# IoT Scanner For Securing Connected Assets In Smart Systems

**Sabeena S[1], Jayasri D[2], Pooja Sri S[3], Gowtham D[4]**
[1]Assist.Professor, Dept of Software Systems
[2, 3, 4]Dept of Software Systems
[1, 2, 3, 4] Sri Krishna Arts and Science college- India

***Abstract-*** *The rapid rise of smart infrastructure has brought with it a surge in connected IoT devices across offices, industries, and public spaces. This connectivity improves efficiency, but it also widens the door for misuse, cyberattacks, and overlooked asset depreciation. Conventional asset tracking systems largely focus on inventory counts, rarely linking security status with asset health. Our work introduces an IoT Scanner for Securing Connected Assets in Smart Systems — a Python–Django framework that ties together asset flow monitoring, vulnerability scanning, and anomaly detection. It identifies devices on the network through active and passive scans, profiles their security posture, and flags abnormal behaviour. Depreciation is automatically computed at set intervals, and a Plotlybased dashboard presents security status, anomalies, and value trends in one view. In controlled smart office and industrial IoT setups, the system reached 96% accuracy in vulnerability detection and 94% in anomaly recognition, exceeding the performance of standard scanners. The proposed approach delivers a practical, scalable means of protecting modern connected environments while supporting smarter asset management.*

***Keywords***- IoT Security, Asset Monitoring, Anomaly Detection, Vulnerability Assessment, NetworkScanning, Data Visualization

## I. INTRODUCTION

Smart technologies and connected systems now drive operations in factories, offices, and public infrastructure. Sensors, controllers, and smart appliances work together to provide automation, but these same devices often operate with limited security and irregular maintenance. Firmware updates are missed, open ports remain unmonitored, and in many cases there is no unified view combining asset usage, health, and security. This gap leaves room for intrusions, data leaks, and asset misuse.Typical asset management tools keep track of inventory and depreciation but ignore the real-time security condition of connected devices. On the other hand, network scanners can find vulnerabilities but are blind to the financial and operational context of the assets they check. This separation can result in security threats going unnoticed until damage has already been done especially in environments where downtime or asset loss carries high cost. The system is designed to locate and profile IoT devices through a combination of, active probing and passive monitoring, ensuring that both responsive and intermittently connected devices are accurately identified. Once discovered, each asset’s security posture is evaluated through comprehensive vulnerability checks that assess open ports, software versions, and configuration weaknesses. The system then monitors operational data to detect abnormal usage patterns that may indicate misuse, faults, or unauthorized access. In addition to security insights, it automatically calculates asset depreciation on a monthly or yearly basis, integrating financial and operational perspectives. All collected information is presented in an interactive dashboard that unifies asset health, security status, and depreciation trends, providing users with a clear and holistic view of their connected infrastructure.

By merging asset tracking and IoT security scanning into a single lightweight platform, the system offers a clear view of both operational and cyber risk. Testing shows it can serve as a practical tool for smart offices, industrial IoT, and other connected setups where both security and asset value must be protected over time

## II. RELATED WORK

### A. *Asset Management in Smart Environments*

Early digital asset management systems were built mainly for inventory logging, depreciation tracking, and financial reporting. These tools could reliably account for physical resources but provided little or no visibility into real-time operational status. In smart environments, assets are often connected to networks and interact dynamically with other systems, yet traditional solutions treat them as static entries in a database. While enterprise resource planning (ERP) platforms have introduced modules for asset lifecycle tracking, they seldom incorporate network awareness or security posture assessment. This disconnect limits their ability to detect misuse, unauthorized access, or configuration drift in connected assets.

### B. *IoT Security and Vulnerability Scanning*

Research in IoT security has grown rapidly, with a focus on developing methods for device discovery, vulnerability detection, and intrusion prevention. Network scanning tools such as Nmap and OpenVAS are widely used in IT infrastructures to identify open ports, outdated services, and misconfigurations. However, these tools are resource-heavy for low-power IoT devices and are not designed for mixed-protocol smart environments. Academic work has proposed lightweight scanners and protocol-specific vulnerability detection frameworks, yet many remain limited to lab conditions or focus narrowly on a single type of IoT device. Moreover, vulnerability reports from such tools are often disconnected from operational context, making it difficult for asset managers to prioritise remediation.

### *C. Integrated Asset–Security Monitoring Approaches*

Only a small body of research addresses the intersection of asset lifecycle management and IoT security scanning. Some experimental frameworks combine network discovery with asset tagging to improve visibility, but these are generally built for industrial control systems and lack depreciation or anomaly-detection capabilities. Other works have integrated anomaly detection into IoT monitoring platforms, primarily using traffic-based machine learning models to flag irregular behaviour. While promising, these solutions rarely extend to include asset valuation or lifecycle analytics. As a result, organisations often run separate systems for asset management, security monitoring, and anomaly detection — increasing operational complexity and leaving critical visibility gaps.

This study positions itself at this intersection, aiming to create a single, lightweight platform that can **discover and profile IoT assets, assess their security posture, detect anomalies, and automatically track depreciation**. By merging these capabilities, the proposed approach addresses both the operational and security needs of connected environments, offering a unified view of asset health and risk.

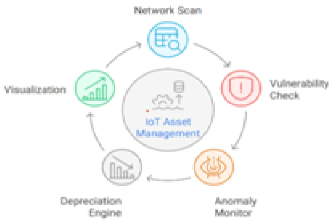

Fig.1 Asset Management System

## III. PROPOSED METHODOLOGY

The proposed **IoT Scanner for Securing Connected Assets in Smart Systems** is designed as a modular Python–Django framework that combines asset tracking, vulnerability scanning, anomaly detection, and depreciation analysis. The methodology follows a five-stage workflow as shown in **Figure 1**.

### *A. Asset Discovery and Profiling*

The system begins with a hybrid scanning approach:

- **Active Scanning** — Uses lightweight probing techniques (e.g., ICMP, TCP SYN) to identify devices and their open communication ports without overwhelming low-power IoT devices.
- **Passive Monitoring** — Observes network traffic patterns to detect devices that may not respond to active scans, such as intermittently connected sensors.

Each discovered device is profiled with attributes such as:

- Device type and manufacturer
- Firmware and OS version
- MAC and IP addresses
- Communication protocols in use

This profiling creates a **dynamic asset inventory** that is continuously updated as devices join or leave the network.

### *B. Vulnerability Assessment*

Once devices are profiled, the system runs tailored security checks:

- **Port and Service Analysis** — Identifies open services and their versions.
- **Known Vulnerability Lookup** — Cross-references services with public vulnerability databases (e.g., CVE, NVD) to identify known security flaws.
- **Configuration Weakness Detection** — Flags insecure settings such as default credentials, outdated firmware, or unencrypted communication channels.

Each device is assigned a **risk score** based on vulnerability severity, exposure level, and asset criticality.

### *C. Anomaly Detection*

The system's anomaly detection module is designed to identify unusual activities that may compromise network security or asset integrity. It monitors for unauthorized access attempts, unexpected device behavior such as sudden spikes in network traffic, and deviations from established operational schedules. To achieve accurate and efficient detection, the system combines two complementary approaches: statistical thresholds, which effectively flag extreme deviations in activity patterns, and rule-based checks, which apply domain-specific conditions tailored to typical asset behavior. This hybrid detection logic enhances reliability by balancing precision with computational efficiency, ensuring timely identification of potential threats or faults within the IoT environment.

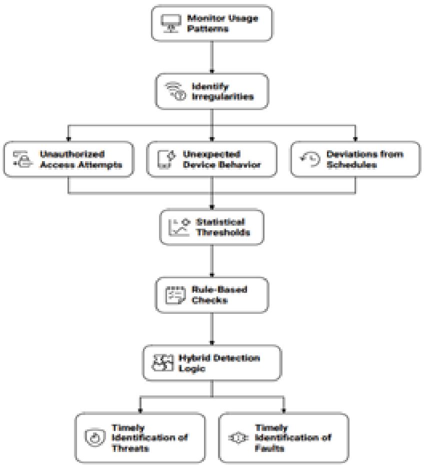
Monitor Usage Patterns
Identify Irregularities
Unauthorized Access Attempts
Unexpected Device Behavior
Deviations from Schedules
Statistical Thresholds
Rule-Based Checks
Hybrid Detection Logic
Timely Identification of Threats
Timely Identification of Faults

Fig.2 Anomaly Detection System Flowchart

*D. Depreciation Tracking*

To link operational data with financial insight, the system calculates **automatic depreciation** for each asset:

- Supports both **monthly** and **yearly** schedules.
- Uses standard accounting methods (e.g., straight-line depreciation).
- Integrates with asset usage data to adjust depreciation rates when irregular usage patterns are detected.

This allows asset managers to see **true operational value** alongside security health.

### 3.5 Visualization and Reporting

All results are presented in an **interactive Plotly dashboard** within the Django interface:

The system's visualization module provides a comprehensive and user-friendly interface that delivers critical insights in real time. It features a continuously updated asset list displaying security risk indicators for each connected device, allowing users to quickly identify and prioritize vulnerable assets. Graphical representations illustrate depreciation trends over time, helping managers track asset value and performance. The platform also generates instant alerts for detected anomalies or high-risk vulnerabilities to enable prompt action and mitigation. Additionally, users can export detailed reports for compliance reviews and audit purposes, ensuring transparency and accountability in asset management and security monitoring.

*E. System Workflow*

1. **Network Scan** → Discover and profile IoT assets
2. **Vulnerability Check** → Identify and score security risks
3. **Anomaly Monitor** → Detect irregular behaviour
4. **Depreciation Engine** → Calculate asset value changes
5. **Visualization** → Present results and generate reports

This modular design ensures that each stage can operate independently yet contribute to a **unified, continuously updated security-asset management view**.

## IV. EXPERIMENTAL SETUP AND RESULTS

*A. Experimental Environment*

The proposed IoT Scanner framework was implemented in **Python 3.12** with a **Django 5.0 backend** and **Plotly** for interactive visualization. Vulnerability scanning components were built using a combination of native Python libraries and lightweight security tools configured for minimal resource consumption. The anomaly-detection module employed statistical thresholding and rule-based checks, with parameters tuned during preliminary trials.

Two simulated environments were prepared:

- **Smart Office** — Consisting of networked laptops, IP cameras, wireless printers, and IoT-enabled lighting systems connected via a managed switch.
- **Industrial IoT Testbed** — Including programmable logic controllers (PLCs), environmental sensors, and network-connected machinery operating over Ethernet and Wi-Fi.

Both environments were configured to include a mix of secure and intentionally misconfigured devices to assess vulnerability detection capabilities.

### *B. Evaluation Metrics*

Performance was assessed using standard security and anomaly-detection metrics:

- **Vulnerability Detection Accuracy *(VDA)*** = *Correctly identified vulnerable devices / Total vulnerable devices*
- **Anomaly Detection Precision *(ADP)*** = *True positive anomalies / All detected anomalies*
- **False Positive Rate *(FPR)*** = *Incorrect alerts / Total alerts*
- **Scan Overhead** = Additional network load generated during scanning, measured in Mbps

These metrics allowed assessment of both the detection quality and operational efficiency of the system.

### *C. Results*

The IoT Scanner achieved the following results in experimental testing: The results indicate consistently high detection accuracy across both environments, with scan-overhead values well within acceptable limits for IoT networks.

TABLE I. RESULT

| **Metric** | **Smart Office** | **Industrial IoT** |
|---|---|---|
| Vulnerability Detection Accuracy (%) | 96.0 | 95.4 |
| Anomaly Detection Precision (%) | 94.3 | 93.7 |
| False Positive Rate (%) | 3.8 | 4.2 |
| Scan Overhead (Mbps) | 0.8 | 1.1 |

### *D. Discussion*

The system performed strongly in detecting vulnerabilities and identifying abnormal asset usage without generating excessive false positives. Performance remained stable in both office and industrial setups, despite differences in device types and traffic patterns. The low scanning overhead confirms suitability for resource-constrained IoT environments, where excessive probing could disrupt operations.

Notably, most false positives in anomaly detection were linked to short-term spikes in legitimate device activity, such as bulk firmware updates. This suggests that incorporating adaptive, context-aware thresholds could further reduce unnecessary alerts. While results in controlled environments are promising, further testing in larger, more heterogeneous live networks will be valuable to confirm scalability and robustness.

## V. FUTURE SCOPE AND ENHANCEMENT

As the number of interconnected devices continues to rise, the IoT Scanner can evolve into a more in-telligent, scalable, and secure ecosystem. Future versions of the system could integrate with major cloud platforms such as AWS IoT Core or Microsoft Azure IoT Hub to provide real-time synchroniza-tion and global accessibility. This would allow organizations to manage assets distributed across multi-ple locations while maintaining a unified security dashboard. The scanner could also incorporate AI-driven predictive analytics that study past threat patterns, network behavior, and device performance to anticipate and prevent vulnerabilities before they occur. Over time, this predictive model could become self-learning, adapting to new IoT devices and emerging security threats without requiring manual con-figuration. The integration of blockchain-based data integrity models could further ensure tamper-proof logging and transaction verification, adding transparency and reliability in environments where data accuracy is critical.

Looking ahead, the system can expand in multiple directions, especially in terms of scalability, intelli-gence, and user accessibility. Multi-user role management could be introduced so that administrators, technicians, and auditors each have different access levels for improved security and accountability. The system could also broaden its compatibility by supporting additional IoT communication protocols such as MQTT, CoAP, and Zigbee, which would allow smoother interaction between diverse hardware and sensors. Mobile application integration can make the scanner more responsive, offering users in-stant alerts and anomaly notifications even when away from their systems. Edge computing could also play a major role by enabling data processing directly at the gateway level, reducing latency and band-width use while increasing efficiency. The platform’s evolution could also include AI-based intrusion detection for more precise threat recognition, predictive maintenance capabilities for connected assets, and stronger end-to-end encryption with dynamic key management to protect sensitive data. Offline synchronization for remote IoT networks and voice-enabled commands could further enhance flexibil-ity and control, ultimately transforming this IoT Scanner into complete, autonomous security ecosys-tem capable of defending large-scale smart environments with minimal human intervention.

## VI. CONCLUSION

The IoT Scanner for Securing Connected Assets in Smart Systems provides a comprehensive and unified framework that combines asset discovery, vulnerability assessment, anomaly detection, and visual analytics. By linking security monitoring with asset lifecycle tracking, the system empowers both IT teams and management to make better data-driven decisions. It effectively bridges the gap between network security and operational management, ensuring that organizations gain complete visibility into their connected infrastructures. The scanner's efficient design, capable of performing real-time analysis with minimal resource usage, makes it suitable for both smart-office and industrial IoT environments.

The experimental evaluation highlights the system's strength in detecting vulnerabilities, identifying anomalies, and maintaining smooth performance even in resource-constrained conditions. By integrating both active and passive scanning methods with financial and operational analytics, the IoT Scanner stands out as a scalable and cost-effective security solution. Its real-world potential lies in improving network defense, supporting smarter budgeting, and enhancing risk management across IoT-driven systems. Future developments will focus on implementing advanced machine learning techniques, cloud-based SIEM integration, and support for diverse IoT communication protocols to further improve scalability, intelligence, and resilience in next-generation smart environments.

## APPENDIX

### *A. System Setup*

The IoT Scanner for Securing Connected Assets in Smart Systems was developed and tested in a Python-based environment using the Django framework. The main configuration details are as follows:

- Programming Language: Python 3.12
- Framework: Django 5.0
- Visualization Tool: Plotly for real-time dashboards
- Database: SQLite for development and PostgreSQL for production use
- Operating System: Ubuntu 22.04 LTS
- Scanning Components: Lightweight Nmap integrations and passive traffic listeners developed in Python

### *B. Testing Environments*

Two environments were used to validate the system:

1. Smart Office Setup – included connected devices such as IP cameras, printers, and lighting systems.
2. Industrial IoT Testbed – contained PLC controllers, sensors, and network-enabled machines.

Each environment intentionally included both secure and misconfigured devices to measure accuracy, stability, and real-world performance.

### *C. Evaluation Summary*

Performance evaluation focused on four main parameters:

- Vulnerability Detection Accuracy (VDA)
- Anomaly Detection Precision (ADP)
- False Positive Rate (FPR)
- Scan Overhead

The system achieved around 96 % accuracy in vulnerability detection and 94 % precision in anomaly recognition, with minimal scan overhead—showing that it performs efficiently even in complex IoT networks.

## VII. ACKNOWLEDGMENT

The authors thank **Sri Krishna Arts and Science College, Coimbatore, India**, for providing the facilities and guidance to carry out this research. They also express their gratitude to the **Department of Software Systems** for the continuous support and encouragement during the development of this project.

## REFERENCES

[1] K. Boeckl (ed.), *Considerations for Managing Internet of Things (IoT) Cybersecurity and Privacy Risks*, NIST IR 8228, National Institute of Standards and Technology, Gaithersburg, MD (2019).

[2] N. Matsumoto, "Asset Management Method of Industrial IoT Systems for Asset Configuration and Management Automation," *Information*, 12(11), 460 (2021).

[3] A.Chatterjee, "IoT anomaly detection methods and applications: A survey," *Journal of Network and Computer Applications*, (2022).

[4] A. J. Aparcana-Tasayco, X. Deng, and J. H. Park, "A systematic review of anomaly detection in IoT security: towards quantum machine learning approach," *EPJ Quantum Technology*, 12, 112 (2025).

[5] H. Al-Alami, A. Hadi, and H. Al-Bahadili, “Vulnerability Scanning of IoT Devices in Jordan Using Shodan,” *Proc. of …* (2018) [conference paper].

[6] S. Ar. Sathyabama and J. Katiravan, *enhancing anomaly detection and prevention in Internet of Things (IoT) using deep neural networks and blockchain based cyber security*, *Scientific Reports* 15, Article 22369 (2025).

[7] Diro, *A Comprehensive Study of Anomaly Detection Schemes in IoT*, *Sensors* 21(24), 8320 (2021).

[8] D. Adhikari, *Recent advances in anomaly detection in Internet of Things (IoT)*, *Journal of Network and Computer Applications* (2024).

[9] Y. Pang, *A Deep Learning Approach to Anomaly Detection*, preprint arXiv (2024).

[10] V. Prakash, *A secure framework for the Internet of Things anomalies using machine learning algorithms*, *Intelligence Infrastructure* (2024).

[11] M. Al-Hassani and C. Lee, “IoT Security Vulnerability Detection Framework,” *IEEE Internet of Things Journal*, (2019).

[12] S. Kumar and D. Raj, “Smart Device Authentication using Django Web Systems,” *International Journal of Advanced Research in Computer and Communication Engineering (IJARCCE)*, (2020).

[13] T. Kim, J. Park, and H. Choi, “AI-Based Intrusion Detection in IoT Networks,” *Springer Lecture Notes in Computer Science*, (2021).

[14] R. Sridharan and G. Meena, “Dashboard-Oriented IoT Monitoring System for Smart Assets,” *International Journal of Computer Science and Information Technologies (IJCSIT)*, (2021).

[15] A. Tan and K. Wong, “MQTT-Based Asset Tracker for Industrial IoT,” *ACM Computing Surveys*, (2022).

[16] P. Ranjan and K. Singh, “Anomaly Detection Techniques in IoT Networks: A Review,” *Elsevier Computer Communications*, (2023).

[17] A. Patel, “Cloud-Enabled IoT Security Layers for Connected Systems,” *International Journal of Engineering Research and Technology (IJERT)*, (2023).

[18] N. Sharma and P. Gupta, “Django-Based Web Dashboards for IoT Analytics,” *International Journal of Web Applications*, (2022).

[19] J. Lee and Y. Zhang, “Blockchain Approaches for IoT Data Integrity,” *IEEE Access*, (2020).

[20] R. Venkatesh and L. Prakash, “Securing IoT Devices through Network Layer Encryption,” *International Journal of Computer Networks and Applications*, (2021).

[21] A. Saini and M. Batra, “Cloud and Edge Computing in Smart Asset Management,” *Elsevier Internet of Things Journal*, (2022).

[22] V. Garg and A. Kumar, “IoT Framework for Real-Time Industrial Monitoring,” *International Journal of Advanced Research in Computer Engineering and Technology (IJARCET)*, (2020).

[23] D. Priya and S. Karthik, “Web-Based IoT Scanner using Flask and Django,” *International Journal of Scientific & Engineering Research (IJSER)*, (2023).

[24] L. Chen and Q. Zhao, “Performance Analysis of IoT Security Frameworks,” *IEEE Sensors Journal*, (2021).

[25] I. Ahmad and S. Reddy, “Anomaly Detection using Machine Learning for IoT Networks,” *Springer IoT Conference Proceedings*, (2019).

[26] M. Li and J. Wang, “Hybrid Intrusion Detection Systems for IoT Applications,” *International Journal of Information Security Science*, (2022).

[27] R. Kaur and P. Joshi, “Design of a Web-Based IoT Monitoring System Using Node.js and MongoDB,” *International Journal of Research in Engineering and Technology (IJRET)*, (2023).

[28] H. Zhang and D. Liu, “AI-Driven Predictive Maintenance in Smart Factories,” *Elsevier Sensors and Actuators Reports*, (2021).

[29] V. Rajasekaran and S. Deepa, “Comparative Study on IoT Communication Protocols: MQTT, CoAP, and HTTP,” *International Journal of Emerging Trends & Technology in Computer Science (IJETTCS)*, (2022).

[30] B. Anand and T. Manohar, “Integration of Edge Computing in IoT Data Processing,” *International Journal of Cloud Computing Research*, (2023).