

IOT Vulnerability Scanner in Smart Home Devices

Lingeshwaran R , Sankara Narayanan S T

[†]Department of Cyber Forensics and Information security

Dr.M.G.R. Educational and Research Institute , Chennai, Tamil Nadu, India

Abstract — The rapid growth of smart home technologies has brought with it a host of Internet of Things (IoT) devices — from smart lights and security cameras to thermostats and door locks. While these devices undeniably make life more convenient, they also open the door to serious cybersecurity risks. Weak passwords, outdated firmware, open ports, and insecure communication protocols are just a few of the vulnerabilities that attackers can exploit. This paper presents an IoT Vulnerability Scanner designed specifically for smart home environments. The proposed system automatically discovers devices on a local network, gathers key details such as IP addresses, MAC addresses, open ports, and active services, and cross-references this data with known vulnerability databases. The scanner is capable of detecting common threats including default credentials, unencrypted communications, outdated software, and exposed services. A clean, user-friendly dashboard presents vulnerability reports along with practical recommendations, enabling homeowners to take meaningful action without needing technical expertise. The system is built using lightweight network scanning tools, protocol analyzers, and AES-based encryption to secure all data exchanges between the monitoring unit and the web interface. Testing across a range of smart home devices demonstrated a vulnerability detection accuracy of approximately 94.7% and a system uptime of 99.8%. These results confirm that the proposed solution offers an effective, affordable, and practical approach to strengthening IoT security in residential settings.

Key Words: *IoT Security, Smart Home, Vulnerability Scanner, AES Encryption, Network Scanning, Firmware Analysis, Cybersecurity, Embedded Systems.*

1. INTRODUCTION

Over the past decade, the Internet of Things (IoT) has fundamentally changed how we live. Homes that once operated on manual switches and physical locks now run on interconnected networks of smart devices — cameras that stream live video to our phones, thermostats that learn our preferences, and door locks that respond to voice commands. This transformation has brought real convenience and efficiency to millions of households around the world.

But this convenience comes at a cost. Most smart home devices are shipped with default usernames and passwords that users never change. Firmware updates, when available, often go uninstalled. Network ports are left open, and many devices communicate over protocols that offer little or no encryption. These weaknesses are not hypothetical — they have already been exploited in real-world attacks. The Mirai Botnet Attack, for instance, hijacked hundreds of thousands of unsecured IoT devices to launch one of the most damaging distributed denial-of-service (DDoS) attacks in history.

The problem is compounded by the fact that most existing security tools are designed for enterprise environments. They require technical expertise to operate, can be expensive to deploy, and are poorly suited for residential use. An ordinary homeowner has no easy way to know whether their smart home network is secure or not.

This paper addresses that gap. We present an IoT Vulnerability Scanner tailored for smart home environments — a system that can be deployed on low-cost hardware, operated without specialized knowledge, and relied upon to provide accurate, actionable security assessments. The scanner automatically discovers all active devices on the local network and analyzes each one for known vulnerabilities. It checks for default credentials, insecure communication channels, outdated firmware, and unnecessarily open ports. Results are presented through an intuitive web dashboard that not only flags problems but also explains what to do about them.

The system is organized into three core components: network discovery, vulnerability analysis, and report generation. AES-128 encryption secures all communication between the Arduino-based monitoring unit and the web interface,

ensuring that sensitive data is never exposed in transit. The following sections describe the system architecture, implementation modules, performance evaluation, and conclusions drawn from our experiments.

2. LITERATURE SURVEY

2.1 Data Center Network Architecture Analysis (Kashif Bilal et al.)

Bilal et al. conducted a detailed comparative study of major data center network (DCN) architectures, specifically fat-tree, DCell, and traditional three-tier designs. Their work examined how each architecture performs under six different traffic patterns. The study involved custom implementations of addressing schemes and routing algorithms — including a two-level routing strategy for fat-tree and a flexible multi-level configuration for DCell. The findings provide a valuable benchmark for understanding trade-offs in network topology design, particularly in terms of scalability and fault tolerance. This work is relevant to IoT vulnerability research as it informs how network discovery and traffic analysis strategies can be optimized at scale.

2.2 Robustness of DCN Topologies (Kashif Bilal et al.)

A follow-up study by Bilal et al. evaluated the robustness of DCN topologies using graph-theoretic models. The authors found that classical robustness metrics — such as average nodal degree and spectral radius — were insufficient for capturing the real-world resilience of data center networks. As a result, they proposed a new metric called 'degradation' to more accurately assess network durability under failure conditions. Their insights into how networks behave under stress are applicable to smart home environments, where device failures and connectivity drops can create exploitable security gaps.

2.3 Energy-Aware Data Replication in Cloud (Dejene Boru et al.)

Boru et al. proposed a data replication strategy for cloud data centers that jointly optimizes energy consumption, network bandwidth, and communication latency. Their model balances the benefits of distributed data replication — such as reduced access delays and improved fault tolerance — against the overhead of synchronization across geographically dispersed data centers. For IoT vulnerability scanning systems that store device data and scan histories in the cloud, this kind of energy-aware architecture offers a practical model for sustainable and responsive deployment.

2.4 Cloud Security and Privacy Considerations (Wayne A. Jansen)

Jansen examined the security and privacy challenges that arise when organizations move their infrastructure to cloud environments. His key argument is that while outsourcing to cloud providers can reduce operational costs, it does not transfer responsibility for security. The hosting organization remains accountable for ensuring that data handled by third-party providers remains protected. This perspective directly informs our design choices — particularly our decision to use AES encryption for all data transfers in the proposed IoT scanner, ensuring that even if data passes through external infrastructure, it remains confidential and tamper-proof.

3. EXISTING SYSTEM

Current approaches to IoT security scanning largely fall into two categories: enterprise-grade network assessment tools and basic consumer applications. Enterprise tools such as Nessus, OpenVAS, and Shodan offer powerful scanning capabilities but are designed for IT professionals. They require significant configuration, produce technical reports that are difficult for non-experts to interpret, and are priced beyond the reach of most residential users.

Consumer-facing applications, on the other hand, tend to be limited in scope. Many only perform basic port scans or ping sweeps without any deeper vulnerability analysis. They rarely check for default credentials, assess firmware versions against known CVE databases, or evaluate the security of device communication protocols. Additionally, data transmitted between the scanning client and remote servers is often unencrypted, introducing its own security risks.

A critical weakness shared by most existing systems is the absence of AES-based encryption for securing scan data in transit. Without encryption, device details — including IP addresses, open ports, and identified vulnerabilities — can be intercepted by a third party on the same network. The proposed system addresses all of these limitations by combining

network-level discovery with protocol analysis, firmware checks, credential testing, and AES-128 encrypted communication, all presented through an interface accessible to ordinary homeowners.

4. PROPOSED SYSTEM

The proposed IoT Vulnerability Scanner is designed to make smart home security monitoring practical, affordable, and understandable for everyday users. At its core, the system pairs an Arduino microcontroller with a software scanning engine and a web-based dashboard. All communications between these components are secured using Advanced Encryption Standard (AES-128) encryption.

When a scan is initiated, the system begins by discovering all devices connected to the local Wi-Fi network. It collects each device's IP address, MAC address, open ports, and active services. This data is then analyzed against a local vulnerability database to identify known weaknesses — including default credentials that were never changed, firmware versions with publicly disclosed security flaws, open ports that expose unnecessary services, and communication over unencrypted protocols.

Encryption plays a central role in the system's design. Before transmitting any device data or vulnerability report to the web interface, the system encrypts it using AES-128. The Arduino monitoring unit decrypts incoming commands, processes them, and re-encrypts the results before sending them back. This bidirectional encryption ensures that even if the local network is compromised, the data exchanged within the scanner remains confidential.

The final output is a structured vulnerability report delivered through the web dashboard. Each identified issue is categorized by severity, and the report includes plain-language recommendations — such as 'Change your router's default admin password' or 'Disable UPnP on your smart TV' — that a non-technical user can act on immediately.

4.1 System Architecture

The system architecture consists of three tightly integrated layers: the hardware monitoring layer (Arduino microcontroller), the scanning and analysis engine (software-based), and the user interface layer (web dashboard). The Arduino unit handles real-time network communication and acts as the encrypted gateway between the scanning engine and the user-facing interface. The scanning engine performs device discovery, service enumeration, and vulnerability matching. The web dashboard visualizes scan results and allows users to trigger scans, view historical reports, and receive security recommendations.

5. MODULE IMPLEMENTATION

The system is divided into three primary modules, each with a clearly defined role. Together, they form a complete pipeline from device discovery to actionable security reporting.

5.1 Network Discovery Module

This module is responsible for identifying all active devices on the smart home network. It performs an ARP sweep across the local subnet to detect devices by their IP and MAC addresses. For each identified device, it conducts a port scan to enumerate open TCP and UDP ports and identifies the services running on each port. The discovery module is designed to be fast and non-disruptive, completing a full scan of a 20-device network in under 60 seconds.

5.2 Vulnerability Analysis Module

Once devices are discovered, the vulnerability analysis module evaluates each one against a set of predefined security rules and a local CVE-based vulnerability database. The module checks for default or commonly-known credentials by attempting authentication with a curated list of manufacturer-default username/password pairs. It also examines firmware version strings — where available — and cross-references them against known vulnerable versions. Open ports are evaluated against a list of ports that should not be publicly accessible on residential devices, and protocols used for device communication are assessed for encryption support.

5.3 Report Generation Module

The report generation module takes the output of the vulnerability analysis and structures it into a clear, human-readable report. Each vulnerability is assigned a severity level — Critical, High, Medium, or Low — based on its potential impact. The report is displayed on the web dashboard and can be exported as a PDF. Alongside each identified issue, the module provides a specific, actionable recommendation written in plain language. All report data is encrypted using AES-128 before being transmitted to the dashboard.

5.4 AES Encryption Module

Security of the scanner's own communications is handled by the AES encryption module. This module runs on both the Arduino unit and the web server, ensuring that all data — including device details, vulnerability reports, and user commands — is encrypted before transmission and decrypted only at the intended destination. AES-128 was chosen for its balance of strong security and computational efficiency on resource-constrained embedded hardware. The module uses a pre-shared symmetric key stored securely on both endpoints.

6. PERFORMANCE ANALYSIS

The system was tested in a realistic smart home environment with a variety of IoT devices connected to a local Wi-Fi network, including smart cameras, smart plugs, thermostats, smart lights, and environmental sensors.

6.1 Device Discovery Accuracy

Table 1 shows the device detection performance across different network sizes.

Connected Devices	Devices Detected	Detection Accuracy
5	5	100%
10	10	100%
15	14	93.3%
20	19	95.0%

Table 1: Device Discovery Performance

The scanner achieved 100% discovery accuracy for smaller networks and maintained above 93% accuracy even as the number of connected devices increased to 20. This is partly because some devices use non-standard network protocols or have power-saving modes that cause them to be briefly offline during the scan window.

6.2 Vulnerability Detection Accuracy

Table 2 presents the detection results across five vulnerability categories.

Vulnerability Type	Actual Cases	Detected	Accuracy
Default Passwords	12	12	100%
Outdated Firmware	10	9	90%
Open Unused Ports	15	15	100%
Unencrypted Communication	8	8	100%
Weak Authentication	6	5	83.3%

Table 2: Vulnerability Detection Accuracy

The average detection accuracy across all vulnerability types was approximately 94.7%. The scanner performed best on clearly defined issues such as open ports and default passwords, where detection is binary. Slightly lower accuracy was observed for weak authentication mechanisms, which can vary significantly across device manufacturers.

6.3 AES Encryption and Decryption Performance

Table 3 shows how quickly the system encrypts and decrypts data at different payload sizes.

Data Size	Encryption Time	Decryption Time
64 Bytes	8 ms	7 ms
128 Bytes	12 ms	10 ms
256 Bytes	18 ms	16 ms
512 Bytes	31 ms	29 ms

Table 3: AES Encryption/Decryption Performance

Even at 512 bytes, encryption completes in under 35 milliseconds — well within acceptable limits for real-time monitoring. These results confirm that AES-128 can be implemented effectively on the Arduino platform without introducing perceptible latency.

6.4 Scan Time by Device Count

Number of Devices	Average Scan Time
5	12 seconds
10	25 seconds
15	38 seconds
20	52 seconds

Table 4: Total Scan Time

Scan time scales approximately linearly with device count. A complete scan of a 20-device network finishes in under a minute, making the system practical for periodic scheduled scans without disrupting normal home network usage.

6.5 Memory Utilization on Arduino

Parameter	Usage
Program Memory	68%
SRAM Usage	54%
EEPROM Usage	12%

Table 5: Arduino Memory Utilization

Memory usage remains well within the Arduino's operational limits, demonstrating that the full system — including the AES encryption module — can run comfortably on a low-cost embedded platform.

6.6 System Reliability (72-Hour Test)

Metric	Result
Successful Scans	100%
Communication Failures	1.2%
False Positive Rate	3.5%
System Uptime	99.8%

Table 6: System Reliability Metrics

The system ran continuously for 72 hours with only a 1.2% communication failure rate and a false positive rate of 3.5%. The 99.8% uptime figure demonstrates that the system is stable enough for always-on residential deployment.

7. CONCLUSIONS

Smart homes are becoming increasingly common, and with them comes a growing attack surface that most homeowners are completely unaware of. The IoT Vulnerability Scanner presented in this paper offers a practical solution to this problem — one that is affordable, easy to use, and effective.

By automating the discovery and analysis of connected devices, the system removes the technical barrier that has traditionally kept residential users from assessing their own network security. The integration of AES-128 encryption ensures that the scanner itself does not introduce new vulnerabilities while doing its job. With a vulnerability detection accuracy of 94.7% and a system uptime of 99.8%, the results speak for themselves.

Looking ahead, future work will focus on expanding the vulnerability database with real-time CVE feeds, adding support for Zigbee and Z-Wave protocol analysis, and developing a mobile companion app for on-the-go monitoring. There is also significant potential for integrating machine learning-based anomaly detection to identify unusual device behavior patterns that may signal a breach in progress.

In a world where the number of connected devices per household continues to grow, tools like the IoT Vulnerability Scanner are not just useful — they are essential. We hope this work contributes to a broader conversation about making cybersecurity accessible to everyone, not just those with technical backgrounds.

ACKNOWLEDGEMENT

The authors would like to express sincere gratitude to the faculty of the Department of Computer Science and Engineering for their guidance and encouragement throughout this project. Special thanks are also due to the institution's network laboratory for providing the hardware and infrastructure required for testing.

REFERENCES

- [1] Biswajit Datta, Akash Roy, Romit Dutta, Samir Kumar Bandyopadhyay, "Secure Communication through Double Layer Security with Efficient Key Transmission," 2018 International Conference on Information Technology (ICIT), IEEE, 2018.
- [2] M. Subbulakshmi, Dr. D. Usha, "Double Layer Encryption Algorithm Key Cryptography for Secure Data Sharing in Cloud," International Journal of Scientific & Engineering Research, Volume 9, Issue 5, May 2018.
- [3] Jyotsna, Janmejai Kumar, Shivani Chauhan, Amit Doegar, "Multiple Layer Text Security using Variable Block Size Cryptography and Image Steganography," 3rd IEEE International Conference on Computational Intelligence and Communication Technology (CICT 2017), IEEE, 2017.
- [4] Geetha D. Devanagavi, Gahan A. V., "An Empirical Study of Security Issues in Encryption Techniques," International Journal of Applied Engineering Research, ISSN 0973-4562, Volume 14, Number 5, 2019.

[5] Santhosha, Ashok Kumar, "Two Layer Security for Data Storage in Cloud," 2015 1st International Conference on Futuristic Trend in Computational Analysis and Knowledge Management (ABLAZE 2015), 2015.

[6] K. Thippeswamy, Naveen N., "Security and Privacy Challenges Using Multi-Layer Encryption Approaches in Cloud Computing Environments," International Journal of Advanced Research in Computer Science and Software Engineering, 2019.
