COLOMBO SCIENTIFIC PUBLISHING

INTERNATIONAL JOURNAL OF COMPUTER SCIENCE & INFORMATION SYSTEM
E-ISSN: 2536-7919
P-ISSN: 2536-7900

DOI: - https://doi.org/10.55640/ijcsis/Volume11Issue05-04 **PAGE NO: 33-39**

Architectural Principles of Zero Trust Privileged Access Management in Modern Corporate Infrastructures

Kolchin Rustam
SoftLine PJSC
Almaty, Kazakhstan

ARTICLE INFO

Article history:

Submission: March 12, 2026
Accepted: April 15, 2026
Published: May 23, 2026
VOLUME: Vol.11 Issue 05 2026

Keywords:
Zero Trust, privileged access management, least privilege, identity security, session governance, PAM architecture, lateral movement, privilege escalation, hybrid infrastructure, auditability.

ABSTRACT

The study examines architectural principles for Zero Trust Privileged Access Management in corporate infrastructures that rely on cloud resources, distributed administration, DevOps pipelines, and machine identities. Privileged access creates risk because administrators, service accounts, automation scripts, and emergency credentials can change infrastructure state across several layers. The research aim is to define a Zero Trust PAM model that connects continuous verification, least privilege, session governance, and audit evidence. The study uses comparative source analysis, conceptual synthesis, typologization, and analytical generalization of standards, peer-reviewed studies, and threat frameworks. The review identifies three outcomes: privileged access moves toward short-lived task sessions, PAM becomes a control point between identity, network, cloud, and monitoring layers, and governance covers human and machine privileges through one evidence trail. The proposed principles help security architects plan PAM modernization without vendor claims, undisclosed deployment metrics, or customer-specific case details. The paper follows a review-plus analytical design for publication.

Introduction

Privileged access gives administrators, service identities, emergency accounts, cloud roles, DevOps secrets, and remote support channels direct influence over infrastructure components. A single approved account can change configurations, read sensitive data, disable controls, create new users, or move from one system to another through existing trust relationships. Classical PAM reduced part of this exposure through credential vaulting, password rotation, session brokering, approval workflows, and recording. These controls remain useful, yet modern corporate infrastructures create access paths that a vault alone cannot govern.

Cloud platforms, container orchestration, remote administration, outsourced operations, CI/CD pipelines, and machine-to-machine communication have changed the shape of privileged work. Administrative power appears in cloud roles, API tokens, build agents, automation scripts, service accounts, emergency credentials, and temporary access grants. Security teams need an architecture that evaluates each privileged operation as a bounded action.

The research aim is to formulate architectural principles for Zero Trust Privileged Access Management in modern corporate infrastructures. Three research objectives guide the study: to compare vault-centric PAM logic with Zero Trust privileged access logic, to define integration points between PAM, IAM, ZTNA, cloud access control, and monitoring systems, and to describe a governance model for reducing lateral movement, privilege escalation, and misuse of privileged identities.

The novelty of the study lies in treating PAM as a policy enforcement and evidence layer inside Zero Trust architecture. The proposed view links privileged access to session-level authorization, task-bound permissions, adaptive verification, automated workflows, and audit traces. The working hypothesis states that Zero Trust PAM strengthens corporate infrastructure security when security architects shift privileged access control from standing account protection to dynamic session governance across human and machine identities.

Materials and Methods

The source corpus combines peer-reviewed studies on Zero Trust Architecture, identity and access management, cloud access control, and implementation barriers with authoritative standards and threat frameworks. The selected materials cover six groups of questions: Zero Trust reference architecture [10], cloud-native and multi-location access control [3], least privilege and privileged account controls [8], Zero Trust maturity domains [4], implementation costs, adoption barriers, survivability, and technical limits of ZTA [2], [5], [6], [7], IAM analytics and adaptive access decisions [1], and adversary techniques linked to valid accounts, privilege escalation, and account manipulation [9]. This corpus suits an analytical article because it connects normative requirements, architectural research, and attacker behavior without using closed deployment metrics.

The study uses comparative analysis to separate classical PAM from Zero Trust PAM, source analysis to extract architectural requirements from standards and academic literature, conceptual synthesis to connect PAM with IAM, ZTNA, SIEM/SOAR, and cloud control planes, typologization to group privileged access scenarios by risk and identity type, and analytical generalization to formulate implementation logic for corporate infrastructures.

Results

Privileged Access Management developed around a practical problem: administrative accounts give broad technical power, and attackers often seek that power after initial compromise. Classical PAM usually begins with a protected vault, controlled credential checkout, password rotation, session brokering, and recording of privileged actions. These controls reduce direct password exposure and preserve evidence after a session. Zero Trust changes the access decision itself. NIST SP 800-207 defines Zero Trust through resource-centered protection and removal of implicit trust based on network location, ownership, or prior access [10]. For privileged access, an approved administrator account does not create sufficient grounds for reaching a server, database, cloud console, Kubernetes namespace, or pipeline secret.

A vault-centric PAM system answers a storage and issuance question: how to protect, rotate, and release privileged credentials. Zero Trust PAM asks whether one privileged session, requested by a specific subject, from a specific device, for a declared task, against a specific target, meets policy at that moment. Prior research on ZTA adoption connects continuous validation with the reduction of implicit trust and discusses cost considerations for organizational implementation [2]. In PAM design, this means that authorization moves from account possession to a decision cycle that checks identity assurance, device posture, request purpose, target sensitivity, session duration, and telemetry.

Research on the technical structure of ZTA warns against treating Zero Trust as a slogan attached to existing products. Fernandez and Brazhuk examine ZTA through security patterns, threats, and reference architecture elements [5]. Their position matters for PAM because privileged access tools can collect features without forming a coherent control model. Security architects need a visible chain: policy decision, policy administration, policy enforcement, session monitoring, log preservation, and risk feedback. A system that records privileged actions after approval still leaves a gap if it cannot constrain or terminate the session near the protected resource.

NIST SP 800-207 gives a clear reference structure through the policy engine, policy administrator, and policy enforcement point [10]. In a PAM architecture, the policy engine evaluates the request. The policy administrator prepares the approved path, for example a brokered SSH session, database console, cloud role assumption, or temporary elevation. The policy enforcement point blocks, permits, constrains, monitors, or terminates the session near the target. Figure 1 presents this control loop for privileged sessions.

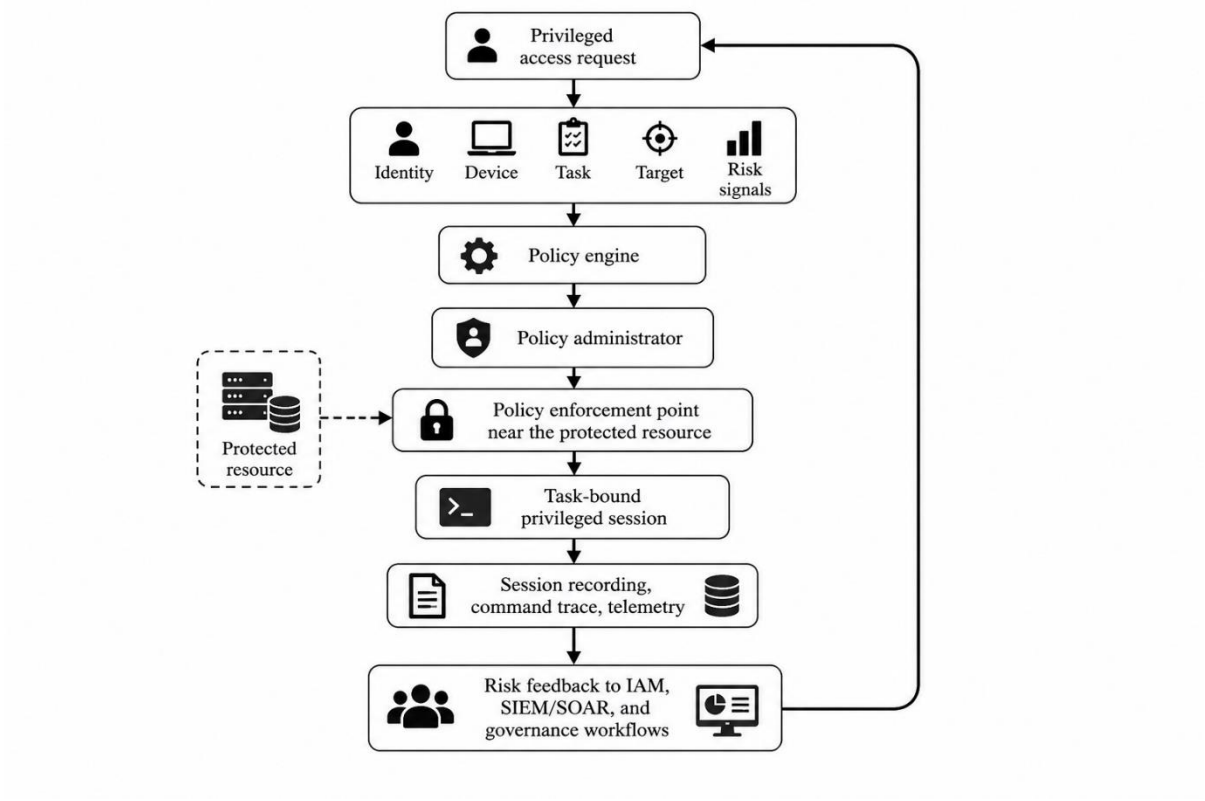

Figure 1. Zero Trust PAM control loop for privileged sessions, adapted from NIST Zero Trust Architecture [10]

The scheme moves the protected object from the credential alone to the privileged session and the target operation. The session carries time limits, task boundaries, monitoring rules, termination criteria, and evidence requirements. Password checkout creates residual exposure when the credential leaves the control plane. Session brokering, just-in-time elevation, ephemeral credentials, step-up verification, and command-level monitoring reduce that exposure because administrators perform privileged work inside enforceable boundaries.

NIST SP 800-207A addresses granular application-level policies for multi-cloud and hybrid environments and emphasizes identity-tier policies for cloud-native access control [3]. This requirement affects PAM because privileged work rarely touches one isolated host. Administrators and automated processes often cross workloads, APIs, databases, storage buckets, container registries, service meshes, CI/CD systems, and remote consoles. Network segmentation can limit reachability, while identity-tier policy defines who or what can perform an operation. Zero Trust PAM therefore needs links with both access paths and identity-based authorization.

Standing roles, stale administrator groups, broad service accounts, long-lived API tokens, and unreconciled emergency access create durable privilege after a task ends. NIST SP 800-53 describes least privilege through restrictions that limit users and processes to authorized access required for assigned tasks [8]. In PAM terms, least privilege requires time-bound elevation, approval rules tied to target sensitivity, separation between daily-use and administrative accounts, controlled use of shared accounts, and revocation after task completion.

Adahman, Malik, and Anwar examine ZTA as an organizational security approach with adoption and cost implications [2]. Ferretti and co-authors propose survivable Zero Trust for cloud computing, where dynamic access-control policies govern flows under cloud conditions [6]. Fernandez and Brazhuk call for stronger technical grounding and threat analysis in ZTA design [5]. Itodo and Ozer report fragmentation in the literature on Zero Trust implementation and point to limited validation across organizational scales [7]. Together, these studies support a restrained claim: Zero Trust PAM cannot eliminate privileged-access risk, but it can reduce uncontrolled pathways through enforceable sessions, cross-layer policy, and auditable governance evidence.

Aboukadri, Ouaddah, and Mezrioui survey machine learning in identity and access management and examine risk-sensitive, behavior-aware access mechanisms [1]. PAM teams can use these signals before and during privileged sessions. Unusual request timing, atypical target selection, unfamiliar device posture, abnormal command sequences, repeated failed elevation, or access from a new geography can trigger step-up verification, narrower session scope, manual approval, or termination. This use gives the policy engine additional evidence before privileged work proceeds.

MITRE ATT&CK describes Valid Accounts as a technique where adversaries abuse existing credentials for initial access, persistence, privilege escalation, or defense evasion [9]. Exploitation for Privilege Escalation covers adversary use of

software vulnerabilities to gain higher privileges [9]. Account Manipulation covers changes that preserve, modify, or elevate adversary access, including credential or permission-group changes [9]. These techniques map directly onto PAM. A vault can protect stored secrets, but an adversary with a valid account, a modified permission group, or an abused cloud role may still act inside approved channels. Zero Trust PAM reduces this path by making privileged access conditional, short-lived, observable, and traceable.

Session recording receives a different function under this threat model. Classical PAM often uses recording for compliance and later investigation. Zero Trust PAM still needs recording, yet live telemetry becomes part of the decision process. Command metadata, target identifiers, source device, ticket reference, elevation reason, policy version, approval path, and revocation status should enter one evidence trail. CISA's Zero Trust Maturity Model organizes Zero Trust maturity across identity, devices, networks, applications and workloads, data, visibility and analytics, automation and orchestration, and governance [4]. PAM contributes to these domains when privileged sessions carry identity proof, device signals, target data, telemetry, automated response options, and reviewable policy evidence.

A second comparison across sources strengthens the architectural conclusion. NIST SP 800-207 frames Zero Trust as an enterprise resource-protection model [10]. NIST SP 800-207A extends that model to cloud-native and multi-location access control [3]. CISA structures maturity through coordinated identity, telemetry, automation, and governance domains [4]. Aboukadri, Ouaddah, and Mezrioui show that IAM increasingly uses behavior and risk signals for access decisions [1]. For PAM, the implication is direct: privileged access control has to connect with IAM for identity assurance, ZTNA or a comparable broker for resource reachability, SIEM/SOAR for telemetry and response, and cloud control planes for entitlement changes. A disconnected vault can reduce password exposure, but it cannot enforce Zero Trust privilege boundaries across distributed administration.

Administrators and developers often need routine task access without long queues, especially in infrastructure operations and incident response. Broad automatic approval recreates standing privilege under another name. Zero Trust PAM resolves this tension through request templates tied to task categories, target sensitivity, approval thresholds, time limits, and evidence requirements. Low-risk maintenance can receive automated just-in-time elevation after identity, device, and ticket checks. Production database access, directory schema change, privileged cloud role assignment, and security-control modification need stronger approval, shorter duration, and richer monitoring.

Three architectural principles follow from the reviewed literature, first, privileged access should be session-bound, short-lived, and tied to declared tasks. Static membership in administrator groups leaves residual power after the task ends. Second, security architects should place PAM between identity, network access, cloud entitlement systems, and monitoring platforms. Privileged work crosses these layers, and governance has to follow the same path. Third, auditability should cover the full decision chain. The trace needs to show why a user or process requested access, which policy approved it, which signals influenced the decision, what happened during the session, when elevation ended, and how the team handled exceptions.

DISCUSSION

Security architects should treat Zero Trust PAM as access-governance architecture. Many enterprises already operate separate components for PAM, IAM, remote access, SIEM, ticketing, and cloud entitlement management. The implementation problem lies in aligning these components around one decision logic. A workable design begins with privileged-access inventory. Security teams cannot enforce least privilege over accounts, service identities, scripts, keys, roles, and tokens that remain outside the access map.

The next step separates privileged identities by type: named human administrator, break-glass account, service account, workload identity, API token, CI/CD secret, outsourced operator, and automated process. Each type needs its own authorization path, lifespan, and evidence requirement. A named administrator may need brokered access with MFA and session recording. A CI/CD pipeline needs scoped secret delivery, expiry, and a link to repository and build identity. A break-glass account requires sealed access, immediate alerting, and mandatory post-use review.

Implementation should proceed through staged modernization. The first stage removes standing privilege by separating daily-use accounts from administrative accounts, disabling stale accounts, rotating shared credentials, and enforcing MFA for privileged access. The second stage introduces just-in-time elevation and task-based approvals. The third stage connects PAM to identity governance, device posture, ticketing, cloud control planes, and monitoring systems. The fourth stage adds feedback from security analytics, privileged session behavior, and incident response. This sequence reduces the risk of deploying advanced policy tools while unmanaged service accounts, shared administrator passwords, and broad cloud roles remain in daily use.

Table 1 compares classical PAM and Zero Trust PAM through control logic. The comparison keeps vaulting within the model, but places vaulting inside a broader policy loop that governs the privileged operation.

Table 1. Architectural comparison of classical PAM and Zero Trust PAM

Comparison criterion	Classical PAM model	Zero Trust PAM model
Main protected object	Privileged credential	Privileged session and target operation
Trust assumption	Approved account receives access after credential control	Each privileged request requires verification
Access duration	Checkout window or standing role often defines access	Task, time, policy, and risk level define access
Authorization basis	Account membership, credential request, approval workflow	Identity assurance, device state, task, target, telemetry, policy version
Session control	Recording and optional command monitoring	Enforcement, recording, termination, and adaptive restriction
Cloud and DevOps coverage	Connectors often add coverage after deployment	Access-control design covers cloud and DevOps from the start
Audit evidence	Credential use and recorded session	Full decision chain from request to revocation
Operational model	Centralized control of privileged secrets	Distributed enforcement with centralized governance

The comparison shows why a PAM modernization program should not stop at vault replacement. Vaulting, rotation, session recording, and approval workflows remain useful, but architects need to connect them to policy decisions near the protected resource. The main architectural risk appears when teams approve broad access once and then rely on after-the-fact recordings. Administrators still gain more power than the task requires, machine identities continue to use long-lived secrets, and cloud privileges expand faster than manual review cycles can remove them.

Security teams should avoid turning every administrative action into a manual approval queue, because excessive friction drives staff toward local credentials, unmanaged scripts, and unofficial emergency paths. A better model assigns control strength to target sensitivity, identity type, and task risk. Routine maintenance on low-sensitivity systems can receive automated approval after identity, device, and ticket checks. Administrative access to production, directory services, security tooling, or cloud root-level functions needs stronger checks, shorter duration, and fuller monitoring. Emergency access remains available, but the access path creates an immediate alert and a mandatory review.

Table 2 translates architectural principles into implementation decisions. It compares common privileged access scenarios by access mechanism, risk control, and evidence requirement.

Table 2. Decision logic for Zero Trust privileged access scenarios

Scenario	Preferred access mechanism	Risk control	Evidence requirement
Named system administrator accessing production server	Brokered session with just-in-time elevation	MFA, device check, task reference, time limit	Session recording, command metadata, policy version
Developer accessing CI/CD secret	Temporary secret injection or scoped token	Repository, pipeline, and workload validation	Secret request log, pipeline identifier, expiry record
Cloud administrator changing privileged role	Short-lived role assumption	Approval threshold, target sensitivity, entitlement review	Role-change record, approver identity, revocation status
Service account used by application	Managed machine identity or rotated credential	Lifecycle ownership, scope restriction, automated expiry	Owner record, rotation history, service dependency map
Break-glass account	Sealed emergency workflow	Dual control, immediate alerting, mandatory post-review	Full session capture, incident ticket, post-use attestation
Outsourced operator performing maintenance	Federated access through brokered session	Contract scope, device posture, time window, target allowlist	External identity record, session transcript, approval chain
Security analyst accessing SIEM or SOAR admin console	Privileged console with command logging	Separation of duties, approval for destructive actions	Console activity log, change record, escalation trail

Many PAM programs focus on whether a user received approval, Zero Trust PAM asks a narrower operational question: which action did the policy allow, under which conditions, and what proof remains after the session ends. This question suits regulated and high-risk infrastructures because it connects administrative work with auditability and incident review.

Security teams should measure the architecture through control-quality indicators. Useful metrics include the share of privileged access delivered through just-in-time elevation, number of standing privileged accounts, average privileged-session duration, number of unmanaged service accounts, percentage of privileged sessions linked to approved tasks, failed step-up verification events, privilege grants without expiry, break-glass account uses, and policy exceptions by business unit. These metrics show whether the organization reduces durable privilege and increases traceable access.

The model has limits, Zero Trust PAM cannot compensate for inaccurate asset inventory, weak identity proofing, poor endpoint hygiene, or fragmented ownership of cloud roles. Integration quality sets a hard boundary. Ticketing, IAM, device posture, cloud APIs, and SIEM data need stable identifiers, because the evidence trail loses value when the team cannot connect a request, a user, a device, a target, and a policy decision. Human factors shape deployment as much as tooling. Administrators may resist excessive approvals, developers may object to broken automation, and operations teams may preserve unofficial access paths for service continuity. Staged implementation, clear exception handling, and measured reduction of standing privileges reduce these adoption risks.

Machine identities require the same governance discipline as human administrators. Service accounts and workload identities often operate with less scrutiny, although their permissions can affect production systems at scale. Zero Trust PAM should assign ownership to each machine identity, bind it to an application or pipeline, replace long-lived secrets where feasible, rotate credentials where replacement is not feasible, and record privileged actions performed by automated processes. A program that secures only human administrators leaves automated privilege sprawl outside the policy loop.

CONCLUSION

Zero Trust changes privileged access from account-centered control to session-centered governance. Classical PAM protects credentials and records privileged sessions. Zero Trust PAM governs each privileged request as a conditional, task-bound, and evidence-generating action. This distinction clarifies the move from vault-centric protection toward policy-based session control.

Privileged access in hybrid infrastructures crosses identity, network, cloud, DevOps, and monitoring layers. PAM therefore needs integration with IAM, access brokers, cloud entitlement systems, ticketing platforms, SIEM/SOAR, and device posture signals. The architecture gains value when these components feed one decision chain.

Lateral movement, privilege escalation, and account manipulation require governance over human and machine privileges. A defensible Zero Trust PAM model uses just-in-time elevation, least privilege, adaptive verification, session monitoring, and complete audit trails. The working hypothesis receives support: Zero Trust PAM strengthens corporate infrastructure security when security architects replace static privileged account protection with dynamic session governance, policy enforcement, and verifiable evidence across distributed systems.

References

1. Aboukadri, S., Ouaddah, A., & Mezrioui, A. (2024). Machine learning in identity and access management systems: Survey and deep dive. *Computers & Security, 139*, 103729. doi: 10.1016/j.cose.2024.103729

2. Adahman, Z., Malik, A. W., & Anwar, Z. (2022). An analysis of zero-trust architecture and its cost-effectiveness for organizational security. *Computers & Security, 122*, 102911. doi: 10.1016/j.cose.2022.102911

3. Chandramouli, R., & Butcher, Z. (2023). *A zero-trust architecture model for access control in cloud-native applications in multi-location environments*. National Institute of Standards and Technology. doi: 10.6028/NIST.SP.800-207A

4. Cybersecurity and Infrastructure Security Agency. (2023). *Zero Trust Maturity Model, Version 2.0*. U.S. Department of Homeland Security.

5. Fernandez, E. B., & Brazhuk, A. (2024). A critical analysis of Zero Trust Architecture (ZTA). *Computer Standards & Interfaces, 89*, 103832. doi: 10.1016/j.csi.2024.103832

6. Ferretti, L., Magnanini, F., Andreolini, M., & Colajanni, M. (2021). Survivable zero trust for cloud computing environments. *Computers & Security, 110*, 102419. doi: 10.1016/j.cose.2021.102419

7. Itodo, C., & Ozer, M. (2024). Multivocal literature review on zero-trust security implementation. *Computers & Security, 141*, 103827. doi: 10.1016/j.cose.2024.103827

8. Joint Task Force. (2020). *Security and privacy controls for information systems and organizations*. NIST Special Publication 800-53, Revision 5. National Institute of Standards and Technology. doi: 10.6028/NIST.SP.800-53r5

9. MITRE Corporation. (2026). *MITRE ATT&CK Enterprise Matrix: Valid Accounts T1078, Exploitation for Privilege Escalation T1068, and Account Manipulation T1098*. Retrieved May 2, 2026.

10. Rose, S., Borchert, O., Mitchell, S., & Connelly, S. (2020). *Zero Trust Architecture*. NIST Special Publication 800-207. National Institute of Standards and Technology. doi: 10.6028/NIST.SP.800-207