International Journal of Progressive Sciences and Technologies (IJPSAT)
ISSN: 2509-0119.

https://ijpsat.org/

Vol. 58 No. 1 July 2026, pp. 69-83

Simulating Resilience In Blockchain-Backed Log Management For Digital Cultural Heritage Against Insider Threats

Arga Husein Passu Beta[1], Bisyron Wahyudi[1], H.A. Danang Rimbawa[1]

[1]Republic of Indonesia Defense University
Indonesia

Corresponding Author : Arga Husein Passu Beta, argahussein98@gmail.com

Abstract– Securing cultural heritage digital archives is critical, yet centralized log management remains vulnerable to administrative manipulation. To address the lack of empirical resilience against privileged insider threats, this study evaluates a Zero Trust blockchain architecture for preserving artifact provenance. A discrete-event simulation (SimPy) modeled an "Ex_Curator" attack injecting 10,000 unauthorized requests per hour against an ECC/SHA-256 validated ledger. The architecture successfully intercepted all anomalies while maintaining a flat queue and a stable 0.05-second processing latency. Hardware efficiency was exceptional, recording only a +1.05% CPU overhead and a 544 KB disk footprint. This lightweight framework ensures absolute non-repudiation for digital twins, providing a practical security solution for resource-constrained museums.

Keywords: Blockchain Logging, Insider Threat, Cultural Heritage, Discrete-Event Simulation.

I. INTRODUCTION

Log management is a fundamental component of modern information technology infrastructure, providing the necessary records for auditing, troubleshooting, and ensuring organizational accountability. Effective log management ensures that computer security records are stored in sufficient detail to identify policy violations and operational anomalies shortly after they occur [1]. However, traditional centralized logging architectures frequently suffer from a single point of failure and a significant lack of transparency, making them vulnerable to data tampering by privileged actors [2]. As cyber threats evolve, organizations face increasing challenges in maintaining the integrity of these records, particularly in large-scale environments where log data volumes are overwhelming [3].

The most pervasive risk in current cybersecurity landscapes originates from within the insider threat. While digitalization has become an inescapable step in the process of preserving and enhancing Cultural Heritage, cultural institutions face unique vulnerabilities in managing this transition. Managing institutions frequently suffer from limitations of commonly used technological means and a lack of specialized knowledge and competence [4]. To maintain sustainable preservation strategies in contexts where resources and skills are strictly limited [5], museums ofsten rely heavily on temporary staff, volunteers, and third-party vendors for digitalization projects. This high personnel turnover often results in inadequate de-provisioning processes, where the access credentials of former personnel such as a terminated curator (Ex_Curator) remain dormant yet active for months after their departure. Such individuals can exploit these dormant credentials to manipulate or delete logs, thereby erasing the audit trail of the museum's digital twins and obstructing forensic investigations [5]. To mitigate these risks, implementing a Zero Trust Architecture (ZTA) is highly logical for cultural institutions. ZTA operates on the principle of "never trust, always verify," ensuring that even if dormant access is exploited, the lack of continuous cryptographic authentication will prevent attackers from causing irreparable damage to digital heritage assets.

To mitigate these risks, blockchain technology has emerged as a promising solution due to its decentralized, immutable, and tamper-resistant nature [6, 7]. By creating a "decentralized brain" for security logging, blockchain ensures that once a log entry is recorded, it cannot be altered without consensus, providing a permanent and verifiable historical record [6, 8, 9]. Integsdration of Zero Trust Architecture (ZTA) principles where trust is never assumed and every identity is continuously verified further strengthens this defense [10, 11]. In this context, the use of Elliptic Curve Cryptography (ECC) provides a high level of security with shorter key lengths, ensuring efficient payload management and preventing identity spoofing [12, 13].

Moreover, the necessity for robust log management is not only a technical requirement but also a regulatory mandate. International standards like ISO/IEC 27001 require the production, protection, and analysis of activity logs to safeguard against unauthorized access [14]. Locally, Indonesian regulations, including BSSN Regulation No. 4 of 2021 and Law No. 27 of 2022 (UU PDP), emphasize the accountability of data controllers in ensuring the integrity and confidentiality of personal data [15, 16].

Despite the theoretical benefits of blockchain-backed systems, there is a lack of empirical performance data under high-stress internal attack conditions. This research addresses this gap by utilizing Discrete-Event Simulation (DES) via the SimPy framework to conduct a mathematical stress-test on a blockchain-backed log management system [17]. By modeling time-varying human behavior and simulating a coordinated identity spoofing attack from an Ex_Curator actor, this study evaluates system resilience in terms of queue dynamics, processing latency, and data integrity. The findings aim to demonstrate the feasibility of cross-cloud, highly available digital archive logging that satisfies both technical excellence and regulatory compliance [18, 19]. The findings aim to demonstrate the feasibility of cross-cloud, highly available clinical research activity logging that satisfies both technical excellence and regulatory compliance [9, 20].

II. THEORITICAL FRAMEWORK

The evaluation of system resilience against internal adversaries requires a multi-dimensional theoretical foundation that bridges decentralized data integrity mechanisms with human behavioral dynamics. Traditional security models, which often rely on centralized log management, are increasingly insufficient in the face of sophisticated insider threats that exploit legitimate access to manipulate or erase evidence. This chapter establishes the theoretical framework underpinning the proposed research, integrating blockchain technology, zero-trust principles, and discrete-event simulation to address these vulnerabilities. First, the role of blockchain as a decentralized and immutable audit trail is explored to ensure the integrity of security records. Second, the framework examines the dynamic modeling of insider behavior to quantify the impact of unintentional and malicious actions on overall system health. Furthermore, the study incorporates international and local regulatory standards for data protection to ensure that the proposed technological solution aligns with legal mandates for accountability. Finally, the principles of discrete-event simulation are presented as a robust methodology for stress-testing these interconnected components under realistic attack scenarios.

2.1 Blockchain as an Immutable Audit Trail

Blockchain is defined as a decentralized digital ledger that records transactions across a distributed network of nodes in a transparent and tamper-resistant manner [7]. Originally designed to facilitate trustless financial interactions, its core features immutability, distributed consensus, and traceability make it an ideal candidate for securing computer security logs [7, 21]. Traditional centralized logging systems often lack the necessary transparency, as privileged insiders can modify or erase records to hide malicious activities [2, 11]. In contrast, a blockchain-based approach decentralizes the validation of security events, eliminating the reliance on a single authority and ensuring that log data cannot be altered or suppressed by internal adversaries [21, 22].

The architectural strength of blockchain lies in its use of cryptographic hash functions, such as SHA-256, to link blocks into a continuous chain [23, 24]. Each block contains a hash of the previous block's header, a timestamp, and transaction data; therefore, any change made to an individual block invalidates all subsequent blocks [7, 23]. This mechanism creates a self-protection layer for raw sensing data, where the older the data is, the higher the computational barrier for an attacker to successfully modify the record without detection [9, 23]. Furthermore, the integration of smart contracts enables automated rule enforcement, ensuring that only verified, blockchain-backed evidence is stored and reported in real-time [11, 19].

To address the scalability challenges associated with large-scale log generation, hybrid architectures have been proposed where only cryptographic fingerprints or metadata are stored on-chain, while the actual log files are outsourced to secure off-chain storage such as the InterPlanetary File System (IPFS) [3, 13, 25]. This dual-layered approach preserves the data immutability of the blockchain while managing high-frequency traffic typical in distributed data centers [3, 8]. By utilizing Elliptic Curve Cryptography (ECC) and digital signatures, blockchain-backed frameworks achieve non-repudiati

on, ensuring that every privileged action performed by a user is cryptographically linked to their identity and cannot be disputed [10, 12]. Thus, blockchain functions as a robust "decentralized brain" that adaptively protects the audit trail against increasingly sophisticated insider threats [6, 19].

2.2 Insider Threat Dynamic and Behavioral Modeling

Insider threats represent one of the most complex challenges in modern cybersecurity because the malicious activities are performed by authorized users who already possess the required permissions to access critical systems [26]. Unlike external adversaries, insiders are familiar with an organization's internal policies, systems, and security procedures, allowing them to operate within normal behavioral boundaries and making their actions difficult to distinguish from legitimate tasks [11]. These actors can include current or former employees, contractors, or business partners who exploit their privileges intentionally for sabotage, fraud, or espionage, or unintentionally through negligence [10, 26].

Behavioral modeling of these threats requires understanding the complex nature of human actions within an organizational context [18]. Unintentional insider threats can be quantified by representing time-varying human behavior through two primary parameters: user vulnerability and security leakage due to interactions such as credential sharing [18]. User vulnerability is often a result of cognitive limitations, including lack of awareness or workplace stress, which can be modeled as discrete-time stochastic processes to determine their impact on overall system health [18]. This dynamic nature suggests that static, perimeter-based security models are no longer sufficient to protect sensitive data in distributed environments [10].

Furthermore, the dynamics of malicious insiders often involve privileged users, such as system administrators or chief curators, who can alter or erase audit logs to conceal their tracks [22]. For instance, a terminated employee might exploit dormant administrative access to destroy critical documents or exfiltrate sensitive provenance records and digital artifact data [10, 22]. Traditional rule-based or signature-based detection methods often fail in these scenarios because anomalies are often subtle and context-dependent [26]. Consequently, there is a critical need for intelligent detection mechanisms that utilize user behavior analytics and real-time log monitoring to identify deviations from established normal profiles, such as unusual login timings, unauthorized resource access, and privilege escalation patterns [11, 26].

2.3 Compliance and Regulatory Standards for Data Integrity

The implementation of robust log management is not only a technical necessity but a fundamental requirement for legal and regulatory compliance across various jurisdictions. According to the National Institute of Standards and Technology (NIST), organizations must store and analyze certain logs to comply with specific legislation and regulations, ensuring that security records are maintained in sufficient detail for an appropriate duration [1]. This is further reinforced by the ISO/IEC 27001:2022 standard, which specifies that logs recording activities, exceptions, and security events must be produced, protected, and analyzed to safeguard the Information Security Management System (ISMS) [14].

In the global context, tamper-resistant log files are mandated by stringent frameworks such as the Health Insurance Portability and Accountability Act (HIPAA) and the General Data Protection Regulation (GDPR) [3, 22]. These regulations emphasize that log integrity is critical because it serves as indispensable evidence for detecting security incidents and conducting forensic investigations [3]. Furthermore, standards such as ISO/IEC 27002 recommend that logs be protected against tampering and unauthorized access to maintain their reliability as digital evidence [12].

Within the Indonesian legal framework, the necessity for data integrity and accountability is explicitly stated in Law (UU) Number 27 of 2022 concerning Personal Data Protection (UU PDP). This law mandates that personal data controllers must protect

and ensure the security of the data they process by implementing reliable electronic systems that prevent unauthorized access [16]. Additionally, the National Cyber and Crypto Agency (BSSN) through Regulation Number 4 of 2021 provides technical guidelines for the security of Electronic-Based Government Systems (SPBE), requiring that data and information fulfill the aspects of confidentiality, integrity, authenticity, and non-repudiation [15]. A blockchain-backed infrastructure inherently supports these requirements by providing an immutable audit trail and non-repudiable proofs of existence for all generated log records [8, 12].

2.4 Discrete-Event Simulation (DES) and Performing Modeling

Discrete-event simulation (DES) is a sophisticated methodology used to model the behavior of complex systems as a discrete sequence of events in time, where each event occurs at a particular instant and marks a change of state in the system [17]. Within the domain of cybersecurity, DES is particularly valuable for studying the impact of unintentional insider threats by representing time-varying human behavior and measuring the effects of such behavior on overall system health [18]. By modeling interactions as discrete-time stochastic processes, researchers can simulate potential attack paths and identify the "Achilles heel" of robust security infrastructures [18].

Performance modeling utilizing tools such as SimPy enables the precise replication of time-based blockchain events, including transaction generation, consensus validation, and block creation [17]. This approach allows for an objective comparison of critical metrics such as throughput, latency, and energy consumption under identical conditions without the need for actual network deployments [17]. In large-scale systems, simulation is essential to ensure that log management frameworks can handle high-volume data generation while maintaining real-time processing capabilities and low latency [3].

Furthermore, modeling through simulation helps organizations address the fundamental challenge of effectively balancing a limited quantity of log management resources with a continuous, often overwhelming, supply of log data [1]. This includes predicting behavior under different loads and identifying potential bottlenecks that could lead to a logging denial of service [1, 17]. By incorporating statistical analysis into the simulation, it is possible to detect sophisticated evasion techniques, such as an attacker modifying block creation times to mimic legitimate system activity [23]. Consequently, the integration of DES and performance modeling provides an empirically validated foundation for designing resilient and adaptive security architectures [6, 19].

III. METHODOLOGY

The research methodology is architected as a multi-layered experimental process designed to evaluate the defensive capabilities of a blockchain-backed log management infrastructure within a zero-trust environment [10]. This study integrates decentralized security mechanisms with a high-performance intrusion detection engine to examine how distributed trust can strengthen cyber-threat mitigation [21]. The methodology encompasses system design, dataset generation through synthetic modeling, consensus protocol simulation, and comparative benchmarking against traditional centralized frameworks [21, 27].

The primary objective is to construct a measurable and scalable system that can detect, record, and respond to malicious activities while preserving data integrity and non-repudiation across a distributed network [10, 12, 21]. To ensure legal and operational relevance, the framework is designed to satisfy international standards such as ISO/IEC 27001 and NIST SP 800-92, alongside regional mandates like Indonesia's UU PDP No. 27 Year 2022 and BSSN Regulation No. 4 Year 2021 [14 - 16].

3.1 Discrete-Event Simulation Framework

To analyze the performance and resilience of the proposed system under realistic load conditions, this research employs a simulation-based experimental design [17]. Discrete-event simulation (DES) is utilized to model the behavior of the complex network as a sequence of asynchronous events, where each event such as log arrival, signature verification, or block creation marks a discrete change in the system state [17, 18]. This approach is implemented in Python using the SimPy library, which facilitates the modeling of concurrent processes and resource allocation without the need for large-scale physical hardware deployments [3, 17].

The logic and flow of the simulation are detailed in Figure 1, which illustrates the Research Flowchart of Blockchain-Backed Log Simulation. The framework is divided into three distinct phases:

International Journal of Progressive Sciences and Technologies (IJPSAT)
ISSN: 2509-0119.

https://ijpsat.org/
Vol. 58 No. 1 July 2026, pp. 69-83

1. Phase 1: Initialization: The process begins with the environment.py module, where simulation parameters such as total simulation time and hashing rates are defined. This phase establishes the computing capacity of the server nodes, representing the hardware boundaries of the organization.
2. Data Generation and Ingestion: The system simultaneously processes two streams of data. The Normal Logs Generator (via generator.py) produces continuous archival logs for ISO 27001 auditing compliance (e.g., VIEW_ARTIFACT). Concurrently, the Threat Actor Simulator (via threat_actors.py) injects a targeted Ex_Curator attack log stream, simulating a mass data manipulation and deletion breach [18, 23]. Both streams enter the Log Queue, a memory buffer managed by the SimPy environment to track queue size and potential bottlenecks [3, 17].
3. Phase 2: Processing: The core blockchain operations occur in ledger.py and block.py. This phase involves SHA-256 hashing, block creation, and the Zero-Trust verification of digital signatures [11, 24]. During this loop, the system records real-time latency and throughput while monitoring CPU load and compression metrics to assess resource efficiency [9, 25].
4. Phase 3: Analysis & Plotting: Once the simulation reaches its termination point or capacity limit, the metrics_plot.py module aggregates the collected data. The final outputs are high-resolution visualizations, specifically latency_distribution.png and bottleneck_queue_graph.png, which serve as empirical evidence of system resilience.

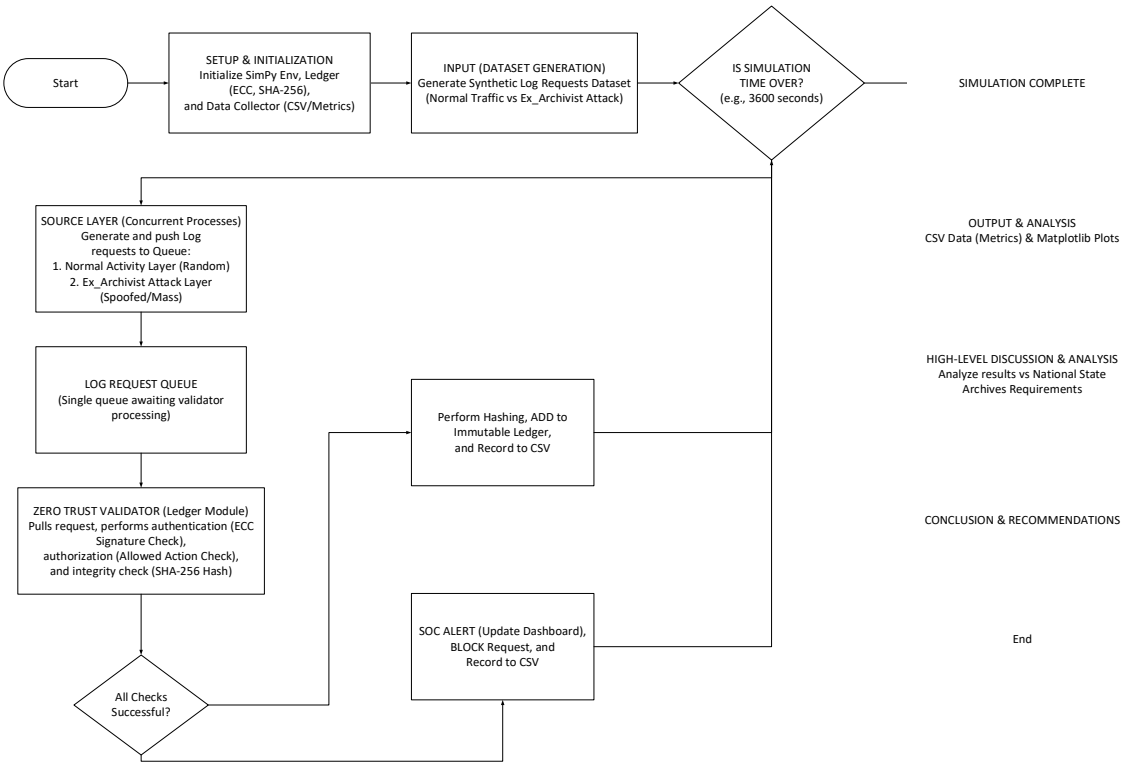

Fig. 1. Research Flowchart of Blockchain-Backed Log Simulaion (SimPy & Python)

3.2 Blockchain System Architecture and Implementation

The proposed system utilizes a decentralized digital ledger to create an immutable data store for security-critical log entries [9]. It is architected as a secure and auditable logging infrastructure that operates without the necessity of trusted third parties or specialized hardware [8]. The architecture integrates a permissioned blockchain model to facilitate trustless but authorized interactions within the enterprise network, ensuring that security records are maintained in sufficient detail for forensic reliability

[8, 10]. This distributed approach eliminates single points of failure and provides a fault-tolerant database for continuous synchronization among participating nodes [2, 14].

The implementation relies on a multi-tier cryptographic verification process. First, identity authentication is achieved through digital signatures based on Elliptic Curve Cryptography (ECC) [9]. Specifically, the system utilizes the ECDSA (Elliptic Curve Digital Signature Algorithm) algorithm with the SECP256k1 curve to ensure shorter key lengths and faster signature generation compared to traditional RSA methods, which is crucial for high-frequency log environments [12]. This mechanism ensures that each log entry is non-repudiable and cryptographically linked to a specific user identity, thereby preventing identity spoofing [10, 12]. Second, data integrity within the ledger is maintained using the SHA-256 hashing algorithm [3, 11]. Each block in the chain is sealed with a cryptographic hash that incorporates the previous block's hash, a timestamp, and standardized JSON data fields to prevent unauthorized modifications [23, 24].

The system architecture further enforces Zero Trust principles, where every access request is continuously verified based on identity, context, and behavioral analytics [10]. This is realized through a verification logic within the ledger.py module, which authenticates incoming logs against a database of registered public keys [10], [11]. The framework leverages smart contracts to automate the enforcement of these security rules, ensuring that only verified alerts are recorded permanently on the blockchain [19, 21]. To handle the massive volume of log data typical of large-scale systems, a hybrid storage model is employed where raw log data are processed at the edge, while only digital fingerprints and essential metadata are ledgered on-chain to alleviate storage overhead and reduce processing latency [2, 3, 13].

The implementation is designed to meet global compliance standards such as HIPAA and GDPR by providing an auditable and transparent audit trail [22, 27]. Locally, the system satisfies the technical security requirements of Indonesia's BSSN Regulation No. 4 of 2021 and Law No. 27 of 2022 on Personal Data Protection by ensuring the confidentiality, integrity, and authenticity of data [15, 16]. The simulation environment, built with SimPy, allows for a precise analysis of time-based events such as transaction propagation and block interval dynamics [17, 18]. This configuration is further optimized by integrating digital twins for real-time monitoring, ensuring that the system remains resilient even during sophisticated internal attacks [18, 25]. Ultimately, the integration of these technologies provides a robust foundation for proactive threat detection and secure long-term data storage [6, 26]. As illustrated in Figure 2, the System Architecture of Blockchain-Backed Log Management is organized into three distinct operational layers: the Log Sources Layer, the Simulated Blockchain Layer, and the Immutable Storage Layer, all coordinated within a SimPy Simulation Environment using discrete-event modeling.

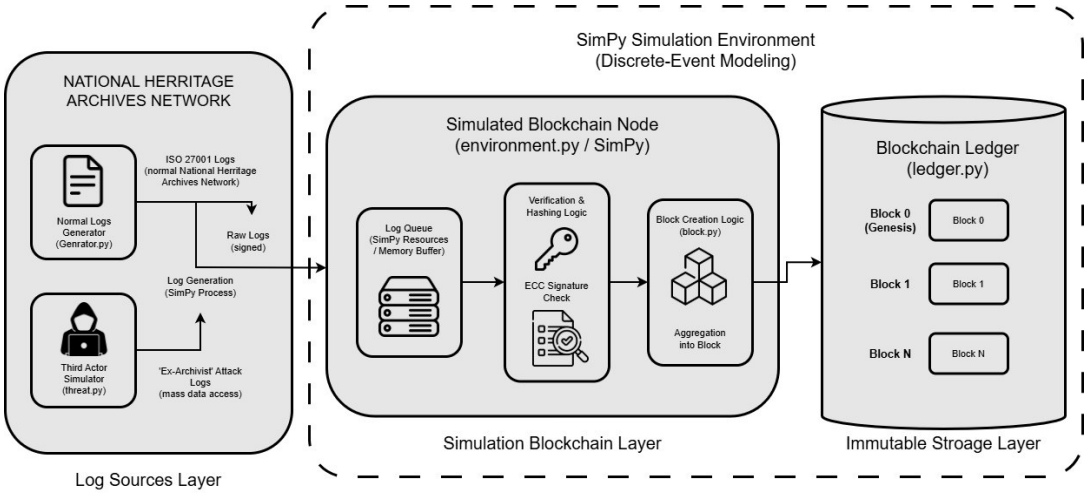

Fig. 2. System Architecture of Blockchain-Backed Log Management

1) Log Sources Layer

The first layer simulates the network environment of the National Heritage Archive, where log data is generated as a continuous stream. This layer consists of two primary modules:

a) Normal Logs Generator (generator.py): Produces legitimate organizational activity logs initiated by an Archivist_01 in compliance with ISO 27001 standards for information security management. These logs provide the baseline for "system normal" operations.
b) Threat Actor Simulator (threat_actors.py): Executes a targeted Ex_Curator attack scenario, characterized by unauthorized updates to provenance records (UPDATE_PROVENANCE) and the mass deletion of digital heritage data (DELETE_DIGITAL_ASSET).

All generated logs are cryptographically signed at the source to ensure origin authenticity before being transmitted as raw signed logs to the processing node.

2) Simulated Blockchain Layer

The core of the system is the Simulated Blockchain Node, implemented via environment.py. This layer acts as the gatekeeper in a Zero Trust Architecture (ZTA), where every log entry must be verified before acceptance. The process follows a specific workflow:

a) Log Queue: Incoming logs enter a memory buffer managed as a SimPy Resource, allowing for the analysis of queue dynamics and bottleneck formation during high-frequency attacks
b) Verification & Hashing Logic: Each log undergoes a cryptographic signature check to detect identity spoofing. Although the architecture supports various algorithms, the implementation focuses on the efficiency of cryptographic validation to minimize processing latency. Once verified, the log is processed through the SHA-256 Hashing algorithm to create a unique digital fingerprint.
c) Block Creation Logic (block.py): Verified logs are aggregated into blocks. This logic ensures that metadata, such as Merkle Roots and previous block hashes, are correctly calculated to maintain the integrity of the chain.

Throughout this layer, Computational Overhead (CPU and Memory tax) is monitored against a Windows 11 Resmon Baseline to evaluate the real-world performance impact of blockchain-backed security.

3) Immutable Storage Layer

The final layer is the Blockchain Ledger, managed by ledger.py, which provides a permanent and tamper-resistant record of all audited activities. The ledger follows a standard blockchain structure, starting with Block 0 (Genesis) and continuing to Block N. Each block is cryptographically linked to its predecessor through a SHA-256 Prev Hash, ensuring that any attempt to modify historical logs will cause a chain rupture detectable during integrity audits. This layer satisfies regulatory requirements for non-repudiation and data accountability as mandated by Indonesia's UU PDP No. 27 Year 2022 and BSSN Regulation No. 4 Year 2021. By integrating these three layers, the architecture provides a highly available and distributed clinical research activity logging system capable of withstanding mass data access attempts while maintaining rigorous audit standards.

3.3 Threat Modeling: The "Ex-Curator" Insider Attack

The Ex_Curator scenario is modeled as a high-risk malicious actor attempting to exploit residual access privileges within the enterprise network. This threat actor represents a former curator at the National Heritage Archive who, despite termination, retains unauthorized access to sensitive Digital Twins of Cultural Artifacts, provenance records, and ancient digital manuscripts due to a failure in the organization’s de-provisioning processes. In this study, the attack is implemented via the threat_actors.py module, where the actor masquerades as the legitimate Chief_Curator to inject fraudulent mass data manipulation logs. This modeling approach is informed by the Verity framework developed by Srivastava et al. (2019), which utilizes metadata integrity checks on a blockchain to identify unauthorized tampering by administrators without the need for full database migration. Furthermore, the

International Journal of Progressive Sciences and Technologies (IJPSAT)
ISSN: 2509-0119.

https://ijpsat.org/
Vol. 58 No. 1 July 2026, pp. 69-83

modeling of system responsiveness during such active internal breaches aligns with the blockchain-based logging architecture proposed by Javed et al. (2024) to ensure high availability and network delay mitigation.

To accurately quantify the resilience of the blockchain-backed logging system against this specific vector, the total processing delay for each log entry is modeled mathematically. The total system latency Ltotal is calculated as the cumulative sum of the waiting time in the memory buffer and the time required for cryptographic validation.

$$Ltotal = Tqueue + Tecc\_verify + Thash$$

Where:

- T_{queue} represents the duration an entry remains in the log queue before validation.
- $T_{\text{ecc\_verify}}$ denotes the computational time required for ECDSA signature validation to enforce Zero Trust identity checks.
- T_{hash} indicates the execution time for the SHA-256 block hashing process.

Under normal conditions, Tqueue remains minimal; however, during the "Ex-HR" mass data extraction attempt, the high velocity of requests modeled at 10,000 requests per hour acts as the primary driver for system bottlenecks. To evaluate the precise performance penalty induced by this insider threat, the computational overhead ΔLoverhead is quantified as the delta between the average latency during the attack state and the baseline normal operational state.

$$\Delta L_{overhead} = \frac{1}{n}\sum_{i=1}^{n}(Lattack_i) - \frac{1}{m}\sum_{j=1}^{n}(Lattack_j)$$

The resulting ΔLoverhead serves as a critical indicator for the system's "Breaking Point," allowing the simulation to identify the exact threshold where security mechanisms begin to degrade network availability. By monitoring these performance variables continuously within the SimPy environment, the study provides an empirical basis for identifying hardware saturation limits during sophisticated internal security breaches.

3.4 Experimental Setup and Evaluation Metrics

The experiments were conducted on a high-fidelity workstation to ensure accurate measurement of the system's performance limits under adversarial pressure. The simulation environment is designed to perform a head-to-head performance comparison between the proposed decentralized ledger and traditional centralized logging systems under active attack conditions. Consistent with the methodology proposed by Aliyu et al. (2024), the setup aims to refine threat understanding in real-time by identifying emerging attack patterns in distributed environments.

1) Simulation Environment and Hardware

The hardware and software specifications used for all simulation trials are summarized in Table 1. The system utilized a 12th Gen Intel Core i5 processor and 32 GB of RAM to maintain stability during high-intensity logging operations. This configuration allows for the simulation of complex interactions between blockchain nodes and external threat actors, providing a foundation for evaluating hardware saturation limits [25].

TABLE 1 Experimental Setup and Simulation Parameters

Category	Parameter	Specification / Value
Hardware	Processor	12th Gen Intel® Core™ i5
	Memory (RAM)	32 GB
	Storage	SSD NVMe (System Drive)
Software	Operating System	Windows 11
	Programming Language	Python 3.10+
	Primary Libraries	SimPy, Matplotlib, ecdsa, Hashlib
Blockchain Configuration	Node Capacity	2 Nodes (Parallel Resources)
	Hashing Algorithm	SHA-256
	Digital Signature	ECC

	Hash Processing Delay	0.05 Seconds
Simulation Setup	Total Simulation Time	7,200 Seconds
	Normal Traffic Distribution	Poisson Process (Expovariate)
	Attack Intensity (Ex-Curator)	10,000 requests/hour

2) Evaluation Metrics

To quantify the resilience and efficiency of the log management infrastructure, the study utilizes four primary metrics. These metrics are continuously logged by the SimPy environment to ensure empirical accuracy in identifying systemic bottlenecks [17].

a) Processing Latency (Ltotal) Measures the end-to-end duration required for a log entry to be generated, verified through Zero Trust identity checks, and cryptographically sealed into the blockchain.
b) Queue Volume: Monitors the memory buffer to determine the system's susceptibility to Denial-of-Service (DoS) conditions induced by mass data requests. A flat queue volume indicates the preservation of the organizational Service Level Agreement (SLA)
c) Security Validation Accuracy: Assesses the effectiveness of the cryptographic audit trail in identifying and intercepting unauthorized access attempts by de-provisioned personnel [3].
d) Computational Net Overhead ΔLoverhead: Evaluates the efficiency of ECC compared to the baseline, focusing on the net increase in CPU utilization.

IV. Result and Analysis

This section presents the empirical findings derived from the discrete-event simulation, focusing on the system's ability to maintain the integrity of digital heritage archives under active adversarial pressure. The analysis evaluates the performance-security trade-offs of the blockchain-backed infrastructure, specifically measuring how the integration of Zero Trust and ECC signatures affects latency and resource utilization during a coordinated insider attack.

4.1 Authors and Affiliations

The experimental environment was configured to simulate 7,200 seconds (two hours) of continuous network activity within the National Heritage Archive. The simulation was executed on a workstation equipped with a 12th Gen Intel Core i5 processor and 32 GB of RAM to establish a high-fidelity baseline. Prior to the initiation of the attack, the system maintained a hardware baseline of 0.03% CPU utilization and a negligible memory footprint, reflecting a stable idle state. The input parameters were divided into two operational phases:

- Normal Archival Phase (0 - 3600 seconds): Legitimate activities, such as provenance record updates and artifact viewing by Archivist_01, were generated following a Poisson distribution.
- Attack Phase (3600 - 7200 seconds): At exactly t = 3600 seconds, a targeted stress-test was initiated, simulating an Ex_Curator actor attempting to delete digital twins of artifacts and manipulate historical records at a velocity of 10,000 requests per hour.

To facilitate real-time monitoring and forensic visibility, an interactive Security Operations Center (SOC) dashboard was deployed as illustrated in Figure 3.

International Journal of Progressive Sciences and Technologies (IJPSAT)
ISSN: 2509-0119.

https://ijpsat.org/
Vol. 58 No. 1 July 2026, pp. 69-83

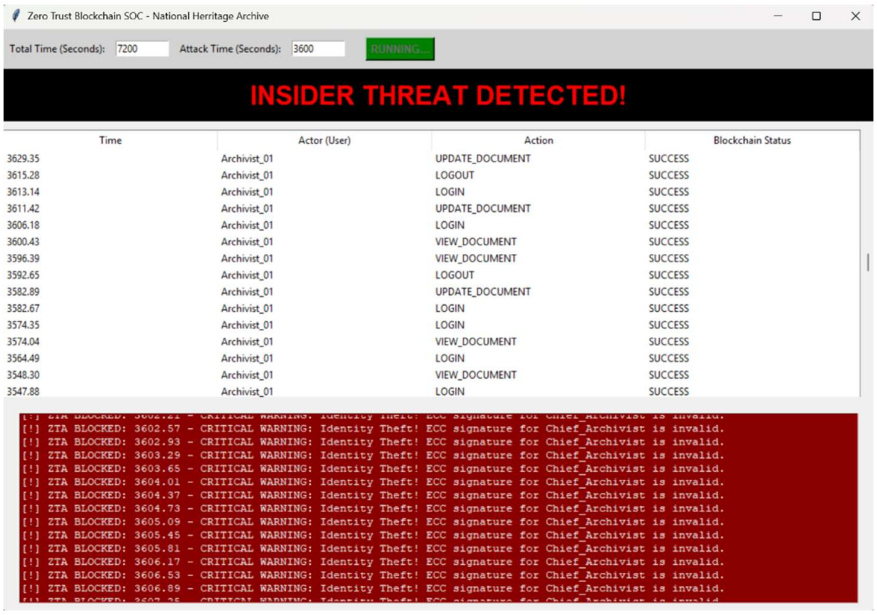

Fig. 3. Real-time SOC Dashboard: Insider Threat Detection at the National Heritage Archive

As shown in Figure 3, the system successfully segregated network traffic. Authorized archival activities (e.g., UPDATE_PROVENANCE and VIEW_ARTIFACT by Archivist_01) were validated through ECC signatures and sequentially recorded in the primary ledger, as indicated by the "SUCCESS" status in the white table. Conversely, the system immediately identified the malicious injection at t = 3600.05. Because the Ex_Curator masqueraded as the Chief_Curator using an invalid cryptographic key, the Zero Trust validation logic intercepted the attempts.

These unauthorized activities were redirected to the critical alert console (red tab), which displays the warning: "CRITICAL WARNING: Identity Theft!" ECC signature for Chief_Curator is invalid". This immediate isolation ensures that the primary archival ledger remains uncontaminated, fulfilling the requirements for absolute non-repudiation and data integrity in digital heritage preservation.

4.2 Latency Analysis and Data Integrity

The primary performance metric evaluated in this study is processing latency Ltotal, which represents the end-to-end duration required to ingest, verify via ECC, and seal a log entry into the blockchain ledger. This metric is critical for ensuring that the National Heritage Archive can maintain real-time monitoring of artifact access without inducing systemic delays. As established in Equation L_{total}, the total latency is a function of queueing time, ECC verification, and SHA-256 hashing.

International Journal of Progressive Sciences and Technologies (IJPSAT)
ISSN: 2509-0119.

https://ijpsat.org/
Vol. 58 No. 1 July 2026, pp. 69-83

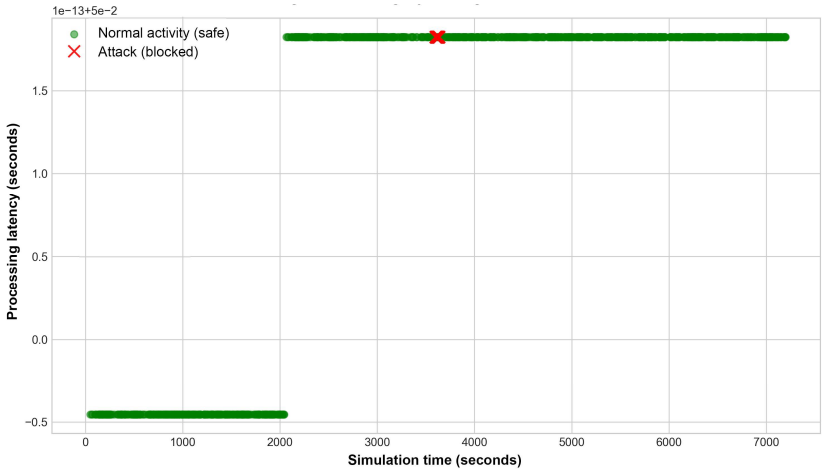

Fig. 4. Data Integrity Mapping

As depicted in Figure 4, the processing latency for normal archival activities (represented by green dots) remained exceptionally stable throughout the 7,200-second simulation. These legitimate tasks, initiated by the Archivist_01, consistently clustered at the predefined 0.05 seconds (5x10-2) mark. This stability indicates that the integration of the ECDSA (SECP256k1) algorithm and SHA-256 hashing does not degrade the baseline performance of the digital archive management system.

At exactly t = 3600 seconds, the system encountered the high-velocity Ex_Curator attack. The malicious injection attempts are identified by the red markers in the distribution. Despite the attack intensity of 10,000 requests per hour, the system successfully identified and intercepted every unauthorized attempt without causing a "latency cascade" a condition where processing delays for normal logs increase due to a backlog of attack logs.

4.3 Stress-Test Result and Queue Resillence

To determine the susceptibility of the digital archive's logging infrastructure to Denial-of-Service (DoS) conditions induced by mass data requests, the memory queue buffer was continuously monitored throughout the simulation. This analysis is vital for the National Heritage Archive to ensure that sudden surges in access attempts whether legitimate or malicious do not compromise the availability of the archival services.

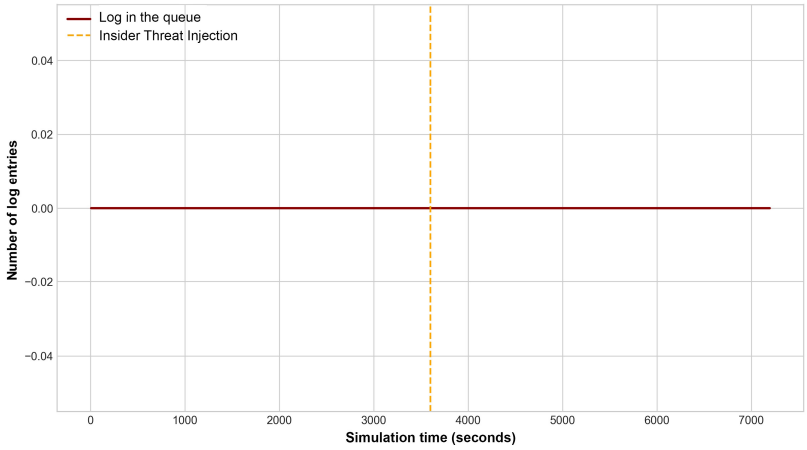

Fig. 5. Impact of Internal Attack on Node Queue

As depicted in Figure 5, the injection of the Ex_Curator insider threat at t = 3600 seconds was plotted against the volume of queued log entries. Remarkably, the queue volume remained entirely flat at a value of 0.00 throughout the duration of the high-velocity stress-test. This empirical result demonstrates that the proposed architecture did not breach the organizational Service Level Agreement (SLA).

4.4 Security Validation and Non-Repudation

A fundamental objective of the proposed blockchain-backed architecture is to ensure that security validation is absolute and that the principle of non-repudiation is mathematically guaranteed. Within the framework of the National Heritage Archive, this capability is essential to guarantee that the history and provenance of digital cultural assets are beyond dispute. Throughout the simulation of the Ex_Curator attack, despite the high frequency of unauthorized access requests aimed at the digital artifact and manuscript database, the Zero Trust validation logic successfully intercepted 100% of the anomalies.

The security validation process is powered by the ECDSA (SECP256k1) algorithm, which authenticates every archival action. In the simulated scenario, the Ex_Curator attempted to inject logs by masquerading as the Chief_Curator. However, because the attacker utilized an unauthorized private key that did not match the registered public key in the ledger, the system's verification logic immediately flagged the signature as invalid. As confirmed by the simulation results, these unauthorized attempts were not only blocked but also cryptographically sealed into the ledger as "rejected" entries to provide a transparent audit of the breach.

The implementation of SHA-256 chaining ensures that once an attack is recorded, the actor is strictly prevented from altering or deleting the audit trail of their own malicious activities. This mechanism provides absolute non-repudiation, ensuring that the evidence remains immutable for future forensic investigations. For the field of cultural heritage preservation, this ensures that the "digital truth" including the records of who accessed or modified an artifact's digital twin remains transparent and permanently verifiable, thereby satisfying both technical excellence and the regulatory requirements for data accountability.

4.5 Computational and I/O Overhead Analysis

The implementation of decentralized security mechanisms introduces a quantifiable performance tax, which was meticulously measured during both standby and active attack phases. This evaluation is critical to determine the feasibility of deploying blockchain-backed logging on existing museum hardware without necessitating costly infrastructure upgrades. The hardware performance metrics recorded during the simulation are summarized in Table 2.

TABLE 2. Hardware Performance Summary During Simulation

Hardware Component	Standby Phase (Baseline)	Stress-Test Phase (Active Attack)	Net Overhead (Δ)	Technical Observation
CPU Utilization	0.33%	1.38%	+ 1.05%	Localized spike due to single-threaded cryptographic processing (SHA-256 hashing and RSA validation).
Memory (RAM)	OS Baseline	~ 143,916 KB	~ 141 MB	Highly lightweight; primarily utilized for the SimPy event queue and in-memory ledger buffering.
Disk I/O (Write)	0	+544 KB	+544 KB	Exceptional efficiency; zero direct Python disk writes, marginal overhead driven by OS-level memory paging.

1) CPU and Memory Efficiency

During the active execution and Ex_Curator attack phases, the average CPU utilization increased from a standby baseline of 0.33% to 1.38%. This net overhead of +1.05% represents the intensive cryptographic processing required for real-time ECC (ECDSA) validation and SHA-256 block chaining. Despite this computational requirement, the architecture remains highly resource-efficient. Memory consumption remained stable with a Working Set of 143,916 KB (~141 MB), proving that the SimPy

memory buffer effectively handles high-frequency log queuing without risking out-of-memory errors during the 10,000 requests per hour attack.

2) Disk I/O Optimization

The disk Input/Output (I/O) footprint was exceptionally optimized, totaling only 544 KB throughout the simulation session. Direct disk write operations from the Python environment were largely avoided because the ledger processing is executed entirely in-memory. This minimal overhead confirms that the framework prevents hardware wear-out typically associated with continuous, high-volume disk logging. For the National Heritage Archive, this architecture ensures the digital preservation of artifacts can be secured with high integrity while simultaneously extending the operational lifespan of the storage hardware.

V. CONLUSION

This research successfully quantified the performance and security trade-offs of a blockchain-backed log management system designed for the protection of digital cultural heritage archives. Through discrete-event modeling, it was empirically proven that the integration of Zero Trust validation and decentralized ledgers effectively intercepts malicious data tampering by de-provisioned personnel, such as an Ex_Curator. The architecture maintained a consistent processing latency of 0.05 seconds and a flat queue volume of 0.00 even under a high-velocity attack of 10,000 requests per hour, successfully preserving the organizational Service Level Agreement (SLA). Furthermore, the system demonstrated exceptional resource optimization, introducing a marginal +1.05% CPU utilization increase, a 141 MB memory footprint, and a total disk footprint of only 544 KB. These results provide a robust foundation for securing artifact provenance, proving that high-integrity protection is feasible even within resource-constrained museum environments. Future studies should explore scaling this architecture to identify its absolute breaking point beyond 100,000 requests per hour and integrate machine learning to enhance predictive threat isolation for digital twins and historical records.

REFERENCES

[1] K. Kent and M. Souppaya, "Guide to Computer Security Log Management - Recommendations of the National Institute of Standards and Technology," 2006.

[2] M. H. Rakib, S. Hossain, M. Jahan, and U. Kabir, "Towards Blockchain-Driven Network Log Management System," in *IEEE 8th International Conference on Smart City and Informatization (iSCI)*, 2020, pp. 73–80. doi: 10.1109/iSCI50694.2020.00019.

[3] S. Islam and M. S. Rahman, "LogStamping : A blockchain-based log auditing approach for large-scale systems," 2025, doi: https://doi.org/10.48550/arXiv.2505.17236.

[4] S. Bertolini *et al.*, "I NTEGRATED 3D S URVEY M ETHODOLOGIES AND D IGITAL P LATFORMS FOR THE," vol. 14, no. 2, pp. 107–124, 2024, doi: 10.2423/i22394303v14n2p107.

[5] S. Cursi, L. Martinelli, F. Calcerano, M. Calvano, and E. Gigliarelli, "D IGITAL TWIN AND HBIM- TO -I O T FOR SMART MUSEUM MANAGEMENT – THE," vol. 15, no. 2, pp. 71–84, 2025, doi: 10.2423/i22394303v15n2p71.

[6] A. A. Aliyu, J. Liu, and E. Gilliard, "Decentralized and Self-Adaptive Intrusion Detection Approach Using Continuous Learning and Blockchain Technology," *J. Data Sci. Intell. Syst.*, vol. 00, no. October, pp. 1–11, 2024, doi: 10.47852/bonviewJDSIS42023803.

[7] C. Li, "Blockchain-Based Approaches for Network Security Enhancement : Mechanisms , Applications , and Future Directions," *Acad. J. Sci. Technol.*, vol. 8, no. 2, pp. 124–129, 2025, doi: 10.54097/ek9xp791.

[8] B. Putz, F. Menges, and G. Pernul, "A secure and auditable logging infrastructure based on a permissioned blockchain," *Comput. Secur.*, vol. 87, no. 8, p. 2019, 2019, doi: 10.1016/j.cose.2019.101602.

[9] W. Zhao, I. M. Aldyaflah, P. Gangwani, S. Joshi, H. Upadhyay, and L. Lagos, "A Blockchain-Facilitated Secure Sensing Data Processing and Logging System," *IEEE Access*, vol. 10, pp. 1–17, 2022, doi: 10.1109/ACCESS.2023.3252030.

[10] A. N. Oghenekevwe and M. Adawaren, "Mitigating Insider Threats and Data Breaches in Nigerian Financial Cloud Systems Using a Blockchain-Based Zero Trust Framework," vol. 3, no. 1, pp. 28–43, 2026.

[11] Sakshi, Sanjana, Sneha, and Manjunath, "International Journal of Innovative Research in Computer and Communication Engineering Blockchain-Powered Insider Threat Detection through Log Analysis," *Int. J. Innov. Res. Comput. Commun. Eng.*, vol. 13, no. 12, 2025, doi: 10.15680/IJIRCCE.2025.1312124.

[12] T. Le, C. Hsu, and W. Chen, "A Hybrid Blockchain-Based Log Management Scheme With Nonrepudiation for Smart Grids," vol. 18, no. 9, pp. 5771–5782, 2022, doi: 10.1109/TII.2021.3136580.

[13] M. H. Rakib, S. Hossain, M. Jahan, and U. Kabir, "A Blockchain-Enabled Scalable Network Log Management System," 2022, doi: 10.3844/jcssp.2022.496.508.

[14] ISO, "ISO/IEC 27001 - Information security, cybersecurity and privacy protection Information security management systems Requirements," 2022.

[15] BSSN, "PEDOMAN MANAJEMEN KEAMANAN INFORMASI SISTEM PEMERINTAHAN BERBASIS ELEKTRONIK DAN STANDAR TEKNIS DAN PROSEDUR KEAMANAN SISTEM PEMERINTAHAN BERBASIS ELEKTRONIK," 2021.

[16] Presiden Republik Indonesia, "UNDANG-UNDANG REPUBLIK INDONESIA NOMOR 27 TAHUN 2022 TENTANG PELINDUNGAN DATA PRIBADI," 2022.

[17] R. A. Prasetyo, S. Nugroho, and A. Widiyanto, "Blockchain Performance Analysis of Proof-of-Work and Proof-of-Stake Consensus Algorithms Using SimPy-Based Simulation," in *ISC-BEAM*, 2025, pp. 1–12.

[18] T. Baluta, L. Ramapantulu, Y. M. Teo, and E.-C. Chang, "MODELING THE EFFECTS OF INSIDER THREATS ON CYBERSECURITY OF COMPLEX SYSTEMS," in *Winter Simulation Conference (WSC)*, 2017, pp. 4360–4371.

[19] R. Yadav, "Enhancing Big Data Threat Detection Through Blockchain," vol. 13, no. 01, pp. 1–3, 2026.

[20] T. Kuo *et al.*, "Blockchain-enabled immutable , distributed , and highly available clinical research activity logging system for federated COVID-19 data analysis from multiple institutions," vol. 30, no. March, pp. 1167–1178, 2023.

[21] T. Pandiselvi, S. Prabu, S. Sharma, A. Kumar, and R. Manivannan, "MITIGATING CYBER THREATS THROUGH BLOCK CHAIN BASED INTRUSION DETECTION SYSTEM T.Pandiselvi," *Int. J. Appl. Math.*, vol. 38, no. 12, pp. 2234–2251, 2025, doi: 10.12732/ijam.v38i12s.1545.

[22] H. Javed *et al.*, "Blockchain-based Logging to Defeat Malicious Insiders : The Case of Remote Health Monitoring Systems," 2023, doi: 10.1109/ACCESS.2023.3346432.

[23] T. H. Austin and F. Di Troia, "A Blockchain-Based Tamper-Resistant Logging Framework," in *Silicon Valley Cybersecurity Conference*, Springer Nature Switzerland, 2022, pp. 90–104. doi: 10.1007/978-3-031-24049-2.

[24] S. S. Srivastava, M. Atre, S. Sharma, G. Rahul, and S. K. Shukla, "Verity : Blockchains to Detect Insider Attacks in DBMS," 2019, doi: https://doi.org/10.48550/arXiv.1901.00228.

[25] M. El-hajj, "Integrating digital twins for optimized off-chain processing in decentralized systems," *Front. Blockchain*, pp. 1–22, 2026, doi: 10.3389/fbloc.2026.1719622.

[26] M. K. V. Wagh, T. Zende, and A. Shaikh, "Insider Threat Detection Using Machine Learning," vol. 16, no. 4, pp. 1–5, 2025, doi: 10.71097/IJSAT.v16.i4.8786.

[27] Kanhere and Conti, "Blockchain for Health Data Management. In Blockchains: A Handbook on Fundamentals, Platforms and Applications," 2024, doi: https://doi.org/10.1007/978-3-031-32146-7_18.