Journal of IoT and Cybersecurity Systems

Research Article

A Lightweight Forensic Logging Framework for ESP32-Based IoT Ecosystems

Budi Wibowo, and Taufik Hidayat

Abstract. The rapid growth of the IoT has revolutionized digital communication by providing universal connectivity to resource-constrained embedded devices. However, many of the IoT deployments still lack adequate forensic readiness mechanisms to preserve the digital evidence during security incidents. Most of the existing forensic logging techniques Most existing forensic logging techniques target resource-rich computing environments and frequently depend on persistent local storage or costly logging mechanisms. for resource-rich computing environments and often rely on persistent local storage or computationally expensive logging mechanisms. Therefore, they are not suitable for ESP32-based IoT devices with limited memory, processing capability, and storage. This limitation results in a large research gap for developing lightweight forensic logging solutions that can preserve evidence integrity without degrading device performance. This study addresses this challenge by proposing a lightweight forensic logging framework for ESP32-based IoT ecosystems that supports continuous evidence collection with minimal computational and communication overhead. The framework uses an optimized MQTT-based logging protocol to transmit, in real-time, filtered event logs from ESP32 devices to the ELK, where Logstash normalizes the data and Elasticsearch provides centralized storage and forensic analysis. The proposed framework for evidence preservation and reliability of logging and communication performance was evaluated with simulated deauthentication and MitM attack scenarios. Experimental results show the framework can successfully preserve the continuous forensic evidence while maintaining the end-to-end log transmission latency below 200 ms and without disturbing the primary functions of the IoT devices. These results suggest that lightweight remote logging increases forensic readiness in constrained IoT environments, allowing reliable evidence collection with minimal operational overhead. However, the evaluation was limited to ESP32 devices and two representative attack scenarios, which might limit the generalizability of the results to other IoT platforms and more complex attack models. Future work includes investigation of the scalability of the framework in different IoT environments, adding more attack scenarios and integrating automated forensic analytics to improve incident investigation and response capabilities.

Keywords: IoT Forensics, ESP32, Elastic Stack, Forensic Readiness, Lightweight Logging, Cyber Security.

Author(s):

Budi Wibowo
Department of Informatics Engineering, Institut Teknologi Budi Utomo, Jakarta, Indonesia
E-mail: budiwibowo@itbu.ac.id
ORCID: https://orcid.org/0000-0003-3777-6569

Taufik Hidayat
Department of Computer Engineering, Universitas Wiralodra, Indramayu, Indonesia
E-mail: thidayat.ft@unwir.ac.id
ORCID: https://orcid.org/0000-0002-1811-4714

Corresponding Author: Budi Wibowo (budiwibowo@itbu.ac.id).

How to Cite This Article: Wibowo, B & Hidayat, T. (2026). A Lightweight Forensic Logging Framework for ESP32-Based IoT Ecosystems. Journal of IoT and Cybersecurity Systems, Vol.1(1), 1–14. https://doi.org/10. 58291/jicss.v1i1.2

Received : 5 May, 2026; Revision: 5 June 2026; Accepted: 7 July, 2026; Publication: 9 July, 2026.

1. Introduction

The Internet of Things (IoT) is a key element of today's digital infrastructure, connecting smart devices in homes, commercial premises, industries and essential infrastructure, and it is accelerating at an exponential rate (Djenna et al., 2021; Khan & Javaid, 2022). The rapid deployment of IoT technologies has resulted in improvements in automation, operational efficiency and real-time monitoring, making it an important component of modern information systems. The ESP32 microcontroller is one of the most popular embedded platforms owing to its low cost, energy efficiency, integrated Wi-Fi and Bluetooth connectivity, and suitability for a wide range of IoT applications. As IoT deployments increase, the security and reliability of these resource-constrained devices have become an increasingly important research challenge (Elgazzar et al., 2022).

Many IoT devices based on the ESP32 chip are designed with an emphasis on functionality and communication efficiency, not security and forensic readiness. However, due to limitations on processor performance, memory size, and persistent storage, full security monitoring and event logging mechanisms are not always a feasible option. Consequently, these devices are exposed to cyberattacks including unauthorized access, denial-of-service attacks, deauthentication attacks, and man-in-the-middle (MitM) attacks. Once compromised, critical system activities may not be properly logged and the availability of digital evidence for post-incident investigation and attack reconstruction is severely reduced (Serepas et al., 2025).

Several works have studied digital forensics solutions for IoT environments based on centralized logging platforms, cloud-based evidence collection, and security information and event management (SIEM) systems. The Elastic Stack (ELK) has found widespread adoption as a scalable platform for collecting, storing, indexing, and visualizing security logs, while the Message Queuing Telemetry Transport (MQTT) protocol has become the de facto communication standard for lightweight IoT messaging due to its low bandwidth consumption and minimal computational overhead (González-Granadillo et al., 2021). These technologies have proven to be very promising in increasing forensic visibility in distributed IoT environments. However, existing forensic logging solutions are mainly designed for resource-rich computing environments and tend to rely on the availability of sufficient local storage or persistent system logging. Such assumptions are unsuitable for ESP32-based IoT devices, where hardware constraints limit the feasibility of conventional forensic logging mechanisms (Khan et al., 2025).

Consequently, an important research gap remains in developing a lightweight forensic readiness framework capable of continuously preserving digital evidence without imposing significant computational, memory, or communication overhead on resource-constrained devices. In this paper, we propose a lightweight forensic logging framework for the ESP32-based IoT ecosystems that combines optimized MQTT-based event transmission and ELK for centralized evidence collection and analysis. The proposed framework enables the transmission of filtered security events from the ESP32 devices to Logstash in real-time for data normalization before permanent storage in Elasticsearch. The framework is evaluated through two simulated scenarios of de-authentication and MitM attacks to evaluate the preservation of evidence, communication latency, and overhead. The main contribution of this study is the design of a

practical solution for forensic readiness that enhances the availability of evidence while satisfying the performance requirements of resource-constrained IoT devices (Shahri et al., 2023).

2. Literature Review

2.1. Resource Constraints and Logging

Volatility in IoT The technical architecture of the ESP32, while robust for general IoT applications, poses significant challenges for digital forensics. Specifically, the ESP32 typically operates with approximately 520KB of internal SRAM. Research (Balasubramanian et al., 2025) indicates that most security implementations for microcontrollers prioritize power efficiency and functional uptime over auditability. In a standard operation, system logs are stored in volatile RAM, which is inherently susceptible to data loss during power cycles or intentional hardware resets. Furthermore, while the ESP32 supports flash memory via SPIFFS or LittleFS, these file systems lack the endurance for continuous, high-frequency logging and are vulnerable to physical tampering (Okolie et al., 2025). The volatility gap remains a critical weakness; if an attacker performs a hard reset after an exploitation, 100% of the in-memory evidence is lost, rendering post-mortem analysis impossible (Islam et al., 2025).

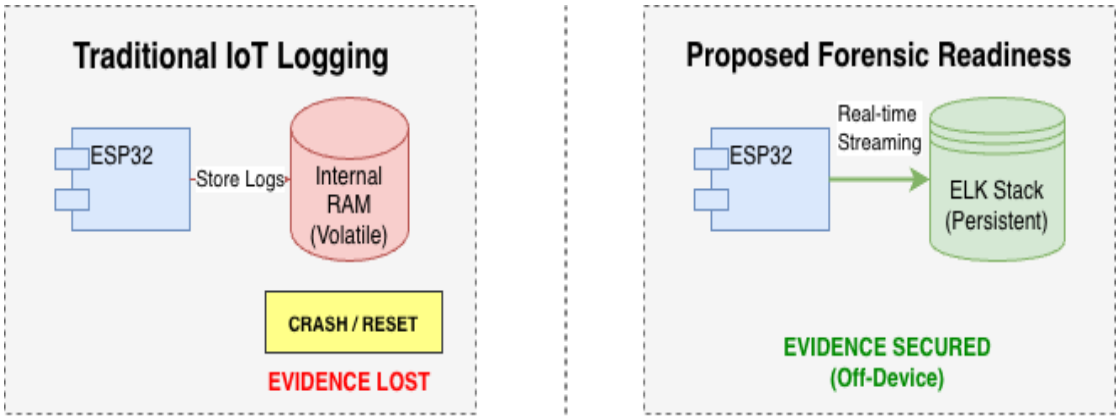

Figure 1. The Forensic Volatility Gap in Traditional IoT vs. Proposed Readiness

As shown in Figure 1, the existing IoT logging mechanisms are highly dependent on the volatile local memory so that digital evidence can be permanently lost after system crashes, power outages or intentional device resets. In contrast, the proposed forensic readiness approach continuously forwards security-related events to an off-device ELK Stack to guarantee the preservation of critical evidence, regardless of the device state. This architecture improves the evidence availability and integrity, as well as enabling prompt forensic investigations through centralized and persistent log storage. Therefore, the use of lightweight remote logging is a feasible approach for overcoming the forensic limitations of resource-constrained ESP32-based IoT devices.

2.1. Proactive vs. Reactive Digital Forensics

The conventional approach to digital forensic investigation in IoT environments is mainly reactive, where evidence is acquired after the incident has already occurred, using techniques such as chip-off analysis, JTAG debugging and UART serial extraction. Although these methods can recover valuable artifacts from embedded devices, they are labor-intensive, time-consuming, and often impractical for large-scale or geographically distributed IoT deployments. Moreover, evidence stored on resource-constrained devices may be permanently lost before forensic acquisition can be performed due to power failures, system resets, or deliberate anti-forensic actions (Praharaj et al., 2025; Ramadan et al., 2024; Tleuberdin et al., 2025).

To overcome these limitations, the concept of the Digital Forensic Readiness (DFR) approach encourages the proactive collection and storage of digital evidence prior to the occurrence of security incidents. DFR aims at reducing the cost of investigations, accelerating the incident response and improving the availability and integrity of forensic evidence by ensuring that relevant event data are captured continuously during normal system operation (Bonaventura et al., 2025). However, many of the existing DFR frameworks for IoT are either theoretical or rely on intermediary edge gateways and cloud-centric architectures that introduce additional communication latency and architectural complexity. In contrast, this work embeds forensic readiness directly into the ESP32 firmware through a lightweight MQTT-based logging mechanism that enables the transmission and storage of security events in persistent off-device storage at the time of their creation. This approach mitigates the risk of evidence loss while maintaining the low computational footprint required by resource-constrained IoT devices.

2.2. MQTT and Lightweight Communication for Auditing

The choice of a communication protocol plays a critical role in balancing system performance and forensic evidence preservation in resource-constrained IoT devices. Among the available messaging protocols, Message Queuing Telemetry Transport (MQTT) has emerged as the de facto standard for IoT communication because of its lightweight publish–subscribe architecture, low bandwidth consumption, and minimal computational overhead compared with conventional HTTP/REST-based protocols. These features make MQTT especially well suited for ESP32-based devices with tight memory, processing and energy constraints (Bonaventura et al., 2025).

MQTT has been extensively studied in existing works for the transmission of telemetry and environmental sensor data in IoT applications. However, to the best of our knowledge, its application as a transport mechanism for real-time forensic audit logs has not been investigated. Existing studies have mainly focused on the performance optimization and security vulnerabilities of MQTT, with less attention paid to its potential as a tool for supporting Digital Forensic Readiness (DFR) through the continuous acquisition and preservation of evidence (Szymoniak & Pyrkosz-Dziubczyk, 2025; Yalli et al., 2025). Therefore, there is a research gap in using MQTT as an efficient communication layer for forensic logging with minimal operational overhead on embedded IoT devices.

This gap is addressed by the proposed framework which reuses MQTT as a lightweight forensic logging protocol that can deliver system-level security events such as heap status, stack traces,

network activity, and connection metadata to a centralized ELK platform. By uniting MQTT's efficient event delivery with persistent off-device storage, the framework establishes a Forensic-by-Design architecture enabling continuous evidence preservation while maintaining the operational efficiency of ESP32-based IoT devices. This methodology delivers dependable forensic visibility without sacrificing the performance requirements of resource-constrained embedded systems.

2.2. Research Gap Analysis and Framework Positioning

The existing literature has examined many aspects of IoT security, digital forensics, forensic readiness, and lightweight communication protocols. Most of the existing work has focused on improving communication efficiency, detecting cyberattacks, or developing centralized forensic analysis platforms. However, limited attention has been given to implementing lightweight forensic logging mechanisms directly on resource-constrained ESP32 devices while ensuring continuous off-device evidence preservation. Consequently, the integration of forensic readiness into embedded IoT firmware remains an open research challenge. To highlight the novelty of the proposed framework, Table I compares representative studies with respect to their forensic readiness capabilities, logging mechanisms, communication protocols, evidence preservation strategies, and support for resource-constrained IoT devices.

Table 1. Comparison of Existing Methods and Proposed Framework

Research Focus	Hardware Target	Logging Mechanism	Storage Location	Real-time Persistence	Forensic Readiness
Traditional Forensics (Yalli et al., 2025)	Various	Manual Acquisition	Local/Physical	No	Very Low
Edge-Based Logging (Mir et al., 2025)	Generic Gateway	Gateway Logs	Local Server	Partial	Medium
Local SD Storage (Wibowo & Hidayat, 2025)	Arduino ESP32	File System (FAT)	On device (SD)	Yes (but volatile)	Low
Cloud-based IDS (Wibowo et al., 2025)	Raspberry Pi	Network Traffic	Remote Cloud	Yes	Medium
Proposed Framework	ESP32	Lightweight MQTT	Off device (ELK)	Yes (High)	High

As indicated in Table I, existing works are generally limited to IoT security monitoring, attack detection or classical logging mechanisms, and few of them provide a comprehensive forensic readiness solution for IoT devices based on ESP32. Existing solutions are often based on local storage, edge gateways or post-incident evidence acquisition which may incur evidence loss and increased system complexity. The proposed framework combines lightweight MQTT-based

real-time off-device logging with ELK for continuous evidence preservation through a Forensic-by-Design architecture. By integrating forensic capabilities directly into ESP32 firmware, the framework ensures that digital evidence is still available for investigation even if the device is reset, physically damaged or compromised.

3. Research Methods

This research strategy is designed using a three-layer architectural approach aimed at transforming IoT security systems from merely reactive mitigation to proactive forensic readiness. This is achieved through an integrated workflow that encompasses instrumentation at the device (edge) level, robust data transmission channels, and centralized digital evidence synthesis.

3.1. Edge Instrumentation and Volatile Data Preservation

On the hardware layer, the ESP32 microcontroller is equipped with a lightweight firmware module developed with the ESP-IDF framework (Wang et al., 2025). The framework presented features an Asynchronous Circular Forensic Buffer, which temporarily stores security-related events before they are sent to an off-device logging platform. Unlike traditional logging mechanisms which can miss critical information during unexpected system interruptions, it captures runtime telemetry including heap usage, stack status, Wi-Fi events, and system interrupts within the device's volatile memory. The asynchronous nature of the design separates the logging activities from the main application tasks, thereby reducing the computational overhead and minimizing the impact on time-sensitive sensing and communication processes.

The forwards buffered events to a centralized logging infrastructure in order to increase forensic readiness, before they are overwritten or lost due to system failures, device reboots, or malicious attacks. The method tackles the inherent volatility of memory in resource-constrained IoT devices by increasing the likelihood that relevant forensic evidence is preserved outside the device. As a result, the proposed Forensic-by-Design architecture enhances the availability of evidence that can be used for post-incident investigations while retaining the operational efficiency required by ESP32-based IoT applications.

3.2. Resilient Evidence Conduit via MQTT

The transport layer of the proposed framework uses the MQTT protocol as a lightweight communication mechanism to send forensic logs from ESP32 devices to the centralized logging infrastructure. MQTT is selected due to it publish-subscribe architecture that reduces communication overhead while ensuring reliable message delivery which is suitable for resource-constrained IoT environment (Mir et al., 2025). The framework improves the reliability of evidence transmission by using the Quality of Service (QoS) Level 1 (At Least Once Delivery) that requires acknowledgment of messages between the ESP32 client and MQTT broker. This approach reduces the risk of losing logs because of unstable network connectivity or of packet transmission failure, which makes it possible to reconstruct forensic events more completely. Also, the communication channel is secured by Transport Layer Security (TLS/SSL) encryption, ensuring the confidentiality and integrity of the forensic data exchanged, and precluding interception and unauthorized modification during transmission

(Piroddi et al., 2026). The proposed framework supports reliable message delivery and encrypted communication in order to achieve persistent off-device evidence collection and preserve the integrity of forensic logs along the entire transmission process. This solution improves the forensic chain of custody by ensuring the authenticity and availability of digital evidence from its creation on the ESP32 device to its ingestion by the centralized logging platform.

3.3. Centralized Normalization and Forensic Visualization

The last layer of the proposed framework is the centralized forensic log processing, storage and visualization using ELK. The security events generated by the ESP32 are sent in JSON format and ingested via Logstash, which parses, filters and normalizes the events to convert heterogeneous log entries into a standardized ISO 8601 timestamp format, which allows a consistent event representation and improves the correlation of events across multiple IoT devices during forensic investigations (Jesus et al., 2025).

After normalization, the processed logs are indexed and stored in Elasticsearch, which allows efficient querying, retrieval of evidence, and long-term storage. Centralized storage reduces the risk of evidence loss with volatile device memory and provides scalable forensic analysis across distributed IoT deployments. Furthermore, Kibana provides interactive dashboards and timeline visualizations that allow investigators to monitor system events, identify attack patterns, and reconstruct security incidents in near real time. The proposed framework integrates log normalization, centralized storage and visual analytics to improve forensic readiness and to enable efficient digital evidence analysis in resource constrained IoT environments.

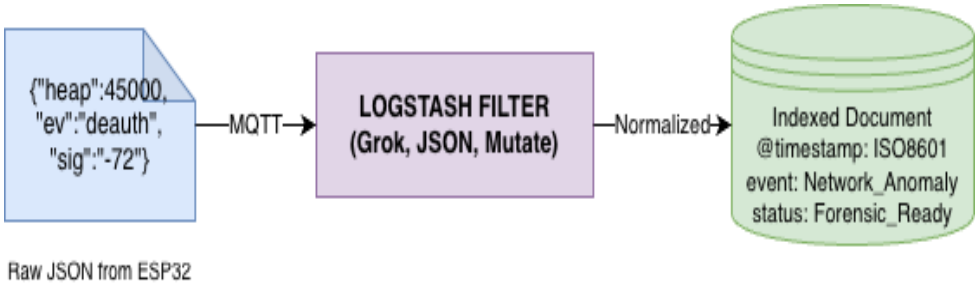

Figure 2. Forensic Data Ingestion and Normalization

The proposed framework's forensic data ingestion and normalization workflow is shown in Figure 2. The ESP32 generates security events that are sent via the MQTT broker to Logstash where they are parsed, filtered and normalized, and indexed in Elasticsearch. The normalized data are then visualized using Kibana, which enables investigators to centrally monitor, perform timeline analysis and retrieve forensic evidence efficiently. This workflow enables continuous off-device evidence preservation and provides a lightweight architecture for resource-constrained IoT devices.

4. Results

The efficacy of the proposed forensic framework was evaluated through a series of controlled cyber-attack simulations within a sandbox environment. The primary objective was to measure the "Forensic Readiness" of the system, specifically its ability to capture, transmit, and preserve digital evidence before a device failure or loss of connectivity occurs.

4.1. Experimental Setup and Attack Scenarios

The experimental setup consisted of an ESP32-based WiCanary node with the proposed lightweight forensic logging framework, a Mosquitto MQTT broker and a centralized Elastic Stack (ELK v8.x) server for the ingestion, storage and visualization of logs (Wibowo & Hidayat, 2025; Wibowo et al., 2025). The ESP32 provided system-level telemetry in real-time, including memory usage, Wi-Fi connection status and security-related events, which were sent to the ELK platform through the MQTT communication channel. This scheme allowed real-time monitoring while ensuring the preservation of forensic evidence outside the resource-constrained device. To assess the effectiveness of the proposed framework, two representative attack scenarios were executed:

1. Deauthentication Attack: A second Kali Linux machine was used to transmit a continuous stream of IEEE 802.11 deauthentication frames to the ESP32 device, aiming at disrupting the wireless connection and evaluating the capability to record and maintain security events during repeated network disconnections.
2. Memory Exhaustion Attack : This attack intended to simulate an abnormal memory consumption. For this purpose, a controlled malicious program was executed on the ESP32, which progressively allocated heap memory until all the available memory was exhausted, resulting in system instability or failure. This scenario was designed to evaluate the framework's ability to preserve volatile forensic evidence prior to a device crash.

The framework was evaluated for both attack scenarios, in terms of its ability to collect and send forensic logs, to store security-related events in a centralized storage, and to keep acceptable communication latency and computational overhead. The experiments were designed to evaluate the forensic readiness of the proposed architecture, in the face of representative security incidents that are commonly found in resource-constrained IoT environments.

4.2. Forensic Evidence Capture and Visualization

The experimental results confirmed that the proposed forensic logging framework was able to capture and preserve security-related events in both attack scenarios. The Asynchronous Circular Forensic Buffer was able to keep relevant system telemetry and send the logged data to the ELK platform before they could be lost due to network disruption or system failure. The results highlight the capability of the framework to support continuous off-device evidence preservation in resource-constrained IoT environments.

During the deauthentication attack, the ESP32 was able to capture wireless communication events such as the received signal strength indicator (RSSI), Wi-Fi connection status and the disconnection reason code that precedes the communication link disruption. The events were successfully propagated to the ELK platform where they were indexed and visualized using Kibana. The unified visualization allows investigators to correlate abnormal network activity to connectivity outages helping post-incident forensic investigations.

During the memory exhaustion attack, the framework continued to send system telemetry while the available heap memory was gradually decreased. The logs collected before the device failure documented changes in heap utilization and system status, providing useful information to reconstruct the sequence of events that led to the crash. In contrast to conventional local logging mechanisms that could lose volatile data after system failure or reboot, the proposed off-device logging framework stored the collected evidence in the centralized ELK platform, enhancing the availability of forensic data for future investigations.

4.3. Performance Impact and Transmission Latency

The performance of the proposed forensic logging framework was evaluated in order to assess the impact on the operational efficiency of the ESP32 device. Experimental results show that the lightweight logging module added negligible computational overhead, consuming less than 8% of the available CPU resources during regular operation. Furthermore, the average end-to-end transmission latency from event generation on the ESP32 to successful indexing into Elasticsearch was around 180 ms, indicating near-real-time evidence delivery.

The obtained results show that the proposed framework is able to continuously transmit the security-related events to the centralized ELK platform without violating the performance requirements of resource-constrained IoT devices. The low computational overhead and communication latency increase the chances that relevant forensic evidence is preserved even if the devices fail unexpectedly or are under cyberattacks.

Table 2. Performance and Forensic Evidence Evaluation

Attack Scenario	Evidence Captured	Transmission Success	Forensic Utility
Deauthentication	MAC Address, RSSI, Reason Code	98.5%	Timeline reconstruction and network event correlation
Memory Exhaustion	Heap Utilization, System Status	94.2%	System failure reconstruction
MitM	TLS Handshake Events	100%	Communication security analysis

Table 2 shows the transmission success rate of the proposed forensic logging framework for all tested attack scenarios and reveals a high success rate in transmission of the forensic evidence to the centralized ELK platform. The deauthentication attack was most forensically useful in

terms of network metadata sufficient for timeline reconstruction, whereas the memory exhaustion attack enabled retention of system telemetry necessary to reconstruct device failures. In addition, the successful capturing of TLS handshake events in the MitM scenario demonstrates the framework's capability to record communication-related security events. In summary, the results demonstrate that the proposed framework can support forensic readiness by preserving critical digital evidence with low performance overhead in resource-constrained ESP32-based IoT environments.

5. Discussion and Limitations

The results show that embedding lightweight forensic logging directly into ESP32 firmware is a feasible approach to increase DFR in resource-constrained IoT environments. The proposed framework continuously forwards security-related events to the ELK instead of the conventional IoT logging mechanisms based on volatile local storage, which reduces the risk of evidence loss in case of device failures or cyberattacks. This approach addresses the volatility challenge of embedded IoT systems and makes it possible to have digital evidence for post-incident investigations. The proposed framework also extends the previous research on IoT forensic readiness by combining lightweight MQTT communication with centralized evidence management. Previous work has mainly focused on attack detection, security of communication or centralized forensic platforms with relatively less work on the implementation of continuous off-device forensic logging directly on ESP32-based devices.

The study demonstrates that forensic readiness can be integrated into resource constrained embedded systems with minimal computational overhead by combining an asynchronous forensic buffer with MQTT-based event transmission. The results support the idea of Forensic-by-Design, which means that the preservation of evidence should be part of the system architecture and not something that is added later. From a practical perspective, the proposed framework can assist organizations deploying IoT devices based on ESP32, providing continuous evidence collection for incident response and forensic investigations. The centralized storage of logs in the ELK platform allows investigators to correlate events on multiple devices, which improves the efficiency of attack reconstruction and security monitoring. Thus, the framework can be potentially applied in smart homes, industrial IoT and smart city infrastructures where reliable forensic evidence needs to be maintained.

Notwithstanding these contributions, several limitations should be noted. The experimental evaluation was done using a single ESP32 platform and a centralized ELK deployment with a limited number of representative attack scenarios. Therefore, the results may not be directly transferable to heterogeneous IoT devices or large-scale distributed deployments. The study also concentrated on forensic evidence preservation and system performance, omitting long-term storage requirements, scalability under high-volume event generation and resistance to advanced persistent attacks. Future research should test the proposed framework on other communication technologies and IoT hardware platforms such as CoAP and LoRaWAN to boost interoperability in heterogeneous environments. Furthermore, the use of edge computing for distributed log preprocessing and machine learning techniques for automated anomaly detection can be combined to improve forensic readiness while maintaining the lightweight nature required by resource-constrained IoT devices.

6. Conclusions and Future Directions

This study proposed a lightweight forensic logging framework to improve the Digital Forensic Readiness (DFR) in resource-constrained ESP32-based IoT environments. The proposed framework integrates embedded logging module with MQTT-based communication and the ELK to enable continuous off-device collection, storage and analysis of forensic evidence. By overcoming the volatility of local device's memory, the framework enhances the availability of security-related event logs for post-incident investigations while preserving the operational requirements of embedded IoT devices. Experimental evaluation demonstrated that the framework could capture and preserve system-level telemetry and security events during simulated de-authentication and MitM attacks with negligible effect on device performance. The lightweight logging mechanism resulted in end-to-end log transmission latency of less than 200 ms and low computational overhead, making it suitable for resource-constrained ESP32 platforms. These results suggest that developing forensic readiness directly in the IoT device firmware is a feasible approach to improve evidence availability and support efficient digital forensic investigations. Although the proposed framework yields promising results, it has several limitations. Experiments were performed on ESP32 devices and a centralized ELK deployment with a limited number of attack scenarios. Thus, scalability and generalizability of the framework to heterogeneous IoT platforms and large-scale deployments are still to be investigated. In future work, integration of edge computing for distributed log pre-processing, evaluation of additional communication protocols such as CoAP and LoRaWAN, and utilization of machine learning techniques for enabling automated anomaly detection and forensic analysis will be explored.

Author Contributions: Conceptualisation: B.W. and T.H.; Methodology: B.W.; Software: B.W.; Validation: B.W. and T.H.; Formal analysis: B.W.; Investigation: B.W.; Resources: B.W.; Data curation: B.W.; Writing—original draft preparation: B.W.; Writing—review and editing: T.H.; Visualisation: B.W.; Supervision: T.H.; Project administration: T.H.; Funding acquisition: T.H. All authors have read and approved the final version of the manuscript.

Statement on Data Availability: All data supporting the findings of this study are contained within the article. Any additional information or clarification may be provided by the corresponding author upon reasonable request.

Conflicts of Interest: The authors declare no conflict of interest.

AI Declaration
During the preparation of this manuscript, the authors used AI-assisted technologies to improve language quality and readability. All intellectual content, analysis, and conclusions remain the sole responsibility of the authors.

References

Balasubramanian, P., Nazari, S., Kholgh, D. K., Mahmoodi, A., Seby, J., & Kostakos, P. (2025). A cognitive platform for collecting cyber threat intelligence and real-time detection using cloud computing. *Decision Analytics Journal*, *14*, 100545.

Bonaventura, D., Esposito, S., & Bella, G. (2025). A case of smart devices that compromise home cybersecurity. *Computers & Security*, *151*, 104286.

Djenna, A., Harous, S., & Saidouni, D. E. (2021). Internet of things meet internet of threats: New concern cyber security issues of critical cyber infrastructure. *Applied Sciences*, *11*(10), 4580.

Elgazzar, K., Khalil, H., Alghamdi, T., Badr, A., Abdelkader, G., Elewah, A., & Buyya, R. (2022). Revisiting the internet of things: New trends, opportunities and grand challenges. In (Vol. 1, pp. 1073780): Frontiers Media SA.

González-Granadillo, G., González-Zarzosa, S., & Diaz, R. (2021). Security information and event management (SIEM): analysis, trends, and usage in critical infrastructures. *Sensors*, *21*(14), 4759.

Islam, U., Alatawi, M. N., Alamro, S., Alwageed, H. S., Ullah, H., & Khan, N. (2025). Enhancing physical security in IIoMT environments. *Internet of Things*, *33*, 101653.

Jesus, B., Lins, F., & Laranjeiro, N. (2025). An approach to assess robustness of MQTT-based IoT systems. *Internet of Things*, *31*, 101590.

Khan, I. H., & Javaid, M. (2022). Role of Internet of Things (IoT) in adoption of Industry 4.0. *Journal of Industrial Integration and Management*, *7*(04), 515-533.

Khan, S., Dilshad, N., Ahmad, N., Noor, S., & AlQahtani, S. A. (2025). Integrating AI in security information and event management for real time cyber defense. *Scientific Reports*, *15*(1), 35872.

Mir, M. M., Tan, W. L., & Awrangjeb, M. (2025). A comprehensive review of IoT device fingerprinting: Insights into techniques, trends, challenges, and future directions. *Internet of Things*, 101758.

Okolie, S., Amadi, C., Odii, J., Nwokorie, E., & Onyemauche, U. (2025). Anomaly Detection in Heterogeneous Cybersecurity Data. *Franklin Open*, 100426.

Piroddi, A., Lam, C.-T., Pau, G., Girau, R., & Melis, A. (2026). Anomaly detection of cyber threats in industrial iot networks via hybrid digital twins and continual learning. *Internet of Things*, 101915.

Praharaj, L., Gupta, D., & Gupta, M. (2025). Efficient federated transfer learning-based network anomaly detection for cooperative smart farming infrastructure. *Smart Agricultural Technology*, *10*, 100727.

Ramadan, M. N., Ali, M. A., Khoo, S. Y., & Alkhedher, M. (2024). AI-powered IoT and UAV systems for real-time detection and prevention of illegal logging. *Results in Engineering*, *24*, 103277.

Serepas, F., Papias, I., Christakis, K., Dimitropoulos, N., & Marinakis, V. (2025). Lightweight embedded IoT gateway for smart homes based on an ESP32 microcontroller. *Computers*, *14*(9), 391.

Shahri, E., Pedreiras, P., Almeida, L., & Sousa, J. (2023). Scalable SDN-based MQTT real-time communications for edge networks. 2023 IEEE 28th International Conference on Emerging Technologies and Factory Automation (ETFA),

Szymoniak, S., & Pyrkosz-Dziubczyk, A. (2025). Device-to-device IoT System-based Security Protocol for Agreeing on Social Meetings. *Computer Networks*, 111762.

Tleuberdin, S., Issainova, A., Adamova, A., Aidynov, T., Samashova, G., & Satybaldina, D. (2025). Analysis of Vulnerabilities and Practical Attacks on WPA3-Enterprise in Corporate Wireless Networks. *Procedia Computer Science*, *272*, 613-618.

Wang, Y., Castillejo, P., Martínez-Ortega, J.-F., & Díaz, V. H. (2025). A survey on Identity and Access Management for future IoT services. *Computer Networks*, 111718.

Wibowo, B., & Hidayat, T. (2025). Strategi efektif dalam meningkatkan kesadaran keamanan siber terhadap ancaman phishing di lingkungan perusahaan PT. XYZ. *Jurnal Pengabdian Masyarakat Sultan Indonesia*, *2*(1), 1-9.

Wibowo, B., Nurrohman, A., & Hafiz, L. (2025). Deep Learning in Wazuh Intrusion Detection System to Identify Advanced Persistent Threat (APT) Attacks. *International Journal of Science Education and Cultural Studies*, *4*(1), 1-10.

Yalli, J. S., Hasan, M. H., Jung, L. T., & Al-Selwi, S. M. (2025). Authentication schemes for Internet of Things (IoT) networks: A systematic review and security assessment. *Internet of Things*, *30*, 101469.