Cloud Computing and Data Science
https://ojs.wiserpub.com/index.php/CCDS/

UNIVERSAL WISER PUBLISHER

Research Article

A Cyber Secure Framework for Modern Database Systems

Hafiz Malik Usman Shahzad, Altaf Hussain*

Department of Computer Science, KICSIT Campus, Institute of Space Technology, Pakistan
Email: altaf.hussain@ist.edu.pk

Received: 22 April 2026; **Revised:** 7 June 2026; **Accepted:** 17 June 2026

Abstract: Database servers remain prime targets for cyberattacks due to their critical role in storing sensitive organizational data and supporting missioncritical applications; however, database security is often implemented through isolated controls, such as access management, encryption, logging, and backup, leading to fragmented protection and exploitable gaps across applications, identity, configuration, monitoring, and governance layers. This study proposes a comprehensive cybersecurity framework for modern database systems based on a layered architecture that integrates prevention, detection, and governance across the database lifecycle. The framework comprises seven coordinated layers: network and perimeter segmentation with mutual Transport Layer Security (mTLS), host and operating system hardening, identity and privileged access management, secure database configuration controls, data protection mechanisms, centralized key management, and continuous monitoring with correlation and anomaly-based detection. Using design science research methodology, a prototype reference deployment was evaluated against a baseline system under realistic workloads and simulated attack scenarios, including credential compromise, privilege escalation, Structures Query Language (SQL) injection via application pathways, data exfiltration attempts, destructive ransomware-like queries, and insider-style bulk data exports. The results indicate that the proposed framework reduces attack surface exposure, limits potential damage, enhances detection readiness, and strengthens containment and recovery capabilities. Although additional controls introduced measurable overhead in latency, throughput, and telemetry volume, these trade-offs remained operationally acceptable with appropriate tuning. The study contributes a validated reference architecture and evaluation framework for secure database deployment across cloud, hybrid, and on-premises environments.

***Keywords*:** database security, cyber-secure architecture, zero trust, privileged access management, encryption and key management, audit logging, anomaly detection

1. Introduction

Database servers constitute the backbone of modern digital enterprises, as they store and process some of the most sensitive and valuable organizational assets, including customer identities, financial records, medical data, intellectual property, and operational telemetry. With the increasing adoption of cloud-native and distributed architectures, database systems are now accessed by a wide range of internal services, external clients, and automated processes. This expanded accessibility introduces a broader attack surface and amplifies the potential consequences of security breaches. Concurrently, the requirements for security governance have intensified; access control decisions must be not

DOI: https://doi.org/10.37256/ccds.72202610054

only enforceable but also auditable, encryption mechanisms must be correctly implemented, and security controls must remain effective across hybrid environments where physical location is no longer a reliable indicator of trust [1, 2].

The database attack surface is no longer confined to the database engine itself but extends across interconnected application and service layers. In real-world scenarios, intrusion attempts often follow an application–Application Programming Interface (API)–database pathway, where vulnerabilities in input validation, insecure query construction, or weak service-to-service authentication mechanisms provide entry points. Structures Query Language (SQL) injections exemplify such chained vulnerabilities, enabling malicious actors to manipulate web requests into unauthorized database operations, data exfiltration, or destructive actions. These risks are particularly severe when database accounts are over-privileged and monitoring mechanisms are inadequate [3, 4]. Furthermore, administrative interfaces, remote access tools, and third-party integrations expand the threat landscape, as compromised credentials in these contexts can grant immediate and high-impact access to both data and control planes [5]. Despite the deployment of standardized security controls such as firewalls, Role-Based Access Control (RBAC), encryption, and backup strategies, vulnerabilities frequently arise at system boundaries. Misconfigurations, inconsistent policy enforcement, and limited cross-layer visibility among network, host, and database components contribute to persistent security weaknesses. Default configurations are seldom secure by design, and operational pressures often lead to the normalization of risky exceptions [6]. Additionally, encryption mechanisms introduce performance trade-offs; while techniques such as transparent data encryption can mitigate risks associated with data theft, they may impose measurable overhead depending on workload and platform characteristics, highlighting the need for performance-aware security design [7]. Crucially, the effectiveness of encryption depends on robust key management practices, including secure storage, regular rotation, and segregation of duties, without which data-at-rest protections can be undermined in both cloud and multi-cloud environments [8].

Effective database security further depends on continuous monitoring, timely detection, and the availability of reliable forensic evidence. Modern database systems generate extensive audit logs, performance metrics, and behavioral indicators; however, these data sources are often underutilized due to the lack of integrated analysis frameworks. Recent research emphasizes the importance of Key Performance Indicator (KPI)-driven and log-based anomaly detection approaches, particularly as database environments grow in complexity. Such approaches require robust evaluation frameworks and multi-metric analysis to avoid over-reliance on isolated signals [9]. Moreover, insider threats and attacks involving valid credentials necessitate advanced monitoring techniques capable of distinguishing between legitimate and malicious behavior, while ensuring the integrity of logs for investigative and compliance purposes [10-13].

A fundamental challenge addressed in this study is that the security of database servers is frequently implemented as a collection of fragmented controls—such as access management, encryption, and monitoring—that do not operate as an integrated, end-to-end system. This disjointed approach often results in residual vulnerabilities, limited detection capabilities, and only partial assurance when systems are exposed to realistic operational workloads and adversarial conditions [5]. To address this limitation, the present study aims to develop a cyber-secure framework for database servers that systematically integrates preventive, detective, and governance-oriented controls across the entire database lifecycle. Specifically, the study seeks to (i) design a layered security model encompassing identity management, configuration hardening, encryption and key governance, and continuous monitoring; (ii) develop a prototype implementation in the form of reference architecture; and (iii) evaluate the model in terms of both security effectiveness and associated performance overhead. In line with these objectives, the study is guided by the following research questions:

1. RQ-1: To what extent does the proposed model reduce the impact of successful attacks compared to baseline deployment?

2. RQ-2: What is the performance overhead introduced by the additional security controls in terms of latency, throughput, and resource utilization?

3. RQ-3: How effectively can monitoring mechanisms detect anomalous or malicious database activities?

The significance of this research lies in its provision of a testable and integrated approach to database security that is applicable across cloud and hybrid environments. Furthermore, it explicitly addresses the critical balance between strengthening security assurance and maintaining operational feasibility, an increasingly important consideration in modern data-intensive systems [7, 9].

2. Literature review

Modern applications are predominantly built upon database servers, making them high-value targets in cybersecurity. Compromising a database server often grants direct access to sensitive assets, including credentials, Personally Identifiable Information (PII), and financial records, while simultaneously enabling service disruption. A common external attack vector is application-layer exploitation, particularly SQL injection, whereby adversaries manipulate query logic to extract or modify data, escalate privileges, or establish a foothold for further compromise [3-4].

Beyond injection-based attacks, threat actors frequently target authentication endpoints and administrative interfaces through techniques such as credential stuffing and password guessing. These attacks exploit weak credential management practices and inconsistencies in identity governance across distributed systems [14]. Furthermore, ransomware campaigns increasingly extend to database environments, affecting not only availability but also data integrity through encryption of database files, destruction of backups, or data exfiltration for extortion, thereby emphasizing the importance of robust recovery mechanisms and immutable audit trails [15].

Insider threats represent another persistent and complex risk category. Privileged users, including database administrators, developers, and third-party contractors, may misuse their elevated access either intentionally or inadvertently. Such actions are particularly challenging to detect, as they often resemble legitimate activity patterns. Recent studies highlight the necessity of integrating both technical and procedural controls for insider threat mitigation, noting that inadequate monitoring and weak governance structures significantly undermine existing safeguards [10].

Misconfiguration and delayed vulnerability remediation further exacerbate database exposure. Common issues include insecure default settings, unnecessary services, permissive network configurations, and insufficient auditing policies. Human factors, such as system complexity and skill gaps, are frequently cited as root causes of misconfigurations [16], while ineffective patch management practices leave systems vulnerable to exploitation of known weaknesses over extended periods [17].

Existing research typically addresses database security as a collection of discrete controls rather than as an integrated system. Key approaches include access control models such as RBAC and Attribute-Based Access Control (ABAC), combined with least-privilege principles [1, 18], and enhanced authentication mechanisms such as Multi-Factor Authentication (MFA), centralized Identity and Access Management (IAM), and Privileged Access Management (PAM) to reduce standing privileges and improve accountability [19-20]. Encryption remains a core safeguard, including Transport-Layer Security (TLS) for data in transit and techniques such as Transparent Data Encryption (TDE) for data at rest; however, performance overhead remains a consistent concern requiring empirical evaluation [7].

Cryptographic key management is equally critical, involving Hardware Security Modules (HSMs), Key Management Systems (KMS), periodic key rotation, and separation of duties. While approaches such as Bring-Your-Own-Key (BYOK) enhance resilience against single-provider compromise, they introduce additional operational complexity [8]. Logging and auditing mechanisms, particularly those incorporating tamper-resistance and immutability, are essential for ensuring accountability and forensic analysis. Emerging approaches, including blockchain-based logging, offer improved integrity but face challenges related to scalability and cost [11].

Detection and response capabilities have also evolved, with increasing emphasis on anomaly detection using database telemetry and logs, coupled with integration into Security Information and Event Management (SIEM) and Security Orchestration, Automation, and Response (SOAR) platforms for automated incident response [9, 21-22]. These approaches align with broader security paradigms such as defense-in-depth and the Confidentiality, Integrity, and Availability (CIA) triad, while Zero Trust models further emphasize continuous verification, identity-centric controls, and enhanced visibility. Although Zero Trust frameworks are conceptually promising, empirical validation in database contexts remains limited [5, 23].

A notable gap in the literature is the absence of a unified, end-to-end database security model that integrates configuration hardening, identity and privilege governance, cryptographic key management, tamper-evident auditing, and detection and response into a cohesive and measurable architecture. Existing studies often evaluate controls in isolation, with insufficient attention to operational overhead, false positives, and insider misuse scenarios. Consequently, there is a growing need for integrated frameworks that quantitatively assess both security effectiveness and performance trade-offs [5, 7, 10].

Islam et al. in [24] further reinforce this gap through a systematic literature review on cybersecurity risk assessment frameworks for engineering databases. Their findings indicate that many existing frameworks are overly generic and

insufficiently tailored to database-specific threats, such as unauthorized access, privilege misuse, data exfiltration, misconfiguration, weak authentication, and inadequate monitoring. The study emphasizes the importance of structured and measurable risk assessment approaches that incorporate asset criticality, threat likelihood, vulnerability exposure, and control effectiveness. These insights directly support the motivation of the present study, which aims to move beyond isolated controls and propose an integrated, layered security framework encompassing identity governance, database hardening, encryption, key management, monitoring, and response. In summary, the literature highlights significant progress in individual aspects of database security, including access control, encryption, monitoring, and detection mechanisms. However, these approaches are predominantly fragmented and lack integration into a unified, end-to-end architecture capable of addressing modern threat landscapes. Additionally, limited empirical evaluation of operational trade-offs and detection effectiveness underscores a critical research gap. This study addresses the limitations by proposing and evaluating a comprehensive, layered database security framework that integrates preventive, detective, and governance controls into a cohesive and measurable system suitable for contemporary cloud and hybrid environments. Table 1 summarizes the literature work conducted in this study.

Table 1. Literature Summary

Category	Key Issues/Threats	Existing Approaches	Limitations	Key References
Application-layer attacks	SQL injection, query manipulation	Input validation, secure coding	Ineffective if privileged access exists	[3-4]
Authentication attacks	Credential stuffing, weak passwords	MFA, IAM, PAM	Inconsistent identity governance	[14, 19]
Ransomware threats	Data encryption, backup destruction	Backup strategies, recovery planning	Weak recovery posture, lack of immutability	[15]
Insider threats	Privilege misuse, legitimate-appearing attacks	Monitoring, governance controls	Difficult detection, weak audit mechanisms	[10]
Misconfiguration	Insecure defaults, patch delays	Hardening, patch management	Human error, complexity, skills gaps	[16-17]
Access control	Over-privileged roles	RBAC, ABAC, least privilege	Fragmented implementation	[1]
Encryption	Data exposure risks	TLS, TDE	Performance overhead concerns	[7]
Key management	Key compromise risks	HSM, KMS, BYOK	Operational complexity	[8]
Logging & auditing	Lack of traceability	Audit trails, blockchain logs	Cost, scalability issues	[11]
Detection & response	Delayed detection	SIEM, SOAR, anomaly detection	False positives, siloed analysis	[9]
Security frameworks	Fragmented controls	Zero Trust, defense-in-depth	Lack of empirical validation	[5]
Risk assessment	Generic frameworks	Structured risk models	Not database-specific	[24]

3. Methodology

This section outlines the methodological framework adopted to design, implement, and evaluate the proposed cyber-secure architecture for database systems.

3.1 *Research design and threat model*

This study adopts a Design Science Research (DSR) methodology integrated with an experimental benchmarking approach to develop and evaluate cyber-secure database architecture. The design science paradigm is particularly appropriate as the primary objective is to construct and validate a practical artifact—namely, an end-to-end layered

cybersecurity model for database servers—that can be instantiated as a reference architecture and assessed for its effectiveness in real-world settings. Complementing this, the experimental component enables systematic measurement of operational trade-offs introduced by security controls, including impacts on latency, throughput, and telemetry generation. Accordingly, the methodology proceeds in two structured phases: (i) model development, involving the specification and implementation of the proposed layered architecture as a functional prototype, and (ii) model evaluation, wherein the secured deployment is compared against a baseline system under controlled, repeatable workloads and adversarial scenarios.

The study is grounded in a comprehensive threat model encompassing three principal adversarial categories: external attackers, insider threats, and compromised application services. External attackers exploit exposed interfaces such as Application Programming Interfaces (APIs), administrative endpoints, and network access points to perform credential-based attacks, injection exploits, vulnerability scanning, and data exfiltration. Insider threats arise from users with legitimate system access, such as employees, database administrators, or contractors, who may intentionally or inadvertently misuse their privileges to access unauthorized data, perform bulk exports, or alter critical records. Additionally, compromised application servers represent a significant and increasingly prevalent attack vector, whereby adversaries gain execution capabilities or token access within the application layer and subsequently pivot to the underlying database by leveraging service credentials. Within this threat context, attackers are assumed to possess several capabilities: (i) access to application endpoints and, potentially, internal services through lateral movement; (ii) possession of stolen credentials or tokens, including API keys and session artifacts; (iii) the ability to attempt privilege escalation via misconfigurations, role abuse, or vulnerable components; and (iv) the use of automated tools for brute-force attacks, large-scale data exfiltration, or execution of destructive queries. The architecture explicitly enforces well-defined trust boundaries between key system domains, including client-to-application, application-to-database, and administrator-to-management plane interactions.

The model operates under several practical assumptions: the database server is not directly exposed to the public internet; cryptographic primitives employed within the system are secure and correctly implemented; and absolute prevention of all attacks is infeasible. Consequently, detection and response mechanisms are treated as integral components of the security architecture rather than optional enhancements, ensuring resilience against both known and emerging threats.

3.2 *Proposed layered cyber-secure architecture for database systems*

The proposed architecture adopts a defense-in-depth paradigm, wherein multiple coordinated security layers are implemented to collectively safeguard database systems against a wide spectrum of cyber threats. As illustrated in the Figure 1, the architecture is organized into seven interdependent layers (A–G), each responsible for mitigating specific risk categories while simultaneously generating telemetry to support continuous monitoring, anomaly detection, and incident response. The model operates within clearly defined trust boundaries between clients, application services, administrative domains, and the protected database environment, thereby enforcing strict access control and visibility across all interaction points. Furthermore, the architecture aligns with core security functions—prevention, detection, containment, and recovery—ensuring holistic lifecycle protection.

3.2.1 *Layer A: network and perimeter security*

The outermost layer focuses on restricting unauthorized network access and limiting lateral movement within the infrastructure. Database servers are deployed within segmented private subnets or Virtual Local Area Networks (VLANs), ensuring that database ports are accessible only to authorized application services and controlled administrative endpoints. The use of firewall allowlists, and micro-segmentation enforces granular traffic policies, effectively reducing the attack surface. Additionally, mutual Transport Layer Security (mTLS) is employed for service-to-service communication to ensure bidirectional authentication and prevent spoofing attacks. Administrative access is tightly controlled through bastion or jump hosts, which act as secure entry points with restricted ingress policies. This layer primarily mitigates risks associated with unauthorized access, network-based attacks, and lateral propagation of threats.

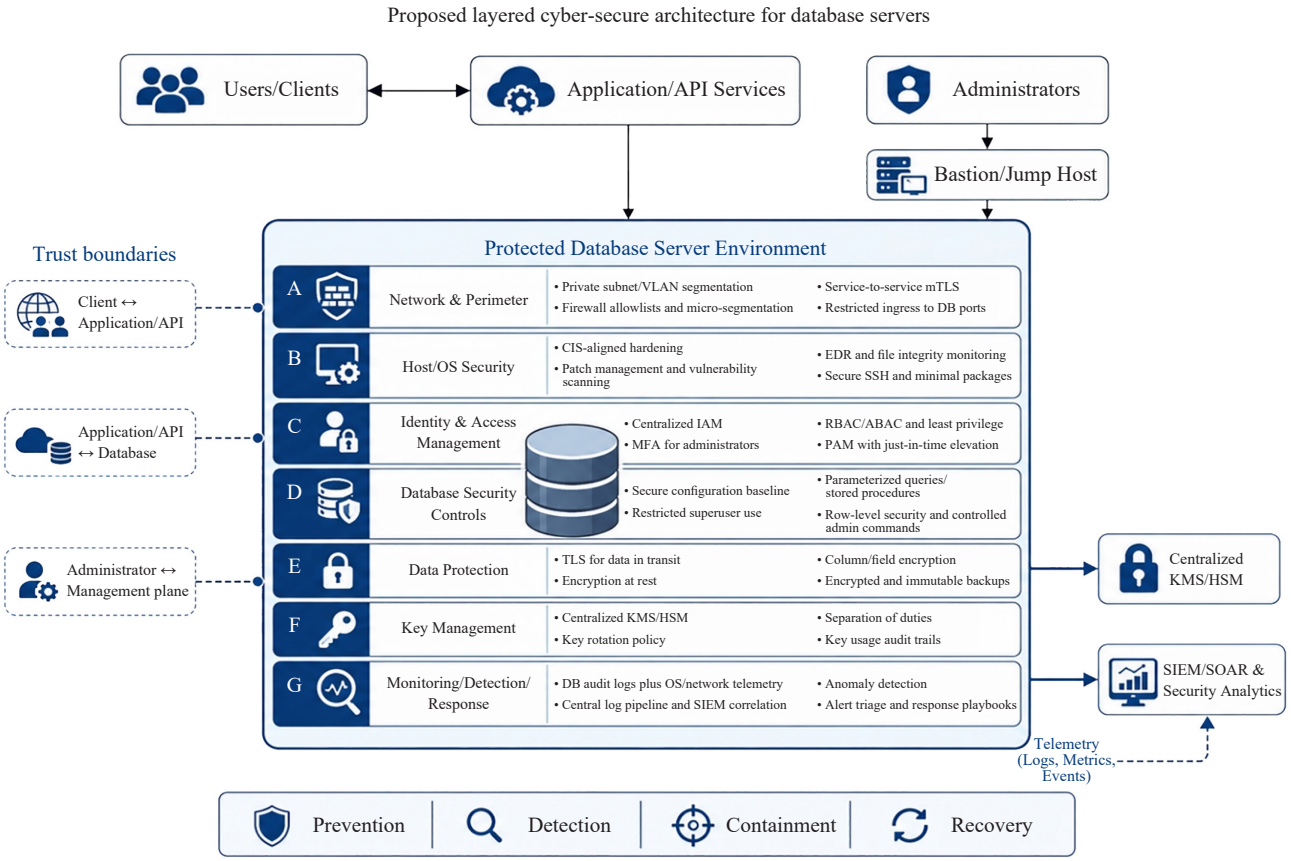

Figure 1. Proposed layered cyber secured architecture for database systems
Note: VLAN: Virtual Local Area Network; mTLS: mutual Transport Layer Security; CIS: Center for Internet Security; EDR: Endpoint Detection and Response; SSH: Secure Shell; IAM: Identity and Access Management; DB: Database; OS: Operating System

3.2.2 *Layer B: host and operating system security*

This layer strengthens the underlying computing environment by implementing Center for Internet Security (CIS) benchmark-aligned hardening practices, including secure configuration of Secure Shell (SSH), enforcement of strict file permissions, and minimization of installed software packages. A continuous patch management process is maintained to address known vulnerabilities, complemented by automated vulnerability scanning tools. Additionally, Endpoint Detection and Response (EDR) solutions provide real-time monitoring of host-level activities, while File Integrity Monitoring (FIM) ensures that critical database binaries, configuration files, and system paths remain unaltered. Features such as secure boot mechanisms further enhance system integrity. Collectively, these controls reduce the risk of host compromise and privilege escalation.

3.2.3 *Layer C: identity and access management*

Identity-centric security is a foundational component of architecture. This layer integrates centralized IAM systems to manage authentication and user lifecycle processes across all services. MFA is enforced for administrative accounts to mitigate credential-based attacks. Fine-grained authorization is implemented using RBAC and ABAC control access models, adhering to the principle of least privilege for both human users and service accounts. The elimination of shared credentials further enhances accountability. A PAM access management framework supports just-in-time privilege elevation, session monitoring, and approval workflows for sensitive operations. These mechanisms collectively address risks related to identity compromise, privilege misuse, and unauthorized access.

3.2.4 *Layer D: database security controls*

The database layer incorporates secure configuration baselines designed to minimize inherent vulnerabilities. This includes restricting superuser privileges, enforcing strong authentication policies, and disabling unnecessary extensions or features. Application interaction with the database is constrained through secure query practices, such as parameterized queries and controlled use of stored procedures, reducing susceptibility to injection attacks. Where supported, row-level security mechanisms are implemented to ensure data isolation in multi-tenant or sensitive environments. Administrative operations are tightly governed, limiting the potential for misuse. This layer directly mitigates database-specific threats such as SQL injection, privilege escalation, and unauthorized data manipulation.

3.2.5 *Layer E: data protection*

Data protection mechanisms ensure confidentiality and integrity across both data-at-rest and data-in-transit states. All communications involving database interactions are secured using TLS, preventing interception and tampering. At rest, encryption techniques such as TDE or full-disk encryption are deployed to safeguard stored data. For highly sensitive attributes, column-level encryption and tokenization provide an additional layer of protection. Backup strategies emphasize encrypted storage, immutability, and regular restoration testing, ensuring resilience against ransomware and data loss scenarios. This layer addresses risks related to data exposure, interception, and unauthorized disclosure.

3.2.6 *Layer F: key management*

Effective cryptographic protection relies on robust key management practices. This layer utilizes centralized KMS or HSMs to securely generate, store, and manage encryption keys. Policies enforce regular key rotation, strict access controls, and separation of duties, reducing the likelihood of key compromises. Architecture explicitly prohibits hard-coded secrets in application artifacts, promoting secure handling of credentials. Additionally, emergency key revocation mechanisms are established to enable rapid containment in the event of a breach. This layer ensures that encryption controls remain reliable and resistant to compromise.

3.2.7 *Layer G: monitoring, detection, and response*

The innermost operational layer focuses on continuous monitoring and adaptive threat detection. Database audit logs, along with host and network telemetry, are aggregated into a centralized logging pipeline. These logs are analyzed using Security Information and Event Management (SIEM) systems, which apply correlation rules to detect known attack patterns such as brute-force attempts, privilege changes, and anomalous query behavior. Advanced anomaly detection techniques further identify previously unseen threats by analyzing deviations from baseline activity patterns. Incident response is guided by structured workflows incorporating severity scoring and predefined playbooks, enabling rapid containment actions such as credential revocation, system isolation, and restoration from verified backups. This layer ensures timely detection, investigation, and mitigation of security incidents.

3.2.8 *Integrated security perspective*

A key strength of the proposed architecture lies in its integrated and telemetry-driven design. Each layer not only enforces preventive controls but also generates actionable security data that feeds into higher-level monitoring and analytics. The architecture enforces segmentation across client, application, database, and administrative domains, ensuring that trust is continuously validated rather than implicitly assumed. By combining layered controls with centralized analytics and response orchestration, the model provides a comprehensive framework capable of addressing both known and emerging threats while supporting operational scalability.

3.3 *Algorithms and detection logic*

The proposed detection mechanism is based on a composite risk-scoring model that integrates multiple contextual and behavioral indicators to systematically evaluate the security posture of database activities. The composite score is

operationalized using following four measurable parameters, each representing a distinct dimension of risk associated with database access and operations.

1. Asset sensitivity is derived from the organization's data classification policy and reflects the criticality of the accessed database object. Data assets are categorized based on their business impact and regulatory importance; for instance, public or operational datasets are assigned relatively low sensitivity scores, whereas tables containing PIIs, financial data, authentication credentials, or other regulated information are assigned higher scores due to their potential impact in the event of compromise. This parameter ensures that risk assessment is aligned with organizational priorities and data value.

2. Privilege level represents the degree of access granted to a user or service and is computed using identity and access management mechanisms, including IAM, RBAC/ABAC models, PAM, and native database role assignments. Lower scores are assigned to read-only or minimally privileged accounts, while higher scores correspond to elevated roles such as database administrators, superusers, or privileged service accounts. This differentiation reflects the increased potential for systemic impact when high-privilege accounts are compromised or misused.

3. Anomaly score captures deviations from established behavioral baselines and is generated through the analysis of database audit logs, operating system logs, network telemetry, and SIEM-based detection outputs. Key indicators include repeated authentication failures, unusual access times, abnormal query frequencies, bulk data extraction patterns, access to previously unused tables, unauthorized privilege modifications, and execution of destructive or high-risk commands. This parameter enables the identification of both known attack patterns and previously unseen anomalous activities.

4. Evidence confidence quantifies the reliability of detected signals based on the degree of correlation across heterogeneous data sources. Isolated or weak signals, such as a single anomalous log entry, result in lower confidence scores, whereas multiple corroborated events across database logs, host telemetry, privileged access logs, and network monitoring systems yield higher confidence values. This mechanism reduces false positives and improves the robustness of detection decisions.

The final composite risk score is normalized on a scale of 0–100 and computed using the following weighted formulation where the *CR_Score, Asset_Sen, Pr_Level, A_Score, Evi_Confidence* represent Computing Risk Score, Asset Sensitivity, Privilege Score, Evidence Confidence, respectively.

$$CR\_Score = 100 \times [0.30 \times Asset\_Sen) + (0.20 \times Pr\_Level) + (0.30 \times A\_Score) + (0.20 \times Evi\_Confidence)]$$

The weighting scheme assigns greater importance to asset sensitivity and anomaly score, reflecting their direct association with business impact and active threat behavior. In contrast, privilege level and evidence confidence serve as complementary factors that enhance prioritization accuracy and mitigate the likelihood of false-positive escalation. This integrated scoring model facilitates risk-aware alert prioritization, efficient incident triage, and automated response orchestration, thereby strengthening the overall detection and response capability of the proposed architecture.

3.4 *Experimental setup, evaluation strategy, and analysis framework*

To validate the proposed architecture and assess its effectiveness under controlled conditions, a functional prototype was implemented within a laboratory-scale environment and a cloud-based Virtual Private Cloud (VPC). The prototype utilizes a widely adopted Relational Database Management System (RDBMS), such as PostgreSQL, MySQL, or Microsoft SQL Server, deployed on hardened Linux virtual machines configured according to established security best practices. This controlled setup facilitates systematic experimentation, reproducibility, and rigorous benchmarking under both normal operational workloads and simulated adversarial scenarios.

3.4.1 *Operational domains*

The prototype architecture is organized into three primary operational domains. First, the Application/API layer emulates real-world user interactions by generating legitimate database queries, thereby replicating realistic workload characteristics. Second, a logically isolated administrative management plan is established, incorporating secure access mechanisms such as bastion (jump) hosts and PAM solutions to enforce strict control and monitoring of

administrative activities. Third, a centralized logging and monitoring layer is integrated to aggregate telemetry across all system components. This layer includes database audit extensions, log forwarding mechanisms, SIEM platforms, and integration with centralized KMS to ensure secure handling of cryptographic operations and comprehensive visibility.

3.4.2 *Evaluation intention*

For evaluation purposes, two distinct configurations are instantiated: (i) a baseline environment, representing conventional deployments with minimal security configurations, and (ii) a secured environment, in which the completely proposed layered cyber-secure architecture is implemented. This comparative design enables systematic assessment of improvements in security effectiveness, detection capability, and operational overhead. Consistency in workload generation and attack execution across both environments ensures a controlled and unbiased evaluation of the proposed model.

3.4.3 *Evaluation strategies*

The evaluation strategy involves the execution of six representative attack scenarios: (1) credential brute-force and credential stuffing attacks; (2) privilege escalation through role misuse and misconfiguration; (3) data exfiltration patterns, including large-scale queries and staged exports; (4) SQL injection through application-layer simulation; (5) ransomware-like destructive behaviors such as mass updates, deletions, or encryption patterns; and (6) insider-style bulk data export using legitimate credentials. These scenarios are executed repeatedly to capture variability and to evaluate both preventive and detective capabilities of the system.

3.4.4 *Evaluation metrics*

A comprehensive set of metrics is employed to quantify system performance across three dimensions. Security effectiveness is measured as detection rate (recall), precision, false positive rate, F1-score, Time-To-Detect (TTD), and Time-To-Contain (TTC). Performance metrics include query response times (p50/p95), throughput (transactions or queries per second), resource utilization (Central Processing Unit (CPU), memory, storage), and telemetry or log volume. Reliability metrics comprise log completeness (event loss), backup integrity through restoration success, and Recovery Time Objective (RTO) under destructive attack conditions. These metrics are collected through database performance views, system monitoring tools, and SIEM event timelines.

3.4.5 *Descriptive and inferential analysis*

The analysis employs both descriptive and inferential statistical techniques to compare baseline and secure deployments. Mean and median differences, along with 95% confidence intervals, are computed, with bootstrapping applied where appropriate. In cases where normality assumptions are not satisfied, non-parametric statistical tests such as the Mann–Whitney U test are utilized, while parametric tests (e.g., t-tests) are applied where applicable. Security metrics are computed per attack scenario and average across multiple runs. Results are reported using both absolute improvements (e.g., increase in detection rate) and relative measures (e.g., percentage reduction in successful attacks).

Finally, the evaluation framework emphasizes the analysis of trade-offs between security effectiveness and operational overhead, particularly in terms of latency, throughput, and system resource consumption. This enables a balanced and evidence-based assessment of the practical deployability and scalability of the proposed architecture in real-world database environments.

4. Experimental results

The performance and effectiveness of the proposed layered cyber-secure architecture were evaluated through a controlled experimental setup, comparing baseline database deployment with the enhanced model across realistic workloads and diverse attack scenarios.

4.1 *Prototype testbed and workload profile*

The experimental evaluation was conducted using a controlled testbed environment comparing a baseline database deployment with the proposed cyber-secure architecture. As summarized in Table 2, both environments utilized identical hardware and Database Management System (DBMS) configurations, while differing in security controls, including segmentation, identity management, encryption, and monitoring mechanisms.

Table 2. Testbed configuration (baseline vs secured)

Component	Baseline Deployment	Proposed Cyber Secure Model Deployment
DBMS	PostgreSQL 15	PostgreSQL 15
Operating System (OS)	Ubuntu Server 22.04 Long Term Support (LTS)	Ubuntu Server 22.04 LTS (CIS-hardened)
Compute	8 vCPU, 32 GB RAM, 500 GB NVMe	8 vCPU, 32 GB RAM, 500 GB NVMe
Network	Flat subnet, DB reachable from app subnet	Segmented network, Database (DB) in private subnet; allowlist; mTLS app↔DB
Authentication	Password-only admin	MFA for admins + PAM ((Just-In-Time) JIT elevation)
Access Control	Basic roles	RBAC + ABAC tags for sensitive tables; least privilege for service accounts
Encryption	TLS optional, disk encryption disabled	TLS enforced; encryption at rest; encrypted backups
Key Handling	Local secrets/config	Centralized KMS (rotation + audit trails)
Auditing	Default DB logs	Full DB audit logging + OS/FIM + centralized SIEM
Detection/Response	Minimal alerts	SIEM correlation + anomaly detection + response playbooks

Table 3 presents the workload characteristics used for benchmarking. The dataset represents realistic, medium-scale transactional system with mixed workloads, high concurrency, and repeated trials to ensure statistical reliability and consistency across experiments.

Table 3. Workload and dataset used in benchmarking

Item	Value
Dataset size	50 GB
Records (customer table)	10,000,000 rows
Records (transactions table)	120,000,000 rows
Concurrent clients	256
Workload mix	70% reads, 25% writes, 5% admin/maintenance
Benchmark duration per trial	30 minutes (steady-state after 5-minute warm-up)
Trials per configuration	10

4.2 *Security outcomes per attack scenario*

A total of 120 attack executions were performed across six representative threat scenarios, supplemented by 240 benign intervals to evaluate false-positive behavior. Table 4 demonstrates that the proposed model significantly reduced successful attacks across all scenarios. Credential brute-force attacks were completely prevented, while other attacks such as SQL injection, data exfiltration, and ransomware-like behaviors showed substantial reductions in both success rate and operational impact. These results confirm that the layered architecture effectively limits attack propagation and reduces potential damage even when partial compromise occurs.

Table 4. Attack success and impact (baseline vs proposed model)

Scenario (20 runs each)	Successful Attack Runs		Impact Metric	Baseline Impact (Median)	Proposed Impact (Median)	Impact Reduction
	Baseline Model	Proposed Model				
Credential brute force (admin login)	12/20	0/20	Time to compromise	11.8 min	—	100% prevented
Privilege escalation (role abuse)	14/20	2/20	Unauthorized privilege gain	Superuser in 9.1 min	Limited role in 14.6 min	86% fewer successes
Data exfiltration pattern (large SELECT/export)	16/20	3/20	Rows extracted	2.6 M rows	120 k rows	95.4% reduction
SQL injection via app simulation	15/20	1/20	Sensitive tables accessed	6 tables	1 table	93.3% reduction
Ransomware-like destructive queries	10/20	1/20	Tables damaged/ dropped	9 tables	1 table	88.9% reduction
Insider bulk export (legit creds)	18/20	6/20	Export volume	3.2 M rows	250 k rows	92.2% reduction

4.3 *Detection performance*

Detection effectiveness was evaluated by measuring the system's ability to generate high-confidence alerts during attack windows. As shown in Table 5, the proposed model achieved consistently high recall and balanced precision across scenarios, with particularly strong performance in brute-force and destructive attack detection.

Table 5. Detection metrics by scenario (proposed model only)

Scenario	Precision	Recall	F1-score	False Positives per 10 benign windows	Median Time-To-Detect (TTD)
Credential brute force	0.91	1.00	0.95	0.3	18 sec
Privilege escalation	0.84	0.90	0.87	0.6	1.9 min
Data exfiltration	0.79	0.85	0.82	0.9	2.7 min
SQL injection path	0.86	0.95	0.90	0.4	34 sec
Ransomware-like destructive	0.93	0.90	0.91	0.2	22 sec
Insider bulk export	0.72	0.80	0.76	1.1	3.6 min

The aggregated confusion matrix in Table 6 indicates an overall precision of 0.835, recall of 0.933, and F1-score of 0.882, demonstrating robust detection capability with a manageable false positive rate (0.092). Table 7 further shows that detection relied primarily on SIEM correlation rules (61%), complemented by anomaly detection mechanisms (39%), highlighting the effectiveness of combining rule-based and behavior-based approaches.

Table 6. Overall confusion matrix (all scenarios aggregated)

	Predicted Attack	Predicted Benign
Actual Attack windows (n = 120)	(True Positive) TP = 112	(False Negative) FN = 8
Actual Benign windows (n = 240)	(False Positive) FP = 22	(True Negative) TN = 218

Table 7. Detection sources (share of true-positive alerts)

Detection source	Share of TP alerts	Typical triggers
SIEM correlation rules	61%	brute force bursts, privilege changes, DROP/ALTER spikes, audit disable attempts
Anomaly detection score	39%	unusual table access, export volume deviation, time-of-day outliers, query burstiness

4.4 *Time-to-contain (response) outcomes*

Response effectiveness was assessed using the time required to execute containment actions after detection. As shown in Table 8, the proposed model enabled rapid containment across scenarios, ranging from 2.4 minutes for credential attacks to 13.2 minutes for insider threats. Containment actions—including credential revocation, session termination, host isolation, and backup restoration—successfully minimizing residual impact in all cases.

These findings demonstrate that timely detection combined with structured response playbooks significantly reduces operational damage.

Table 8. Response outcomes (proposed model)

Scenario	Median Time-To-Contain (TTC)	Containment action–most often used	Residual impact after containment
Credential brute force	2.4 min	account lock + (Internet Protocol) IP block + MFA enforcement	No compromise
Privilege escalation	7.8 min	revoke elevated token + rollback role grants	Minor config changes only
Data exfiltration	10.6 min	export throttle + session terminate + key rotation	Partial export (≤ 120 k rows median)
SQL injection	5.1 min	block endpoint + rotate app creds + Web Application Firewall (WAF) rule	Minimal table exposure
Ransomware-like destructive	4.0 min	isolate host + stop DB service + restore check	1 table median affected
Insider bulk export	13.2 min	PAM session stop + export block + review	Partial export (≤ 250 k rows median)

4.5 *Performance overhead*

The introduction of layered security controls resulted in measurable but controlled performance overhead. Table 9 shows latency increases of approximately 20–28% (p95) across query types, while Table 10 indicates a 15.9% reduction in throughput and moderate increases in CPU, memory, and I/O utilization.

Despite these overheads, system performance remains within acceptable operational limits for secure database environments.

Table 9. Query latency (milliseconds) baseline vs proposed model

Query type	Baseline p50	Baseline p95	Proposed p50	Proposed p95	p95 Overhead
Point SELECT (by id)	6.2	18.4	7.4	22.1	+ 20.1%
Range SELECT (last 30 days)	14.8	45.6	17.9	58.2	+ 27.6%
INSERT transaction	10.6	32.7	12.8	39.1	+ 19.6%
UPDATE status	9.8	29.4	11.7	35.8	+ 21.8%
Bulk export (COPY/SELECT large)	62.3	210.5	71.6	256.8	+ 22.0%

Table 10. Throughput and system overhead

Metric	Baseline	Proposed Model	Change
Throughput (Queries Per Second (QPS)/Transactions Per Second (TPS) equivalent)	6,800	5,720	− 15.9%
DB CPU utilization (avg)	52%	63%	+ 11 pp
DB RAM utilization (avg)	19.2 GB	21.6 GB	+ 2.4 GB
Disk write Input/Output Operations Per Second (IOPS) (avg)	2,100	2,980	+ 41.9%
Network egress from DB	120 Mbps	145 Mbps	+ 20.8%

4.6 *Monitoring and logging overhead*

Enhanced auditing and monitoring significantly increased telemetry generation. As illustrated in Table 11, total log volume increased from 5.0 GB/day to 25.0 GB/day, with corresponding growth in SIEM event ingestion. This increase reflects improved visibility and forensic readiness, although it necessitates careful storage and retention planning.

Table 11. Log volume and retention impact

Item	Baseline	Proposed Model
DB logs generated	4.2 GB/day	18.6 GB/day
OS + FIM logs	0.8 GB/day	6.4 GB/day
Total security telemetry	5.0 GB/day	25.0 GB/day
SIEM events ingested	0.9 M/day	4.8 M/day
Retention configured	14 days	30 days (hot 7, warm 23)

4.7 *Reliability and recovery*

The proposed model demonstrated superior reliability and recovery capabilities under destructive attack scenarios. Table 12 shows that encrypted and immutable backups achieved a 100% restoration success rate, compared to 85% in the baseline, while reducing Recovery Time Objective (RTO) from 4.6 hours to 1.3 hours.

These improvements highlight the importance of secure backup design and key management in enhancing system resilience.

Table 12. Backup and recovery results (destructive/ransomware scenario)

Metric	Baseline	Proposed Model
Backup encryption	No	Yes
Immutable backup copies	No	Yes (Write Once, Read Many (WORM)/immutable retention)
Restore success rate (20 trials)	17/20 (85%)	20/20 (100%)
Median Recovery Time Objective (RTO)	4.6 hours	1.3 hours
Median Recovery Point Objective (RPO)	45 minutes	10 minutes
Post-restore integrity check failures	2 trials	0 trials

4.8 *Ablation study*

An ablation analysis was conducted to evaluate the contribution of individual layers to overall system effectiveness. As shown in Table 13, removing critical components such as Identity (Layer C) and Monitoring/Response (Layer G) significantly degraded performance, with the latter causing a substantial drop in recall (0.41) and increased time-to-detect.

This confirms that the proposed architecture derives its strength from tight integration of all layers, rather than reliance on isolated controls.

Table 13. Ablation impact (security + detection)

Configuration	Successful attack runs (out of 60)	Overall Recall	Precision	Median TTD
Full model (A–G)	6	0.93	0.84	1.2 min
Without Layer C (Identity/PAM/MFA)	18	0.89	0.81	1.4 min
Without Layer E (Data protection)	15	0.92	0.83	1.3 min
Without Layer F (KMS/HSM governance)	13	0.92	0.82	1.3 min
Without Layer G (Monitoring/Response)	24	0.41	0.77	6.8 min
Without Layer A (Segmentation/mTLS)	16	0.88	0.82	1.5 min

5. Discussion

This section interprets the experimental findings in relation to the research questions (RQ1–RQ3), highlighting the effectiveness, efficiency, and practical implications of the proposed layered cyber-secure architecture. While the proposed framework increases daily security telemetry from approximately 5 GB to 25 GB, the associated storage cost remains modest. Based on current cloud object storage pricing (≈ $0.023 per GB/month), the additional 20 GB/day with 30-day retention translates to roughly $8–15 per month per database server. This cost can be further optimized through log sampling, compression, and tiered storage policies without significantly affecting detection capabilities.

5.1 *Impact on attack success and damage mitigation (RQ1)*

RQ1: To what extent does the proposed model reduce the impact of successful attacks compared to baseline deployment?

The results provide strong empirical evidence that the proposed model significantly reduces both the likelihood of successful attacks and their resulting impact. As demonstrated in Table 3, complete prevention was achieved in the case of credential brute-force attacks (0/20 successful runs), while other attack scenarios exhibited substantial reductions in success rates, including SQL injection (93.3%), data exfiltration (95.4%), and ransomware-like destructive activities (88.9%). Even in scenarios where attacks were not fully prevented—such as insider-enabled bulk exports—the architecture effectively constrained the magnitude of damage, reducing extracted data volumes by over 92%.

This reduction can be attributed to the layered defense strategy, where controls across network segmentation, identity governance, database configuration, and data protection collectively limit adversarial progression. Notably, the ablation study (Table 12) further confirms that the absence of critical layers, particularly identity management (Layer C) and monitoring/response (Layer G), significantly increases attack success rates. This highlights that security effectiveness arises from the interdependency of layers, rather than isolated control mechanisms.

Overall, the findings confirm that the proposed architecture not only prevents a substantial proportion of attacks but also limits the blast radius of successful compromises, thereby enhancing system resilience.

5.2 *Performance overhead and operational trade-offs (RQ2)*

RQ2: What is the performance overhead introduced by the additional security controls in terms of latency, throughput, and resource utilization?

The implementation of comprehensive security controls introduces measurable but manageable performance overhead. As shown in Table 8, query latency increased by approximately 20–28% at the p95 level, while Table 9 indicates a throughput reduction of 15.9% and moderate increases in CPU, memory, disk I/O, and network utilization.

In addition to running time overhead, monitoring and auditing controls significantly increased telemetry volume (Table 10), with total log generation rising from 5.0 GB/day to 25.0 GB/day. While this increase enhances visibility and forensic readiness, it also necessitates scalable storage and efficient log management strategies.

Despite these overheads, the system remains operationally feasible for production-like environments. Importantly, the trade-off between performance and security is justified by the substantial improvements observed in attack prevention, detection, and recovery. Moreover, performance degradation remains within acceptable thresholds for most enterprise workloads, particularly where security assurance is prioritized over marginal performance gains.

Thus, the findings indicate that the proposed architecture achieves a balanced trade-off, where enhanced security controls impose overhead that is quantifiable, predictable, and practically manageable through tuning and resource scaling.

5.3 *Detection effectiveness and monitoring accuracy (RQ3)*

RQ3: How effectively can monitoring mechanisms detect anomalous or malicious database activities?

The proposed detection framework demonstrates high effectiveness and reliability across multiple attack scenarios. As reported in Table 4, recall values range from 0.80 to 1.00 across scenarios, indicating strong capability in identifying malicious activities. The aggregated results (Table 5) show an overall recall of 0.933 and precision of 0.835, resulting in

an F1-score of 0.882. These metrics reflect a well-balanced detection system with high sensitivity and controlled false-positive rates (False Positive Rate (FPR) = 0.092).

Time-based performance is equally significant; median Time-To-Detect (TTD) ranges from seconds (18 seconds for brute-force attacks) to a few minutes for more complex anomalies such as insider activity (3.6 minutes). This rapid detection enables timely response, which is further validated by containment results (Table 7).

An important insight from Table 6 is the complementary role of SIEM correlation rules (61%) and anomaly detection mechanisms (39%). Rule-based detection provides robustness against known attack patterns, while anomaly-based approaches enhance visibility into previously unseen or insider-driven threats. This hybrid detection strategy contributes to improved overall coverage and reduced reliance on any single detection technique.

However, relatively lower precision in insider scenarios suggests that behavioral ambiguity remains a challenge, indicating potential areas for further refinement, such as adaptive baselining or user behavior analytics.

5.4 *Integrated perspective and practical implications*

Collectively, the findings demonstrate that proposed architecture delivers substantial improvements in security effectiveness, robust detection capabilities, and resilient recovery mechanisms, while maintaining operational feasibility. The results validate the importance of integrating preventive, detective, and responsive controls into a unified architecture rather than relying on isolated security measures.

The ablation analysis further underscores that monitoring and response capabilities are indispensable, with their removal leading to significant degradation in detection accuracy and response time. This reinforces the argument that modern database security must extend beyond prevention to include continuous monitoring and automated response.

From a practical standpoint, organizations adopting the proposed model must carefully consider resource allocation, telemetry management, and performance tuning to optimize the balance between security and efficiency. Nonetheless, the demonstrated reduction in attack success and impact strongly supports the viability of deploying such layered architectures in cloud, hybrid, and on-premises environments.

6. Conclusion and future work

In conclusion, this study has presented and empirically validated a layered cyber-secure architecture for database systems that integrate preventive, detective, and governance controls within a unified framework. Through a design science–driven prototype and controlled experimental evaluation, the findings demonstrate that the proposed model significantly reduces attack success rates and limits the impact of successful compromises across multiple realistic threat scenarios, thereby enhancing overall system resilience. The architecture further achieves high detection effectiveness, with strong recall, balanced precision, and rapid response capabilities, supported by a hybrid detection approach combining rule-based correlation and anomaly analysis. Although the integration of comprehensive security controls introduces measurable overhead in terms of latency, throughput, and telemetry volume, these trade-offs remain operationally feasible and justifiable given the substantial improvements in security assurance, visibility, and recovery performance. The results also highlight the importance of tightly coupled layers—particularly identity governance and monitoring/response—in achieving robust protection, as confirmed by the ablation analysis. Overall, the study contributes to practical, testable, and scalable reference architecture that addresses a critical gap in the literature by moving beyond fragmented security measures toward an integrated, end-to-end database protection model suitable for cloud, hybrid, and on-premises environments. Future research should also explore more advanced human-in-the-loop mechanisms versus fully automated response playbooks. While automation can reduce dwell time, our findings suggest that high-severity actions (e.g., host isolation or mass credential revocation) benefit from human approval to avoid unintended operational impact.

Conflict of interest

The authors declare no competing financial interest.

References

[1] Golightly L, Modesti P, Garcia R, Chang V. Securing distributed systems: A survey on access control techniques for cloud, blockchain, IoT and SDN. *Cyber Security and Applications.* 2023; 1: 100015. Available from: https://doi.org/10.1016/j.csa.2023.100015.

[2] Adahman Z, Malik AW, Anwar Z. An analysis of zero-trust architecture and its cost-effectiveness for organizational security. *Computers & Security.* 2022; 122: 102911. Available from: https://doi.org/10.1016/j.cose.2022.102911.

[3] Paul A, Sharma V, Olukoya O. SQL injection attack: Detection, prioritization & prevention. *Journal of Information Security and Applications.* 2024; 85: 103871. Available from: https://doi.org/10.1016/j.jisa.2024.103871.

[4] Lu D, Fei J, Liu L. A semantic learning-based SQL injection attack detection technology. *Electronics.* 2023; 12(6): 1344. Available from: https://doi.org/10.3390/electronics12061344.

[5] Buck C, Olenberger C, Schweizer A, Völter F, Eymann T. Never trust, always verify: A multivocal literature review on current knowledge and research gaps of zero trust. *Computers & Security.* 2021; 110: 102436. Available from: https://doi.org/10.1016/j.cose.2021.102436.

[6] Loureiro S. Security misconfigurations and how to prevent them. *Network Security.* 2021; 5: 13-16. Available from: https://doi.org/10.1016/S1353-4858(21)00053-2.

[7] Carvalho M, Sá F, Bernardino J. Evaluation of the impact of AES encryption on query read performance across Oracle, MySQL, and SQL Server databases. *Cryptography*. 2025; 9(4): 77. Available from: https://doi.org/10.3390/cryptography9040077.

[8] Hedabou M. Cloud key management based on verifiable secret sharing. In: Yang M, Chen C, Liu Y. (eds.) *Network and System Security (NSS 2021) Lecture Notes in Computer Science*. Heidelberg, Berlin: Springer Cham; 2022. p.289-303. Available from: https://doi.org/10.1007/978-3-030-92708-0_18.

[9] Gao X, Yuan P, Han S, Liu Y, Xia X, Zhang H, et al. Insights into KPI-based performance anomaly detection in database systems: A comprehensive study. *Expert Systems with Applications.* 2025; 285: 127959. Available from: https://doi.org/10.1016/j.eswa.2025.127959.

[10] Inayat U, Farzan M, Mahmood S, Zia MF, Hussain S, Pallonetto F. Insider threat mitigation: Systematic literature review. *Ain Shams Engineering Journal.* 2024; 15(12): 103068. Available from: https://doi.org/10.1016/j.asej.2024.103068.

[11] Li W, Feng Y, Liu N, Li Y, Fu X, Yu Y. A secure and efficient log storage and query framework based on blockchain. *Computer Networks.* 2024; 252: 110683. Available from: https://doi.org/10.1016/j.comnet.2024.110683.

[12] Morillo Reina JD, Mateo Sanguino TJ. Decentralized and secure blockchain solution for tamper-proof logging events. *Future Internet.* 2025; 17(3): 108. Available from: https://doi.org/10.3390/fi17030108.

[13] Tian T, Zhang C, Jiang B, Feng H, Lu Z. Insider threat detection for specific threat scenarios. *Cybersecurity*. 2025; 8(1): 17. Available from: https://doi.org/10.1186/s42400-024-00321-w.

[14] Yu W, Yin Q, Yin H, Xiao W, Chang T, He L, et al. A systematic review on password guessing tasks. *Entropy.* 2023; 25(9): 1303. Available from: https://doi.org/10.3390/e25091303.

[15] Begovic K, Al-Ali A, Malluhi Q. Cryptographic ransomware encryption detection: Survey. *Computers & Security.* 2023; 132: 103349. Available from: https://doi.org/10.1016/j.cose.2023.103349.

[16] Nobles C. Investigating cloud computing misconfiguration errors using the human factors analysis and classification system. *Scientific Bulletin.* 2022; 27(1): 59-66. Available from: https://doi.org/10.2478/bsaft-2022-0007.

[17] Dissanayake N, Jayatilaka A, Zahedi M, Babar MA. Software security patch management-A systematic literature review of challenges, approaches, tools and practices. *Information and Software Technology*. 2022; 144: 106771. Available from: https://doi.org/10.1016/j.infsof.2021.106771.

[18] Arshad H, Johansen C, Owe O. Semantic attribute-based access control: A review on current status and future perspectives. *Journal of Systems Architecture.* 2022; 129: 102625. Available from: https://doi.org/10.1016/j.sysarc.2022.102625.

[19] Glöckler J, Sedlmeir J, Frank M, Fridgen G. A systematic review of identity and access management requirements in enterprises and potential contributions of self-sovereign identity. *Business & Information Systems Engineering.* 2024; 66: 421-440. Available from: https://doi.org/10.1007/s12599-023-00830-x.

[20] Alruwies MH, Mishra S, AlShehri MAR. Identity governance framework for privileged users. *Computer Systems Science and Engineering.* 2022; 40(3): 995-1005. Available from: https://doi.org/10.32604/csse.2022.019355.

[21] López Velásquez JM, Martínez Monterrubio SM, Sánchez Crespo LE, Garcia Rosado D. Systematic review of SIEM technology: SIEM-SC birth. *International Journal of Information Security.* 2023; 22(3): 691-711. Available from: https://doi.org/10.1007/s10207-022-00657-9.

[22] Lee M, Jang-Jaccard J, Kwak J. Novel architecture of security orchestration, automation and response in internet

of blended environment. *Computers, Materials & Continua.* 2022; 73(1): 199-223. Available from: https://doi.org/10.32604/cmc.2022.028495.
[23] Yeoh W, Liu M, Shore M, Jiang F. Zero trust cybersecurity: Critical success factors and a maturity assessment framework. *Computers & Security*. 2023; 133: 103412. Available from: https://doi.org/10.1016/j.cose.2023.103412.
[24] Islam MT, Mission MR, Refat TK, Kynatun M. Cybersecurity risk assessment frameworks for engineering databases: A systematic literature review. *Strategic Data Management and Innovation.* 2025; 2(1): 224-243. Available from: https://doi.org/10.71292/sdmi.v2i01.22.