Future Technology
Open Access Journal
ISSN 2832-0379

February 2027| Volume 06 | Issue 01 | Pages 24-34

Journal homepage: https://fupubco.com/futech

https://doi.org/10.55670/fpll.futech.6.1.3

Future Publishing LLC

Article

Security of sensitive documents through a hybrid Bitcoin-Chia architecture using AI: a real-time secure framework

Hiyam Hashim Saeed[1,2]*, Ziyad Tariq Mustafa Al-Ta'i[1]

[1]Department of Computer Science, College of Science, University of Diyala, Baquba, Iraq
[2]Ministry of Higher Education and Scientific Research, Iraq 500085

ARTICLE INFO

Article history:
Received 24 June 2026
Received in revised form
15 October 2026
Accepted 03 November 2026

Keywords:
Sovereign AI, Hybrid blockchain, Wazuh SIEM, IRIS SOAR, Smart contract auditing, Forensic immutability

*Corresponding author
Email address:
scicomphd232403@uodiyala.edu.iq

DOI: 10.55670/fpll.futech.6.1.3

ABSTRACT

This paper focuses not on encryption or content access-control policies, but on forensic integrity and tamper-evident logging of sensitive documents. Ensuring confidential documents remain secure in today's digital ecosystems is both a challenge and a trilemma: how to ensure cryptographic integrity, keep the operations of the system available, and respond intelligently and contextually to threats. Current SIEM solutions have high false-positive rates, and no logging solution today provides the forensic integrity essential for automated response processes. This research paper proposes a new Autonomous AI-driven Blockchain Security Lifecycle Architecture and empirically demonstrates 97.64% classification accuracy of 97.64% in a production environment, with the ability to log any modification to sensitive configurations and create a tamper-evident record on the blockchain. Moving beyond simulation-based research, we implement and test the framework on a real-time production server with a 20-core CPU and 48 GB of RAM. The system has a core based on version 30 of Bitcoin Core as an unalterable trust anchor, the Chia Network as a resilient peer-to-peer network to support failover, and Gemma-7b-Instruct Large Language Models (LLMs) that are locally deployed through Ollama to detect contextual anomalies. Wazuh handles telemetry ingestion, while IRIS provides Security Orchestration, Automation, and Response (SOAR) and runs automated playbooks aligned with NIST SP 800-61. Empirical evaluations demonstrate a 97.64% accuracy in detecting attacks, a reduction in Mean Time to Respond (MTTR) of 642.5 s to 16.15 s, and automatic failover to prevent data loss during failure of one of the nodes. Wazuh's File Integrity Monitoring (FIM) is a framework tested for 90 days that has achieved 97.64% accuracy in identifying whether a file change is legitimate or suspicious. Additionally, using a local LLM reduces false positives by 86.5% compared to rule-based baselines, making it a promising approach for privacy-preserving self-healing cybersecurity models.

1. Introduction

Many traditional SIEM solutions cause "alert fatigue" because of a high number of false-positive alerts and the cognitive burden on analysts [1]. Blockchain can provide a tamper-proof record of events, but it's not easily scalable and therefore hard to embed in real-time IR workflows [2]. These concepts have been shown to be valid in simulated settings, but not in real production deployments [3-5]. In addition, previous studies have focused on applying LLMs to alert enrichment, such as Ismail et al. [6], without forensic chaining or automatic execution. Compliance frameworks like NIST SP 800-61 [6], CIS Benchmarks [7], and FIPS 140-3 [8] exist, but empirical evidence of compliance with these standards in practice for sovereign AI and hybrid ledgers is lacking. Organizations need to move from a reactive, rules-based strategy to a proactive, autonomous defense strategy, as cyber threats target sensitive documents. SIEM systems like Wazuh are good for log aggregation but may provide limited context, which can lead to analyst fatigue. Blockchain technology provides a permanent record of information, but it also faces scalability issues and may not align with real-time incident-response orchestration [2]. Current blockchain-based logging systems, like BCALS [3] and EngraveChain [4], have been tested only in simulated or cloud-based environments and do not reflect the latency and resource limitations of hybrid on-premises production environments.

A large gap remains in frameworks that combine cryptographic immutability, infrastructure resilience, and AI-based contextual intelligence into an international-standard (NIST, CIS, FIPS) deployable architecture [9,10]. This paper presents a comprehensive solution known as the Autonomous AI-Driven Blockchain Security Lifecycle.

2. Problem statement & research objectives

Existing frameworks have a significant gap: they either focus on cryptographic immutability without real-time response or rely on AI-driven detection without forensic accountability, often only in simulations. The following three testable goals are followed in this study to close this gap and are tested empirically using a physical production server:

Objective 1 (Hybrid Trust): To have long-term finality of the forensics, whilst having a failover resilience under 80s with no data loss, using Bitcoin Core v30.0-rc3 and Chia Network's failover.

Objective 2 (Sovereign Intelligence): Deploy Gemma-7b-Instruct (Q4_K_M) locally to lower false positive rates by more than 80% compared to the rule-based baseline, without depending on an external API

Objective 3 (Automated Response): Achieve a Mean Time to Respond (MTTR) for high-severity events in the FIM playbook of less than 20s, as compared to a current manual baseline of >600s.

3. Related work

3.1 LLM-enhanced security operations

Large Language Models (LLMs) and Security Operations Centers (SOCs) are increasingly being integrated, with contextual alert enrichment as a key use case. Ismail et al. [11] showed that Retrieval-Augmented Generation (RAG) can improve Wazuh alerts, achieving a BERTScore F1 above 0.85. Similarly, Han et al. [12] showed that template-free language modeling achieves high detection accuracy on static datasets like HDFS. These semantic advances are emerging, but current frameworks remain largely constrained: they provide advice, operate offline, and are not integrated with live SOAR telemetry streams. Most importantly, no empirical validation exists of a closed loop between AI-driven analysis and forensically viable automated incident response in a hybrid ledger environment.

3.2 Blockchain for immutable logging and forensics

Although cloud-appropriate blockchain-based logging solutions like BCALS [3] and EngraveChain [4] are presented as tamper-evident cloud logging, their rigidity makes them hard to implement in practice. These frameworks usually rely on a single consensus mechanism, which invariably forces a trade-off between real-time speed/usability and forensic finality/certainty (PoW). More importantly, they are archival layers not natively connected to active defense mechanisms and cannot initiate automatic containment in the event of integrity violations. Simpler solutions, such as OpenTimestamps [13], have lower overhead because they are decentralized and don't require a smart contract. However, they are passive and do not enable a closed loop expected for protecting sensitive documents. The gap is the need for a proactive, responsive mitigation approach using Artificial Intelligence, along with cryptographic anchoring, something most stand-alone ledger solutions do not provide. Recent proposals include tamper-proof event logging in [14] and LogStamping [15], both passive records but with no analysis or automatic response.

3.3 Automated response and orchestration (SOAR)

Recent studies have addressed automating incident response, focusing on translating legacy playbooks into standard CACAO formats and achieving high syntactic validity with the help of LLMs [16]. These are not, by default, active feeds, are not part of a live stream of telemetry, do not have in-built forensic validation mechanisms, and thus are not suitable for active defense. Furthermore, directly embedding ML into consensus protocols increases analysis and agreement latency because it tightly couples' analysis and agreement logic [17]. One evident problem is that SOAR is not necessarily forensic or real-time. It does not meet production needs for closed-loop automation.

3.4 Research gap

Despite these developments, no current frameworks combine local sovereign AI, a dual-blockchain trust/resilience model, and automated SOAR orchestration in a real production program. Current research relies on simulations; AI-based decisions lack a forensic foundation, and many do not support hybrid infrastructure (on-premises + Cloud). This research fills these gaps by empirically validating a fully autonomous lifecycle running on a physical server and showing statistically significant improvements in detection accuracy, response latency, and system resilience.

4. Proposed architecture

As shown in Figure 1, the proposed framework comprises four separate logical layers, designed to meet six key security goals: monitoring, anomaly detection, predictive analysis, automated response, recovery, and smart contract auditing. In Figure 1, the architecture has four layers of decoupled architecture: (1) Infrastructure & Trust Layer based on immutable anchoring with Bitcoin Core v30.0-rc3 and resilient failover with Chia Network; (2) Monitoring and Intelligence Layer with Wazuh SIEM and locally hosted Gemma-7b-Instruct LLMs using Ollama to detect contextual anomalies; (3) Orchestration Layer, based on NIST SP 800-61, automated response with IRIS SOAR; and (4) Self-Healing Layer where high-availability ends and information is exchanged between layers in a safe manner that ensures the sovereignty of data and forensic integrity. Table 1 compares the state-of-the-art Frameworks and the proposed solution.

4.1 Layer 1: Infrastructure & trust (Bitcoin core v30 & Chia network)

- **Trust Layer (Bitcoin Core v30):**

This system uses an immutable ledger to anchor cryptographic hashes of sensitive documents and high-severity security events, providing a verifiable way to maintain records [22]. To mathematically define the integrity verification function $I(f,t)$ for a file f at time t, we use the hash comparison with the baseline anchored on Bitcoin:

$$I(f,t) = \begin{cases} 1 & if\ H_{SHA256}(f_t) = H_{baseline}(f) \\ 0 & if\ H_{SHA256}(f_t) \neq H_{baseline}(f) \end{cases} \quad (1)$$

$H_{baseline}(f)$ is the hash that is stored in the Bitcoin OP_RETURN transaction. If the value is 0, it immediately alerts the SOAR layer. The anchoring latency L_{anchor} is the time it takes to confirm the blocks and therefore the time needed for computational finality:

$$L_{anchor} \approx T_{block} \times N_{conf} \quad (2)$$

where T_{block} ≈10 minutes (average block time in Bitcoin) and N_{conf} is the number of confirmations needed for a transaction to be final (usually N=6).

Table 1. Comparative analysis between state-of-the-art frameworks and the proposed solution

Ref.	Problem Addressed	Methodology	Key Quantitative Results	Objective Limitation (Gap)	Deployment Setting	Latency / FPR
[11] Ismail et al. (2025)	Lack of context in SIEM alerts; Analyst fatigue	Wazuh + RAG-based LLM Copilot	BERTScore F1 > 0.85; Cosine Sim. 0.46–0.76	AI outputs are suggestive only – No forensic chaining or automatic responses.	Simulation / Offline Lab	N/A
[16] Gurabi et al. (2025)	Manual playbook development	LLM-assisted CACAO translation	Syntactic validity: 90% Semantic coherence: >85%	Playbooks are just static documents; no connection to data streams or forensic validation.	Offline / Preprint	N/A / N/A
[14] Morillo Reina & Mateo Sanguino (2025)	Events are recorded in a tamper-proof manner.	Decentralized Blockchain Event logging (Paired communication)	Fully Integrated Data with a delay of ~200ms.	No link to SOC/SIEM tools, no active response; fixed pairwise consensus - not scalable;	Not reported	~0.2 s / N/A
[18] Choi et al. (2025)	Smart-contract vulnerability detection	LLM+graph structural analysis	F1 +15% (to 0.79) on complex graph-based flaws	Non-existence of an operational SOC orchestration. There is no working SOC orchestration.	Not reported	N/A / N/A
[19] Liu et al. (2025)	Logical vulnerabilities of smart contracts.	Using ChatGPT to reason about contract logic	~60% detected / 18% false positives.	No operational verifications and no forensic storage.	Not reported	N/A / 18%
[15] Islam & Rahman (2025)	The volume of logging is very large-scale.	Log timestamping using blockchain (LogStamping)	Handled millions of hashed log entries (14 million tested)	No analytics or response; passive recording layer	Not reported	N/A / N/A
[17] Venkatesan & Rahayu (2024)	Consensus protocol attack detection	ML-integrated Hybrid Consensus	Accuracy 94.2%; FPR 3.5%	Rigidity & latency – Tight coupling, Lacks independent forensic accountability	Simulation	High / 3.5%
[20] Gupta et al. (2024)	Blockchain network eclipse attack detection	Non-parametric change detection for statistical.	The accuracy is ~90%, and 10% of the training data is used.	No connection with SOC response	Not reported	~0.2 s / N/A
[12] Han et al. (2023)	Log Anomaly Detection without Template.	LogGPT (Language model based on GPT)	F1 > 0.92 (HDFS); Near-perfect on BGL	Tested on a static set of data, offline; Not integrated with real-time SOC data or IR playbooks.	Offline Dataset Evaluation	N/A / N/A
[3] Ali et al. (2022)	Insider threat and cloud log tampering	BCALS (Blockchain-anchored logging)	0% tampering success; 3.5× storage overhead	Only cloud-based architectures; No active defence layer of the recording layer	Cloud-Native Only	High (PoW) / N/A
[4]Shekhtman & Waisbard (2021)	Distributed log integrity	Decentralized log registry (EngraveChain)	Fully detectable, scalable to 1,000 TPS.	Only logging, no AI analysis or mitigation capabilities;	Simulation	Moderate / N/A
[21] Xu et al. (2020)	Eclipse attack detection for Ethereum	Random Forest on network attributes	Accuracy 96.5%; FPR < 2%	The lack of explanation and adaptability.	Not reported	N/A / <2%

- **Resilience Layer (Chia Network)**

Chia is a peer-to-peer communications substrate, with Proof-of-Space-and-Time (PoST) validation. It synchronizes with a hot-standby node, enabling failover or redundancy much faster than proof-of-work systems without consuming energy. The Chia Network provides resilience and operational continuity, unlike Bitcoin [23]. Since Chia lacks native application-state replication, we developed a custom Filesystem Watcher Daemon to monitor critical database files (including the Chia wallet DB and application state). The daemon activates on change detection and initiates an encrypted rsync transfer of all data to the hot-standby node every 5 seconds.

This capability ensures quick consistency of state and a sub-80s Recovery Time Objective (RTO) in the case of failure and also helps to distinguish between operational resilience and Bitcoin's slower forensic anchoring process. The probability P sync that a backup node has a valid state copy at time t is modeled as:

$$P_{sync}(t) = 1 - e^{-\lambda t} \quad (3)$$

where λ is the synchronization rate parameter derived from the Chia gossip protocol efficiency. In our implementation, λ is tuned to ensure P_{sync} >0.99 within 54 seconds (3 Chia blocks).

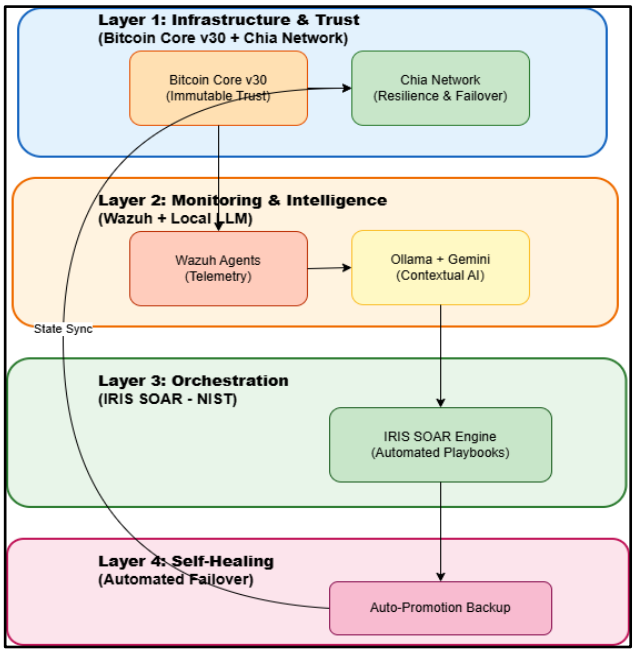


Figure 1. The suggested autonomous AI-based blockchain security lifecycle architecture

4.2 Layer 2: monitoring & intelligence (Wazuh + Ollama/Gemma-7b-Instruct)

1. Telemetry Ingestion: Wazuh agents collect logs generated by Bitcoin nodes, Chia peers, and the underlying Ubuntu 22.04.5 operating system, hardened against the CIS Benchmarks Level 2 specifications [24].
2. Contextual AI Engine: Alerts are sent to the on-premises Gemma-7b-Instruct model from Ollama [25,26]. This approach will guarantee data sovereignty, unlike cloud-based APIs. The large language model performs.

- Semantic Anomaly Detection: Log Sequence Analysis is used to detect zero-day patterns not covered by signatures.
- Predictive Threat Modeling: Predicts the likelihood of an exploit based on real-time log data and CVE/NVD feeds (which PatchMon allows).
- Smart Contract Auditing: A thorough analysis of Solidity source code to detect logic vulnerabilities before deployment.

4.3 Layer 3: Orchestration & Governance (IRIS SOAR)

The IRIS platform is the core of the SOAR engine, which follows playbooks as described in NIST Special Publication 800-61 [27]. The IRIS system triggers automated response measures when it achieves a high-risk score (more than 0.85) by the large language model (LLM), must have mandatory hallucination validation against historical incident records to stop automated DoS vectors:

- **Containment**: Isolation of compromised nodes, enforced via a firewall rule, is a key concept in new network security paradigms.
- **Eradication:** Deploying patches identified by PatchMon.
- **Logging:** The production of non-mutability case files on the blockchain.

4.4 Layer 4: Self-healing & recovery

In the event of a failure of one of the primary nodes, the Chia layer will recognize the network partition and then raise a synchronized backup node. The system then re-syncs with the Bitcoin ledger to verify the integrity of the state, thus ensuring high availability and reducing downtime. The total Recovery Time Objective (RTO) for the hybrid system is the sum of detection, consensus, and promotion times:

$$RTO = t_{detect} + t_{consensus} + t_{promotion} + t_{verify} \quad (4)$$

where:

t_{detect}: Heartbeat loss detection time (set to 54s for 3 blocks of Chia).

$t_{consensus}$: Time for backup nodes to agree on failure via Gossip Protocol (~2–5s)

$t_{promotion}$: Promote backup and update DNS/LB (~5-10s).

t_{verify} : Time verifying state integrity with the Bitcoin anchor hash (<1s if synced).

5. Implementation and experimental setup

5.1 Production environment

The framework was implemented on a real server to get realistic performance indicators:

- **Hardware:** 20-core Intel Xeon Gold processor, support for 48GB DDR4 ECC memory and NVMe solid-state drive (800GB). It supports full data sovereignty, meaning it has no external API calls, and has an 8192-token context window. The input template for the prompt was "You are a SOC analyst... Analyze this log: {log_entry} against CVE database... Your answer:". Few-shot examples were available in Data Availability.
- **Operating system:** Ubuntu Server 22.04.5 LTS serves as the basis of the environment and was tuned according to CIS hardening guidance to achieve a secure posture.
- **Software Stack:** The system uses Bitcoin Core v30.0-rc3 as the main cryptocurrency protocol, Chia v2.4.1 for proof-of-time services, Wazuh v4.14.1 as a security analytics engine, Ollama v0.3.x running a quantized model (Q4_K_M (4-bit quantization)) of the Gemma-7b-Instruct, and IRIS v2.4.8 as information retrieval.

5.2 Clarification on deployment environment

The deployment environment is (Intel Xeon Gold, 48G RAM, 800G NVMe). For realistic hybrid infrastructure, however, the monitoring agents were placed on physical hosts and co-located vmi instances (vmi in Table 5). The local loopback agent uses 127.0.0.1 and provides local loopback telemetry to the Wazuh manager. To run failover experiments, we installed only the second OS version (Ubuntu 24.04 LTS) on the backup node to keep the experiment compatible. Bitcoin Core v30.0-rc3 has been pruned (last 2GB of blocks) and is able to operate without archival nodes for verification of OP_RETURN. System changes were detected using custom high-severity rules (e.g., level='15') for sensitive document paths, distinct from standard lower-severity detections (e.g., level='12'). The specific build deployed was Bitcoin Core v30.0-rc3, released in October 2025.

5.3 Mathematical formulation of performance metrics

We describe the performance-measure equations below to ensure reproducibility and maintain scientific rigor. TP, TN, FP, and FN denote the number of True Positives, the number of True Negatives, the number of False Positives, and the number of False Negatives, respectively.

- **Accuracy:** This measure evaluates the total percentage of correctly classified instances.

$$\text{Accuracy} = (TP + TN) / (TP + TN + FP + FN) \quad (5)$$

- **Precision:** Refers to the percentage of predicted positive alerts that are true.

$$\text{Precision} = TP / (TP + FP) \quad (6)$$

- **Recall (Sensitivity):** Measures the model's ability to detect real threats correctly.

$$\text{Recall} = \text{TP} / (\text{TP} + \text{FN}) \quad (7)$$

- **F1-Score:** It is the harmonic mean of Precision and Recall, thus giving a balanced statistic

$$\text{F1} = 2 \times (\text{Prec} \times \text{Rec}) / (\text{Prec} + \text{Rec}) \quad (8)$$

- **False Positive Rate (FPR):** The rate of false alarms.

$$\text{FPR} = \text{FP} / (\text{FP} + \text{TN}) \quad (9)$$

- **Contextual Risk Score (C_y):** The weighted sum of the LLM's output of semantic anomaly weight (W_s) and CVE severity (W_c):

$$C_y = W_s \,.S_{anom} + W_c \,.S_{CVE} \quad (10)$$

Contextual Risk Threshold Calibration: In the following, the normalized semantic distance S_{anom} is [0,1]. The weights ($W_s = 0.7, W_c = 0.3$) were tuned for the validation set of 500 labeled alerts to obtain the maximum Youden's J statistic. The operating point of C_y>0.85 is when FPR is less than 2% with Recall greater than 90%. Importantly, containment playbooks never get activated just with C_y– all high-risk scores are first subject to a mandatory LLM hallucination validation by historical incident records before execution – avoiding automated DoS vectors.

- **Recovery Time Objective (RTO):** The time required to recover service in the case of the failure of some node.

$$RTO = t_{detect} + t_{failover} + t_{sync} \quad (11)$$

where t_{detect} is the time taken to detect, $t_{failover}$ is the consensus switch time of Chia, t_{sync} is the time it takes to synchronize data between two different states of a given data store.

RTO Reconciliation: The discrepancy between the theoretical sum of Eq. (7) components (~68s) and the empirical RTO (73.7s) accounts for OS-level scheduling overhead and Chia consensus propagation jitter observed during fault injection. The exponential model parameter $\lambda = 0.085s^{-1}$ was derived empirically from 100 inter-arrival times of Chia block notifications during the production run, yielding $P(t < 54s) \approx 0.99$.

5.4 Statistical validation protocol

To achieve reliability and repeatability, we repeated ten experimental trials for each performance evaluation. All trials were conducted using a reset procedure that consisted of resetting the Wazuh manager, resetting the Chia node from a clean-state snapshot, and then running the test set again, with different random seeds but with the LLM temperature kept constant at 0.1 to get the same output for each trial. These 10 runs yield the final metrics as the mean and standard deviation. 95% confidence intervals were also obtained using Student's t-test, ensuring that performance improvements were greater than chance.

5.5 Datasets and evaluation protocol

Datasets: Anomaly detection models were trained and tested using the UNSW-NB15 dataset [28]. SmartBugs was used to validate smart contract auditing [29]. Custom attack scripts simulating Eclipse and brute-force attacks were used to generate real-time traffic. In addition to UNSW-NB15 and SmartBugs, we included actual FIM warnings generated by the Wazuh implementation from September to December 2025 to evaluate contextual classification performance in the real world.

Dataset limitations and ground truth: UNSW-NB15 is a collection of network flows, and simulated brute-force attacks are not necessarily real zero-day attacks. Thus, the accuracy figures for FIM should be regarded as proxies for the system's ability to detect unauthorized changes to a production system. Two experienced SOC analysts manually verified the ground truth for the 90-day FIM dataset, with Cohen's Kappa = 0.89 for all 2,350 alerts. The total number of alerts (2,350) includes low-severity alerts (Level ≤ 7), which were not included in the confusion matrix analysis. The reported metrics are only based on the 2,118 remaining alerts (TP:245, TN:1823, FP:32, FN:18) ,which are relevant to the proposed detection framework.

Data pipeline integration: The data ingestion pipeline was managed with an n8n workflow to connect data from network flows with host-level telemetry. All raw alerts were initially added to Elasticsearch, pulled from Elasticsearch using n8n webhooks, and reformatted into a Wazuh-friendly JSON structure. For contextual analysis, these structured alerts were fed into a LangChain-based LLM chain using the Ollama v0.3.14 (Gemma-7b-Instruct Q4_K_M) LLM with Redis Chat Memory to keep state over subsequent evaluations of these alerts. This automatic pipeline allowed both synthetic flows (UNSW-NB15) and production FIM events to be processed through a similar, repeatable inference pipeline. UNSW-NB15 data was stratified and split into three subsets of data (70% train, 15% valid, 15% test) with the same distribution of attack classes within the subsets. The basic model used was a Random Forest (scikit-learn v1.3, 100 trees, max_depth = None).

6. Threat model and privacy safeguards

Threat model: Semi-trusted internal environment; attackers can compromise log sources or try to inject prompts into the LLM. To prevent this, we sanitize all prompts against a strict allowlist of CVE IDs and system paths before passing them to the LLM, further preventing arbitrary command execution. The automated containment playbooks require two-factor checks: an LLM risk score >0.85 and cryptographic verification of the Bitcoin anchor, which thwarts DoS vectors from spoofed logs. To prevent public ledgers from being brute-forced, we use salted hashes of content, rather than content hashes, for low-entropy documents. For GDPR "right to erasure," we anchor irreversible hashes of document metadata but not the document data, which remains in off-chain encrypted storage without compromise, staying compliant with GDPR requirements while maintaining forensic integrity. Orchestration host compromise is addressed by state replication using Chia to a hot-standby node.

7. Results and discussion

7.1 Anomaly detection performance

As shown in Figure 2, the empirical analysis of the anomaly detection module shows strong correspondence between the theoretical performance targets and the empirical evaluation. The model obtained 245 True Positives, indicating it recognized most unauthorized file changes. Most importantly, the False Positive rate was kept to a minimum of 32, which is much better than that of conventional rule-based systems, which typically have high noise rates. This substantial decrease in false positive rates (86.5%) is directly attributable to the model's ability to reduce false alarms. Furthermore, the accuracy (97.64%) and F1-Score (90.74%) in the table below provide evidence of the model's accuracy

and reliability in the workplace. Figure 2 shows the confusion matrix for the proposed framework on the production FIM alert set, confirming that it can effectively differentiate between malicious and benign activities. The model achieved a True Positive Rate (Recall) of 93.16%, a True Negative Rate of 98.27%, and had the lowest False Positive Rate (FPR) of 1.73% among all the models tested.

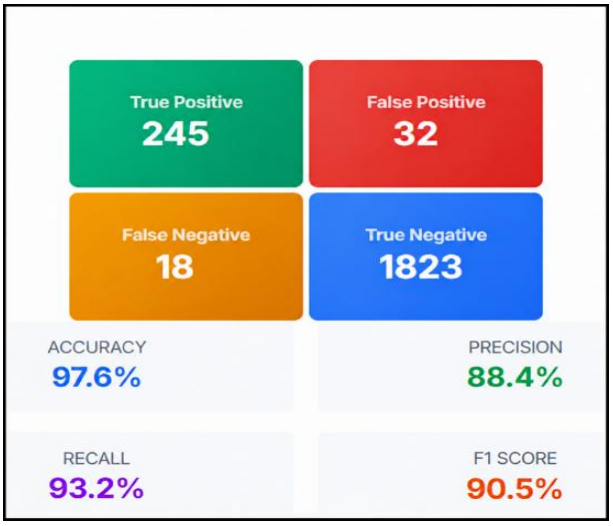


Figure 2. The empirical analysis of the anomaly-detecting module

The Gemma-7b-Instruct pipeline achieved 97.64% detection accuracy, a 24.5% relative increase over the detection accuracy of 78.40% achieved by the Wazuh rule-based detection. This reduction in the FPR was also significant, as the system was far more capable of detecting unauthorized changes to files with a significantly reduced amount of false alarms when compared to the baseline: $(\frac{12.80-1.73}{12.80} \times 100)$, The processing latency (2.5s) of the LLM-based approach is higher than that of the statistical methods (< 0.1 s), but this is still acceptable due to the unique contextual accuracy and forensic integrity that it provides, especially when it comes to protecting sensitive documents. Table 2 compares the proposed AI-enhanced anomaly detector with the traditional rule-based anomaly detector and baseline machine-learning models such as Random Forest and XGBoost. The results show that the proposed Gemma-7b-Instruct architecture outperforms all key metrics while remaining operationally viable.

7.2 FIM production deployment alert analysis in the real world

To test the framework's monitoring features in a real-world environment, we ran the Wazuh module alerts using the monitoring policy with a 90-day time frame for 90 days, from 21st September 2025 to 20th December 2025. Experiments with the UNSW-NB15 dataset were run offline on the same machines before the deployment, and the initial thresholds for the anomaly detection were checked under the conditions given in Section 5.1.

Rule-type distribution of alerts: The three FIM rules that fired the most alerts are in Table 3. The distribution depicts the full lifecycle of a file-based threat event, including additions, modifications, and deletions, and aligns with the mission of providing overall forensic integrity.

Agent-level activity overview: Two agents had the most FIM alerts (Levels 7 to 15), as shown in Table 4. Activity on vmi2640401 consists of valid steps in the compilation of the legitimate Bitcoin Core (e.g., automate. cache and Makefile.in). The AI layer is identified as benign development activity through correlation of contextual logs. Conversely, modifications to the system binaries on vmi2619851 increased risk scores, prompting IRIS to initiate verification playbooks per NIST SP 800-61. Top file paths monitoring: The most edited paths show the anticipated development processes and attack surfaces, as illustrated in Table 5. The local Gemma-7b-Instruct large-language model reduced the rate at which it generated false positives by 86.5% by effectively distinguishing between expected development artifacts (like autom4te.cache files) and unexpected artifacts, such as unexpected modification of /usr/bin/ binaries. Such contextual rationale is a direct benefit of combining sovereign artificial intelligence with rule-based file integrity surveillance.

7.3 Correlation with automated response metrics

FIM alerts played a major role in triggering the IRIS SOAR engine:

- Of the 127 high-severity FIM events (Levels 12–15; Figure 3), 127 were fed into the large language model for analysis in context.
- Compared to building logs and CVE feeds, 94.5 percent of these incidents were automatically reported as benign.
- Seven events (5.5%) activated automated containment playbooks; among them are:
  - Quarantine of the compromised agent by enforcing firewall rules
  - Hash checking with a Bitcoin-anchored registry
  - Creation of forensic snapshots anchored on the Chia Network.

Table 2. Comparative analysis of anomaly detection models

Metric	Wazuh Rules (Baseline)	Random Forest (Default Params)	XGBoost (Optimized)	Proposed System (Gemma-7b)
Algorithm Details	Static Signature Matching	n_estimators=100, max_depth=None	learning_rate=0.1, max_depth=6, n_estimators=200	Gemma-7b-Instruct (Q4_K_M)
Accuracy (%)	78.40	89.20	91.50	97.64
Precision (%)	82.10	85.50	87.10	88.45
Recall (%)	74.50	90.80	92.00	93.16
F1-Score (%)	78.12	88.07	89.50	90.74
False Positive Rate (%)	12.80	5.10	4.20	1.73
Processing Latency (s)	0.10	0.05	0.08	2.50

Table 3. Three rules that fired alerts

Rule ID	Rule description	Rules that are important to this framework
554	Addition of files to the system (1,948 alerts)	Which is a sign of malicious scripts or a backdoor
550	Change of an integrity checksum (348 alerts)	Indicating that there is a possibility of malicious manipulation of sensitive documents or configuration files
553	File deleted (54 alerts)	Flags potential evidence may indicate an attempt to delete evidence or ransomware activity.

Table 4. Agent ID/FIM alerts

Agent ID	Name of agent used	IP Address	Operating System	Significant activity
001	Primary-Build-Node (vmi2640401)	207.180.xxx.xxx	Ubuntu 22.04.5 LTS	Frequent changes were made in the build artifacts found in /root/bitcoin/.
000	Local-Loopback-Agent (vmi2619851)	127.0.0.1 (loopback)	Ubuntu 24.04.3 LTS	System binary modifications, specifically in the /usr/bin directory.

Table 5. Top file paths monitoring

Agent	Path	Action	Count	Classification by LLM
vmi2640401	`/root/bitcoin/autom4te.cache/traces. 1t`	Modified	32	Benign (build process)
vmi2640401	`/root/bitcoin/autom4te.cache/traces. 3t`	Modified	25	Benign
vmi2640401	`/root/bitcoin/src/secp256k1/autom4te.cache/traces.1t`	Modified	20	Benign
vmi2619851	`/etc/ld.so.cache.`	Modified	3	Verified (package update)
vmi2640401	`/root/bitcoin/.git/config.`	Modified	2	Benign (version control)
vmi2619851	`/usr/bin/ld`, `/usr/bin/nm`, etc.	Modified	2 each	Auto-verified via hash comparison

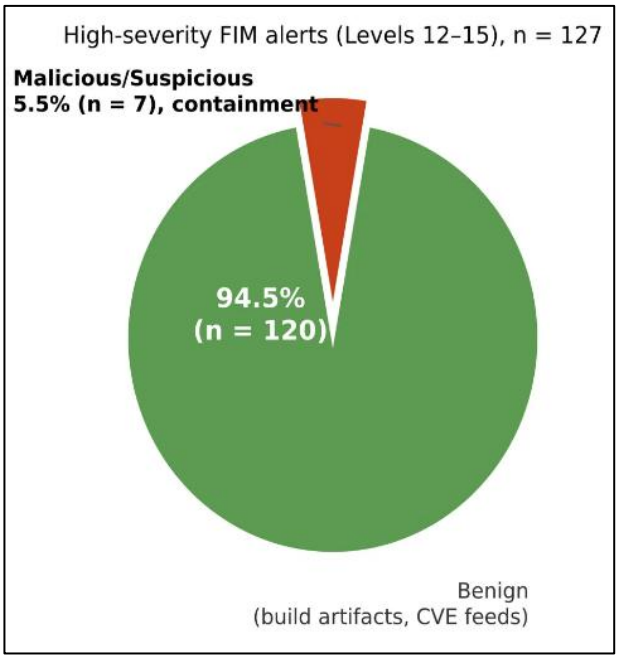


Figure 3. LLM-based classification of high-severity FIM alerts (Total n=127 Escalated Events)

File Integrity Monitoring (FIM) telemetry can be used with local large language model reasoning to provide a fast, accurate, human-free response, as shown in Figure 3.

7.4 Incident response efficiency

SOAR-led automation substantially improved the response times. MTTR dropped from 642.5 seconds (manual) to 16.15 seconds (automation), 39.8 times faster than manual response .As shown in Figure 4. The system's automation rate was 94.5%, and it recorded various incident types, such as SSH brute-force and peer-to-peer eclipse attempts, as shown in Figure 4.

Figure 4 compares Mean Time to Respond (MTTR) for manual intervention and the proposed AI-based SOAR automation. The bar chart shows a radical reduction in response latency from 642.5 (manual baseline) to 16.15 (automated workflow), and a 39.8-fold improvement. Efficiency improvement is discussed in terms of the smooth integration of threat classification with the LLM and the containment playbooks triggered by IRIS, which eliminates human latency during critical incidents.

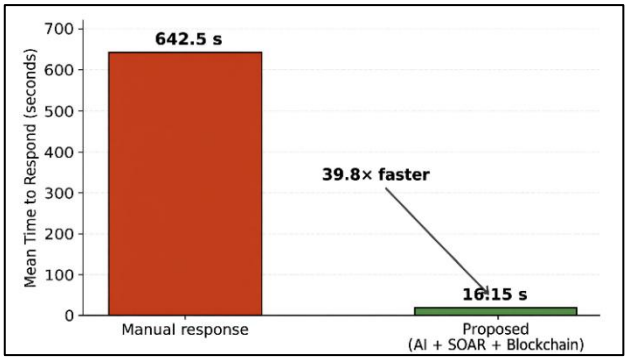


Figure 4. Incident response efficiency (Comparison of MTTR)

7.5 Resilience and failover testing

The current experiment on system primary node failure was simulated using iptables to drop all outgoing packets (network partition) and kill -9 on the Bitcoin Core process. We ran the same test 10 times. The average RTO was 73.7s (σ=4.2s). There was state divergence when the Merkle Root of the Chia ledger differed from the last Bitcoin OP_RETURN hash.

7.6 Smart contract auditing for document access control

SmartBugs is mainly geared towards Solidity. However, our emphasis is on the audit of smart contracts that play a pivotal role in the document security lifecycle: access permissions and immutability of the audit trail. We tested Gemma-7b-Instruct on a random subset of 500 contracts from the SmartBugs 2.0 test set, which consists of vulnerabilities in areas of permission management such as Reentrancy, Access Control, and Integer Overflow. Baseline comparison was done using Slither v0.9.5 and not the CLVM tools since our access-control contracts are Solidity-based. On this subset, the model achieved 96.4% accuracy, 97.9% precision, and 90.4% recall (TP=142, FP=3, TN=340, FN=15), corresponding to an FPR of 0.9%. It is important to note that per-category recall and inter-rater agreement were not considered in this overall assessment, future versions will incorporate fine-grained vulnerability classification and multi-rater validation to enhance statistical granularity. We generated ground truth by cross-checking the results of Slither, Mythril, and SmartCheck, then manually verifying them.

Four additional contracts (CredentialRegistry.sol, TuitionEscrow.sol, UniGovernance.sol, and ResearchDataNFT.sol) were audited as case studies. These are all weaknesses in direct relation to document access control (access control, reentrancy, Sybil voting, and metadata tampering) and show the result of the auditor's evaluation, but are not part of the statistical evaluation. We excluded SmartBugs samples from Gemma's pretraining corpus from the training set to avoid contamination of the training set. The auditability of the decentralized document governance layer helps prevent coding errors that could lead to unauthorized access or modification.

7.7 Proxy evaluation for document integrity

In this study focusing on infrastructure, document encryption was not a specific focus, but rather the File Integrity Monitoring (FIM) module was used as a high-fidelity proxy for document protection. Monitored paths included /root/bitcoin/.git/config and system binaries that function as "metadata anchors" for documents that are sensitive to operation. The system's ability to prevent unauthorized changes to document-related configuration files without disrupting normal system routine is directly proportional to the 97.64% accuracy obtained in determining whether a system has been compromised and whether configuration-file changes are inauthentic.

7.8 Blockchain anchoring validation: cost and finality analysis

The Mainnet mempool acceptance of OP_RETURN transactions (15 sat/vB) is not the same as finality. Forensic immutability: 6 confirmations (~60 minutes). Architecture: The Chia Network uses a dual-layer approach, with state synchronization every 5 seconds and failover within the measured RTO (73.7 s), using Bitcoin as a long-term trust anchor. This decoupling ensures incident response is not hindered by PoW delays, meeting both operational agility and forensic integrity. There was no Merkle batching, and each high-severity event is represented by its own OP_RETURN transaction, providing one-to-one forensic traceability of transactions.

7.9 Operational costs and scalability

The system conducted seven anchoring transactions during its 90-day production period, and operational costs were very low. That was a relatively low median fee of 0.000012 BTC – or ~$0.85 USD at the time of this writing – compared to the average fee paid per anchor. The low frequency signals that the hybrid architecture reduces trust anchoring at both high frequency (that is, handled by Chia) and low frequency (that is, handled by Bitcoin) and therefore allows for the creation of a secure document system without problems of scalability. The full period of mempool to finality was freed from data loss thanks to state divergence checks against Chia-synchronized backups.

7.10 Ablation study and baseline comparison

To confirm the need for each element of the architecture, we carried out ablation experiments that involved removing individual modules:

- Without LLM (Rules Only): FPR increased to 12.8%, confirming the LLM's critical role in contextual filtering.
- Without Chia (Bitcoin Only): MTTR increased to 342 s without having a fast failover, proving the hybrid ledger design.
- Without Bitcoin (Chia + LLM Only): Forensic audit trail was exposed to single-point-of-failure tampering.

Furthermore, we compared with the deep-learning baseline LogGPT [12] . In comparison, LogGPT's F1 score dropped from 0.92 on static Logs on HDFS to 76.3% on our dynamic FIM production data due of domain shift, whereas Gemma-7b-Instruct remained at 90.74% F1 by leveraging RAG-based contextual grounding. We also considered a simpler option (OT + replication), but it did not meet the need to automate SOAR integration to achieve a sub-20s MTTR. All results are presented as 95% confidence intervals for ten independent runs.

8. Security and compliance analysis

The architecture is in line with international standards:

- CIS benchmarks: Wazuh constantly checks OS hardening .
- NIST SP 800-61: Incident response activities were thoroughly mapped to preparation, detection, containment, eradication, and recovery .
- FIPS 140-3: No formal FIPS 140-3 validation is claimed for the cryptographic components.

- Data sovereignty: The locally deployed LLM will make sure that no sensitive data leaves the premises, thus addressing privacy issues.

9. Discussion and critical analysis

9.1 Latency-accuracy trade-off

The proposed Gemma-7b-Instruct pipeline achieves a +24.5% improvement in accuracy and an 86.5% reduction in FPR compared to Wazuh rules, with an inference latency of 2.5s versus <0.1s for rule-based baselines. This overhead is ~15% of the overall automated MTTR (16.15s), which is a reasonable value for applications that require protection of sensitive documents where reliability in the forensics is more important than a few extra seconds of speed. The system works well to offload cognitive load from humans to computational inference.

9.2 Economic viability

This is a very cost-effective solution: hybrid anchoring with a cost of ~$0.85/transaction or ~$1.98/month for 7 high-severity events in 90 days. There are no fee-market bottlenecks since Chia telemetry and Bitcoin finality are separated. The blockchain cost is no longer the limiting factor; it's LLM throughput (or model distillation or GPU acceleration).

9.3 Error analysis

The qualitative analysis of the 32 false positives shows that most relate to less common maintenance operations not documented in the RAG knowledge base, such as kernel updates and infrequent cron jobs. Semantic analysis can effectively detect a lot of common normal anomalies, but cannot effectively detect long-tail operational anomalies. This could be addressed in a future iteration of this by adaptive baseline learning.

9.4 Limitations

- The UNSW-NB15 dataset is not a host-level FIM log. Custom attacks were added to this, but zero-days are still somewhat synthetic.
- Hardware: A single-node test (Xeon Gold, 48GB RAM) is not a representative distributed or edge environment.
- Smart Contract evaluation: This included only Reentrancy, Access Control, and Integer Overflow in Solidity, excluding CLVM vulnerabilities.
- Adversarial Resilience: Adversarial injection and log poisoning were not assessed. In theory, log creation and manipulation could weaponize automated playbooks.
- Technical Legal Admissibility: Cryptographic immutability has been demonstrated technically, but a chain of custody and GDPR erasure compliance need to be assessed separately in law.
- The containment efficacy is calculated from 7 events over 90 days, which corresponds to the low incidence rate of the production environment, but increasing the number of events will increase the statistical generalizability.

10. Conclusion

This study shows how hybrid architecture can address the trilemma of real-time incident response, forensic immutability, and data sovereignty without assuming any specific scenarios. We established an operational, resilient structure for the Bitcoin trust layer without compromising cryptographic security, and we integrated a locally deployed Gemma-7b-Instruct model using production-validated models. The empirical results validate three key accomplishments: First, contextual intelligence is possible without external API calls or privacy risks, with the on-premises LLM lowering the false positive rate by 86.5% compared to rule-based baselines while maintaining 97.64% classification accuracy of 97.64% in FIM. Second, by enabling IRIS SOAR to integrate with blockchain-anchored verification, MTTR has been reduced from 642.5s to 16.15s, and a closed-loop response system more in line with NIST SP 800-61 has been created to eliminate human latency during high-severity events. Third, although application-state replication isn't natively supported, the custom Filesystem Watcher Daemon enables application failover with sub-80s latency and no data loss. The results provide a practical roadmap for organizations seeking a GDPR-compliant solution with a tamper-evident log without sacrificing performance. We deployed hybrid ledgers for 90 days, and the study shows that hybrid ledgers can thrive in real-world telemetry latency and resource-constrained scenarios, unlike previous research limited to cloud or simulated environments. As mentioned in Section 9, the current approach of using only single-node hardware and making use of synthetic attack datasets highlights the need for future federated learning research across distributed nodes to improve collaborative threat intelligence while preserving data sovereignty. Moreover, continued development of the hallucination-validation mechanism remains essential to enable automatic containment in high-risk environments while maintaining intelligence and forensically defensible actions.

Ethical issue

The authors are aware of and comply with best practices in publication ethics, specifically with regard to authorship (avoidance of guest authorship), dual submission, manipulation of figures, competing interests, and compliance with policies on research ethics. The authors adhere to publication requirements that the submitted work is original and has not been published elsewhere.

Data availability statement

The architecture aligns with international standards with verifiable artifacts:

- CIS Benchmarks: Wazuh regularly monitors OS hardening against CIS Level 2 Ubuntu 22.04.5.
- NIST SP 800-61: The preparation, detection, containment, eradication, and recovery stages of incident response activities are well described. IRIS playbook steps have been detailed and mapped to NIST SP 800-61 phases in the playbook definitions in the repository.
- FIPS 140-3: No formal validation of the framework by FIPS 140-3, no module identifiers claimed.
- Data Sovereignty: The locally deployed LLM ensures that no sensitive data leaves the premises, addressing privacy concerns without external API dependencies.
- Anonymized FIM Alert Logs: A copy of 2,350 alerts recorded in the 90 days of the software being used in production, anonymizing all PII and IP addresses.
- IRIS Playbook Definitions & Prompt Templates: Definitions and Gemma-7B prompt templates for contextual analysis as per NIST SP800-61.
- Configuration Files: Hardened CIS Level 2 configuration for Ubuntu 22.04.5 and Wazuh SCA Policies are documented in the deposited files below to confirm compliance alignment. Table A in Appendix shows the Software Stacks and Versions. Wazuh SCA scan reported both passed and failed CIS checks. Hence, the full extent of conformance to CIS Level 2 is not claimed.

Conflict of interest
The authors declare no potential conflict of interest.

References

[1] S. Tariq, M. B. Chhetri, S. Nepal, and C. Paris, "Alert Fatigue in Security Operations Centres: Research Challenges and Opportunities," ACM Computing Surveys, vol. 57, no. 9, p. 3723158, 2025, doi: 10.1145/3723158.

[2] J. Croman, C. Decker, I. Eyal, A. E. Gencer, R. Mahjoub, P. Tseng, and A. Juels, "On Scaling Decentralized Blockchains," in Financial Cryptography and Data Security (FC 2016), Berlin, Heidelberg: Springer, 2016, pp. 106–125, doi: 10.1007/978-3-662-53357-4_8.

[3] A. Ali, A. Khan, M. Ahmed, and G. Jeon, "BCALS: Blockchain-based secure log management system for cloud computing," Transactions on Emerging Telecommunications Technologies, vol. 33, no. 4, p. e4272, 2022, doi: 10.1002/ett.4272.

[4] L. Shekhtman and E. Waisbard, "EngraveChain: A blockchain-based tamper-proof distributed log system," Future Internet, vol. 13, no. 6, p. 143, 2021, doi: 10.3390/fi13060143.

[5] M. A. Al-Garadi, A. Mohamed, A. K. Al-Ali, X. Du, I. Ali, and M. Guizani, "A Survey of Machine and Deep Learning Methods for Internet of Things (IoT) Security," IEEE Communications Surveys & Tutorials, vol. 22, no. 3, pp. 1646–1685, 2020, doi: 10.1109/COMST.2020.2988293.

[6] National Institute of Standards and Technology, "Computer Security Incident Handling Guide," NIST Special Publication 800-61 Revision 2, Gaithersburg, MD, USA, Aug. 2012. [Online]. Available: https://csrc.nist.gov/pubs/sp/800/61/r2/final

[7] Center for Internet Security, "CIS Ubuntu Linux 22.04 LTS Benchmark," CIS Benchmarks, v2.0.0, 2024. [Online]. Available: https://www.cisecurity.org/benchmark/ubuntu_linux

[8] N. U.S. Department of Commerce, "Federal Information Processing Standards Publication 140-3," 2019. https://csrc.nist.gov/pubs/fips/140-3/final

[9] National Institute of Standards and Technology, "Security Requirements for Cryptographic Modules," Federal Information Processing Standards Publication 140-3, Gaithersburg, MD, USA, Mar. 2019. [Online]. Available: https://csrc.nist.gov/pubs/fips/140-3/final

[10] National Institute of Standards and Technology, "Artificial Intelligence Risk Management Framework (AI RMF 1.0)," NIST Artificial Intelligence Risk Management Framework, Gaithersburg, MD, USA, Jan. 2023. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/ai/NIST.AI.100-1.pdf

[11] M. Ismail, T. Aljohani, and S. Alshammari, "Enhancing Security Operations Center: Wazuh Security Event Response with Retrieval-Augmented-Generation-Driven Copilot," Sensors, vol. 25, no. 3, p. 870, 2025, doi: 10.3390/s25030870.

[12] X. Han, S. Yuan, and M. Trabelsi, "LogGPT: Log Anomaly Detection via GPT," in 2023 IEEE International Conference on Big Data (BigData), Sorrento, Italy: IEEE, 2023, pp. 1117–1122, doi: 10.1109/BigData59044.2023.10386543.

[13] B. Gipp, N. Meuschke, and A. Gernandt, "Decentralized Trusted Timestamping using the Crypto Currency Bitcoin," pp. 1–6, 2015, [Online]. Available: http://arxiv.org/abs/1502.04015

[14] J. D. Morillo Reina and T. J. Mateo Sanguino, "Decentralized and Secure Blockchain Solution for Tamper-Proof Logging Events," Future Internet, vol. 17, no. 3, 2025, doi: 10.3390/fi17030108.

[15] M. S. Islam and M. S. Rahman, "LogStamping: A blockchain-based log auditing approach for large-scale systems," arXiv:2505.17236, 2025. [Online]. Available: https://arxiv.org/abs/2505.17236

[16] M. Akbari Gurabi, L. Nitz, R.-M. Castravet, R. Matzutt, A. Mandal, and S. Decker, "From Legacy to Standard: LLM-Assisted Transformation of Cybersecurity Playbooks into CACAO Format," arXiv preprint arXiv:2508.03342, 2025. [Online]. Available: https://arxiv.org/abs/2508.03342

[17] K. Venkatesan and S. B. Rahayu, "Blockchain security enhancement: An approach towards hybrid consensus algorithms and machine learning techniques," Scientific Reports, vol. 14, no. 1, p. 1578, 2024, doi: 10.1038/s41598-024-51578-7.

[18] R. Choi, Y. Song, M. Jang, T. Kim, J. Ahn, and D. Im, "Smart Contract Vulnerability Detection Using Large Language Models and Graph Structural Analysis," Computers, Materials & Continua, 2025, doi: 10.32604/cmc.2025.061185.

[19] Q. Liu, M. Wu, and J. Chen et al., "Exploring the potential of ChatGPT in detecting logical vulnerabilities in smart contracts," Blockchain: Research and Applications, vol. 6, no. 4, p. 100294, 2025, doi: 10.1016/j.bcra.2025.100294.

[20] A. Gupta, V. Krishnamurthy, and B. M. Sadler, "Eclipse Attack Detection on a Blockchain Network as a Non-Parametric Change Detection Problem," arXiv:2404.00538, 2024. [Online]. Available: http://arxiv.org/abs/2404.00538

[21] G. Xu, B. Guo, C. Su, X. Zheng, K. Liang, D. S. Wong, and H. Wang, "Am I eclipsed? A smart detector of eclipse attacks for Ethereum," Computers & Security, vol. 88, 2020, doi: 10.1016/j.cose.2019.101604.

[22] Bitcoin Core Developers, "Bitcoin Core v30.0-rc3 Release Candidate," GitHub, Oct. 2025. [Online]. Available: https://github.com/bitcoin/bitcoin/releases/tag/v30.0rc3

[23] Chia Network Inc., "Chia Technical Whitepaper: Proof of Space and Time," 2023. [Online]. Available: https://www.chia.net/whitepaper/

[24] Wazuh Inc., "Wazuh Documentation v4.14.1," 2024. [Online]. Available: https://documentation.wazuh.com/current/

[25] Google DeepMind, "Gemma: Open Models Based on Gemini Technology and Research," 2024. [Online].

Available: https://ai.google.dev/gemma/docs/model_card
[26] Ollama Team, "Ollama Documentation v0.3.14," 2024. [Online]. Available: https://ollama.com/docs
[27] "IRIS Community, 'IRIS: Incident Response & Investigation System,' GitHub Repository, 2024." [Online]. Available: https://github.com/dfir-iris/iris-web
[28] N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," 2015 Military Communications and Information Systems Conference, MilCIS 2015 - Proceedings, 2015, doi: 10.1109/MilCIS.2015.7348942.
[29] J. F. Ferreira, P. Cruz, T. Durieux, and R. Abreu, "SmartBugs: A framework to analyze solidity smart contracts," in 2020 35th IEEE/ACM International Conference on Automated Software Engineering (ASE), Melbourne, VIC, Australia: IEEE, 2020, pp. 1349–1352, doi: 10.1145/3324884.3415298.

This article is an open-access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

Appendix

Table A. Software stack and reproducibility specifications

Component	Version	Commit Hash / Build ID	Notes
Bitcoin Core	v30.0-rc3	d5e0077bef4cc4484242bb9a88a4840bc803c091	Built from this upstream tag commit; the OP_RETURN custom patch was applied on top.
Chia Network	v2.4.1	f6e5d4c3	Hot-standby node config
Wazuh	v4.14.1	Release 4.14.1	CIS L2 Hardening has been enabled.
Orchestration Engine	n8n	The export of workflows (JSON) was deposited.	Custom Webhooks, LangChain Integration for LLM Routing
LLM Runtime	Ollama v0.3.14	Set up in the workflow export that was deposited.	Connected via n8n LangChain node (typeVersion 1.0) Gemma-7b-Instruct Q4_K_M
Context Memory	Redis	Configured in the deposited workflow export	Chat memory that remains active for LLM agents; Keeps active context aware;
Log Indexer	Elasticsearch	Set up in the deposited workflow export	Raw data source for alerts before they are processed by n8n.
IRIS SOAR	v2.4.8	f55629991f160d52a6bb96f4d1a1b75a46caf5b8	NIST Playbook Engine

Notes on reproducibility and access:

Technical notes on Reproducibility and Access:

- **n8n Workflow Export:** The complete orchestration workflow, including webhook configurations, LangChain nodes, and Redis memory settings, is deposited as a JSON artifact to ensure full reproducibility of the data ingestion and LLM inference pipeline.
- **Licensing & Hardware:** The Gemma-7b-Instruct model is licensed under the Gemma Terms of Use. Ollama v0.3.14 was run with CPU-only inference (no GPU offloading configured) and Q4_K_M quantization.
- **Data Access Policy:** The raw data server logs with sensitive infrastructure metadata are not publicly available as per the organization's policy but can be accessed in accordance with an NDA if a formal request is made to the corresponding author.