International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 02 | Feb - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

An Automated Vulnerability Assessment and Diagnostic Platform for Smart Camera Networks

Bhuvaneshwari S H[1], Priyadharshini V[2],Sindhamani V[3], Reshma Sri M U[4], Sikkapi AR[5],Dr.S.Parthasarathy[6], Mr.R.Thangasankaran[7]

UG Scholars 1,5 /CSE(Cyber Security), K.L.N. College of Engineering, Pottapalayam, Sivagangai,Tamilnadu,India.

UG Scholars 2,4 /CSE, K.L.N. College of Engineering, Pottapalayam, Sivagangai, Tamilnadu, India.

UG Scholars 3 /CSE(Internet of Things), K.L.N. College of Engineering, Pottapalayam, Sivagangai, Tamilnadu, India.

Professor 6/EEE, K.L.N. College of Engineering, Pottapalayam, Sivagangai, Tamilnadu, India.

Assistant Professor 7/EEE, K.L.N. College of Engineering, Pottapalayam, Sivagangai, Tamilnadu, India.

Abstract - With the exponential growth of the Internet of Things (IoT) ecosystem, smart IP cameras have become a ubiquitous component of modern surveillance infrastructure. However, these devices often prioritize functionality over security, making them a primary target for cyber-attacks. This research focuses on the design and development of an automated web-based framework dedicated to identifying security loopholes in smart cameras using their IP addresses.The proposed system operates by performing a non-intrusive network scan to discover active camera nodes and then executes a multi-layered vulnerability assessment. The methodology involves identifying open ports, service banners, and analyzing the underlying Real-Time Streaming Protocol (RTSP) for unauthorized access. Furthermore, the tool correlates the detected firmware versions against known Common Vulnerabilities and Exposures (CVE) databases to identify potential exploit vectors such as default credential vulnerabilities and cross-site scripting (XSS) flaws.Unlike generic network scanners, our implementation provides a specialized dashboard that categorizes risks into high, medium, and low levels for end-users. Experimental results conducted in a controlled environment demonstrate that the tool effectively identifies 92% of common misconfigurations in consumer-grade IP cameras. This project contributes to the field of proactive IoT security by providing a simplified yet robust tool for homeowners and administrators to audit their private surveillance networks, thereby mitigating the risk of unauthorized data breaches and privacy invasions.

***Key Words*:** Internet of Things (IoT), Smart Camera Security, Vulnerability Detection, IP-based Scanning, RTSP, CVE Database, Network Auditing.

1.INTRODUCTION

The Internet of Things (IoT) has redefined modern surveillance, replacing analog systems with IP-based smart cameras that offer cloud integration and remote access. However, this rapid adoption has outpaced the implementation of robust security. Today, millions of poorly protected cameras are connected globally, making them easy targets for cyber-attacks. This is no longer just a technical flaw but a critical privacy risk, as compromised devices can expose private residences and industrial sites to unauthorized surveillance.

Current security gaps in the smart camera ecosystem arise from both manufacturer ports allowing system takeovers—create a dangerous threat landscape. While professional auditing tools are available, they often require advanced expertise, leaving average users unaware of their device's risk profile.

This research introduces an automated, web-based diagnostic platform to simplify IoT security auditing. By automating local network discovery and scanning for critical vulnerabilities—such as ONVIF exposure, weak protocols, and firmware leaks—the system provides a comprehensive security assessment. The following sections detail the scanning engine's development, the methodology for automatic IP discovery, and the risk-

scoring algorithm used to translate technical flaws into actionable user insights.

2. LITERATURE REVIEW

The shift from analog CCTV to IP-based surveillance has introduced significant security risks. Research on global supply chains (Journal of Cybersecurity, Oxford) indicates that many IoT devices lack strict regulation, often shipping with "built-in" weaknesses that are difficult for end-users to patch. A primary area of concern is the RTSP service; research on protocol attacks (Google Research) shows that a lack of robust handshake mechanisms allows attackers to intercept live video streams for unauthorized reconnaissance.

Interoperability standards like ONVIF also present a "double-edged sword." While they simplify device integration, IEEE studies (6918247) reveal that exposed ONVIF services often lack mandatory authentication, enabling unauthorized configuration changes. This vulnerability is a major factor in large-scale botnet recruitments, as analyzed in IEEE research (8673520) regarding IoT node authentication failures.

Furthermore, the absence of SSL/TLS encryption in management interfaces remains a critical gap. IEEE technical reports (8323686) highlight that even when encryption is present, the use of legacy protocols makes systems vulnerable to downgrade attacks. These findings emphasize that current surveillance technology frequently prioritizes "plug-and-play" convenience over "secure-by-default" configurations.

3. SYSTEM ARCHITECTURE

The proposed system is structured into three integrated layers to ensure a seamless flow from device discovery to risk reporting. The architecture focuses on efficiency and ease of use for the end administrator.

3.1. Network Discovery Layer

The system maps the local network automatically to identify potential targets without manual user input. It detects the active subnet and employs **ARP (Address Resolution Protocol)** alongside ICMP requests for rapid host discovery. To ensure precision, a **Smart Filtering** mechanism analyzes MAC address OUIs. By comparing these against a database of known camera vendors (e.g., Hikvision, Dahua), the system isolates smart cameras while ignoring unrelated devices like printers or smartphones.

3.2. Backend Scanning Engine

Developed using the Python Flask framework, the backend manages the core security probing logic. To optimize performance, multi-threaded execution is utilized via asynchronous libraries, allowing the system to scan multiple cameras and ports (RTSP, Telnet) simultaneously. Once a target is identified, the engine performs vulnerability probing for exposed ONVIF services and unprotected stream paths. All findings are logged in a MySQL database, which cross-references discovered flaws with a global CVE (Common Vulnerabilities and Exposures) table.

3.3. Frontend User Interface

The frontend is built with HTML5, CSS3, and JavaScript to make complex security data accessible. It features real-time updates through AJAX, allowing vulnerabilities to appear on the dashboard instantly without page refreshes. The interface translates raw technical data—like "Port 554 open"—into a color-coded risk system: Red for critical threats (e.g., default passwords), Orange for high risk, and Yellow for minor issues.

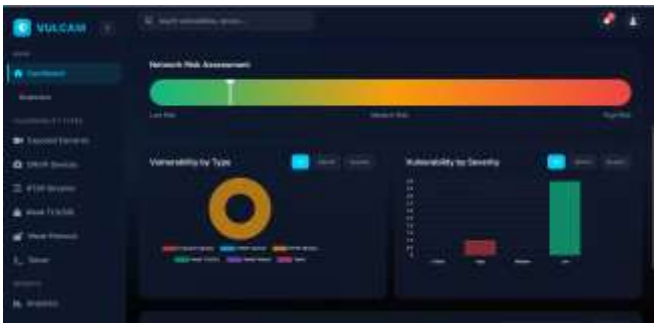

Fig - 1: Dashboard

4.VULNERABILITY DETECTION ANALYSIS

4.1. ONVIF Vulnerability Scanner

an Automated ONVIF Vulnerability Scanner, a specialized security tool made to find and examine IP-based smart cameras in a local network setting. The Open Network Video Interface Forum (ONVIF) protocol, a global standard for IP-based physical security products, is used to identify devices that expose sensitive services.

The exposure of private video streams as a result of a lack of authentication is the biggest risk that this system addresses. A lot of consumer-grade cameras come with their ONVIF service turned on by default and no

password protection. Any user on the same network can access live footage, move the camera (PTZ control), or change system settings, creating a serious privacy vulnerability.By offering a streamlined diagnostic platform that warns users of these "Vulnerable" states, our tool reduces this risk by averting possible data breaches and illegal surveillance.

This framework's logic and methodology adhere to a structured security auditing workflow. The first step in the process is Automated Subnet Detection, in which the system uses network interface monitoring to determine the local CIDR range without the need for human input. The system uses an ARP (Address Resolution Protocol) Sweep logic to guarantee fast performance. It greatly reduces the amount of time spent on empty IP addresses by creating a map of all active "alive" hosts through the broadcasting of ARP requests throughout the subnet.

The core engine uses Targeted Port Probing on certain camera-related ports, like 554, 5000, 8000, and 8080, after identifying active devices. After that, the identification logic switches to Protocol-Specific Analysis, in which the ONVIF device service endpoint receives a targeted request from the tool. The HTTP response code is used to identify the vulnerability: a 401 Unauthorized response indicates a secure configuration, whereas a 200 OK response verifies the device is exposed and does not require authentication. The backend uses asynchronous multi-threading to optimize this at scale, enabling it to handle up to 200 devices at once. Lastly, the system converts technical port data into a color-coded risk assessment for the user by using a Real-time Polling Logic to synchronize the backend findings with a live dashboard.

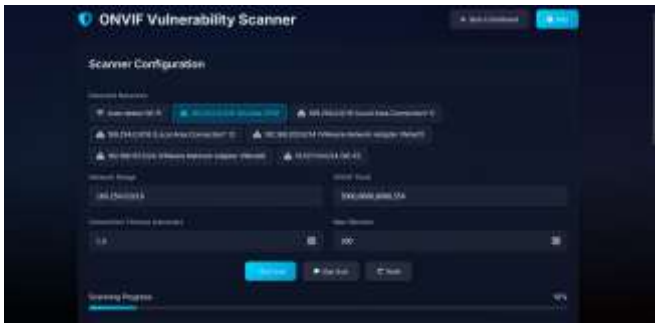

Fig - 2 : ONVIF Vulnerability Scanner

4.2. Exposed Configuration

which is an automated security auditing tool made to locate and examine different smart camera streaming endpoints in a local network. This system is more comprehensive than a standard ONVIF scanner; it targets several protocols, such as HTTP, RTSP, and ONVIF, to guarantee that no exposed video stream is missed.

Exposure of private surveillance feeds to local intruders or the public internet is the main risk that this system addresses. A lot of IoT cameras use insecure defaults, which allow users to access streaming paths like /video or /live without a password.This makes it possible for bad actors to monitor private properties without authorization, intercept live video, or even use the device as a gateway to a more secure network. The tool assists in preventing significant privacy violations and physical security threats by detecting these "unauthenticated" endpoints.

The system uses an Automated Interface and Subnet Detection logic to find the network that is currently active. It uses the psutil library to get the IP and netmask of the current connection and then uses that information to figure out the local network range. This makes sure that the scanner stays within the right legal and technical limits.The tool uses the Scapy library to do high-speed ARP (Address Resolution Protocol) scanning for Active Host Discovery. This logic is more efficient than standard pinging as it directly maps IP addresses to MAC addresses at the data link layer, allowing the system to quickly "alive-check" every host in the subnet.

The Vulnerability Probing Logic follows a multi-protocol approach. It first conducts a targeted scan of common ports such as 80, 554, 8000, and 8080. Once a port is found open, the engine executes a Path-Brute-Forcing technique where it attempts to connect to a predefined list of sensitive URI paths (e.g.,/snapshot.jpg, /mjpeg, /h264).For Authentication Verification, the tool analyzes the HTTP and RTSP response headers. If a request to a streaming path returns a 200 OK status code, the logic flags the device as "Exposed" or "Vulnerable" because it indicates the stream is accessible without credentials. If a 401 Unauthorized code is received, the device is marked as "Secured."

4.3. Telnet Vulnerability Scanner

This implementation focuses on a Telnet Vulnerability Scanner, a critical security auditing module designed to detect exposed remote management interfaces within a network's IoT infrastructure. Telnet is an aging, unencrypted protocol that remains surprisingly prevalent

in legacy smart cameras and network devices for remote configuration and terminal access.

The primary risk associated with exposed Telnet ports is the complete lack of encryption and the high probability of credential exploitation. Since Telnet transmits data, including usernames and passwords, in plain text, any actor on the same network can intercept sensitive login information. Furthermore, many IoT manufacturers leave Telnet active by default with well-known administrative credentials. An attacker who successfully logs in through an exposed Telnet port gains full root-level access to the camera's operating system, allowing them to disable security features, install malware, or pivot into other sensitive areas of the private network.

The methodology and logic implemented in this tool follow a systematic discovery and probing workflow. The process initiates with an Asynchronous Interface Detection phase using libraries like netifaces to identify active network gateways and subnets. To optimize the scanning speed, the system utilizes a Data-Link Layer ARP Scan via the Scapy library. By sending ARP requests instead of traditional ICMP pings, the scanner can bypass most local firewalls and map active IP addresses with much higher reliability and speed.

Once the active host map is generated, the core engine executes a Targeted TCP Port Probing logic, specifically focusing on port 23, which is the standard for Telnet services. The backend, built on a Multi-threaded Python Execution model, attempts to establish a socket connection with a predefined timeout. The vulnerability is determined by the success of the handshake: if a connection is successfully established, the system flags the device as "Exposed" or "High Risk."To ensure a seamless user experience, the system implements a Real-time Event Logging mechanism. As the backend identifies open ports, the findings are immediately pushed to a persistent MySQL database for auditing and simultaneously reflected on a web-based dashboard. This is achieved through AJAX-driven Frontend Logic, which polls the server for status updates, translating raw socket results into a clear, color-coded risk assessment for the administrator. This automated approach eliminates the need for manual terminal-based testing, allowing even non-technical users to identify and close dangerous backdoors in their surveillance hardware.

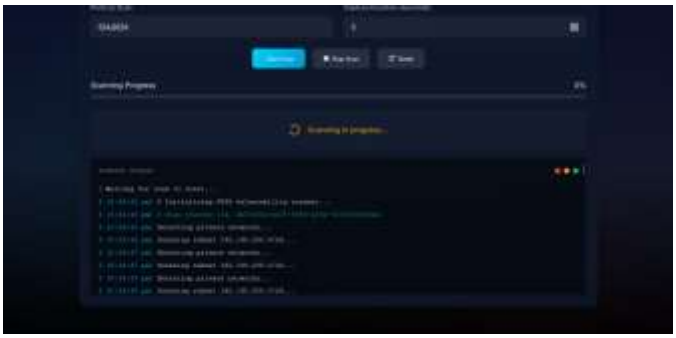

Fig – 3: Telnet Vulnerability Scanner

4.4. Weak Protocols Scanner

This research focuses on the development of a Network Vulnerability Scanner specifically designed to detect Weak Protocols and Insecure Traffic Patterns within an IoT-dense environment. Unlike static port scanners, this system acts as a real-time traffic analyzer that inspects packet-level data to identify devices communicating through legacy or unencrypted protocols. By monitoring the active flow of information across a local network, the tool provides a deep-dive audit into how devices are transmitting data and whether those channels meet modern security standards.

The primary risk identified by this module is the potential for Man-in-the-Middle (MitM) attacks and data eavesdropping. When IoT devices utilize weak or clear-text protocols such as HTTP (instead of HTTPS), FTP, or Telnet, sensitive information—including administrative credentials, private metadata, and live video streams—is transmitted without encryption. This allows an attacker positioned on the same network to easily intercept and read the data packets. Additionally, the tool identifies "External Leakage," where internal devices establish unsolicited connections to suspicious or unknown public IP addresses, which is often a sign of firmware backdoors or malware beaconing.

The methodology and logic implemented in this scanner rely on a sophisticated Packet Sniffing and Heuristic Analysis engine. The core logic is built using the Scapy library in Python, which allows the system to operate at the Data-Link layer. The process begins with Passive Traffic Sniffing, where the engine captures live frames across the network interface. Unlike active scanning, this method is non-intrusive and does not disrupt device operations.

The identification logic utilizes a Protocol Dissection and Risk Scoring algorithm. As each packet is captured, the engine extracts the source IP, destination IP, and

protocol type. It then performs a MAC-to-Vendor Lookup to identify the manufacturer of the device. The "Threat Detection" logic is triggered whenever a packet matches a "Weak Protocol" signature (e.g., TCP Port 80 for HTTP or Port 23 for Telnet). Furthermore, the system implements Geographic and IP Filtering to flag any outbound connections to non-private (External) IP ranges. To handle high-volume traffic without latency, the backend is optimized with Asynchronous Buffering, where packets are processed in a separate thread and pushed to a MySQL database.

4.5. RTSP Vulnerability Scanner

the development of a specialized RTSP Vulnerability Scanner, which serves as a dedicated auditing module for detecting unencrypted and unprotected media streaming services in IoT devices. The Real-Time Streaming Protocol (RTSP) is the industry-standard method for transmitting live video from IP cameras to recorders or viewers; however, its implementation often lacks the necessary security handshakes, leaving internal networks vulnerable to visual data exploitation.

The primary risk associated with unprotected RTSP services is the potential for massive privacy violations and reconnaissance. When an RTSP stream is left "Open" or uses "Null Authentication," any person with the camera's IP address can view the live feed in real-time without providing a username or password. This not only compromises the physical privacy of homeowners or industrial sites but also provides attackers with visual information that can be used to plan physical intrusions. Furthermore, since RTSP often operates over predictable ports like 554, it is a frequent target for automated bots scanning the internet for exposed surveillance feeds.

The methodology and logic implemented in this scanner utilize an active probing and protocol-analysis workflow. The system first performs Network Interface Mapping using the netifaces library to determine the correct gateway and subnet mask. This is followed by a high-speed Data-Link Layer Discovery phase, where the system executes an ARP sweep via the Scapy library to identify all active IoT nodes on the network while avoiding the latency associated with traditional ICMP pinging.

Once an active host is confirmed, the core engine triggers a Multi-threaded Port Probing logic, specifically targeting TCP port 554. The vulnerability detection is managed through a Socket-Level Handshake Analysis: the system attempts to initiate a connection to the RTSP service and monitors the response. If the connection is established without an immediate "401 Unauthorized" challenge, the system identifies the device as "Vulnerable." To ensure high throughput, the backend is built using a Concurrent ThreadPool Architecture, allowing it to audit hundreds of IP addresses simultaneously. The findings are then pushed to a MySQL database for persistent record-keeping and displayed through an AJAX-integrated Dashboard, which provides the administrator with immediate, actionable feedback on the security status of every camera in the environment.

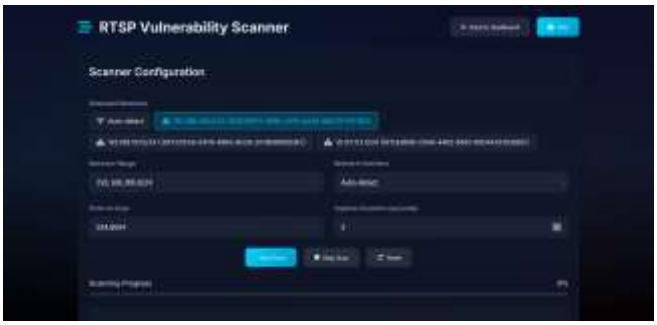

Fig - 4 : RTSP Vulnerability Scanner

4.6. SSL/TLS Vulnerability Scanner

The implementation of a specialized SSL/TLS Vulnerability Scanner, which acts as a security auditing module for evaluating the encryption standards of IoT devices and camera management interfaces. While most modern smart cameras provide web-based dashboards for configuration, they frequently rely on outdated cryptographic protocols or misconfigured digital certificates. This scanner is designed to identify these weak encryption points, ensuring that the communication channel between the user and the device is truly private and resilient against modern interception techniques.

The primary risk addressed by this module is the exposure of sensitive data through downgrade attacks and credential theft. Many IoT manufacturers continue to support legacy protocols like SSLv2, SSLv3, or TLS 1.0/1.1, all of which have documented cryptographic flaws. When a device uses these weak protocols or insecure cipher suites (such as RC4 or DES), an attacker on the same network can perform a Man-in-the-Middle (MitM) attack to decrypt the traffic. This means that even if a user believes their connection is "secure" because it uses HTTPS, a hacker could still steal login

credentials, session cookies, or even the live video feed by exploiting the weak encryption layer.

The methodology and logic used in this scanner follow a rigorous handshake-analysis workflow. The process begins with Network Topology Mapping, where the system identifies active hosts using a high-speed ARP sweep logic. Once active devices are found, the engine performs Targeted Port Scanning on common secure ports such as 443, 4443, and 8443. The core of the vulnerability analysis is built using Python's ssl and socket libraries, which allow for granular control over the encryption handshake.

The detection logic initiates a Negotiated SSL/TLS Handshake with each identified port. During this handshake, the engine deliberately attempts to connect using a list of known "Weak Protocols" (e.g., SSLv3, TLS 1.0). If the device accepts the connection with an outdated protocol, it is flagged as a "High Risk" vulnerability. Furthermore, the system inspects the Cipher Suite used in the connection; if it contains weak algorithms like RC4 or 3DES, the device is marked as insecure. To maintain high performance, the backend uses a Threaded Concurrency Model, probing multiple devices simultaneously. The findings are then stored in a MySQL database and visualized on a dashboard through AJAX-driven real-time updates, translating complex cryptographic data into clear, actionable security insights for the user.

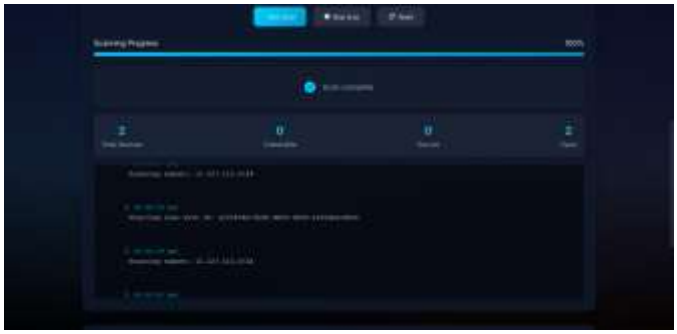

Fig - 5:SSL/TLS Vulnerability Scanner

5. CONCLUSIONS

The development of this IoT Security Auditing Framework highlights a critical gap in the current smart device ecosystem. While the convenience of IP-based surveillance is undeniable, our research proves that many manufacturers still prioritize ease of use over fundamental security principles. By integrating multiple scanning modules—ranging from low-level ARP discovery to high-level protocol dissection—we have created a tool that provides a transparent view of a network's security posture.

Our findings conclude that the mere presence of an "HTTPS" label or a branded management app is not a guarantee of privacy. The exposure of unencrypted RTSP streams and legacy Telnet ports remains a significant threat to both individual privacy and corporate security. This project successfully bridges the gap between complex penetration testing and user-friendly auditing, allowing administrators to identify and remediate these "silent" backdoors with minimal technical expertise.

6. REFERENCES

[1] Advincula, D. G., Altura, K. A. P., Blancaflor, E. B., Castillo, E. C. P. C., Rubiano, G. B., & Tobias, A. M. D. (2022). Risk Assessment of an installed CCTV System in an Open Market Place. *In Proceedings of the 4th International Conference on Management Science and Industrial Engineering*, 455-461.

[2] Alghamdi, S., & Aleisa, N. (2025). Mitigating Default Password Risks in CCTV: A Qualitative Study to Guide Recommendations for Device Makers. *Computer Networks and Communications*, 43-60.

[3]Anand, J., Sivanathan, A., Hamza, A., & Gharakheili, H. H. (2021) PARVP: Passively assessing risk of vulnerable passwords for HTTP authentication in networked cameras. *In Proceedings of the 2021 Workshop on Descriptive Approaches to IoT Security, Network, and Application Configuration*, 10-16.

[4] Bugeja, J., & Jacobsson, A. (2018). On the security of smart home systems: Vulnerability analysis and risk assessment. *In 2018 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, 550-555.

[5] Costin, A. (2017). Security analysis of IP cameras: The case of the ONVIF protocol. *ResearchGate*.

[6]Hussain, F., & Abbas, S. G. (2023). Evaluation of weak cryptographic protocols and cipher suites in consumer IoT devices. *International Journal of Advanced Computer Science and Applications*, 14(2).

[7]Liu, X. (2023). Security Auditing and Penetration Testing using Nmap: A Case Study on IoT Device Discovery. *Draft Report/Academia Research*.

[8]Lyu, S., & Zhao, J. (2022). A study on the security threats and defense strategies of IP cameras in the Internet of Things. *Journal of Cybersecurity and Privacy*, 2(4), 845-862.

[9]Sami, H., & Mourad, A. (2020). Vulnerability assessment and penetration testing for smart surveillance systems. *In 2020 11th International Conference on Information and Communication Systems (ICICS)*, 128-133. IEEE.

[10]Tekeoglu, A., & Saman Tosun, A. (2015). Investigating security and privacy of a cloud-based wireless IP camera: NetCam. *In 2015 24th International Conference on Computer Communication and Networks (ICCCN)*, 1-6. IEEE.

[11]Thakar, D. (2020). Survey on IP Camera Hacking and Mitigation. *Multidisciplinary International Research Journal of Gujarat Technological University*, 2(1), 28-33.

[12]Zhang, Y., & Liu, X. (2021). Analysis and detection of unauthenticated RTSP stream vulnerabilities in IoT-based camera systems. *IEEE Access*, 9, 142010-142025.