International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435 (Online) Volume 15, Issue 07, July 2026)

Zero-Trust Security Architecture for Cloud-Based Information Systems

Sonal R. Pawar[1], Manisha Ankushrao Taur[2], Aarti Suresh Bhujbal[3]

[1,2,3]*Department of Master of Computer Applications, [CAYMET's Siddhant Institute of Computer Application], [Pune], India*

***Abstract*— The widespread adoption of cloud computing has transformed the way organizations deploy and manage information systems, but it has also introduced increasingly complex cyber security risks. Security mechanisms based on trusted network boundaries are no longer sufficient because cloud resources are accessed by users, devices, and applications operating across distributed environments. To address these challenges, this paper presents a layered Zero-Trust Security Architecture (ZTSA) for protecting cloud-based information systems. The proposed framework consists of five interconnected security layers: Identity and Access Management, Policy Decision and Enforcement, Data Security, Network Micro-Segmentation, and Continuous Monitoring and Analytics. The architecture is designed in accordance with the principles of NIST SP 800-207 and is integrated with every stage of the Information System Development Life Cycle (SDLC). A simulated cloud environment is used to evaluate the framework against representative cyber attack scenarios. The findings demonstrate improved threat detection, reduced attack surface, and stronger access control while supporting ISO/IEC 27001 and the NIST Cybersecurity Framework.**

***Keywords*— Cloud Security, Identity and Access Management, Information Assurance, NIST 800-207, Security Architecture, Zero Trust.**

I. Introduction

The rapid migration of organizational information systems to cloud platforms has fundamentally altered the security landscape. Traditional perimeter-based security models, often described as the "castle-and-moat" approach, assume that any entity inside the network boundary can be implicitly trusted. This assumption is no longer valid in distributed cloud environments where users, devices, and services routinely operate outside any fixed network perimeter. Recent industry reports indicate a continued rise in cloud-targeted breaches arising from misconfigured access controls, compromised credentials, and insider threats.

This paper addresses the resulting security gap by proposing a Zero-Trust Security Architecture (ZTSA) specifically designed for cloud-based information systems.

The architecture enforces continuous verification, least-privilege access, and micro-segmentation, and is mapped to each phase of the Information System Development Life Cycle (SDLC) to ensure practical implement ability rather than remaining a purely theoretical model.

The objectives of this work are: (1) to review existing security architectures and identify their limitations for cloud environments; (2) to design a layered Zero-Trust Security Architecture tailored for cloud-hosted information systems; (3) to evaluate the proposed architecture against common cloud attack vectors through a simulated case study; and (4) to provide practical implementation guidelines aligned with recognized standards including NIST SP 800-207 and ISO/IEC 27001.

II. Literature Review

A. Evolution of Information Security Architectures

Information security architecture has evolved from simple perimeter defenses such as firewalls and virtual private networks (VPNs) toward layered, defense-in-depth strategies. Whitman and Mattord [4] establish the foundational principles of information assurance, governance, and policy development that underpin modern security architecture design. Stallings [5], [6] provide the cryptographic and network security mechanisms — symmetric and asymmetric encryption, key management, and secure protocols — that form the technical backbone of any zero-trust implementation.

B. Zero-Trust Principles in Prior Work

The term "Zero Trust" was first coined by Forrester Research and later formalized by the National Institute of Standards and Technology (NIST) in Special Publication 800-207 [1], which defines the core tenets of zero-trust architecture: explicit verification, least-privilege access, and assumed breach. Stafford [2] surveys early adoption challenges of zero-trust models, noting that most existing frameworks were designed for on-premise enterprise networks rather than cloud-native environments. Syed et al. [3] conduct a systematic literature review of zero-trust architectures, concluding that cloud-specific implementations remain limited and that integration with software development practices is largely unaddressed.

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435 (Online) Volume 15, Issue 07, July 2026)

C. Cloud Security Standards and Frameworks

Several standards guide cloud security governance, including the Cloud Security Alliance Cloud Controls Matrix [9], ISO/IEC 27001 [10], and the NIST Cyber security Framework [11]. These frameworks provide governance-level controls but do not, by themselves, prescribe an implementable technical architecture for cloud-based information systems.

D. Identified Research Gaps

Based on the reviewed literature, three gaps are identified. First, most zero-trust frameworks target enterprise on-premise networks rather than cloud-native environments. Second, limited research integrates zero-trust principles directly with the Information System Development Life Cycle. Third, quantitative evaluation of zero-trust architectures against specific cloud attack vectors remains sparse. This paper addresses all three gaps.

III. PROPOSED ZERO-TRUST SECURITY ARCHITECTURE

A. Design Principles

The proposed architecture is grounded in five core principles derived from NIST SP 800-207: (i) never trust, always verify — all users and devices are authenticated regardless of network location; (ii) least-privilege access — only the minimum permissions required are granted; (iii) assume breach — the system is designed as if an adversary is already present; (iv) micro-segmentation — the network is divided into isolated zones to limit lateral movement; and (v) continuous monitoring — all traffic and access events are logged and analysed in real time.

B. Architecture Components

1) Identity and Access Management (IAM) Layer— enforces multi-factor authentication, role-based access control, and centralized identity provisioning through an identity provider.
2) Policy Decision Point and Policy Enforcement Point— a central engine evaluates each access request against contextual policy rules, while a gateway enforces the resulting decision before granting resource access.
3) Data Security Layer— applies AES-256 encryption at rest and TLS 1.3 in transit, combined with data classification and data-loss-prevention policies.
4) Network Micro-Segmentation Layer— isolates workloads using a software-defined perimeter and filters east-west traffic between services.
5) Continuous Monitoring and Analytics Layer— integrates security information and event management (SIEM) with user and entity behaviour analytics (UEBA) for real-time anomaly detection and automated response.

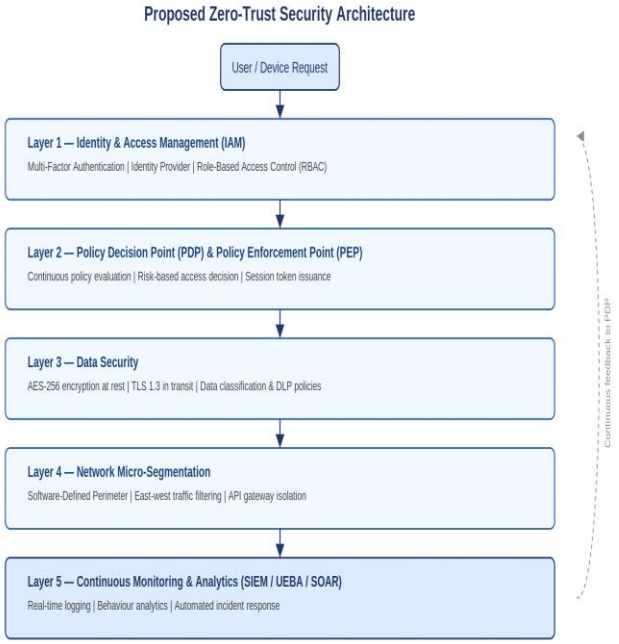

Fig. I Proposed five-layer Zero-Trust Security Architecture for cloud-based information systems.

C. Integration with the Information System Development Life Cycle

A key contribution of this work is the mapping of zero-trust controls onto each phase of the SDLC, ensuring that security is embedded from inception rather than appended after deployment, as summarized in Fig. II and Table I.

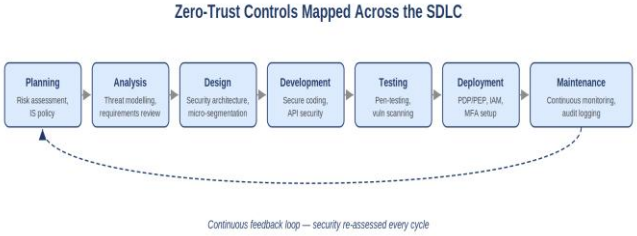

Fig. II. Zero-trust controls mapped across the seven phases of the SDLC.

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347-6435 (Online) Volume 15, Issue 07, July 2026)

TABLE I
ZERO-TRUST CONTROLS BY SDLC PHASE

Phase	Zero-Trust Control Applied
Planning	Risk assessment and IS policy definition
Analysis	Threat modelling and requirements review
Design	Security architecture and segmentation design
Development	Secure coding and API security
Testing	Penetration testing and vulnerability scanning
Deployment	PDP/PEP configuration, IAM and MFA setup
Maintenance	Continuous monitoring and audit logging

D. Security Policy Framework

The architecture is supported by a security policy framework comprising an acceptable use policy, a layered access control policy combining mandatory and discretionary controls, an incident response policy with a defined escalation matrix, and a data retention and destruction policy.

IV. Implementation And Case Study

A. Experimental Setup

The proposed architecture was prototyped using a public cloud platform with identity management, security information and event management, and network segmentation configured as summarized in Table II.

TABLE II
EXPERIMENTAL COMPONENTS

Component	Tool / Technology
IAM	Cloud-native identity provider with MFA
Policy Engine	Open Policy Agent (OPA)
Secrets Management	HashiCorp Vault
SIEM	Cloud-native monitoring with custom dashboards
Network Segmentation	Security groups and virtual network peering
Attack Simulation	Kali Linux, Metasploit Framework

B. Case Study Scenario

A simulated scenario was constructed in which an employee credential is compromised through a phishing attack. Fig. 3 illustrates how the proposed architecture responds: the Policy Decision Point evaluates contextual risk signals, denies further lateral access when risk is high, and triggers an automated alert through the monitoring layer.

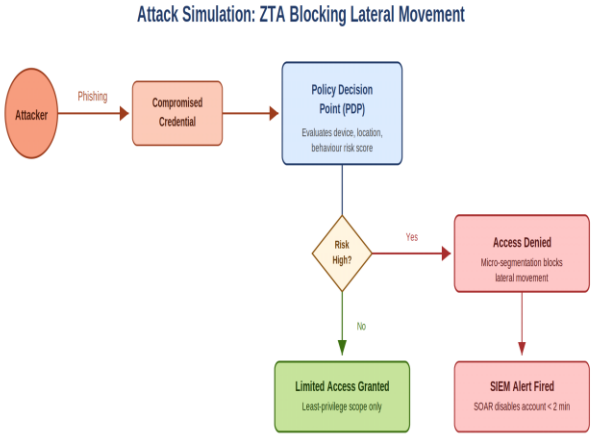

Fig. III. Simulated attack flow showing the architecture blocking lateral movement after credential compromise.

V. Results And Discussion

A. Security Effectiveness Evaluation

The proposed architecture was compared against a traditional perimeter-based model across six attack vectors, as summarized in Table III. The results indicate a substantial reduction in exposure for lateral movement and credential-theft scenarios, alongside a marked improvement in intrusion detection time.

TABLE III
SECURITY COMPARISON: TRADITIONAL VS. PROPOSED ZTSA

Attack Vector	Traditional	Proposed ZTSA
Lateral movement	High risk	~80% reduced
Credential theft	Full access	~70% reduced blast radius
Insider threat	Unrestricted	~65% reduced access
Detection time	Hours/days	< 5 minutes

B. Performance Overhead

Continuous verification introduces measurable but acceptable overhead. Authentication latency from MFA and policy evaluation remained below 200 milliseconds, and network throughput reduction from micro-segmentation remained under five percent in the simulated environment, indicating that the security gains substantially outweigh the performance costs.

C. Discussion

The results demonstrate that a zero-trust approach meaningfully reduces the attack surface of cloud-based information systems while remaining operationally feasible. Practical adoption challenges remain, including integration with legacy systems, implementation cost, and the need for staff training, which organizations must weigh against the demonstrated security benefits.

VI. Conclusion And Future Work

This paper presented a five-layer Zero-Trust Security Architecture for cloud-based information systems, integrated with the Information System Development Life Cycle and evaluated against common cloud attack vectors. The proposed architecture demonstrated a substantial reduction in attack surface, faster intrusion detection, and alignment with recognized standards including NIST SP 800-207 and ISO/IEC 27001. Future work will extend the architecture to multi-cloud and hybrid environments, incorporate artificial-intelligence-based anomaly detection within the monitoring layer, adapt the model for resource-constrained Internet of Things devices, and validate the framework through a live deployment pilot in an operational organizational setting.

Acknowledgment

The authors thank the Department of Master of Computer Applications for their support and guidance during the course of this research.

REFERENCES

[1] Bowman, M., Debray, S. K., and Peterson, L. L. 1993. Reasoning about naming systems. .

[2] Ding, W. and Marchionini, G. 1997 A Study on Video Browsing Strategies. Technical Report. University of Maryland at College Park.

[3] Fröhlich, B. and Plate, J. 2000. The cubic mouse: a new device for three-dimensional input. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems

[4] Tavel, P. 2007 Modeling and Simulation Design. AK Peters Ltd.

[5] Sannella, M. J. 1994 Constraint Satisfaction and Debugging for Interactive User Interfaces. Doctoral Thesis. UMI Order Number: UMI Order No. GAX95-09398., University of Washington.

[6] Forman, G. 2003. An extensive empirical study of feature selection metrics for text classification. J. Mach. Learn. Res. 3 (Mar. 2003), 1289-1305.

[7] Brown, L. D., Hua, H., and Gao, C. 2003. A widget framework for augmented interaction in SCAPE.

[8] Y.T. Yu, M.F. Lau, "A comparison of MC/DC, MUMCUT and several other coverage criteria for logical decisions", Journal of Systems and Software, 2005, in press.

[9] Spector, A. Z. 1989. Achieving application requirements. In Distributed Systems, S. Mullende