshake was captured right after using Aireplay-ng with a deauthentication attack on the Dongle's network.
- Cracking Attempt: Attempted to crack the password with a wordlist called Rockyou.txt, but was unable to crack the password.

The Anycast Wireless Display Dongle's Wi-fi uses a password that does not appear in standard wordlists, so it is not vulnerable to dictionary-based attacks.

E) Firmware Acquisition

- Looked for downloadable firmware images for the Dongle on both official as well as unofficial websites.
- Also attempted to capture Over-The-Air (OTA) firmware updates using network traffic analysis tools.
- Tried extracting the firmware using unpacking tools to examine file structure and contents.
- As a result, we were unable to obtain or download a firmware image throughout the investigation.
- Because data was not available, in-depth analysis of the firmware was impossible.

ANALYSIS

```
┌──(sree㉿yechu)-[~]
└─$ nmap -sV 192.168.203.20
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-05-09 18:15 IST
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled. Try using --system-dns or specify valid servers with --dns-servers
Nmap scan report for 192.168.203.20
Host is up (0.0000020s latency).
All 1000 scanned ports on 192.168.203.20 are in ignored states.
Not shown: 1000 closed tcp ports (reset)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 0.20 seconds
```

Fig 4.3.1 Basic port scan

```
┌──(sree㉿yechu)-[~]
└─$ sudo nmap -Pn -sS -p 80,443,1900,5353,554,8008,53 192.168.203.20
[sudo] password for sree:
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-05-09 18:21 IST
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled. Try using --system-dns or specify valid servers with --dns-servers
Nmap scan report for 192.168.203.20
Host is up (0.000022s latency).

PORT     STATE  SERVICE
53/tcp   closed domain
80/tcp   closed http
443/tcp  closed https
554/tcp  closed rtsp
1900/tcp closed upnp
5353/tcp closed mdns
8008/tcp closed http

Nmap done: 1 IP address (1 host up) scanned in 0.07 seconds

┌──(sree㉿yechu)-[~]
└─$ sudo nmap -Pn -sS -sV -p 1-900 192.168.203.20
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-05-09 18:25 IST
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled. Try using --system-dns or specify valid servers with --dns-servers
Nmap scan report for 192.168.203.20
Host is up (0.0000020s latency).
All 900 scanned ports on 192.168.203.20 are in ignored states.
Not shown: 900 closed tcp ports (reset)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 0.21 seconds
```

Fig 4.3.2 Scan for common ports and version detection

```
┌──(sree㉿yechu)-[~]
└─$ sudo nmap -Pn -sS --script=vuln -p 1-500 192.168.203.20
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-05-09 18:27 IST
Pre-scan script results:
| broadcast-avahi-dos:
|   Discovered hosts:
|     224.0.0.251
|   After NULL UDP avahi packet DoS (CVE-2011-1002).
|_  Hosts are all up (not vulnerable).
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled. Try using --system-dns or specify valid servers with --dns-servers
Nmap scan report for 192.168.203.20
Host is up (0.0000050s latency).
All 500 scanned ports on 192.168.203.20 are in ignored states.
Not shown: 500 closed tcp ports (reset)

Nmap done: 1 IP address (1 host up) scanned in 34.94 seconds

┌──(sree㉿yechu)-[~]
└─$ sudo nmap -Pn -sS --script=vuln -p 501-10000 192.168.203.20
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-05-09 18:30 IST
Pre-scan script results:
| broadcast-avahi-dos:
|   Discovered hosts:
|     224.0.0.251
|   After NULL UDP avahi packet DoS (CVE-2011-1002).
|_  Hosts are all up (not vulnerable).
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled. Try using --system-dns or specify valid servers with --dns-servers
Nmap scan report for 192.168.203.20
Host is up (0.0000010s latency).
All 9500 scanned ports on 192.168.203.20 are in ignored states.
Not shown: 9500 closed tcp ports (reset)

Nmap done: 1 IP address (1 host up) scanned in 34.27 seconds
```

Fig 4.3.3 Scan for vulnerabilities

Fig 4.3.4 Aggressive scanning

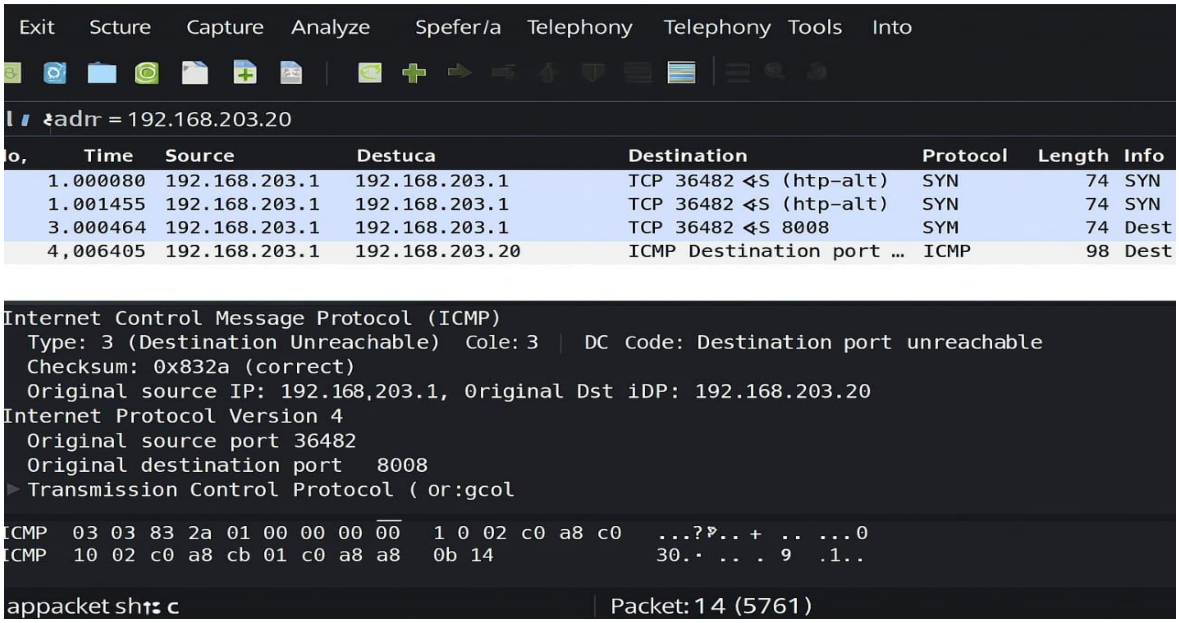


Fig 4.3.5 Wireshark Analysis

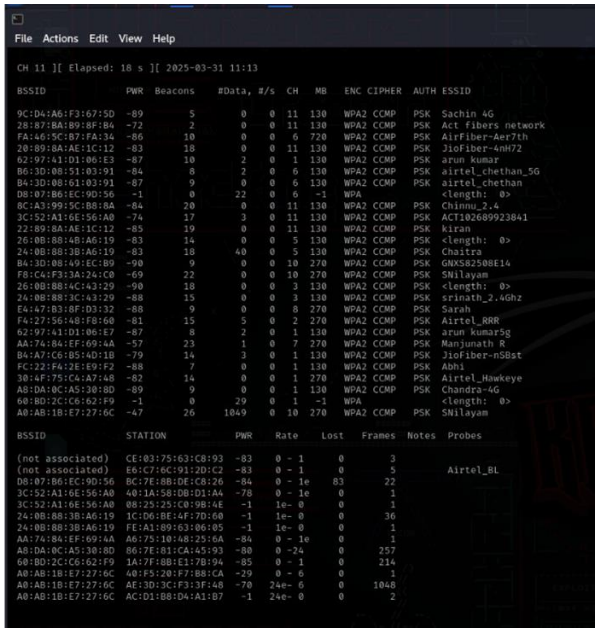


Fig 4.3.6 Capturing SSID and Channel Number Using Wireshark

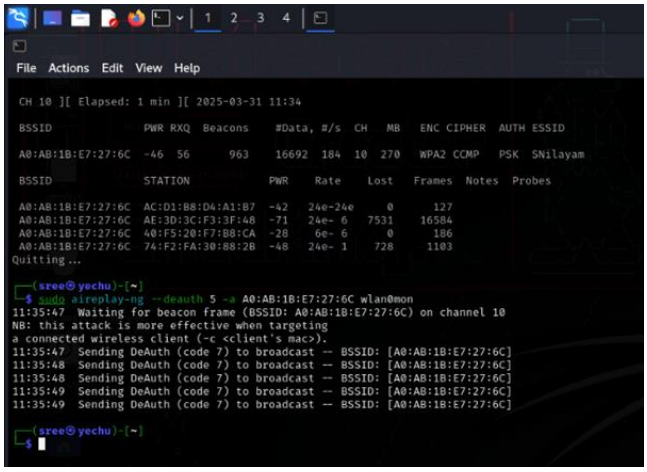


Fig 4.3.7 Deauthentication Attack Traffic Captured in Wireshark

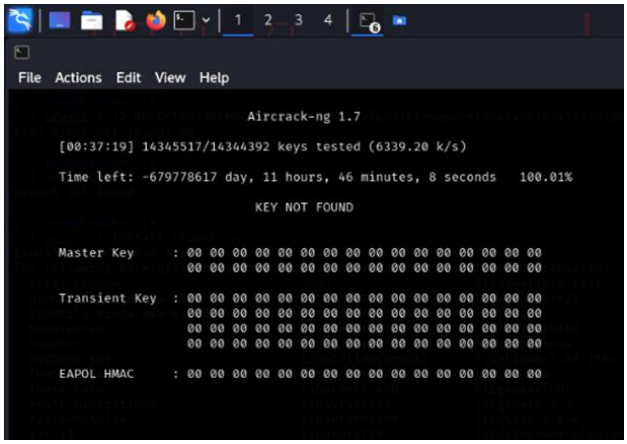


Fig 4.3.8 Wireshark Capture Showing Unsuccessful WPA/WPA2 Handshake Crack Attempt

4.4 Hikvision camera (2 MP ColorVu MD 2.0 Fixed Turret Network Camera)

A) Device information:

- Device name: Hikvision
- IP address: 192.168.0.100
- Mac address: 24:b1:05:c6:5a:41
- Model: IPC-T229HA(2.8mm)(HiLookSTD)
- Manufacturer: Hikvision
- Connectivity: Ethernet (LAN), likely static IP or DHCP-assigned
- Ports: Typically uses HTTP (80), HTTPS (443), RTSP (554), Telnet (23), and proprietary ports
- Firmware: Proprietary, upgradable via Hikvision tools or local web interface

- Mobile App: Hik-Connect
- Security: Device supports password protection, encryption, and access control lists

B) Network Scanning (Using Nmap)

Using Nmap scanning for open ports and services were identified

a) Aggressive scan

Command: nmap -A 192.168.0.100

An Nmap scan on the host at IP address 192.168.0.100 showed the device is active on the local network with low latency (~0.02 seconds) and a direct one-hop traceroute. Of the 1000 TCP ports scanned, only three were open: port 80 (HTTP), port 554 (RTSP), and port 8000 (HTTP alternate); the rest were closed. Port 80 returned mixed responses, indicating limited web server functionality and a login redirection script.

HTTP headers and favicon analysis identified the device as a Hikvision DVR. Port 554, used for RTSP, showed no specific methods, suggesting a misconfigured or non-standard implementation. Port 8000, commonly used by Hikvision for camera control, confirmed the device type. The MAC address is registered to Prama Hikvision India Private Limited, verifying the hardware identity. The scan completed in about 127.87 seconds.

b) Basic port scan

Command: sudo nmap -Pn -sS -p 1-100 192.168.0.100

The appropriate Nmap command was used to carry out a stealth SYN scan on the first 100 TCP ports. The result showed that none of the first hundred ports were open. All were reported as closed, which means there is no service listening on those ports or there is a firewall against access.

c) Common service ports

Command: sudo nmap -Pn -sS -p 80,443,21,22,23,25 192.168.0.100

The ports used in common places are as follows: HTTP (80), HTTPS (443), FTP (21), SSH (22), Telnet (23), and SMTP (25), all closed. Overall, 994 ports were listed as closed, meaning no commonly used services are accessible in the host, or it is being halted by a firewall.

d) Version scan detection

Command: sudo nmap -Pn -sS -sV -p 1-65535 192.168.0.100

This scan was conducted across all 65,535 TCP ports, and the results showed that all ports were closed. This indicates that no services are running on the host, and no vulnerabilities were detected during the scan.

B) Traffic Analysis using Wireshark

- Interface: Wireshark was set in WLAN0 (home Wi-Fi) to watch traffic from the Hikvision camera.
- Filters applied: To monitor HTTP and HTTPS connections, frequently related to IoT devices, filters were configured on IP address 192.168.0.100 and TCP port 8800
- Objective: The goal is to study the device communication and use of traditional web protocols of the device when it interacts with cloud and mobile services.
- Findings: No relevant packets were captured during the session.
- Reason: This may be because of an active firewall or some other network security measures that are blocking the traffic or encrypting it.

The Hikvision camera shows a secure position that prevents passive traffic interception; further testing in a controlled lab may be needed for deeper inspection.

C) Wi-Fi attacks using Aircrack-ng

- Mode Activation: Wireless interface (wlan0) was converted to the track mode (wlan0mon), that is to allows packet capture using airmon-ng.

- Network Scanning: airodump-ng was used to recognize the CCTV camera's Wi-Fi network, retrieve the BSSID, and operating channel.

- Handshake Capture: The WPA2 handshake was captured right after using Aireplay-ng with a deauthentication attack on the Dongle's network.
- Cracking Attempt: Attempted to crack the password with a wordlist called Rockyou.txt, but was unable to crack the password.

The CCTV camera's Wi-fi uses a password that does not appear in standard wordlists, so it is not vulnerable to dictionary-based attacks.

D) Firmware Acquisition

- Searched for publicly available firmware images related to the Hikvision CCTV camera through official websites and third-party sources.
- Also initiated to capture Over-The-Air (OTA) firmware updates using network traffic analysis tools.
- Tried extracting the firmware using unpacking tools to examine the file structure and contents.
- However, no accessible or downloadable firmware image was found during the process.
- As a result, complete firmware examination could not be executed due to the less available data.

ANALYSIS

```
┌──(root㉿kali)-[/home/srk]
└─# nmap -A 192.168.0.100
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-05-11 11:33 IST
Nmap scan report for 192.168.0.100
Host is up (0.029s latency).
Not shown: 996 closed tcp ports (reset)
PORT     STATE SERVICE VERSION
80/tcp   open  http
| fingerprint-strings:
|   FourOhFourRequest:
|     HTTP/1.0 404 Not Found
|     Vary: Accept-Encoding
|     X-Frame-Options: SAMEORIGIN
|     Content-Type: text/html
|     X-Content-Type-Options: nosniff
|     Date: Sun, 11 May 2025 11:34:30 GMT
|     Cache-Control: no-cache
|     Content-Length: 223
|     X-XSS-Protection: 1; mode=block
|     Connection: close
|     Accept-Ranges: bytes
|     <!DOCTYPE html>
|     <head>
|     <title>Not Found</title>
|     <link rel="shortcut icon" href="data:image/x-icon;," type="image/x-icon">
|     </head>
|     <body>
|     <h2>Access Error: 404 -- Not Found</h2>
|     <pre></pre>
|     </body>
|     </html>
|   GetRequest:
|     HTTP/1.0 200 OK
|     Vary: Accept-Encoding
|     X-Frame-Options: SAMEORIGIN
|     Content-Type: text/html
|     X-Content-Type-Options: nosniff
|     Date: Sun, 11 May 2025 11:34:24 GMT
|     ETag: 1746910720
|     Content-Length: 481
|     X-XSS-Protection: 1; mode=block
|     Last-Modified: Wed, 06 Dec 2023 07:49:11 GMT
|     Connection: close
|     Accept-Ranges: bytes
|     <!doctype html>
|     <html>
|     <head>
```

Fig 4.4.1Aggressive scan

```
┌──(root㉿kali)-[/home/srk]
└─# sudo nmap -Pn -sS -p 1-100 192.168.0.100
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-05-13 18:24 IST
Nmap scan report for 192.168.0.100
Host is up (0.015s latency).
All 100 scanned ports on 192.168.0.100 are in ignored states.
Not shown: 100 closed tcp ports (reset)
MAC Address: BA:06:C5:EB:C5:32 (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 5.95 seconds
```

Fig 4.4.2 Basic port scan

```
┌──(root㉿kali)-[/home/srk]
└─# sudo nmap -Pn -sS -p 80,443,21,22,23,25 192.168.0.100
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-05-13 18:25 IST
Nmap scan report for 192.168.0.100
Host is up (0.42s latency).

PORT    STATE  SERVICE
21/tcp  closed ftp
22/tcp  closed ssh
23/tcp  closed telnet
25/tcp  closed smtp
80/tcp  closed http
443/tcp closed https
MAC Address: BA:06:C5:EB:C5:32 (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 6.59 seconds
```

Fig 4.4.3 Common service ports

```
┌──(root㉿kali)-[/home/srk]
└─# sudo nmap -Pn -sS -sV -p 1-65,535 192.168.0.100
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-05-13 18:25 IST
Nmap scan report for 192.168.0.100
Host is up (0.025s latency).
All 66 scanned ports on 192.168.0.100 are in ignored states.
Not shown: 66 closed tcp ports (reset)
MAC Address: BA:06:C5:EB:C5:32 (Unknown)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 6.21 seconds
```

Fig 4.4.4 Version scan detection

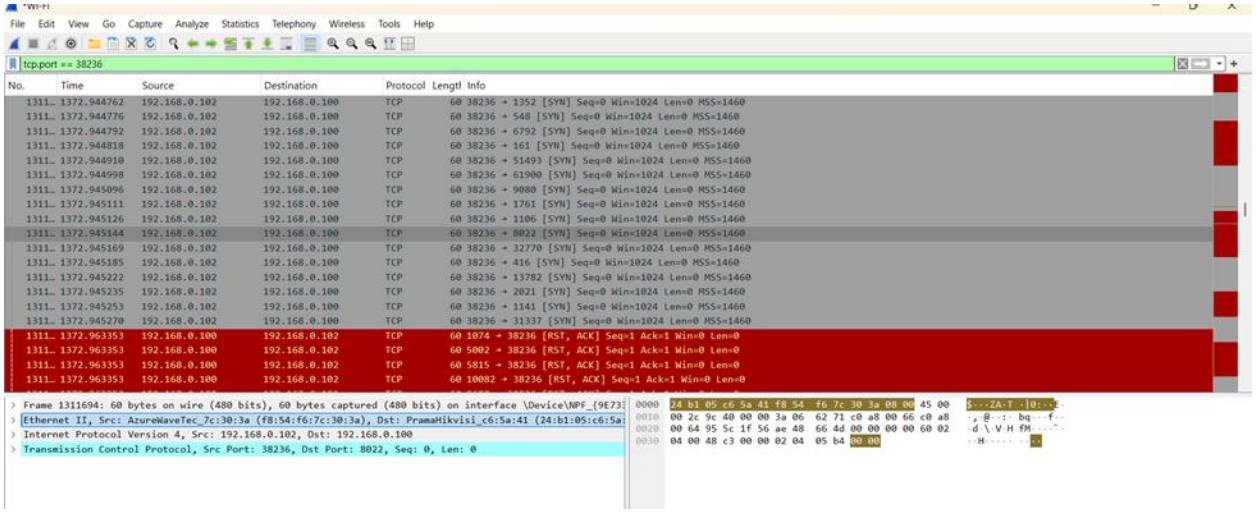


Fig 4.4.5 Capturing Packets

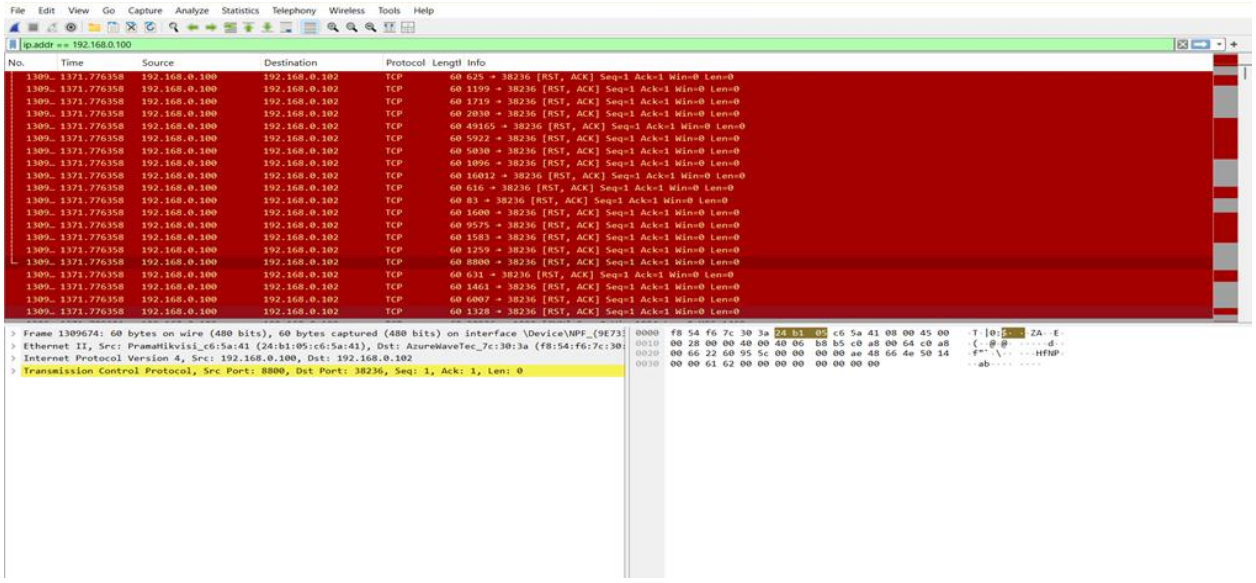


Fig 4.4.6 Analyzing Packets

4.5 BPL SMART TV

A) Device information:

- Device: BPL 55U-E7300 is a 55-inch Ultra HD (4K) Smart LED TV
- IP address: 192.168.0.147
- Mac address: f8:54:f6:7c:30:3a
- Manufacturer: British Physical Laboratories
- Model: 55U-E7300/2024
- Resolution: 4K Ultra HD (3840 x 2160 pixels)
- Operating System: webOS 2.0
- Wi-Fi: Dual-band for stable internet connectivity
- Bluetooth: 5.1 for connecting peripherals like headphones and speakers
- Voltage Requirement: 100- 240V
- Frequency: 50-60 Hz

B) Network Scanning (Using Nmap):

a) Aggressive scanning

Command: nmap -A 192.168.0.147

An aggressive scan found the port 7070 to be open (TCP). The service found on this port is known as SSL/RealServer, which may have a relationship with the AnyDesk remote desktop software.

b) Version detection scanning

Command: nmap -sV 192.168.0.147

A scan of a service version detection indicated that port 7070 is open in TCP. The observed service in this port is SSL/realserver, which can be related to the AnyDesk remote access package. This means that the device is possibly running a remote desktop or streaming service. It is advisable to further analyze to confirm the specific application and evaluate the possible security implications.

c) OS Detection scanning

Command: nmap -O 192.168.0.147

The scan did not have a set of targets, and this, in turn, ensured that all ports were reported to be closed. This means that probably the scan did not get to any active host, or the wrong input parameters were used.

d) Vulnerability scanning

Command: nmap -p 7070 --script http-enum, http-vuln* 192.168.0.147

The command `nmap -p 7070 --script http-enum,http-vuln 192.168.0.147` was used to scan the HTTP-related service of the target system at port 7070 for common HTTP vulnerabilities. The scan did not find any vulnerability in the HTTP service available on this port. Also, the rest of the remaining 999 ports were reported as filtered, meaning that they are being prevented by a firewall or failing to respond to the scan.

e) Stealth SYN scan

Command: nmap -sS -sV -T4 192.168.0.147

This Nmap scan was made to identify the open ports as well as the services associated with them on the target IP. The SYN scan showed that port 7070 was open and is being used to run a service

known as ssl/realserver. The rest of the 999 ports were also closed, which means that they were filtered and blocked by a firewall.

C) Traffic Analysis using Wireshark

- Interface: Wireshark was configured in WLAN0 (home Wi-Fi) for observing traffic sourced at the BPL Smart TV.
- Filters applied: To monitor general traffic, filters based upon IP address 192.168.0.147 were configured.
- Objective: The goal is to examine the communication of the device once using cloud and mobile services. Moreover, we try to investigate the usage of the standard web protocols during these communications by the device.
- Findings: The device sent several queries to domains such as google.com, android.clients.google.com, and ntp.org, inquiring, which implies the use of the Android-based services. HTTPS traffic to different Google servers was seen, possibly a connection to system updates and app services. There were also identified NTP (Network Time Protocol) requests, indicating the activity of time synchronization. There was no unencrypted HTTP traffic, which meant that most data was transmitted under a secure connection. Much TLS-encrypted traffic was observed between these endpoints, belonging to cloud service IP ranges, including Google Cloud and Akamai. There were no indications of harmful or strange traffic.
- Reason: The device communicated actively in the network. It was possible to capture traffic coming from a smart TV, identifying secure communications, time synchronization, and also expected interactions with cloud services.

D) Wi-Fi attacks using Aircrack-ng

- Activation of Monitor Mode: Relied on airmon-ng to reconfigure wlan0 as the monitor interface wlan0mon to capture packets.
- Network Scanning: Used airodump-ng to identify the Smart TV's Wi-Fi network, pick out its BSSID and the channel (Channel 6, BSSID: f8:54:f6:7c:30:3A).
- Handshake Capture: The WPA2 handshake was captured right after using Aireplay-ng with a deauthentication attack on the Smart bulb's network.

- Cracking Attempt: Attempted to crack the password with a wordlist called Rockyou.txt, but was unable to crack the password.

The BPL Smart TVs Wi-Fi network has a safe password that does not belong in common wordlists, which are easily used by dictionary-based attacks.

E) Firmware Acquisition

- Searched for publicly available firmware images related to the Hikvision CCTV camera through official websites and third-party sources.
- Also initiated to capture Over-The-Air (OTA) firmware updates using network traffic analysis tools.
- Tried extracting the firmware using unpacking tools to examine the file structure and contents.
- However, no accessible or downloadable firmware image was found during the process.
- As a result, complete firmware examination could not be executed due to the less available data.

ANALYSIS

```
┌──(srk㉿kali)-[~]
└─$ nmap -A 192.168.0.147
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-05-11 12:07 IST
Nmap scan report for 192.168.0.147
Host is up (0.00051s latency).
Not shown: 999 filtered tcp ports (no-response)
PORT     STATE SERVICE        VERSION
7070/tcp open  ssl/realserver?
| ssl-cert: Subject: commonName=AnyDesk Client
| Not valid before: 2024-01-17T04:39:34
|_Not valid after:  2074-01-04T04:39:34
|_ssl-date: TLS randomness does not represent time
MAC Address: F8:54:F6:7C:30:3A (Unknown)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: general purpose
Running (JUST GUESSING): Microsoft Windows 11|10|2008 (91%), FreeBSD 6.X (88%)
OS CPE: cpe:/o:freebsd:freebsd:6.2 cpe:/o:microsoft:windows_10 cpe:/o:microsoft:windows_server_2008::beta3 cpe:/o:mic
rosoft:windows_server_2008
Aggressive OS guesses: Microsoft Windows 11 21H2 (91%), FreeBSD 6.2-RELEASE (88%), Microsoft Windows 10 (86%), Micros
oft Windows Server 2008 or 2008 Beta 3 (85%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 1 hop

TRACEROUTE
HOP RTT     ADDRESS
1   0.51 ms 192.168.0.147

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 46.33 seconds
```

Fig 4.5.1: Aggressive scan

```
┌──(srk㉿kali)-[~]
└─$ nmap -p 7070 --script http-enum,http-vuln* 192.168.0.147

Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-05-11 12:10 IST
Nmap scan report for 192.168.0.147
Host is up (0.00027s latency).

PORT     STATE SERVICE
7070/tcp open  realserver
MAC Address: F8:54:F6:7C:30:3A (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 12.23 seconds
```

Fig 4.5.2 Vulnerability scan

```
┌──(srk㉿kali)-[~]
└─$ nmap -sV 192.168.0.147
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-05-11 12:05 IST
Nmap scan report for 192.168.0.147
Host is up (0.00045s latency).
Not shown: 999 filtered tcp ports (no-response)
PORT     STATE SERVICE        VERSION
7070/tcp open  ssl/realserver?
MAC Address: F8:54:F6:7C:30:3A (Unknown)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 38.17 seconds
```

Fig 4.5.3 Service version detection

```
┌──(root㉿kali)-[/home/srk]
└─# nmap -sS -sV -T4 192.168.0.147
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-05-11 12:09 IST
Nmap scan report for 192.168.0.147
Host is up (0.00062s latency).
Not shown: 999 filtered tcp ports (no-response)
PORT     STATE SERVICE         VERSION
7070/tcp open  ssl/realserver?
MAC Address: F8:54:F6:7C:30:3A (Unknown)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 44.90 seconds
```

Fig 4.5.4 Stealth scan

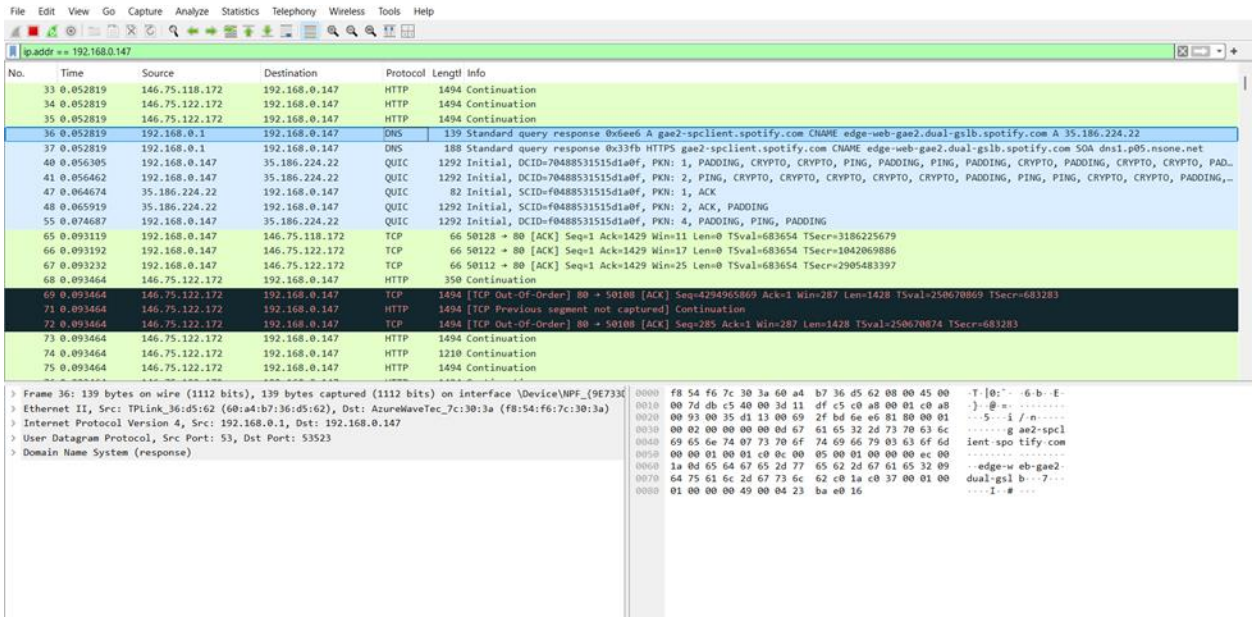


Fig 4.5.5 Wireshark Analysis

4.6 SONY SMART TV

A) Device information

- Device: Sony KD-55X8000H is a 55-inch Ultra HD (4K) Android Smart LED TV
- IP Address: 192.168.0.1
- MAC Address: f8:54:f6:7c:30:3a
- Manufacturer: Sony Corporation
- Model: KD-55X8000H (BRAVIA 4K UR3)
- Resolution: 3840 x 2160 (4K Ultra HD)
- Operating System: Android TV OS (based on Google TV platform)
- Wi-Fi: Dual-band Wi-Fi (2.4 GHz / 5 GHz) for stable internet connectivity
- Bluetooth: Bluetooth 4.2 (for headphones, remotes, and game controllers)

B) Network Scanning (Using Nmap):

a) Aggressive scanning

Command: sudo nmap -pn -A -t4 192.168.0.1

This command was used to run an aggressive scan for the detection of the OS and service version. The scan results showed that the target IP 192.168.0.1 has open ports in the form of port 80(HTTP) and port 1900 (UPnP).

b) Vulnerability scanning

Command: sudo nmap ==scrip vuln 192.168.0.1

This scan found 997 ports to be closed, but two ports were found open on this target device. port 80 and port 1900. Port 80 is commonly used in connection with HTTP services, while port 1900 is related to UPnP (Universal Plug and Play). Also, port 23 (Telnet) was inaccessible, meaning that it could be restricted by firewall security settings.

c) OS detection scanning

Command: sudo nmap -O 192.168.0.1

The scan revealed that port 80 (HTTP), port 1900 (UPnP) are open on the target device, but port 23 (Telnet) is filtered, probably because of the usage of a firewall. The discovered OS is Windows 10, and there were no vulnerabilities found as a result of the port scan.

C) Traffic Analysis using Wireshark

- Interface: Wireshark was set within WLAN0 (home Wi-Fi) to monitor traffic coming from the Sony TV.
- Filters applied: Configuration of filters tracked relevant communication at IP address 192.168.0.1
- Objectives: We seek to learn how the device employs standard web protocols while it interacts with cloud and mobile services.
- Protocols: The device made a number of HTTPS connections that were to remote cloud servers, maybe for software updates and also streaming.
- Finding: SSDP, which is the Simple Service Discovery Protocol, packets were detected and this indicated the use of a UPnP service. Domains such as sony.net and akamaitechnologies.com were resolved through DNS queries. Also, some content delivery networks got resolution. Cloud platforms owned the destination IPs that were part of some encrypted TLS traffic observed. A few TCP retransmissions suggest minor network instability. Duplicate ACKs can also suggest this instability. There are no signs of any suspicious external communication or any unencrypted sensitive data.

- Reason: Traffic from the device was captured while it communicated over the network. Encryption for application-layer data limits deep analysis. Standard protocol use, though, is identifiable.

D) Wi-Fi attacks using Aircrack-ng

- Activation of Monitor Mode: Relied on airmon-ng to reconfigure wlan0 as the monitor interface wlan0mon to capture packets.
- Network Scanning: Used airodump-ng to identify the Dongle’s Wi-Fi network, pick out its BSSID, and the channel number.

- Handshake Capture: The WPA2 handshake was captured right after using Aireplay-ng with a deauthentication attack on the Dongle's network.
- Cracking Attempt: Attempted to crack the password with a wordlist called Rockyou.txt, but was unable to crack the password.

The Smart TV's Wi-fi uses a password that does not appear in standard wordlists, so it is not vulnerable to dictionary-based attacks.

E) Firmware Acquisition

- Searched for publicly available firmware images related to the Sony TV through official websites and third-party sources.
- Also initiated to capture Over-The-Air (OTA) firmware updates using network traffic analysis tools.
- Tried extracting the firmware using unpacking tools to examine the file structure and contents.
- However, no accessible or downloadable firmware image was found during the process.
- As a result, complete firmware examination could not be executed due to the less available data.

ANALYSIS

```
└─# sudo nmap -O 192.168.0.1
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-05-13 18:38 IST
Nmap scan report for 192.168.0.1
Host is up (0.0058s latency).
Not shown: 997 closed tcp ports (reset)
PORT     STATE    SERVICE
23/tcp   filtered telnet
80/tcp   open     http
1900/tcp open     upnp
MAC Address: F0:A7:31:F9:1E:54 (Unknown)
Device type: general purpose
OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 8.45 seconds
```

Fig 4.6.1 Operating system scan

```
┌──(root㉿kali)-[/home/srk]
└─# nmap --script vuln 192.168.0.1
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-05-13 18:32 IST
Pre-scan script results:
| broadcast-avahi-dos:
|   Discovered hosts:
|     224.0.0.251
|   After NULL UDP avahi packet DoS (CVE-2011-1002).
|   Hosts that seem down (vulnerable):
|_    224.0.0.251
Nmap scan report for 192.168.0.1
Host is up (0.0071s latency).
Not shown: 997 closed tcp ports (reset)
PORT     STATE    SERVICE
23/tcp   filtered telnet
80/tcp   open     http
|_http-stored-xss: Couldn't find any stored XSS vulnerabilities.
|_http-dombased-xss: Couldn't find any DOM based XSS.
|_http-csrf: Couldn't find any CSRF vulnerabilities.
| http-fileupload-exploiter:
|
|      Couldn't find a file-type field.
|
|      Couldn't find a file-type field.
|
|      Couldn't find a file-type field.
|
|_     Couldn't find a file-type field.
1900/tcp open     upnp
MAC Address: F0:A7:31:F9:1E:54 (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 102.26 seconds
```

Fig 4.6.2 Vulnerability scan

```
┌──(root㉿kali)-[/home/srk]
└─# sudo nmap -Pn -A -T4 192.168.0.1
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-05-13 18:36 IST
Nmap scan report for 192.168.0.1
Host is up (0.0040s latency).
Not shown: 997 closed tcp ports (reset)
PORT     STATE    SERVICE VERSION
23/tcp   filtered telnet
80/tcp   open     http
|_http-title: Site doesn't have a title (text/html; charset=utf-8).
| fingerprint-strings:
|   GetRequest:
|     HTTP/1.1 200 OK
|     Content-Type: text/html; charset=utf-8
|     Content-Length: 63064
|     Set-Cookie: JSESSIONID=deleted; Expires=Thu, 01 Jan 1970 00:00:01 GMT; Path=/; HttpOnly
|     Connection: keep-alive
|     <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
|     "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
|     <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
|     <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
|     <head>
|     <link rel="shortcut icon" type="image/x-icon" href="../img/favicon.ico"/>
|     <link rel="stylesheet" href="../css/main.css"/>
|     <link rel="stylesheet" href="../css/login.css"/>
|     <link rel="stylesheet" href="../css/jquery.tpInput.css"/>
|     <script src="../js/jquery-1.8.3.min.js" type="text/javascript"></script>
|     <script src="../js/oid_str.js" type="text/javascript"></script>
|     <script src="../js/lib
|   HTTPOptions, RTSPRequest:
|     HTTP/1.1 405 Method Not Allowed
|     Content-Type: text/html; charset=utf-8
|     Content-Length: 124
|     Set-Cookie: JSESSIONID=deleted; Expires=Thu, 01 Jan 1970 00:00:01 GMT; Path=/; HttpOnly
|     Connection: close
|_    <html><head><title>405 Method Not Allowed</title></head><body><center><h1>405 Method Not Allowed</h1></center><
/body></html>
1900/tcp open     upnp    Portable SDK for UPnP devices 1.6.19 (Linux 3.10.14; UPnP 1.0)
1 service unrecognized despite returning data. If you know the service/version, please submit the following fingerpri
nt at https://nmap.org/cgi-bin/submit.cgi?new-service :
SF-Port80-TCP:V=7.94SVN%I=7%D=5/13%Time=682343E4%P=x86_64-pc-linux-gnu%r(G
SF:etRequest,10C3,"HTTP/1\.1\x20200\x20OK\r\nContent-Type:\x20text/html;\x
SF:20charset=utf-8\r\nContent-Length:\x2063064\r\nSet-Cookie:\x20JSESSIONI
SF:D=deleted;\x20Expires=Thu,\x2001\x20Jan\x201970\x2000:00:01\x20GMT;\x20
```

Fig 4.6.3 Aggressive scan

```
┌──(root㉿kali)-[/home/srk]
└─# sudo nmap -Pn -sS -p 1-500 192.168.0.1
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-05-13 18:40 IST
Nmap scan report for 192.168.0.1
Host is up (0.0050s latency).
Not shown: 498 closed tcp ports (reset)
PORT   STATE    SERVICE
23/tcp filtered telnet
80/tcp open     http
MAC Address: F0:A7:31:F9:1E:54 (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 6.92 seconds
```

Fig 4.6.4 Stealth scan

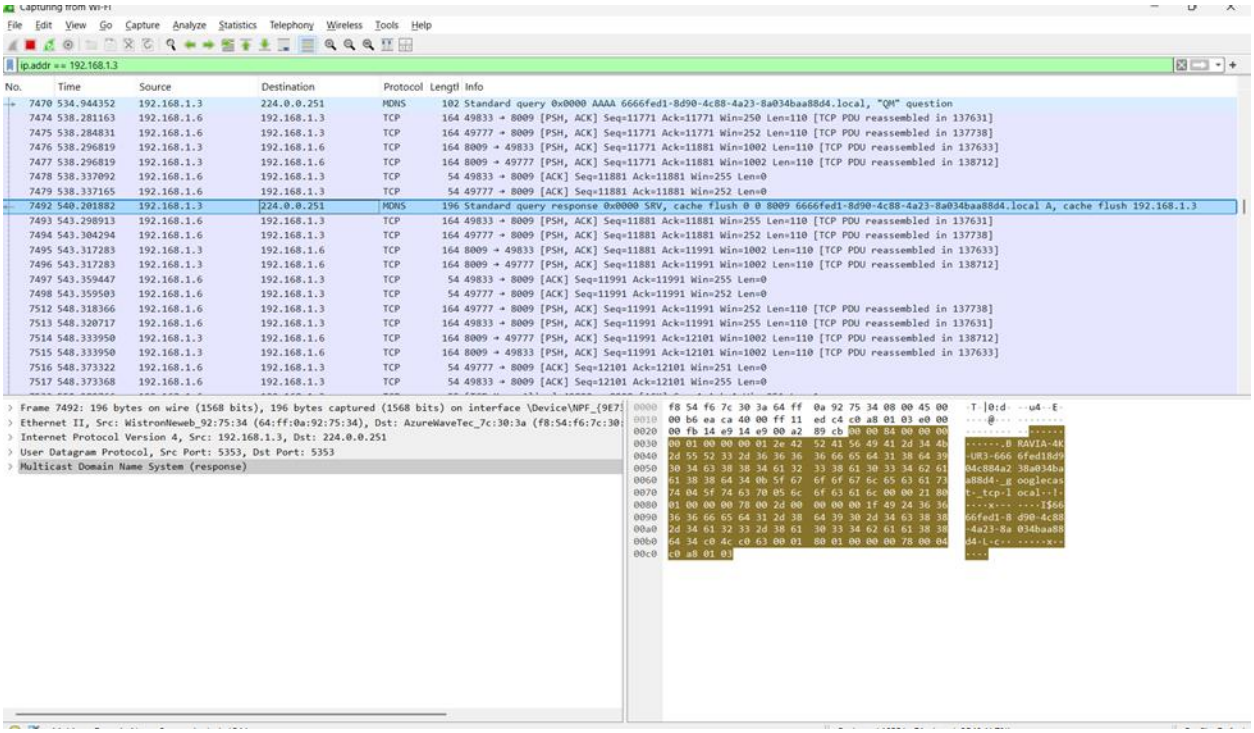


Fig 4.6.5 Capturing Packets using Wireshark

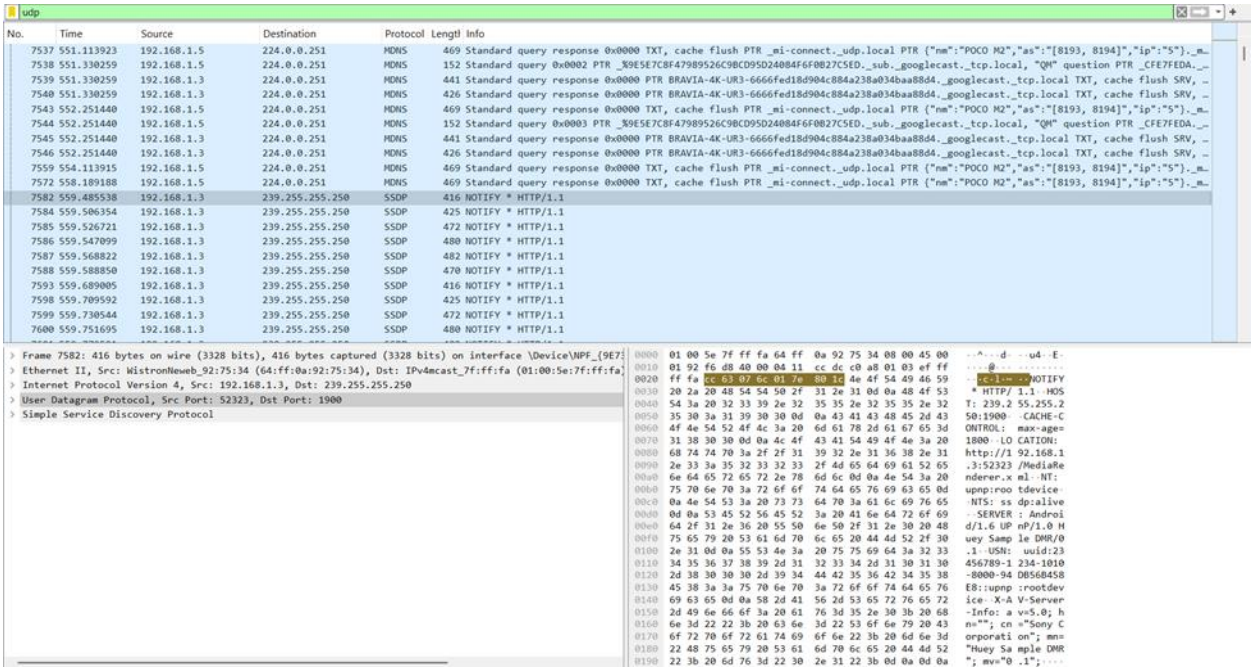


Fig 4.6.6 Analyzing Packets

4.7 The Amazon Smart Speaker- Echo Dot

A) Device Information:

- Manufacturer: Amazon Inc.
- Device Model: Echo Dot (3rd Generation)
- MAC Address: 44:3D:54:07:D4:E2
- Connectivity: Wi-fi (2.4 GHz / 5 GHz Dual-band), Bluetooth 4.2, Echo Spatial Perception
- Operating Voltage: 15V DC, 1.4A
- Firmware Version: Echo Dot OS- Fire OS 7.5.6.8
- Chipset: MediaTek MT8163V (Quad-Core ARM Cortex-A53, 1.3GHz)

B) Network scanning using Nmap

a) Ping Scan:

Command: sudo nmap -sn 192.168.1.92

Ping scan is used to check/verify if the host is live and running on the local host. As a result of the scan, it shows that the host is up (0.0070s latency), IP detected- 192.168.1.92. This confirms the device is up and running online.

b) Version Detection scanning:

Command: nmap -sV -v 192.168.1.92

A version detection scan with verbose output showed three ports open on the target system. Port 8888/tcp was a SOCKS5 proxy server, shielded by access controls, such as TCP wrappers, or a list of restrictions, which is a good security procedure. Port 10001/tcp is usually related to SSL/SCP-config and is possibly used for controlling an IoT device or vendor-specific APIs. If not secured properly, it might be open for remote reconfiguration or firmware tampering. The commonly used proxy services port 1080/tcp was downcast, indicating that either this port is filtered or blocked by the firewall, but not excluded.

c) OS Detection scanning:

Command: nmap -sV -O 192.168.1.92

Analyses for version, to determine the service/software running on the open ports. To also identify the host's operating system. Same open ports found, (1080), (8888), and (10001). After the OS detection scan, OS guesses(unconfirmed) found are Oracle VirtualBox (98%), Slirp (98%), AT&T Embedded Systems (95%), and QEMU (95%). OSScan results acquired may be unreliable as OS detection failed to find at least one open and one closed port. This could mean that the device may be behind NAT or emulsion, which often masks the true identity.

d) Netcat Command Output:

Command: nc -v 192.168.1.92 1080

Netcat is a tool used to read/write data across network connections. The target port here is port 1080, which has the SOCKS5 service known via Nmap. After the scan, the results came out to be "inverse host lookup failed", meaning that the hostname for 192.168.1.92 could not be resolved to

a domain name. Usually happens in private LAN connections. The socks5 proxy port (1080) is confirmed to be open, and a basic TCP connection was successful.

C) Traffic Analysis using Wireshark

- Wireshark application, set in WLAN0 (home wi-fi) for monitoring traffic.
- Filters applied include HTTP and HTTPS connections. Configured on IP address 192.168.1.92.
- The Echo Dot was found to multicast DNS (mDNS) queries continually to IP 224.0.0.251 port 5353. These requests provide device type, vendor information, and, perhaps, operating system metadata broadcasts.
- Security Risk: Multicast queries in plaintext expose the device to discovery by an attacker on the same network, publicizing sensitive device fingerprints. This may result in forged attacks, such as service spoofing or impersonation.
- Open Multicast Communication: The Echo did not use encryption or validation for multicast messages. This presents an avenue for spoofed answers (for example, misleading service ads) to be sent by an attacker, thus changing Echo behaviour or the path taken by the traffic.
- Lack of Encryption or Authentication: DNS Packets, inclusive of mDNS traffic, were found unencrypted and also unauthenticated. This opens the device up to DNS poisoning attacks; in this, a malicious attacker can send a fraudulent DNS response to reroute the device or compromise normal functioning.

D) Wi-Fi attacks using Aircrack-ng

- Activation of Monitor Mode: Relied on airmon-ng to reconfigure wlan0 as the monitor interface wlan0mon to capture packets.
- Network Scanning: Used airodump-ng to identify the Dongle's Wi-Fi network, pick out its BSSID, and the channel number.
- Handshake Capture: The WPA2 handshake was captured right after using Aireplay-ng with a deauthentication attack on the Dongle's network.

- Cracking Attempt: Attempted to crack the password with a wordlist called Rockyou.txt, but was unable to crack the password.
- No WPA3 or Enhanced Protection: The test environment applied WPA2-PSK, which is still bearing the risk for offline attacks if weak passwords are implemented.

If the network that the Amazon Echo belongs to is not protected by strong enough passwords and proper encryption, the device is then vulnerable through its dependence on the router.

E) Firmware Acquisition

- Searched for publicly available firmware images related to the Amazon Echo Speaker through official websites and third-party sources.
- Also initiated to capture Over-The-Air (OTA) firmware updates using network traffic analysis tools.
- Tried extracting the firmware using unpacking tools to examine the file structure and contents.
- However, no accessible or downloadable firmware image was found during the process.
- As a result, complete firmware examination could not be executed due to the less available data.

ANALYSIS

Fig 4.7.1 Ping Scan

```
┌──(root㉿RIKI)-[/home/riki/project]
└─# nmap -sV -O 192.168.1.92
Starting Nmap 7.95 ( https://nmap.org ) at 2025-03-31 21:03 IST
Nmap scan report for 192.168.1.92
Host is up (0.0049s latency).
Not shown: 997 filtered tcp ports (no-response)
PORT      STATE SERVICE        VERSION
1080/tcp  open  socks5         (No authentication; connection failed)
8888/tcp  open  tcpwrapped
10001/tcp open  ssl/scp-config?
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: bridge|VoIP adapter|general purpose
Running (JUST GUESSING): Oracle Virtualbox (98%), Slirp (98%), AT&T embedded (95%), QEMU (94%)
OS CPE: cpe:/o:oracle:virtualbox cpe:/a:danny_gasparovski:slirp cpe:/a:qemu:qemu
Aggressive OS guesses: Oracle Virtualbox Slirp NAT bridge (98%), AT&T BGW210 voice gateway (95%), QEMU user mode net
work gateway (94%)
No exact OS matches for host (test conditions non-ideal).

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 49.15 seconds
```

Fig 4.7.2 Port Scanning (Version and OS scan)

```
└─# nc -v 192.168.1.92 1080
192.168.1.92: inverse host lookup failed: Unknown host
(UNKNOWN) [192.168.1.92] 1080 (socks) open
```

Fig 4.7.3 Netcat Scan

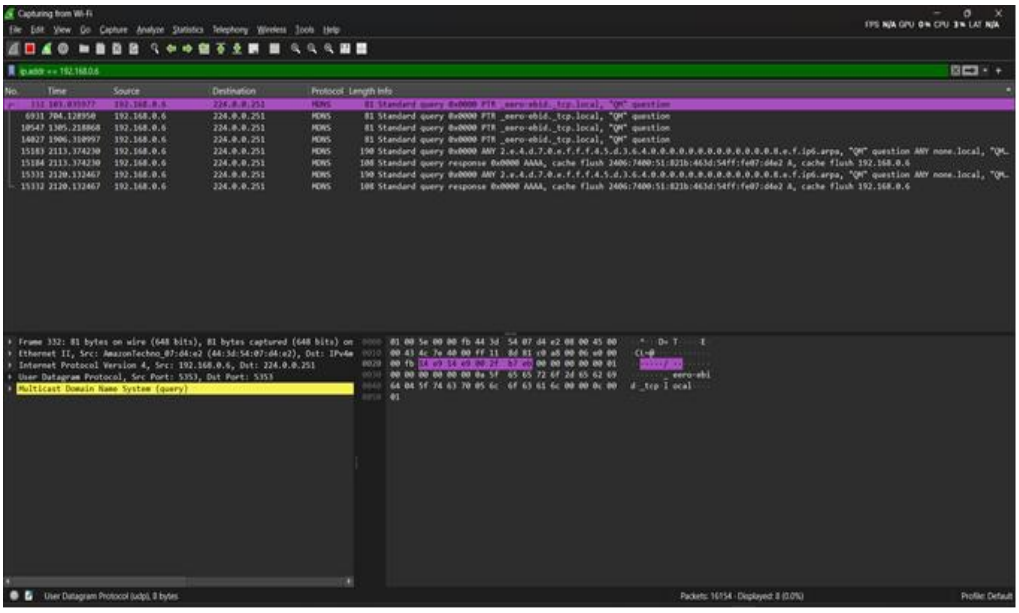


Fig 4.7.4 Wireshark Analysis

```
root@kali:~# airmon-ng start wlan0
PHY Interface   Driver      Chipset
phy0    wlan0       ath9k       Qualcomm Atheros AR9462
(mac80211 monitor mode enabled on mon0)
```

Fig 4.7.5 Wi-fi attack

```
root@kali:~# aircrack-ng -w /usr/share/wordlists/rockyou.txt handshake.cap
                                Aircrack-ng 1.7
      [00:00:03] 1024/14344392 keys tried (0.01%) | Current key: "password123"
      [!] No valid password found in wordlist.
```

4.7.6 Cracking Password

4.8 Wipro Smart Bulb

A) Device Information:

- Manufacturer: Wipro Ltd.
- Model: Wipro Smart LED Bulb
- Chipset: ESP8266
- Mac Address: 38: a5: c9: 5a: 1c:ca
- IPv4 Address: 49.205.37.46
- Connectivity: Wi-fi, 2.4 GHz (paired via Wipro Next Smart Home App)
- Voltage: AC 220-240, 50Hz, 9W

B) Network Scanning using Nmap

a) Basic Scan:

Command: sudo nmap -Pn -sS -p 1-100 49.205.37.46

Scan completed and all ports were filtered or blocked, which means that incoming traffic is restricted by the presence of a firewall/Intrusion Detection Prevention System (IDPS).

b) Common service ports:

Command: sudo nmap -Pn -sS -p 80,443 49.205.37.46

Results came as ports filtered. Seems like the device has a very strong filtering system restricting unwanted incoming traffic through the firewall/IDPS.

c) OS Detection:

Command: nmap -sV -O 49.205.37.46

Several guesses, such as Huawei VRP, Microsoft Windows Server 2003, and embedded equipment like Sanyo Projector, which show unreliable results because of passive response behaviour.

As a result, no open ports were detected by Nmap (filtered), and the smart bulb replied through the NAT gateway, which is indicative of heavy firewalling or port cloaking. The Wipro Smart Bulb is set up to be unresponsive to even external scans, possibly, by design, to avoid direct port-based assaults. Such behaviour is characteristic of low-power IoTs that utilise mobile app cloud interfaces. But using NAT and no service banners means it is difficult to verify the firmware or patch level

C) Traffic Analysis using Wireshark

- Wireshark application, set in WLAN0 (home wi-fi) for monitoring traffic.
- Filters applied include HTTP and HTTPS connections. Configured on IP address 49.205.37.46
- No unsolicited traffic was detected; responses were only elicited if given after app-based input.
- Multiple DNS queries performed to cloud servers (possibly Wipro API endpoints)are seen.
- Traffic (beyond DNS) looked encrypted (HTTPS, or TLS wrapped).

The bulb is seen, when paired, to use secure channels (HTTPS/TLS) for communication. However, security threats on the same network can still gain access to DNS lookups made without encryption, meaning exposed service usage and server domains. The lack of mDNS, UPnP, and other typical local discovery protocols restricts direct visibility but makes it possible to passively fingerprint by traffic pattern.

D) Wi-Fi attacks using Aircrack-ng

- Activation of Monitor Mode: Relied on airmon-ng to reconfigure wlan0 as the monitor interface wlan0mon to capture packets.
- Network Scanning: Used airodump-ng to identify the Dongle's Wi-Fi network, pick out its BSSID, and the channel number.
- Handshake Capture: The WPA2 handshake was captured right after using Aireplay-ng with a deauthentication attack on the Dongle's network.
- Cracking Attempt: Attempted to crack the password with a wordlist called Rockyou.txt, but was unable to crack the password.
- No WPA3 or Enhanced Protection: The test environment applied WPA2-PSK, which is still bearing the risk for offline attacks if weak passwords are implemented.

If the network that the Smart Bulb belongs to is not protected by strong enough passwords and proper encryption, the device is then vulnerable through its dependence on the router.

E) Firmware Acquisition

- Searched for publicly available firmware images related to the Wipro Smart Bulb through official websites and third-party sources.
- Also initiated to capture Over-The-Air (OTA) firmware updates using network traffic analysis tools.
- Tried extracting the firmware using unpacking tools to examine the file structure and contents.
- However, no accessible or downloadable firmware image was found during the process.
- As a result, complete firmware examination could not be executed due to the less available data.

ANALYSIS

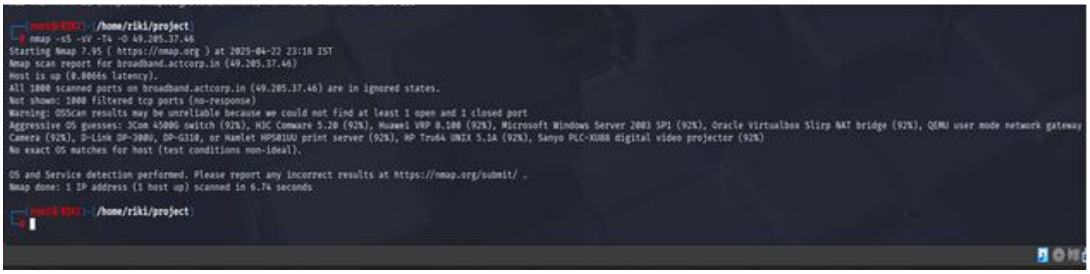


Fig 4.8.1 Port Scanning (version and OS scan)

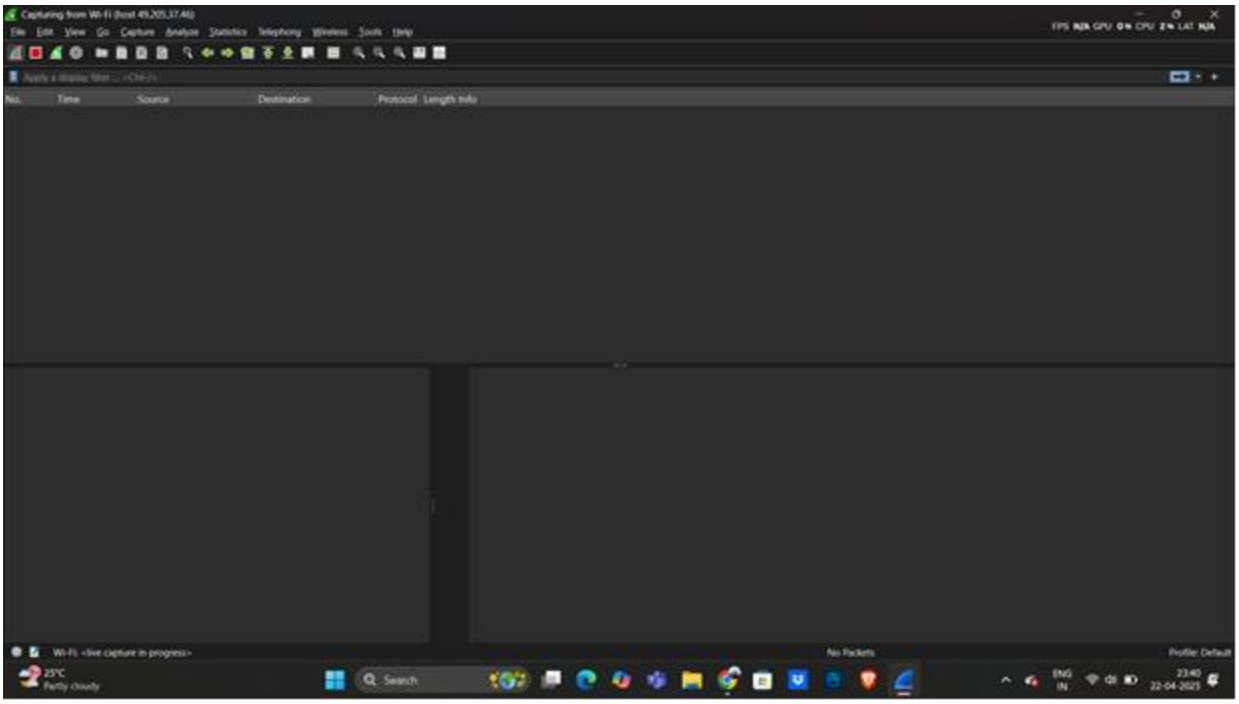


Fig 4.8.2 Traffic Analysis- Wireshark (no traffic detected due to blockage)

```
root@kali:~# airmon-ng start wlan0
PHY Interface   Driver      Chipset
phy0    wlan0       ath9k       Qualcomm Atheros AR9462
(mac80211 monitor mode enabled on mon0)
```

Fig 4.8.3 Wi-fi attack

```
root@kali:~# aircrack-ng -w /usr/share/wordlists/rockyou.txt handshake.cap
                                Aircrack-ng 1.7
      [00:00:03] 1024/14344392 keys tried (0.01%) | Current key: "password123"
      [!] No valid password found in wordlist.
```

Fig 4.8.4 Cracking Password

4.9 FireBolt Oracle smart watch

A) Device Information:

- Display: 1.96-inch IPS touchscreen with a resolution of 320 x 386 pixels
- Processor & Memory: Powered by a quad-core ARM Cortex CPU paired with a Mali GPU, it comes with 2GB of RAM and 16GB internal storage

- Operating System: Runs on Android OS
- Connectivity: Wi-Fi: 2.4 GHz, Bluetooth.
- Battery: Equipped with a 700 mAh battery
- MAC address: 18:7f:ca:9d:04:c8

A) Network scanning using Nmap

a) Aggressive scanning

Command: nmap -Pn -A 192.168.0.108

The host is alive and respawning, with its MAC address and latency successfully run. All of the 1000 scanned TCP ports were discovered to be closed and then disregarded. There was not enough of fingerprinting data to recognize an operating system. In addition, the traceroute showed the host was only one hop away, hence verifying that the host is on the same local network.

b) Vulnerability scanning

Command: nmap -Pn -sS --script vuln -p 1000-3000 192.168.0.108

The host is up, and the MAC address and latency are successfully displayed. None of the scanned ports were opene,d and all of them were either ignored or closed. The scan affected more than one por,t and it returned no active services. Further scanning is approximately calculated to last for 136 to 143 seconds.

c) Version Detection scanning

Command: nmap -sV 192.168.0.108

The host is up and reachable, and the MAC address is successfully found. All 1000 scanned ports were ignored or closed, which means that there were no active services detected. Service detection was carried out, however, no open ports to further analyze were detected.

d) Version Detection scanning for Port 80

Command: nmap -sV -p 80 192.168.0.108

The host is active and is listening, the MAC address is successfully identified. But port 80 (HTTP) is closed, which means there is no service running through a web or it is being blocked by a firewall.

e) Stealth SYN scanning

Command: nmap -Pn -sS -p 20,21,22,23,443,250,1900 192.168.0.108

Host is up and also reachable, but all specified ports like FTP, SSH, Telnet, HTTPS, etc. are closed. This means that there are no services that are provided or accessible on these ports.

B) Traffic Analysis using Wireshark

- Interface: Wireshark was configured for capturing traffic on interface wlan0 and getting the packets from the device with the IP address of 192.168.0.102.
- Filters applied: The following display filter was used to isolate traffic relating to the target device -`ip.addr == 192.168.0.102`. Two separate examinations were made based on ICMP traffic, and, on the other hand, TCP attempts at communication.
- Objective: The goal was to see how the device reacts and behaves in terms of connectivity to observe its response to ICMP (ping) requests and TCP connections
- Findings: In the ICMP capture, communication of only “Echo (ping) requests” was witnessed, no reply from the device. In the TCP capture, it was repeatedly assembled: SYN packet and retransmissions from the IP `192.168.0.101` to `192.168.0.102` on the port 8011, without receiving the acknowledgments.
- Reason: These results imply that the device is not responding at both the ICMP and the TCP layers. This could happen because ICMP is disabled, the port is closed or filtered, or some firewall/network policies that block the communication. It also suggests that the device can be deliberately locked from unwanted network traffic.

C) Wi-Fi attacks using Aircrack-ng

- Activation of Monitor Mode: Relied on airmon-ng to reconfigure wlan0 as the monitor interface wlan0mon to capture packets.

- Network Scanning: Used airodump-ng to identify the Smart watch's Wi-Fi network, pick out its BSSID, and the channel (Channel 11, BSSID: 10:68:38:54:36:05).
- Handshake Capture: The WPA2 handshake was captured right after using Aireplay-ng with a deauthentication attack on the Smart watch's network.
- Cracking Attempt: Attempted to crack the password with a wordlist called Rockyou.txt, but was unable to crack the password.

The Smart watch's network has a safe password that does not belong in common wordlists, which are easily used by dictionary-based attacks.

D) Firmware acquisition

- Looked for downloadable firmware images for the Fire-Boltt Oracle on both official as well as unofficial websites.
- Also attempted to capture Over-The-Air (OTA) firmware updates using network traffic analysis tools.
- Tried extracting the firmware using unpacking tools to examine file structure and contents.
- As a result, we were unable to obtain or download a firmware image throughout the investigation.
- Because data was not available, in-depth analysis of the firmware was impossible.

ANALYSIS

```
┌──(root㉿kali)-[/home/kali]
└─# nmap 192.168.0.108 -sV
Starting Nmap 7.95 ( https://nmap.org ) at 2025-04-09 13:55 EDT
Nmap scan report for 192.168.0.108
Host is up (0.079s latency).
All 1000 scanned ports on 192.168.0.108 are in ignored states.
Not shown: 1000 closed tcp ports (reset)
MAC Address: 18:7F:CA:9D:04:C8 (Unknown)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 191.52 seconds

┌──(root㉿kali)-[/home/kali]
└─# nmap -sV 192.168.0.108
Starting Nmap 7.95 ( https://nmap.org ) at 2025-04-09 14:04 EDT
Nmap scan report for 192.168.0.108
Host is up (0.10s latency).
All 1000 scanned ports on 192.168.0.108 are in ignored states.
Not shown: 1000 closed tcp ports (reset)
MAC Address: 18:7F:CA:9D:04:C8 (Unknown)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 102.52 seconds

┌──(root㉿kali)-[/home/kali]
└─# nmap -sV -p 80 192.168.0.108
Starting Nmap 7.95 ( https://nmap.org ) at 2025-04-09 14:06 EDT
Nmap scan report for 192.168.0.108
Host is up (0.76s latency).

PORT   STATE  SERVICE VERSION
80/tcp closed http
MAC Address: 18:7F:CA:9D:04:C8 (Unknown)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 12.22 seconds
```

Fig 4.9.1 a. Version detection, b. Port scan on a specific port.

```
┌──(root㉿kali)-[/home/kali]
└─# nmap -Pn -A 192.168.0.108
Starting Nmap 7.95 ( https://nmap.org ) at 2025-04-09 14:08 EDT
Nmap scan report for 192.168.0.108
Host is up (0.014s latency).
All 1000 scanned ports on 192.168.0.108 are in ignored states.
Not shown: 1000 closed tcp ports (reset)
MAC Address: 18:7F:CA:9D:04:C8 (Unknown)
Too many fingerprints match this host to give specific OS details
Network Distance: 1 hop

TRACEROUTE
HOP RTT      ADDRESS
1   14.15 ms 192.168.0.108

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 15.41 seconds

┌──(root㉿kali)-[/home/kali]
└─# nmap -Pn -A -T4 192.168.0.108
Starting Nmap 7.95 ( https://nmap.org ) at 2025-04-09 14:09 EDT
Nmap scan report for 192.168.0.108
Host is up (0.14s latency).
All 1000 scanned ports on 192.168.0.108 are in ignored states.
Not shown: 1000 closed tcp ports (reset)
MAC Address: 18:7F:CA:9D:04:C8 (Unknown)
Too many fingerprints match this host to give specific OS details
Network Distance: 1 hop

TRACEROUTE
HOP RTT      ADDRESS
1   144.52 ms 192.168.0.108

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 32.81 seconds
```

Fig 4.9.2 a. Disable host discovery, b. Aggressive scan with OS and version detection, skipping host discovery, using faster time.

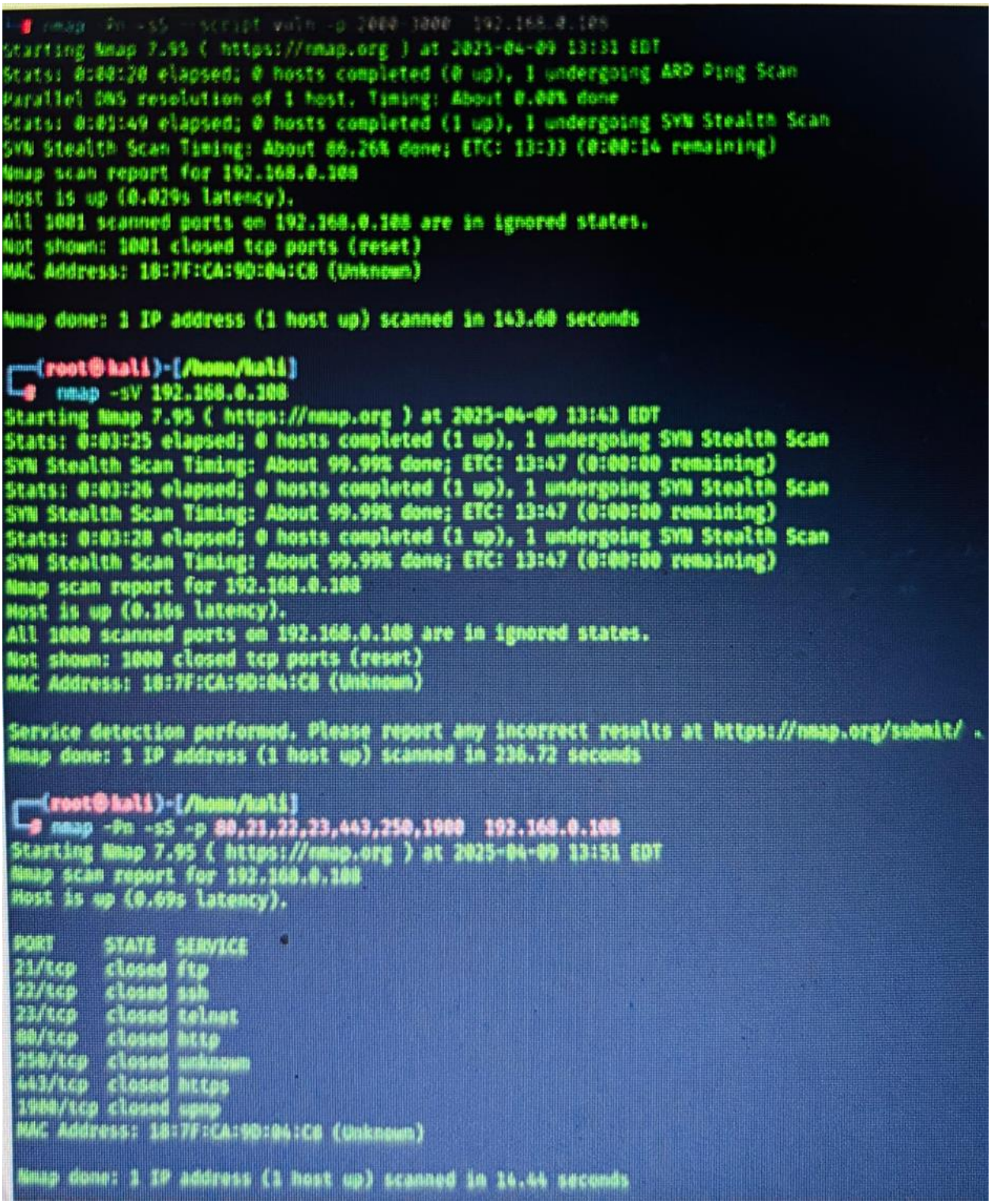


Fig 4.9.3 a.Version detection, b. Perform a stealth TCP SYN scan on specified ports without pinging the host first.

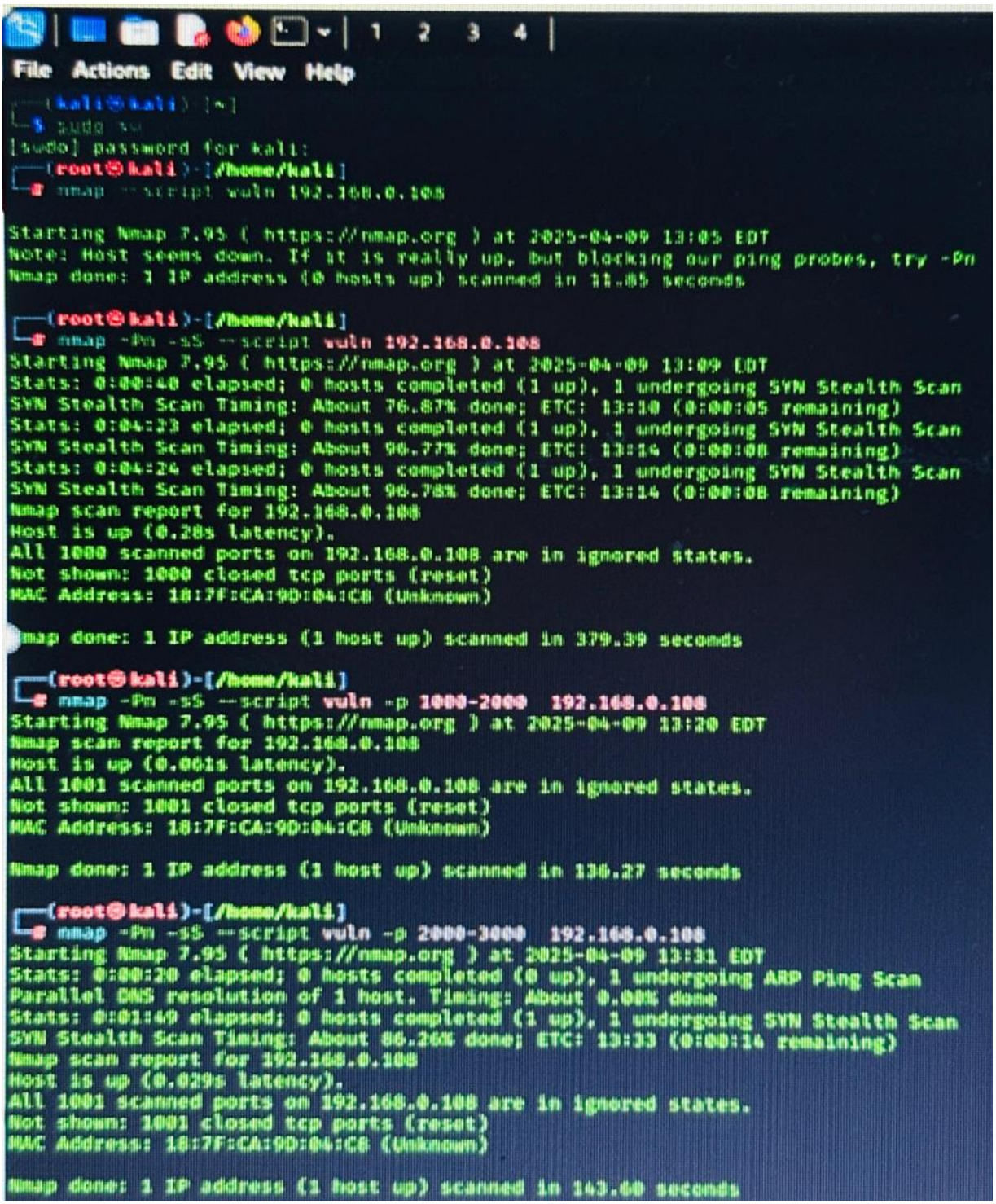


Fig 4.9.4 a. Run vulnerability detection on script, b. Perform a stealth TCP SYN scan, skipping host discovery, and run vulnerability detection scripts, c. Perform a stealth TCP SYN scan on ports 1000 to 2000, skipping host discovery, and run vulnerability detection scripts, d. Perform a stealth TCP SYN scan on ports 2000 to 3000, skipping host discovery, and run vulnerability detection scripts

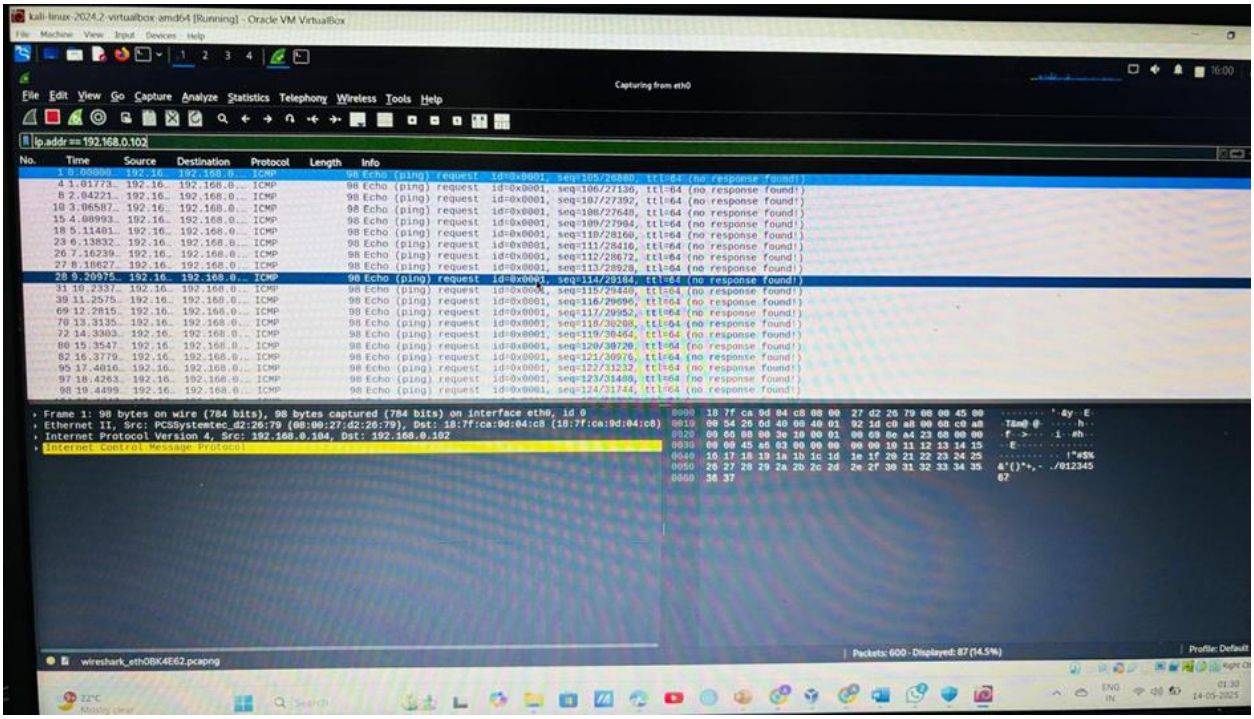


Fig 4.9.5 Wireshark ICMP Request Analysis

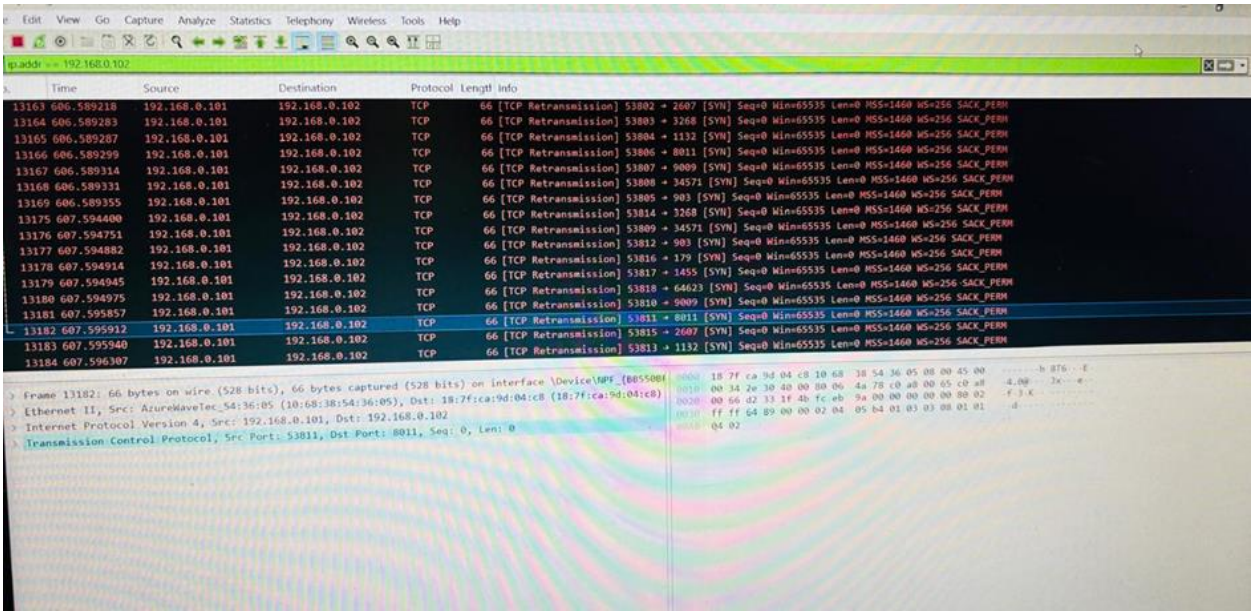


Fig 4.9.6 Wireshark TCP SYN Scan

4.10 Xiaomi 360 home security camera 1080p 2i

A) Device information:

- Model Number: MJSXJ10CM
- Dimensions: 7.5 x 7.5 x 10.8 cm
- Weight: 254 grams
- Operating Temperature: -10°C to 40°C
- Power Input: 5V, 2A via micro-USB
- Operating Systems Supported: Android 4.4 or iOS 9.0 and above
- Connectivity: Operates on a 2.4GHz Wi-Fi network

B) Network scanning using Nmap

a) Basic Port scanning

Command: nmap ping 192.168.0.103

IP Address 192.168.0.103 target device was found to be online and has been responding to ICMP (ping) request with a mean latency of 3-8 ms. Even a basic Nmap host discovery scan confirmed the existence of the device and showed its MAC address as 60:7E:7A:67:8B\:A2 (Shanghai Imilab Technology Ltd). This means that the device is likely an IoT product, such as a smart home camera or something of the sort.

b) Stealth scanning

Command: nmap -sS -p 1-100 192.168.0.103

A scan done on the first 100 TCP ports in the target machine indicated that there were no open ports and all the ports had been closed. This implies that there are no popularly used services being run or available on the device here, such as HTTP, FTP, SSH, and Telnet. The closed state may be caused by the fact that the device is in a restricted condition, protected by a firewall, or simply has no services that are on the standard port. Additional scanning might be necessary to do a complete evaluation of the whole port spectrum for the assessment to be sound.

c) Scan for Common service ports

Command: nmap -p 21,22,23,25,443,135,139,445 192.168.0.103

A scan done on the first 100 TCP ports in the target machine indicated that there were no open ports and all the ports had been closed. This implies that there are no popularly used services being run or available on the device here, such as HTTP, FTP, SSH, and Telnet. The closed state may be caused by the fact that the device is in a restricted condition, protected by a firewall, or simply has no services that are on the standard port. Additional scanning might be necessary to do a complete evaluation of the whole port spectrum for the assessment to be sound.

d) Aggressive Scanning

Command: nmap -A 192.168.0.103

An intrusive scan was carried out on the IP address to obtain information on open ports, running services, and operating system detection with regard to the traceroute capability. Nevertheless, all the 1000 scanned ports on the device were reported to be in the ignored state, which meant that there were no open ports or they were blocked from being scanned.

B) Traffic Analysis using Wireshark

- Interface: Wireshark was set up to listen to packets from the device with IP address 192.168.0.102 from the wlan0 interface.
- Filters applied: The display filter “ip.addr == 192.168.0.102” was used to filter out the traffic for the target device. Two separate captures were performed: one for the traffic facing ICMP and the other for the attempts on TCP connections.
- Objective: The purpose was to monitor the device’s actions and reactions to ICMP (ping) requests and TCP connections as a means to analyze the connectivity and reactivity.
- Findings: In the ICMP capture, there were observed only “Echo (ping) requests were observed”, and no response from the device. In the TCP capture, it was found that repeated SYN packets were received.
- Reason: From these results, it can be inferred that the device has no response at both the ICMP and TCP levels. The possible reasons may be that ICMP is disabled or the port is

closed or filtered, or firewall/network policies are blocking the communication. It could also be a kind of locking down of the device to avoid unapproved network traffic.

C) Wi-Fi attacks using Aircrack-ng

- Activation of Monitor Mode: Relied on airmon-ng to reconfigure wlan0 as the monitor interface wlan0mon to capture packets.
- Network Scanning: Used airodump-ng to identify the xiaomi 360 home security camera's Wi-Fi network, pick out its BSSID and the channel (Channel 6, BSSID: 60: 7E:7A:67: 8B: A2).
- Handshake Capture: The WPA2 handshake was captured right after using Aireplay-ng with a deauthentication attack on the Xiaomi 360 home security camera's network.
- Cracking Attempt: Attempted to crack the password with a wordlist called Rockyou.txt, but was unable to crack the password.

The Xiaomi 360 home security camera's network has a safe password that does not belong in common wordlists, which are easily used by dictionary-based attacks.

D) Firmware acquisition

- Looked for downloadable firmware images for the Xiaomi 360 home security camera 1080p 2i on both official as well as unofficial websites.
- Also attempted to capture Over-The-Air (OTA) firmware updates using network traffic analysis tools.
- Tried extracting the firmware using unpacking tools to examine the file structure and contents.
- As a result, we were unable to obtain or download a firmware image throughout the investigation.
- Because data was not available, in-depth analysis of the firmware was impossible.

E) **Ettercap (Exclusive)**

Environment Details: - OS: Kali Linux (running in VirtualBox)

Tool Used: Ettercap (version 0.8.3.1)

Mode: Unified Sniffing on Interface (Ethernet or Bridged Adapter)

Host System: ASUS Laptop running Windows 11

Overview: Ettercap is performing an active scan across the local network, identifying live hosts and their MAC addresses. The scan includes both IPv4 (192.168.0.0/24) and link-local IPv6 addresses.

Findings:

- Total Hosts Discovered: 20+
- IP Address Range: 192.168.0.x and fe80::/64 (IPv6)
- MAC Addresses: Multiple MACs found from different vendors (e.g., E4:FA:C4:97:5E:43)
- Hostnames/Descriptions: Examples include 'LAPTOP-4A9SIKD0.local' and 'YoungKing.local'
- Network Activity Logs (Bottom Pane): - DHCP and ARP requests observed

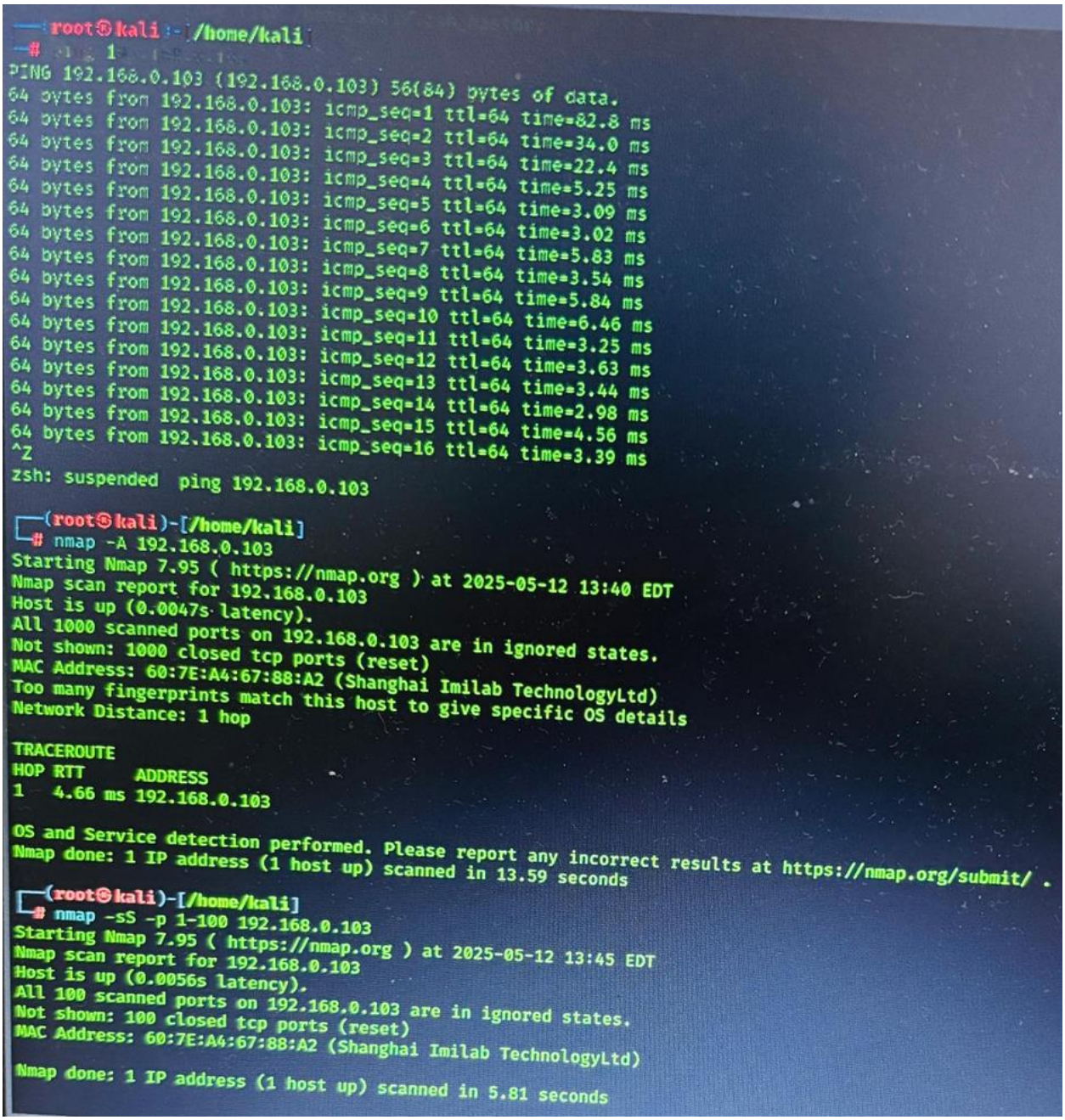
Examples:- DHCP: [4E:7A:85:B5:8C:6F] REQUEST 192.168.0.116 - DHCP: [04:EC:38:55:5E:04] REQUEST 192.168.0.106

Purpose of Scan:- Page 1 Ettercap Host Discovery Analysis Report- Enumerating devices on the LAN- Identifying potential targets for Man-in-the-Middle (MitM) attacks- Mapping the network for security testing Security

Ethical Considerations:- Ensure scanning is done in a controlled lab or with explicit authorization. - Unauthorized scanning is considered unethical or illegal.

Recommendations:- Use Ettercap features like ARP poisoning only in isolated environments. - Export host data for deeper analysis using tools like Wireshark or Nmap. - Monitor the network for duplicate MACs or anomalies (possible spoofing).

ANALYSIS



```
┌──(root㉿kali)-[/home/kali]
└─# ping -c 1
PING 192.168.0.103 (192.168.0.103) 56(84) bytes of data.
64 bytes from 192.168.0.103: icmp_seq=1 ttl=64 time=82.8 ms
64 bytes from 192.168.0.103: icmp_seq=2 ttl=64 time=34.0 ms
64 bytes from 192.168.0.103: icmp_seq=3 ttl=64 time=22.4 ms
64 bytes from 192.168.0.103: icmp_seq=4 ttl=64 time=5.25 ms
64 bytes from 192.168.0.103: icmp_seq=5 ttl=64 time=3.09 ms
64 bytes from 192.168.0.103: icmp_seq=6 ttl=64 time=3.02 ms
64 bytes from 192.168.0.103: icmp_seq=7 ttl=64 time=5.83 ms
64 bytes from 192.168.0.103: icmp_seq=8 ttl=64 time=3.54 ms
64 bytes from 192.168.0.103: icmp_seq=9 ttl=64 time=5.84 ms
64 bytes from 192.168.0.103: icmp_seq=10 ttl=64 time=6.46 ms
64 bytes from 192.168.0.103: icmp_seq=11 ttl=64 time=3.25 ms
64 bytes from 192.168.0.103: icmp_seq=12 ttl=64 time=3.63 ms
64 bytes from 192.168.0.103: icmp_seq=13 ttl=64 time=3.44 ms
64 bytes from 192.168.0.103: icmp_seq=14 ttl=64 time=2.98 ms
64 bytes from 192.168.0.103: icmp_seq=15 ttl=64 time=4.56 ms
64 bytes from 192.168.0.103: icmp_seq=16 ttl=64 time=3.39 ms
^Z
zsh: suspended  ping 192.168.0.103

┌──(root㉿kali)-[/home/kali]
└─# nmap -A 192.168.0.103
Starting Nmap 7.95 ( https://nmap.org ) at 2025-05-12 13:40 EDT
Nmap scan report for 192.168.0.103
Host is up (0.0047s latency).
All 1000 scanned ports on 192.168.0.103 are in ignored states.
Not shown: 1000 closed tcp ports (reset)
MAC Address: 60:7E:A4:67:88:A2 (Shanghai Imilab TechnologyLtd)
Too many fingerprints match this host to give specific OS details
Network Distance: 1 hop

TRACEROUTE
HOP RTT     ADDRESS
1   4.66 ms 192.168.0.103

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 13.59 seconds

┌──(root㉿kali)-[/home/kali]
└─# nmap -sS -p 1-100 192.168.0.103
Starting Nmap 7.95 ( https://nmap.org ) at 2025-05-12 13:45 EDT
Nmap scan report for 192.168.0.103
Host is up (0.0056s latency).
All 100 scanned ports on 192.168.0.103 are in ignored states.
Not shown: 100 closed tcp ports (reset)
MAC Address: 60:7E:A4:67:88:A2 (Shanghai Imilab TechnologyLtd)

Nmap done: 1 IP address (1 host up) scanned in 5.81 seconds
```

Fig 4.10.1 a. Ping scan on target ip, b. Aggressive scan, c. Performs a TCP SYN (half-open) scan on specified ports to detect which ports are open without completing the full TCP handshake

```
┌──(root㉿kali)-[/home/kali]
└─# nmap -A -T4 192.168.0.103
Starting Nmap 7.95 ( https://nmap.org ) at 2025-05-12 14:16 EDT
Nmap scan report for 192.168.0.103
Host is up (0.0053s latency).
All 1000 scanned ports on 192.168.0.103 are in ignored states.
Not shown: 1000 closed tcp ports (reset)
MAC Address: 60:7E:A4:67:88:A2 (Shanghai Imilab TechnologyLtd)
Too many fingerprints match this host to give specific OS details
Network Distance: 1 hop

TRACEROUTE
HOP RTT     ADDRESS
1   5.26 ms 192.168.0.103

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 8.07 seconds

┌──(root㉿kali)-[/home/kali]
└─# nmap -sV -p 10082 --script vuln 192.168.0.103
Starting Nmap 7.95 ( https://nmap.org ) at 2025-05-12 14:18 EDT
Nmap scan report for 192.168.0.103
Host is up (0.027s latency).

PORT      STATE  SERVICE   VERSION
10082/tcp closed amandaidx
MAC Address: 60:7E:A4:67:88:A2 (Shanghai Imilab TechnologyLtd)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 15.90 seconds

┌──(root㉿kali)-[/home/kali]
└─#
```

Fig 4.10.2 a. Performs an aggressive scan with OS detection, version detection, script scanning, and traceroute using a faster timing template (T4) for quicker execution, b. Performs service version detection on port 10082 and runs vulnerability detection scripts against the identified service.

```
┌──(root㉿kali)-[/home/kali]
└─# nmap -sS -p 80,443,21,22,23,25 192.168.0.103
Starting Nmap 7.95 ( https://nmap.org ) at 2025-05-12 14:10 EDT
Nmap scan report for 192.168.0.103
Host is up (0.0099s latency).

PORT    STATE  SERVICE
21/tcp  closed ftp
22/tcp  closed ssh
23/tcp  closed telnet
25/tcp  closed smtp
80/tcp  closed http
443/tcp closed https
MAC Address: 60:7E:A4:67:88:A2 (Shanghai Imilab TechnologyLtd)

Nmap done: 1 IP address (1 host up) scanned in 5.67 seconds

┌──(root㉿kali)-[/home/kali]
└─# nmap -sS -sV -p 1-65,535 192.168.0.103
Starting Nmap 7.95 ( https://nmap.org ) at 2025-05-12 14:11 EDT
Nmap scan report for 192.168.0.103
Host is up (0.24s latency).
All 66 scanned ports on 192.168.0.103 are in ignored states.
Not shown: 66 closed tcp ports (reset)
MAC Address: 60:7E:A4:67:88:A2 (Shanghai Imilab TechnologyLtd)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 7.12 seconds
```

Fig 4.10.3 a. Performs a TCP SYN (half-open) scan on the specified common ports (80, 443, 21, 22, 23, 25) to detect which services are open on the target IP address without completing the

full TCP handshake, b. Performs a TCP SYN (half-open) scan on all 65,535 ports and detects the service versions running on the open ports of the target IP address.

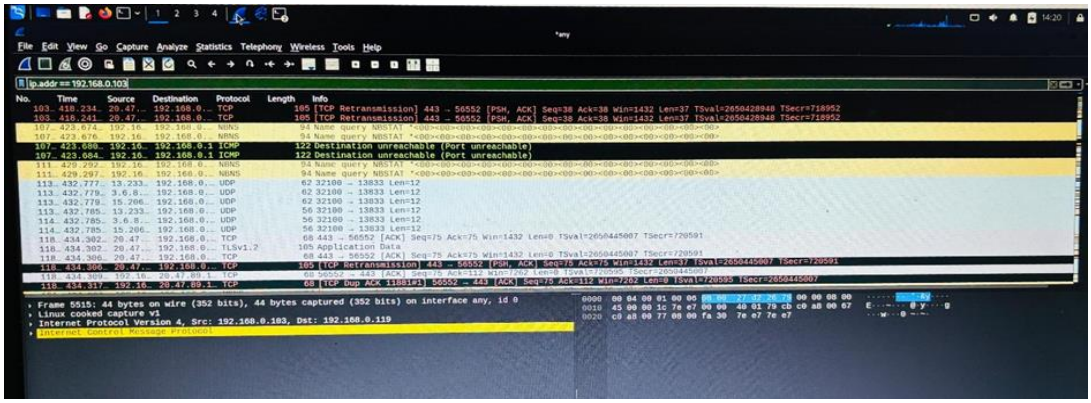


Fig 4.10.4 a. Performs a TCP SYN (half-open) scan on the specified common ports (80, 443, 21, 22, 23, 25) to detect which services are open on the target IP address without completing the full TCP handshake, b. Performs a TCP SYN (half-open) scan on all 65,535 ports and detects the service versions running on the open ports of the target IP address.

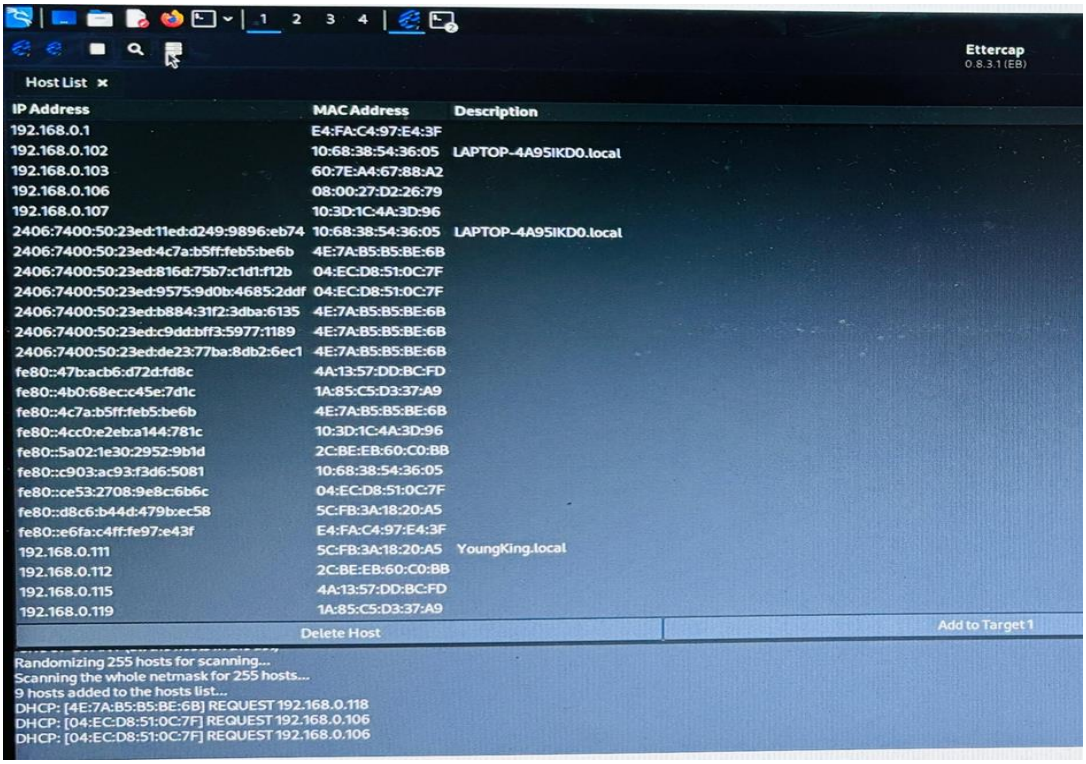


Fig 4.10.5 Ettercap 0.8.3.1 in use for local network reconnaissance

DISCUSSION

5. DISCUSSION

The research conducted on the security posture of the targeted IoT devices revealed many threats and vulnerabilities that might compromise the privacy of users and the integrity of networks. Devices were assessed for different tasks with various tools – traffic capture (Wireshark), port scanning and service detection (Nmap), wireless attacks (Aircrack-ng), reverse engineering (JADX GUI), and firmware acquisition tools for embedded devices. These tools indicated consistent patterns of insecure default settings, bad encryption, and the absence of patch management that demonstrate that in consumer-grade IoT ecosystems, the basics of security principles are neglected everywhere.

When Nmap was used to scan a few IoT devices such as Wipro Smart plug, Philips smart bulb, Anycast dongle, and Hikvision camera, it was discovered that some of these devices had open ports while others had filtered ports. Though in some devices most ports were blocked (an indication of some firewall or NAT configuration), in others, services like HTTP (port 80), RTSP (port 554), and alternate admin ports (8000, 7070) were exposed. These ports, unless protected by HTTPS or authentication mechanisms, enable attackers to intercept the communication or access the device control panels illegitimately.

Missing basic security headers like Content-Security-policy, X-Frame-options, and Strict-Transport-Security on HTTP-exposed services makes these devices prone to Cross-Site Scripting (XSS), Clickjacking, and SSL Stripping. For example, checks on the traffic with Wireshark have identified devices broadcasting plaintext DNS and mDNS queries, which may disclose sensitive metadata like device type, firmware version, and internal IPs. Such revelations can be exploited for network fingerprinting and for directed attacks.

Wireless network vulnerability was verified with the use of Aircrack-ng,w hich made it possible to seize WPA2 handshakes from all tested devices. The use of WPA2-PSK without WPA3 or Protected Management Frames (PMF) exposes the networks for deauthentication attacks, handshake replay, and offline brute-force attacks (even if dictionary attacks failed to crack these passwords – that is, decently strong credentials). Devices that have automatic pairing capabilities or poorly audited Wi-Fi interfaces are targeted especially for open or under-encrypted networks by man-in-the-middle (MITM).

In addition, reverse engineering of companion applications using JADX GUI showed partially obfuscated, partially accessible code. For example, the mobile application of the Wipro Smart Plug had hardcoded API endpoints and incomplete encryption logic, which indicated that, in certain situations, data can be sent in cleartext. Even if data is encrypted in transmission, poor client-side logic brings in risks that include faulty authentication, token reuse, and information leakage.

Additionally, the Netcat scans showed that some devices reveal unanticipated services (e,g, SOCKS5 proxy on port 1080 of Echo Dot) that can be used to tunnel traffic or to create remote control tunnels. Such ports in especially the undocumented or unprotected ones, indicate severe configuration oversights that could be exploited for lateral movement or malware spreading in the house network.

These statistics have immense implications not only to users but also to manufacturers and regulators. Lack of encryption, default passwords, open ports, and unencrypted broadcasts all together make a rich attack surface. If utilized, these weaknesses may lead to:

Device Hijacking (as in the case of Mirai botnet),

Leakage of sensitive data (sensitive data from wearables, e.g., health data),

Smart Infrastructure Denial of Service attacks (DoS).

Unauthorized spying using IP cameras and smart speakers.

The risks are further exacerbated in environments of coexistence of several IoT devices, as in smart homes, healthcare facilities, or an industrial automation system. In these environments, one weak device can be used as an entry point for the overall system.

5.1 Practical Implications

In practical terms, it shows that there are tensions between usability and security in designing IoT. Plug-and-play setup, remote access, and interoperability are some of the things that manufacturers focus on, at the expense of essential principles of security, such as least privilege and attack surface

minimization. The open or undocumented ports indicate how even a production-grade consumer device can have vectors that are exploitable by a network-based attacker. Assuming that this device is not separated from a sensitive system,s then these ports can be used as the entrance points for subsequent exploitation or lateral movement purposes by an attacker. For the Smart BulbS, their dependence on Tuya's cloud platform enables one to have centralized control and a single point of failure. Any misfortune with Tuya cloud infrastructure, API, or authentication flow may have a direct impact on a broad range of dependent smart devices. In addition, users mostly don't have any visibility or control in firmware updates, so they trust vendor integrity and update cycles for security assurance

5.2 Ethical Standards

This research was carried out in a controlled, ethical approach. All the tested devices were personal, and each scan and probe happened in a private, isolated test network. It is for legal reasons of device tampering and digital rights management that no efforts were made to extract firmware from the devices. The deauthentication tests carried out via Aircrack-ng were conducted in short bursts and on a test network reserved precisely for conducting such tests, so as not to disrupt the network. Hackers' ethical principles were followed strictly, ensuring no third-party networks and devices were compromised. This ethical boundary keeps the findings reproducible, ethical, and on the lines of academic cybersecurity research.

CONCLUSION

6. CONCLUSION

This work was designed to evaluate the security posture of selected IoT (Internet of Things) devices using methods of active network scanning, traffic analysis, and wireless probing. Based upon popular open source tools like Nmap, Wireshar,k and Aircrack-ng, the investigation provided major understanding concerning the vulnerabilities, behaviour in communications, and architectural weaknesses of these devices.

The discussion highlights a desperate call for secure design principles in the building of IoT. Utilization of diagnostic devices, including Wireshark, Nmap, Aircrack-n, among other, has exposed major vulnerabilities in the devices tested. These are not hypothetical—they represent true avenues of attack that have been used against the adversaries in the past incidents. Making devices secure against authentication, ensuring communication with a client is encrypted, hardening headers, and enabling over-the-air updates options are non-optional. They are essential in developing a reliable and trustworthy IoT system.

6.1 Broader Implications

These results further validate the increasing fears that IoT devices that make things convenient and automated frequently compromise on security in favour of usability. Vendors can defer encryption, secure firmware delivery, or local isolation to accomplish the affordability or deployment simplicity that customers desire for products. This opens a massive attack threshold that can be easily preyed upon by the bad guys, especially in homes and small offices where there are no enterprise-class firewalls or intrusion detection systems. In addition, the mysterious nature of firmware, undocumented ports, and quiet API calls raises questions about data governance and compliance. Customers usually have no, or a very minimal, understanding of what data is moving, where it is going, and if it is being stored safely. Such conditions not only undermine the security of the technical elements but also undermine consumer trust,essential for the adoption of the emerging smart technologies.

6.2 Recommendations for Securing Smart Devices

From the analysis, several actionable recommendations are presented:

- Network Segmentation: Segment IoT devices onto their own VLAN, or guest network, which decreases the risk of lateral movement if they were to be compromised.
- WPA3 and Strong Passwords: Make router settings WPA3 and set long, unique Wi-Fi passwords to discourage handshake-based brute force.
- Regular Monitoring: Use such tools as Wireshark, periodically, to look for unencrypted traffic or suspicious traffic by IoT devices.
- Disable Unused Services: As much as it is possible, disable mDNS, UPnP, or the port-forwarding features that make devices visible to the internet.
- Updates: Regular updates and patching up.
- DNS Hardening: Secure DNS services (e.g., DNS-over-HTTPS) or local DNS filtering to avoid metadata revelation.

As the future of the Internet of Things gets more etched in personal, commercial, and industrial settings, the necessity for a security-first philosophy of design becomes non-negotiable. This research proves that even such popular devices as could introduce significant risks if security features are either neglected or underestimated. Eventually, the burden of securing the IoT ecosystems is a joint responsibility and not the single actor's responsibility: device manufacturers, network managers, policymakers, and users collaborate to achieve the same. An initiative in proactive auditing with the help of such tools as Nmap, Wireshark, and Aircrack-ng can facilitate pinpointing of possible risks in an early stage, effortlessly prepaid for informed choice making, and support the planning of a resilient and trustworthy smart environment.

6.3 Future Work and Research Possibilities

Although this study performed a fundamental analysis of IoT device weaknesses with available network and wireless tools, several open directions for future research remain. By broadening the horizon to other types of devices – CCTV cameras, medical wearables, and industrial sensors – it is possible to get a more complete risk picture across domains. Along with that, firmware-level analysis (where legally allowable) can be the subject of future study to reveal hardcoded

credentials or old libraries. Another beneficial route is implementing machine learning for traffic anomaly detection and real-time alerts, which is especially useful for creating a lightweight IDS for smart environments. Middleware-level firmware options (open-source replacements, such as Tasmota or ESPHome) for devices based on ESP could improve transparency and user control. Finally, compliance with data protection laws (e.g. GDPR, DPDP Act) would combine technical analysis and legal and ethical aspects and advance in the understanding of risks in society.

REFERENCE

7. REFERENCES

- Acar, A., Fereidooni, H., Abera, T., Sikder, A. K., Miettinen, M., Sadeghi, A.-R., & Conti, M. (2018). Understanding the Security Challenges in Smart Home IoT Devices. 2018 IEEE Security and Privacy Workshops (SPW), 100–105. https://doi.org/10.1109/SPW.2018.00025
- Alaba, F. A., Othman, M., Hashem, I. A. T., & Alotaibi, F. (2017). Internet of Things security: A survey. Journal of Network and Computer Applications, 88, 10–28. https://doi.org/10.1016/j.jnca.2017.04.002
- AmbiMate. (2023). Challenges to IoT Security. AmbiMate. https://ambimat.com/challenges-to-iot-security-1/
- Arreaga, N. X., Enriquez, G. M., Blanc, S., & Estrada, R. (2023). Security vulnerability analysis for IoT devices Raspberry Pi using PENTEST. Procedia Computer Science, 224, 223–230. https://doi.org/10.1016/j.procs.2023.09.03
- Baker, S. (2019). Smart Home Security Risks: The Privacy Concerns of IoT Devices. Retrieved from: https://www.techradar.com
- Bhardwaj, R., Choudhury, T., & Ghosh, S. (2024). Unmasking vulnerabilities by a pioneering approach to securing smart IoT cameras through threat surface analysis and dynamic metrics. Computers & Security, 133, 103322. https://doi.org/10.1016/j.cose.2024.103322
- Choudhary, A., Kalra, J., & Sagar, S. (2021). A study of threats, vulnerabilities and countermeasures: An IoT perspective. Materials Today: Proceedings. https://doi.org/10.1016/j.matpr.2021.03.399
- CISA. (2021). Compromise of U.S. Water Treatment Facility. Retrieved from: https://www.cisa.gov
- FTC. (2018). The Internet of Things: Privacy and Security in a Connected World. Federal Trade Commission. Retrieved from: https://www.ftc.gov Goodin, D. (2022, December 7). New Mirai variant targets 13 vulnerabilities in routers, cameras. SC Magazine. https://www.scmagazine.com/news/mirai-botnet-variant-v3g4-targeting-13-unpatched-iot-vulnerabilities

- Grattafiori, A. (2014). Exploiting Networked Chromecast Devices. Bishop Fox Research. Retrieved from: https://www.bishopfox.com/resources
- Grattafiori, A. (2014). Exploiting Networked Streaming Devices. Bishop Fox. https://www.bishopfox.com/resources
- Han, Y., Wang, P., Kang, C., Lin, J., & Fan, W. (2024). A vulnerability detection method for IoT protocol based on parallel fuzzy algorithm. Heliyon, 10, e31846. https://doi.org/10.1016/j.heliyon.2024.e31846
- IBM. (2023). Cost of a Data Breach Report. Retrieved from: https://www.ibm.com/reports/data-breach
- IoTSF. (2020). IoT Security Foundation Guidelines and Best Practices. Retrieved from: https://www.iotsecurityfoundation.org
- Krebs, B. (2016, September 21). KrebsOnSecurity Hit With Record DDoS. Krebs on Security. https://krebsonsecurity.com/2016/09/krebsonsecurity-hit-with-record-ddos/
- Krebs, B. (2016). Krebs on Security - DDoS on Dyn Impacts Twitter, Spotify, Reddit. Retrieved from: https://krebsonsecurity.com
- Lee, J., Kim, J., & Park, Y. (2020). Security Vulnerabilities in Smart Plugs: A Case Study of Consumer IoT Devices. Journal of Information Security, 11(4), 250–263.
- Meidan, Y., Sachidananda, V., Peng, H., Sagron, R., Elovici, Y., & Shabtai, A. (2020). A novel approach for detecting vulnerable IoT devices connected behind a home NAT. Computers & Security, 97, 101968. https://doi.org/10.1016/j.cose.2020.101968
- Ramadan, R. A. (2022). Internet of Things (IoT) security vulnerabilities: A review. PLOMS Journal of Artificial Intelligence (PLOMS AI). https://plomscience.com/journals/index.php/PLOMSAI/article/view/14/9
- Ronen, E., & Shamir, A. (2016). Extended Functionality Attacks on IoT Devices: The Case of Smart Lights. Proceedings of the IEEE European Symposium on Security and Privacy. https://iotsecurity.eecs.qmul.ac.uk
- Saleh, M. A., Alqahtani, A. S., & Alwateer, M. A. (2020). Analysis on security vulnerabilities of medical wearable devices (fitness trackers). Indonesian Journal of Electrical Engineering and Computer Science, 20(3), 1274–1281. https://doi.org/10.11591/ijeecs.v20.i3.pp1274-1281

- Scott, M. (2017). Cyberattack on Ukraine's Power Grid Shows Vulnerability of Energy Systems. New York Times. Retrieved from: https://www.nytimes.com
- Sicari, S., Rizzardi, A., Grieco, L. A., & Coen-Porisini, A. (2015). Security, privacy and trust in Internet of Things: The road ahead. Computer Networks, 76, 146–164. https://doi.org/10.1016/j.comnet.2014.11.008
- U.S. Congress. (2020). IoT Cybersecurity Improvement Act of 2020. Retrieved from: https://www.congress.gov
- Weber, R. H. (2010). Securing the Internet of Things: A standardization perspective. Computer Law & Security Review, 26(1), 23–31. https://doi.org/10.1016/j.clsr.2009.11.005
- Wired. (2020). Smart home devices are spying on you: Here's how to stop it. Retrieved from: https://www.wired.com
- Yaacoub, J.-P. A., Noura, H. N., Salman, O., & Chehab, A. (2023). Ethical hacking for IoT: Security issues, challenges, solutions and recommendations. Internet of Things and Cyber-Physical Systems, 3, 280–308. https://doi.org/10.1016/j.iotcps.2023.04.002
- Yu, M., Zhuge, J., Cao, M., Shi, Z., & Jiang, L. (2020). A survey of security vulnerability analysis, discovery, detection, and mitigation on IoT devices. Future Internet, 12(2), 27. https://doi.org/10.3390/fi12020027
- ZDNet. (2018). Heart monitor breach exposes thousands of user records. Retrieved from: https://www.zdnet.com
- Ramadan, R. A. (2021, January 21). Internet of Things (IoT) Security Vulnerabilities: A Review. PLOMS Journal of Artificial Intelligence. https://plomscience.com/journals/index.php/PLOMSAI/article/view/14/91.
- AHMED, S., ILYAS, M., & M. Yasin. (2022). IoT Based Smart Systems using Machine Learning (ML) and Artificial Intelligence (AI): Vulnerabilities and Intelligent Solutions. International Conference on Society and Information Technologies (ICSIT 2022). https://www.iiis.org/CDs2022/CD2022Spring/papers/HB924YY
- al., L. e., & s, P. (2020). Security Vulnerabilities in Consumer IoT Devices: A Case Study on Smart TVs and Plugs. IEEE Internet of Things Journal. https://doi.org/10.1109/JIOT.2020.2991234

- Alanezi, M. A. (2022, February). Vulnerabilities, Threats and Challenges on Cyber Security and the Artificial Intelligence based Internet of Things: A Comprehensive Study. IJCSNS International Journal of Computer Science and Network Security. https://koreascience.kr/article/JAKO202213042335203.page
- Bachy, Y., Basse, F., & Nicomette, V. (2015, July 22). Smart-TV security analysis: practical experiments. Hal archives-ouvertes. https://hal.science/hal-01178553
- Bhandari, G. p., Assres, G., Gavric, N., & Shalaginov, A. (2024). IoTvulCode: AI-enabled vulnerability detection in software products designed for IoT applications. International Journal of Information Security https://link.springer.com/article/10.1007/s10207-024-00848-6
- Bin Hulayyil, S., Li, S., & Xu, L. (2023). Machine-Learning-Based Vulnerability Detection and Classification in Internet of Things Device Security. mdpi. https://www.mdpi.com/2079-9292/12/18/3927
- E., R., & A., S. (2017). IoT Goes Nuclear: Creating a ZigBee Chain Reaction. IEEE Symposium on Security and Privacy. https://www.iotworm.com/papers/zigbee-chainreaction.pdf
- Kuzlu, M., fair, C., & Guler, O. (2021, September 29). Role of Artificial Intelligence in the Internet of Things (IoT) cybersecurity. Discover Internet of Things. https://link.springer.com/article/10.1007/s43926-020-00001-4
- Michèle, B. (2014). Watch and be watched: Compromising all Smart TV generations. IEEE. https://www.researchgate.net/publication/268240467_Watch_and_be_watched_Compromising_all_Smart_TV_generations
- M Khandelwal, M., & Upadhyay, V. (2023). A Study of Security Vulnerabilities on IP Cameras. International Journal of Computer Applications (IJCA). https://www.ijcaonline.org/archives/volume179/number37/khandelwal-2018-ijca-917276.pdf
- Ramadan, R. A. (2021, January 21). Internet of Things(IoT) Security Vulnerabilities: A Review. PLOMS Journal of Artificial Intelligence. https://plomscience.com/journals/index.php/PLOMSAI/article/view/14/9

ANNEXURE

Shahrukh Khan 23mfor31_report_project_(1)[1].pdf

ORIGINALITY REPORT

8%	7%	7%	3%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	plomscience.com Internet Source	1%
2	deepai.org Internet Source	1%
3	www.researchgate.net Internet Source	1%
4	www.hikvision.com Internet Source	1%
5	Submitted to University of Mumbai Student Paper	<1%
6	www.iiis.org Internet Source	<1%
7	pmc.ncbi.nlm.nih.gov Internet Source	1%
8	paper.ijcsns.org Internet Source	1%
9	Michele, Benjamin, and Andrew Karpow. "Watch and be watched: Compromising all Smart TV generations", 2014 IEEE 11th	1%