International Journal of Research and Applied Innovations (IJRAI)

| ISSN: 2455-1864 | www.ijrai.org | editor@ijrai.org | A Bimonthly, Scholarly and Peer-Reviewed Journal |

||Volume 7, Issue 5, September–October 2024||

DOI:10.15662/IJRAI.2024.0705004

Oracle-Driven Cybersecurity and Real-Time ERP Automation: Zero-Downtime BMS and the EDAS Method

Malek Bashir Al-Amin

Site Reliability Engineer, Libya

ABSTRACT: Enterprise Resource Planning (ERP) platforms underpin core business operations — finance, procurement, HR, and supply chain — and many organizations now run automated, always-on ERP processes in cloud environments. While automation increases speed and resilience, it also compresses the attack window and amplifies damage from cyber incidents, making business continuity dependent on both rapid detection and automated containment. This paper proposes an Oracle-driven framework that fuses zero-trust principles, Oracle Cloud telemetry and enforcement primitives (database activity monitoring, Data Safe, IAM/adaptive authentication), and streaming AI/ML detection to secure automated ERP operations and preserve business continuity. The framework ingests audit trails, database activity streams, API gateway logs, and identity events into a real-time feature pipeline where sequence-aware and ensemble detectors score activity on sliding windows. High-confidence anomalies trigger policy-driven automated playbooks: graded actions range from soft quarantine and adaptive multi-factor authentication challenges to temporary account suspension and automated workflow rollbacks, with human-in-the-loop gating for high-impact financial operations. The prototype, implemented on an Oracle testbed simulating procure-to-pay and payroll workflows, shows substantial reductions in time-to-detect and time-to-respond compared to baseline static rule engines while preserving auditable evidence and compliance mappings. Key operational challenges identified include model explainability, latency/compute overhead, data-privacy constraints on centralized modeling, and Oracle licensing/cost tradeoffs. The paper therefore recommends a phased, risk-prioritized deployment (start high-value workflows), feature masking for privacy, explainability layers for operator trust, and governance controls that map automated actions to audit evidence. By tightly integrating Oracle native controls and streaming AI detection with policy orchestration, organizations can automate ERP workflows safely and maintain business continuity under fast-moving cyber threats.

KEYWORDS: Oracle ERP Cloud, business continuity, real-time detection, streaming ML, Oracle Data Safe, zero-trust, automated remediation, audit trails, adaptive authentication

I. INTRODUCTION

Modern enterprises rely on ERP systems to coordinate mission-critical processes. As organizations migrate ERP workloads and automation pipelines to cloud platforms, transaction velocity increases and traditional, periodic security controls become insufficient. Automated scripts, API integrations, and bulk workflows can execute many high-value transactions within minutes; if compromised, these flows can inflict rapid operational and financial damage and disrupt business continuity. Consequently, protecting automated ERP platforms requires controls that operate at the same tempo as automation: continuous telemetry ingestion, streaming detection, and policy-driven automated containment close to the data and workflows.

Zero-trust architecture provides the conceptual foundation for this approach: continuous verification of identity and device posture, least-privilege access, and per-request authorization reduce implicit trust and lateral movement inside ERP ecosystems. NIST's Zero Trust Architecture codifies these principles and shows how continuous policy evaluation and enforcement points can be applied across services. Oracle's Cloud stack supplies telemetry and enforcement primitives well suited for an integrated design: Data Safe and database auditing for activity capture and data discovery, IAM and adaptive authentication for identity control, and orchestration APIs for automated remediations. When these primitives are combined with streaming feature engineering and AI/ML detectors that operate on ERP audit logs and DB activity, organizations can detect sequence-based abuse (credential replay, automated invoice fraud, collusive supplier activity) and take automated, auditable containment actions to preserve business continuity. This paper presents an Oracle-centric architecture, prototype evaluation, and operational guidance for deploying such systems.

International Journal of Research and Applied Innovations (IJRAI)

| ISSN: 2455-1864 | www.ijrai.org | editor@ijrai.org | A Bimonthly, Scholarly and Peer-Reviewed Journal |

||Volume 7, Issue 5, September–October 2024||

DOI:10.15662/IJRAI.2024.0705004

II. LITERATURE REVIEW

ERP systems concentrate sensitive business data and high-impact functionality, making them attractive targets for attackers. Early academic work identified recurring ERP weaknesses—misconfiguration, segregation-of-duties lapses, and poor audit hygiene—that create persistent exposures (Grabski et al., 2011). Industry surveys and practitioner reports highlight that common breach vectors include phishing-driven credential theft, insider misuse, and misconfiguration of cloud resources; these vectors persist as ERP moves to cloud and API-driven architectures (SANS, ISACA). As perimeter models erode in multi-tenant and distributed deployments, identity- and data-centric controls along with continuous monitoring become cornerstone defenses.

NIST's Zero Trust Architecture reframes defenses around resources and per-request policy decisions rather than network boundaries, emphasizing continuous verification and least privilege—principles that map directly to ERP security needs because privileged ERP roles affect many business processes. Oracle and other vendors have responded with capabilities for adaptive authentication, activity auditing, data discovery, and masking. Oracle Data Safe (and related DB auditing features) provide unified capabilities for sensitive-data discovery, activity collection, SQL firewalling, and masking — yielding rich signals for detection systems and enforcement points for containment. Leveraging vendor-native telemetry improves signal fidelity relative to peripheral logging alone and enables direct automated enforcement via cloud APIs.

In the research domain, log-based and sequence-aware anomaly detection have matured: recurrent autoencoders, predictive auto-regression, and hybrid ensembles have been used to detect insider misuse and transaction anomalies in ERP and financial systems. These approaches show higher adaptability and recall than static rule engines for sequence-based attacks, but they introduce explainability issues and false positives that operational teams must manage. Surveys of log-based deep learning outline preprocessing, representation, and model classes that work well for heterogeneous audit logs and identify tradeoffs in latency and compute cost.

Operational literature emphasizes several pragmatic constraints: (1) streaming ML systems add compute and latency that can affect transaction throughput if not engineered with bounded windows and selective prioritization; (2) model explainability is essential for operator trust and for auditors when automated actions affect financial records; (3) data-privacy and residency constraints may prohibit centralizing raw PII for model training; and (4) vendor lock-in and licensing affect cost and deployment choices. Hybrid architectures—using vendor native telemetry for enforcement and enriched detection signals, computing derived/masked features in streaming pipelines, and applying ensembles with human checkpoints for high-impact actions—emerge in the literature as the most practical pattern. This study builds on these prior findings to present an Oracle-driven, business-continuity oriented architecture and prototype evaluation.

III. RESEARCH METHODOLOGY

1. **Problem scoping and gap analysis.** Performed a structured review of academic literature, industry reports (ISACA, SANS), and Oracle product documentation to identify specific gaps: the lack of integrated Oracle-centric frameworks that combine streaming AI detection with automated remediation and clear compliance evidence for ERP automation.
2. **Objectives.** Defined measurable objectives: (a) cut time-to-detect (TTD) for high-risk automated transaction anomalies by ≥50% versus a baseline rule engine; (b) enable automated containment (where safe) to reduce time-to-respond (TTR); (c) preserve audit trails and compliance artifacts for every automated action; (d) limit average per-transaction latency impact to <200 ms at target throughput.
3. **Architecture design.** Designed a layered, Oracle-centric architecture: telemetry sources (ERP audit exports, Oracle DB audit/Data Safe activity streams, IAM/adaptive-auth events, API gateway logs) feed a message bus. A streaming feature engine computes behavioral and transactional features over sliding windows. The detection tier runs ensemble models in parallel (statistical baselines, sequence-aware models such as predictive auto-regression and recurrent autoencoders, supervised classifiers where labeled data exist). A policy engine maps confidence and contextual risk scores to graded automated playbooks and human-in-the-loop gates.
4. **Prototype implementation.** Built a proof-of-concept on an Oracle testbed simulating procure-to-pay, payroll, and supplier-onboarding workflows. Used Oracle audit exports/Data Safe for DB activity telemetry, an open-source stream

| ISSN: 2455-1864 | www.ijrai.org | editor@ijrai.org | A Bimonthly, Scholarly and Peer-Reviewed Journal |

||Volume 7, Issue 5, September–October 2024||

DOI:10.15662/IJRAI.2024.0705004

processor for feature computation, and Python microservices for detection models. Playbooks used Oracle IAM and ERP workflow APIs to enact remediations (adaptive MFA triggers, account suspension, workflow rollback).

5. **Dataset generation and labeling.** Generated labeled datasets by simulating normal operations and injecting adversarial scenarios: credential replay, scripted invoice injection, rapid privilege escalation, and collusive supplier fraud. Labels supported supervised components and evaluation; unsupervised detectors used injected anomalies for validation.
6. **Evaluation metrics and experiments.** Measured detection performance (precision, recall, F1), operational KPIs (TTD, TTR), system latency overhead, and compute utilization. Benchmarked against a baseline rule-based detector and conducted stress tests at scaled loads.
7. **Governance and compliance mapping.** Ensured all automated actions produced immutable audit records and human-review checkpoints for high-impact remediations. Mapped automated controls to typical compliance requirements (data masking, retention, and audit evidence collection) and documented escalation policies.
8. **Iterative tuning and validation.** Performed multiple cycles of feature selection, threshold tuning, retraining cadence adjustments, and operator workshops to validate explainability artifacts and refine escalation flows. The methodology balanced engineering rigor, empirical evaluation, and governance to assess feasibility and operational tradeoffs.

Advantages

- Detects sequence-based abuse and collusive fraud patterns that static rules often miss.
- Shortens detection and response windows, improving business continuity under attack.
- Uses Oracle native telemetry and enforcement primitives for high-fidelity signals and direct remediation.
- Enables policy-driven, graded containment actions that can be tuned to business risk.
- Produces auditable evidence trails required for compliance and post-incident analysis.

Disadvantages

- Streaming ML and continuous telemetry introduce compute and latency overhead; careful engineering (bounded windows, selective detection) is required.
- False positives risk operational disruption; human-in-the-loop and soft-quarantine patterns are necessary to limit business impact.
- Explainability requirements add development and operational complexity.
- Data-privacy and residency rules may limit centralization of raw data for modeling; feature masking or federated approaches may be required.
- Advanced Oracle modules and telemetry retention can drive licensing and operational cost, which may constrain adoption for smaller organizations.

IV. RESULTS AND DISCUSSION

In the Oracle testbed prototype, ensemble detection (sequence-aware models + statistical baselines) detected injected insider sequences and automated invoice fraud with higher recall than a static rule engine while maintaining precision above typical operational targets after threshold tuning. Median time-to-detect for high-confidence anomalies dropped by roughly 50–65%, and automated containment playbooks (adaptive MFA, temporary account suspension, or workflow rollback) achieved median time-to-respond under two minutes for high-confidence incidents in test scenarios.

Streaming feature computation added measurable per-transaction latency (ranges observed in tests: ~80–160 ms depending on feature complexity and load). Mitigations—bounding sliding-window sizes, using approximate/streaming aggregates, prioritizing detection for high-risk transaction types, and offloading heavy models to asynchronous scoring for lower-risk events—kept latency within acceptable operational limits in the prototype.

Two operational themes emerged. First, explainability is essential: operators demanded human-readable rationales (feature attributions, sequence excerpts, and similar evidence) before allowing fully automated rollback for financial workflows. Introducing explainability layers and a two-stage containment model (soft quarantine → human release) materially reduced operator pushback. Second, compliance and privacy constrained feature design: sensitive fields were masked or tokenized and only derived behavioral features were centralized. Cost analysis showed telemetry retention, streaming compute, and advanced Oracle licensing as the main cost drivers — suggesting a phased deployment focused on highest-value workflows as the practical adoption path.

| ISSN: 2455-1864 | www.ijrai.org | editor@ijrai.org | A Bimonthly, Scholarly and Peer-Reviewed Journal |

||Volume 7, Issue 5, September–October 2024||

DOI:10.15662/IJRAI.2024.0705004

Overall, the results indicate that an Oracle-driven, streaming-AI approach can materially improve ERP resilience and business continuity when paired with careful engineering, explainability, privacy controls, and governance.

V. CONCLUSION

Securing automated online ERP systems for business continuity requires controls that operate in real time and close to the data. An Oracle-centric architecture that combines zero-trust principles, native telemetry (Data Safe, DB auditing, IAM), streaming feature pipelines, and sequence-aware/ensemble ML detectors enables faster detection and policy-driven automated containment while preserving compliance evidence. Practical deployment demands a phased approach, feature masking for privacy, explainability layers to build operator trust, and governance that maps automated playbooks to auditable trails. With these elements in place, organizations can retain the productivity benefits of ERP automation while substantially strengthening continuity and resilience against modern exploitation paths.

VI. FUTURE WORK

1. Research privacy-preserving collaborative training (federated learning) to enable cross-organization model improvements without sharing raw sensitive records.
2. Investigate graph-neural-network (GNN) approaches for collusive, multi-entity fraud detection across supplier and user graphs.
3. Develop standardized explainability artifacts aligned with auditor evidence requirements for automated remediation decisions.
4. Conduct longitudinal field studies in production Oracle ERP deployments to quantify model drift, retraining cadence, and real-world ROI.
5. Evaluate hybrid, cross-vendor architectures that combine Oracle native enforcement with cloud-agnostic detection layers to reduce vendor lock-in risk.

REFERENCES

1. Grabski, S. V., Leech, S. A., & Schmidt, P. J. (2011). A review of ERP research: A future agenda for accounting information systems. *Journal of Information Systems, 25*(1), 37–78.
2. Rajendran, Sugumar (2023). Privacy preserving data mining using hiding maximum utility item first algorithm by means of grey wolf optimisation algorithm. Int. J. Business Intell. Data Mining 10 (2):1-20.
3. Kindervag, J. (2010). *No more chewy centers: Introducing the zero trust model of information security*. Forrester Research.
4. Rose, S., Borchert, O., Mitchell, S., & Connelly, S. (2020). *Zero Trust Architecture* (NIST Special Publication 800-207). National Institute of Standards and Technology.
5. Oracle Corporation. (2019). *Secure critical data with Oracle Data Safe* (White paper).
6. Oracle Corporation. (2023). *Cybersecurity solutions and best practices to protect your organization* (Oracle white paper).
7. ISACA. (2021). *ERP security and controls* (ISACA Professional Practices Paper).
8. Vinay Kumar Ch, Srinivas G, Kishor Kumar A, Praveen Kumar K, Vijay Kumar A. (2021). Real-time optical wireless mobile communication with high physical layer reliability Using GRA Method. J Comp Sci Appl Inform Technol. 6(1): 1-7. DOI: 10.15226/2474-9257/6/1/00149
9. SANS Institute. (2019). *ERP security: Understanding and mitigating risks* (SANS white paper).
10. Subramanian, G. H. (2017). Cloud ERP implementation and the impact of cloud computing on ERP. *International Journal of Enterprise Information Systems, 13*(4), 21–34.
11. T. Yuan, S. Sah, T. Ananthanarayana, C. Zhang, A. Bhat, S. Gandhi, and R. Ptucha. 2019. Large scale sign language interpretation. In Proceedings of the 14th IEEE International Conference on Automatic Face Gesture Recognition (FG'19). 1–5.
12. Vaidya, S., & Seetharaman, P. (2020). Artificial intelligence applications in ERP systems. *Information Systems Frontiers, 22*(2), 475–491.
13. Yu, J., Kim, M., Oh, H., & Yang, J. (2021). Real-time abnormal insider event detection on enterprise resource planning systems via predictive auto-regression model. *IEEE Access, 9*, 62276–62284.

| ISSN: 2455-1864 | www.ijrai.org | editor@ijrai.org | A Bimonthly, Scholarly and Peer-Reviewed Journal |

||Volume 7, Issue 5, September–October 2024||

DOI:10.15662/IJRAI.2024.0705004

14. Chunduru, V. K., Gonepally, S., Amuda, K. K., Kumbum, P. K., & Adari, V. K. (2022). Evaluation of human information processing: An overview for human-computer interaction using the EDAS method. SOJ Materials Science & Engineering, 9(1), 1–9.
15. Zwilling, M., Lesjak, D., & Kovačič, A. (2020). Cyber security threats and vulnerabilities in ERP systems. *Procedia Computer Science, 176*, 2242–2250.
16. Kumbum, P. K., Adari, V. K., Chunduru, V. K., Gonepally, S., & Amuda, K. K. (2023). Navigating digital privacy and security effects on student financial behavior, academic performance, and well-being. Data Analytics and Artificial Intelligence, 3(2), 235–246.
17. Bakumenko, A., & Aivazian, V. (2022). Detecting anomalies in financial data using machine learning. *Systems, 10*(5), 130.
18. Konda, S. K. (2024). Zero-Downtime BMS Upgrades for Scientific Research Facilities: Lessons from NASA's Infrared Telescope Project. International Journal of Technology, Management and Humanities, 10(04), 84-94.
19. Dong Wang, Lihua Dai (2022). Vibration signal diagnosis and conditional health monitoring of motor used in biomedical applications using Internet of Things environment. Journal of Engineering 5 (6):1-9.
20. Landauer, M. (2023). Deep learning for anomaly detection in log data: A survey. *Journal/Survey* (survey article).
21. NeuroQuantology. (2022). Review on security and privacy of cloud ERP systems. *NeuroQuantology, 20*(15), 303–315.
22. Sangannagari, S. R. (2024). Design and Implementation of a Cloud-Native Automated Certification Platform for Functional Testing and Compliance Validation. International Journal of Technology, Management and Humanities, 10(02), 34-43.
23. Forrester Research. (2021). *The state of zero trust adoption*. Forrester.