International Journal of Engineering, Science and Information Technology
Volume 5, No. 3 (2025) pp. 628-635
eISSN: 2775-2674
Website: http://ijesty.org/index.php/ijesty
DOI: https://doi.org/10.52088/ijesty.v5i3.1294
Research Paper, Short Communication, Review, Technical Paper

IJESTY
International Journal of Engineering, Science and Information Technology

Failure of Preventive Security Controls in Cloud-Native Systems: Revisiting Governance Enforcement

Muhammad Daffa Ramadhan*, Ahmad Nurul Fajar

Master of Information Systems Management, Universitas Bina Nusantara, Jakarta, Indonesia

**Corresponding author Email: daffaromadon@gmail.com*

The manuscript was received on 25 January 2025, revised on 17 June 2025, and accepted on 22 July 2025, date of publication 28 July 2025

Abstract

Cloud-native architectures have introduced a fundamental shift in how security and governance are applied within modern IT environments. While traditional preventive IT General Controls (ITGCs) were designed for static, centralised systems, their application in dynamic, decentralised, and automated cloud-native systems remains ambiguous and often ineffective. This study investigates the patterns of failure in preventive controls across cloud-native environments and analyses the extent to which governance frameworks fail to enforce security proactively. Employing a meta-synthetic approach, this research reviews documented cloud breach incidents from 2021 to 2024 to extract recurring failure patterns. These incidents were analysed and mapped against major security control domains, including identity and access management, configuration hardening, and observability. The findings highlight systemic gaps in the implementation of preventive measures, particularly in areas where infrastructure is governed as code, and runtime dynamics alter control effectiveness. Furthermore, the study examines how existing governance frameworks such as ISO 27001, COBIT, and NIST CSF are often too abstract or outdated to directly translate into executable policies within CI/CD pipelines and cloud-native infrastructures. The study reveals that misconfigurations, inadequate identity management, and runtime blind spots are among the most common contributors to control failures. These issues are compounded by the lack of real-time enforcement mechanisms and the misalignment between policy design and operational realities. Based on these findings, the paper proposes a shift toward Governance-as-Code and continuous control validation as critical strategies for modern preventive governance. In conclusion, the paper demonstrates that traditional ITGCs, while still conceptually relevant, require operational reengineering to remain effective in cloud-native ecosystems. A governance model that is executable, context-aware, and runtime-integrated is essential for proactive security and sustained compliance in modern digital infrastructure.

***Keywords**: Cloud Computing, Cybersecurity, Information Governance, Security Controls, System Misconfiguration.*

1. Introduction

Cloud computing has radically transformed the operational landscape of enterprise IT, enabling elasticity, automation, and distributed architecture at scale [1][2]. However, the same features that make cloud-native systems efficient also introduce complexity that traditional governance models struggle to manage [3][4]. In particular, Identity and Access Management (IAM), misconfiguration control, and runtime monitoring have emerged as persistent weak points in cloud security, despite organisations' formal adherence to widely accepted IT governance frameworks such as ISO/IEC 27001, NIST CSF, COBIT 2019, and CSA CCM [5]–[7].

Recent industry breach reports and empirical studies consistently highlight that a significant proportion of cloud-related incidents are not caused by unknown vulnerabilities or advanced persistent threats, but by governance failures, where declared controls exist on paper yet remain unenforced or misaligned with technical realities [8][9]. This contradiction raises a critical question: Why do breaches continue to occur in domains that are already covered by formal frameworks?

Academic literature increasingly suggests that traditional control frameworks fail to capture the operational complexity and automation inherent in modern DevSecOps pipelines and ephemeral environments [10]–[13]. In such settings, control enforcement requires real-time validation, context-aware configuration, and architectural alignment, none of which are typically enforced through static compliance audits. This contradiction between formal compliance and the recurrence of preventable breaches calls for a closer examination of how governance is actually enforced in practice. What patterns and root causes explain the failure of preventive controls in cloud-native security incidents? And to what extent can existing frameworks such as ISO 27001, COBIT, NIST CSF, and the Cloud Controls Matrix support effective enforcement in dynamic, automated environments?

This paper addresses these questions through a meta-synthetic analysis of cloud breach incidents from 2021 to 2024, drawing from publicly available incident reports, threat intelligence repositories, and published postmortems [14]–[16]. The objective is to identify recurring breach patterns, classify them into control domains, and evaluate whether existing frameworks adequately cover these domains both in definition and in practical enforceability.

The contribution of this paper is twofold: first, it quantifies the frequency of control domain failures across a multi-year sample of real-world cloud security incidents; second, it presents a comparative mapping between those incidents and the corresponding governance controls across four major frameworks. The findings demonstrate that the presence of controls in documentation does not equate to effectiveness in practice, especially in cloud-native environments characterised by ephemeral workloads, decentralised management, and rapid deployment cycles [17][18].

By exposing this gap, the study aims to inform practitioners and policymakers of the urgent need to shift from compliance-based governance toward enforceable, lifecycle-aligned control implementations that reflect the distributed, automated, and transient nature of modern cloud systems [19][20].

2. Literature Review

2.1. IT Governance Frameworks and Cloud Context

Information Technology Governance (ITG) frameworks such as COBIT 2019, ISO/IEC 27001, and NIST CSF were developed to help organisations formalise control objectives, manage risks, and ensure compliance in traditional IT environments. These frameworks typically articulate principles such as accountability, risk management, auditability, and information assurance through a set of generalised control categories [21][22].

However, as organisations migrate to cloud-native infrastructure, these frameworks have faced increasing criticism for their limited contextual alignment with modern computing paradigms. Cloud-native systems are characterised by rapid deployment, immutable infrastructure, ephemeral workloads, and decentralised service ownership. These conditions challenge legacy assumptions about perimeter-based control, static audit evidence, and hierarchical accountability [23].

Although ISO/IEC 27001 was updated in 2022 to include considerations for cloud computing, it remains primarily document-oriented and lacks prescriptive guidance for control execution in distributed environments. Similarly, COBIT 2019 emphasises enterprise alignment and governance objectives, but its abstraction level can leave implementation details vague, particularly in relation to automated DevOps pipelines and shared responsibility models inherent in cloud ecosystems [24][25].

NIST’s Cybersecurity Framework Version 2.0 attempts to modernise its categories to address cloud environments, including the introduction of subcategories on continuous monitoring and third-party supply chain risk. Nonetheless, its practical enforceability in cloud-native architectures remains a concern due to the non-binding nature of the recommendations and their dependence on organisational maturity [26].

The Cloud Security Alliance’s CCM (Cloud Controls Matrix) aims to address these gaps with control mappings explicitly designed for cloud infrastructure. It offers more granular controls for areas such as containerization, CI/CD security, and identity federation. However, its adoption remains uneven across industries and often supplements rather than replaces traditional frameworks [27].

Overall, while governance frameworks provide foundational control schemas, their translation into enforceable mechanisms in cloud-native environments requires reinterpretation, augmentation, or automation to remain effective.

2.2. IT Governance Frameworks and Cloud Context

Cloud environments have introduced a new attack surface and threat model that diverges significantly from traditional on-premise systems. While common breach causes such as phishing and credential theft remain relevant, cloud-specific patterns have emerged, such as misconfigured storage buckets, insecure APIs, and overprivileged machine identities [28].

One of the most recurring breach vectors is misconfiguration of cloud resources. This includes public exposure of data storage (e.g., Amazon S3), permissive security groups, and overly broad IAM policies. These misconfigurations are often unintentional but can result in massive data leakage or lateral privilege escalation, especially when combined with stolen credentials [29][30].

Credential abuse is another dominant pattern, exacerbated by poor key rotation, use of static access tokens, and lack of multifactor authentication enforcement. Once credentials are compromised, attackers can exploit cloud consoles, APIs, or automation tools such as Terraform and Kubernetes to propagate laterally or exfiltrate data [31].

A third category involves supply chain risks and third-party compromise. In multi-tenant or federated cloud environments, a compromise in one partner or SaaS provider can propagate across connected systems due to weak vendor governance or excessive trust boundaries. This was evident in several high-profile cases where access tokens issued by third-party CI/CD platforms were used to inject malicious artefacts [32].

Insecure APIs and weak monitoring controls also play a role in cloud breaches. APIs are often under-protected and exposed to the internet by default, making them prime targets for enumeration and injection attacks. Without centralised logging or behaviour baselining, such activities often go undetected for weeks [33][34].

Interestingly, many breach patterns map to control domains already covered in major frameworks, such as IAM, monitoring, and third-party risk management, yet still occur frequently in practice. This suggests a persistent enforcement gap between control declaration and execution, often driven by the speed and abstraction of cloud-native deployment models.

2.3. Governance Implementation Gaps in Practice

While governance frameworks provide structured sets of control objectives, their real-world application often falters due to architectural, operational, and organisational disconnects. These “governance gaps” are not simply oversights, but systemic limitations in translating policy into enforceable technical mechanisms within modern cloud-native environments [35][36].

One major challenge lies in the separation between governance teams and technical implementation teams. In many enterprises, security and governance functions operate independently from DevOps and cloud platform teams, leading to a loss of contextual control enforcement. Controls such as IAM policies, workload segmentation, or configuration standards are declared centrally but executed variably, or not at all, across teams, accounts, or cloud service providers [37].

Cloud-native characteristics exacerbate this problem. Features such as auto-scaling, container orchestration, immutable infrastructure, and CI/CD pipelines create environments where changes occur continuously and often automatically. This velocity renders static governance controls obsolete without automation or runtime verification mechanisms. For instance, while ISO 27001 mandates configuration control, it does not define how to validate that these controls remain intact in continuously deployed infrastructure [38].

Moreover, evidence from breach postmortems frequently shows that governance processes exist in policy but are not validated in runtime. IAM misconfigurations, unrestricted permissions, and unmonitored data flows often exist in systems that had passed formal compliance audits just months earlier [39][40]. This pattern suggests that audits and documentation-based approaches cannot keep up with the rate of change in cloud-native platforms.

Frameworks such as COBIT and NIST CSF include categories for continuous monitoring, auditability, and risk management, but the methods to operationalise these concepts remain vague. A lack of tooling and integration between governance declarations and infrastructure-as-code practices further contributes to gaps [41][42]. Without embedding governance into code and pipelines, organisations rely on retroactive assessments that miss ephemeral violations.

The proliferation of multi-cloud and hybrid environments also introduces inconsistencies in the control application. Different cloud providers implement similar services with varied security models, and organisations often fail to standardise governance across platforms. This heterogeneity increases cognitive and operational load, weakening governance integrity across the stack [43].

As a result, many governance failures in cloud security are not caused by the absence of policy, but by the inability to enforce controls dynamically across ephemeral and automated environments. While recent approaches aim to embed control logic into the software delivery lifecycle, closing the gap between policy and enforcement, challenges remain. Despite the availability of runtime-aware controls and Policy-as-Code implementations, there is a persistent drift wherein declared policies degrade during automation and infrastructure deployment. This phenomenon, referred to in this paper as governance enforcement drift, describes the widening gap between formally defined governance policies and their actual enforceability within cloud-native systems. Figure 1 visualises this drift across layers of abstraction, from framework declarations to deployment pipelines and runtime execution.

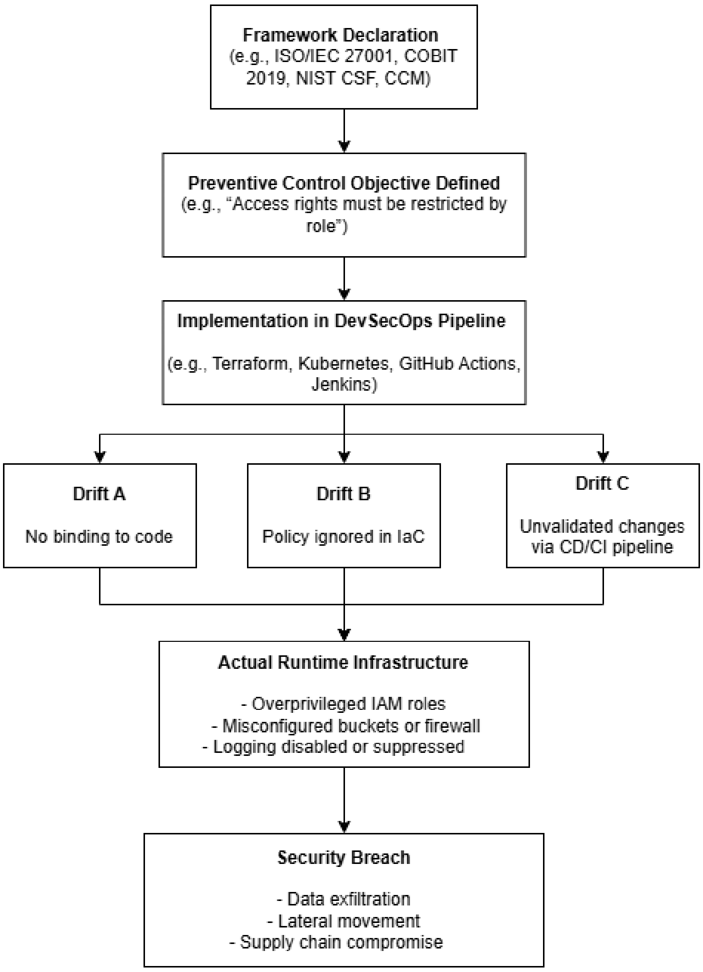
Framework Declaration
(e.g., ISO/IEC 27001, COBIT 2019, NIST CSF, CCM)
Preventive Control Objective Defined
(e.g., "Access rights must be restricted by role")
Implementation in DevSecOps Pipeline
(e.g., Terraform, Kubernetes, GitHub Actions, Jenkins)
Drift A
No binding to code
Drift B
Policy ignored in IaC
Drift C
Unvalidated changes via CD/CI pipeline
Actual Runtime Infrastructure
- Overprivileged IAM roles
- Misconfigured buckets or firewall
- Logging disabled or suppressed
Security Breach
- Data exfiltration
- Lateral movement
- Supply chain compromise

Fig 1. Governance Enforcement Drift in Cloud-Native Systems

Control declarations in frameworks such as ISO 27001, COBIT, NIST CSF, and CSA CCM are often weakened during implementation. Drifts occur when policies are not embedded as code, are bypassed by automation, or are misaligned with runtime infrastructure, leading to enforcement failure and eventual security breach.

2.4. Toward Runtime and Lifecycle-Aware Controls

To address the growing disparity between control documentation and actual security posture, there is a shift in the literature and industry toward runtime and lifecycle-aware governance mechanisms. These approaches aim to embed policy enforcement directly into system behaviours and deployment pipelines, rather than relying on periodic audits or static declarations [44][45].

Governance-as-Code (GaC) has emerged as a promising paradigm in this context. Analogous to Infrastructure-as-Code (IaC), GaC treats governance policies as programmable artefacts that can be versioned, tested, and enforced automatically during system lifecycle events. This model enables controls such as IAM policy constraints, compliance validation, and workload segmentation to be applied dynamically in CI/CD pipelines [46]. Additionally, Policy-as-Code (PaC) tools, such as Open Policy Agent (OPA) and HashiCorp Sentinel, allow for declarative policy definitions that can be embedded within infrastructure orchestration tools like Terraform, Kubernetes, and Helm. Studies show that such integrations reduce configuration drift and increase compliance consistency across distributed cloud environments [47][48].

Another key aspect is behavioural baselining, wherein runtime monitoring systems model expected workload behaviour and flag deviations as potential policy violations. This is particularly useful in detecting lateral movement or privilege misuse that would otherwise bypass static access control models. Tools combining observability with governance (e.g., Falco, Datadog Compliance Monitoring) are increasingly used to enforce real-time alerts and automated responses [49].

However, challenges remain in achieving fine-grained and scalable governance in highly dynamic multi-cloud systems. Existing frameworks still lack the semantic structures needed to support policy abstraction across heterogeneous cloud providers, and integration with software-defined perimeters (SDPs) or service meshes is still emerging [50][51].

Despite these limitations, the movement toward automatable and runtime-validatable controls represents a critical step toward securing cloud-native environments. Rather than treating governance as an overlay, these approaches embed control logic into the software delivery lifecycle, closing the gap between policy and enforcement.

3. Methods

This study employs a meta-synthetic approach to analyse patterns of governance control failure in cloud-native security breaches from 2021 to 2024. The methodology integrates qualitative content analysis with comparative framework mapping, allowing a structured synthesis of publicly available breach data against preventive control categories within major IT governance frameworks.

A total of 52 documented cloud-related security incidents were extracted from authoritative secondary sources, including technical postmortems, threat intelligence reports, and cybersecurity advisories published by organisations such as the Cloud Security Alliance, ENISA, Unit42, and IBM Security. To ensure consistency and relevance, only incidents meeting the following criteria were included: (1) involved cloud-native infrastructure (e.g., containerised workloads, IaC-based deployments, CI/CD environments), (2) reported with sufficient technical detail to determine the proximate cause, and (3) publicly disclosed between January 2021 and March 2024.

In addition to the above inclusion criteria, incidents were screened for source credibility and cross-verifiability. Only reports published by authoritative entities (e.g., IBM X-Force, ENISA, CSA, Unit42) or corroborated by multiple technical write-ups were retained. Incidents lacking technical depth, clarity of root cause, or traceable evidence chains were excluded. For each included case, triangulation was performed using vendor advisories, threat intelligence platforms, and independent forensic blogs to ensure the validity of the incident narrative and minimise interpretive bias.

For each incident, root causes were identified and classified into one or more of four predefined preventive control domains: Identity and Access Management (IAM), Configuration and Deployment Management, Runtime Monitoring, and Third-Party/Supply Chain Governance. These categories reflect recurring areas of concern in both academic literature and industry frameworks. Each classified incident was then cross-referenced against control objectives declared in ISO/IEC 27001:2022, NIST Cybersecurity Framework v2.0, COBIT 2019, and the CSA Cloud Controls Matrix v4.0 to evaluate alignment between the stated controls and the failed implementations. Control failures were tallied and visualised to reveal dominant breach patterns and areas of control redundancy, vagueness, or ineffectiveness. For example, incidents caused by IAM misconfigurations were mapped to specific sub-controls in ISO/IEC 27001 Annexe A (e.g., A.9 Access Control) and evaluated for coverage versus enforceability gaps. Observed discrepancies were interpreted as failures in preventive IT general control design or operationalisation.

Despite efforts to ensure data integrity through strict inclusion criteria and multi-source triangulation, this study is subject to certain limitations. It relies exclusively on publicly disclosed incidents, which may introduce reporting bias, especially for breaches undisclosed by vendors or under NDA. The absence of internal audit data and confidential forensic reports constrains the depth of root cause verification. Additionally, the classification into preventive control domains, while grounded in framework semantics, involves interpretive judgment. These limitations, though inherent to qualitative cybersecurity research, are mitigated through rigorous source validation and consistent analytical coding.

4. Result and Discussion

4.1. Overview of Control Failure Distribution

The analysis of 52 documented cloud-native security incidents reveals consistent failure patterns across a limited set of preventive control domains. Table 1 summarises the distribution of breach causes across seven domains aligned with traditional ITGCs: Identity and Access Management (IAM), Configuration and Deployment, Runtime Monitoring, Third-Party/Supply Chain Governance, Data Lifecycle Controls, API Governance, and Infrastructure Automation Controls. While IAM and Configuration/Deployment dominate the dataset, failures in newer domains, such as API and automation governance, reflect the growing complexity and interdependence of modern cloud systems.

Table 1. Distribution of Preventive Control Failures Across Cloud-Native Breaches (2021–2024)

Control Domain	No. of Incidents	Primary Failure Type	Representative Framework Coverage
Identity & Access Management (IAM)	15	Excessive privileges, static credentials, weak MFA	ISO 27001 A.9, COBIT BAI08, NIST ID.AM
Configuration & Deployment	12	Public S3 buckets, unvalidated changes, drifted IaC	ISO A.12, NIST PR.IP, CSA CCM IVS
Runtime Monitoring	9	Absent logging, no anomaly detection, delayed alerting	ISO A.16, COBIT DSS05, CSA CCM SEF01
Third-Party / Supply Chain	6	CI/CD compromise, token leakage, vendor backdoors	NIST ID.SC, CSA A&A-01, ISO A.15
Data Lifecycle Protection	5	Orphaned data, lack of encryption at rest, retention gaps	ISO A.8.3, NIST PR.DS, COBIT DSS01
API Governance	3	Unauthenticated endpoints, no rate limiting, exposed keys	CSA CCM IVS05, NIST PR.AC, COBIT DSS06
Infrastructure Automation	2	Script reuse, unscoped automation, insecure defaults	ISO A.12.1.2, COBIT BAI03, CSA IVS06

As summarised in Table 1, control failures were most prevalent in Identity and Access Management (IAM), followed by configuration-related issues and deficiencies in runtime monitoring. To aid visual interpretation of this distribution, Figure 2 below illustrates the proportional breakdown of failure domains based on incident frequency across the sample.

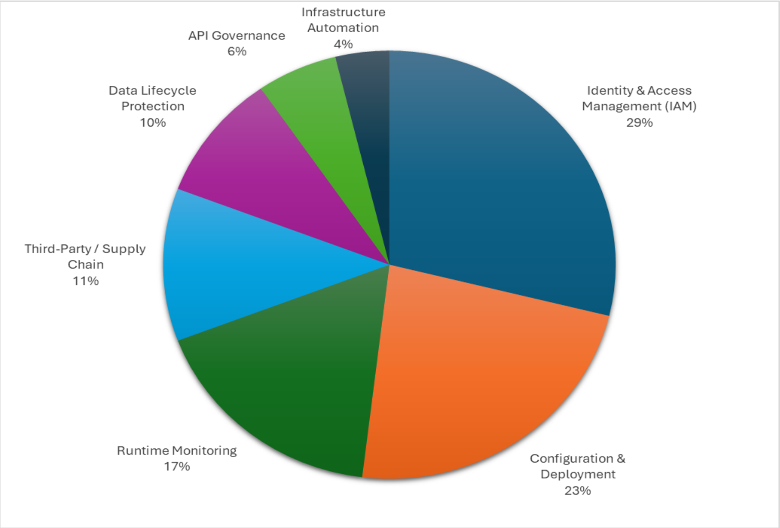

Fig 2. Proportional Distribution of Preventive Control Failures across Cloud-Native Security Incidents (2021–2024)

This visualisation reinforces the quantitative finding that governance failures are not evenly distributed across domains, but rather concentrated in a few systemic weak points. The dominance of IAM and configuration-related breaches suggests a structural fragility in how access controls and deployment policies are operationalised in cloud-native contexts. Despite being explicitly covered in all major frameworks, these domains continue to suffer from implementation drift and a lack of automated enforcement. The relative scarcity of incidents in other categories, such as API governance or infrastructure automation, may indicate either improved tooling in those areas or lower visibility due to underreporting, pointing to possible blind spots in breach disclosure practices.

IAM-related misconfigurations emerged as the most frequent cause, often manifesting in the form of overprovisioned roles or exposed credentials embedded in deployment pipelines. Although ISO/IEC 27001 and NIST CSF prescribe controls for access governance, these breaches occurred even in "compliant" environments, highlighting a gap between declared policies and their enforceable presence.

Configuration and deployment issues, ranging from overly permissive security groups to unsynchronised IaC changes, comprised the second most prevalent category. These failures frequently stemmed from inconsistencies between declared configuration baselines and real-time infrastructure states, underscoring a critical misalignment between static governance documentation and the velocity of change in DevOps pipelines.

Runtime monitoring failures involved disabled logging, improperly scoped detection policies, or alerting delays due to suppressed thresholds during scale testing. Although frameworks like COBIT DSS05 and CSA SEF01 address the need for real-time observability, they lack actionable enforcement mechanisms, leading to control gaps.

Third-party and supply chain-related compromises, though fewer, had a significant impact, including CI/CD token leakage and poisoned builds. These risks challenge the perimeter-based assumptions of older governance models and demand granular enforcement at the software supply chain level.

Many incidents reflected cascading failures across multiple domains. For example, an over-permissioned IAM role allowed for unauthorised modification of deployment scripts, which disabled runtime monitoring. Similarly, unscoped automation scripts deployed misconfigured API endpoints. These relationships highlight that preventive ITGC failures often do not exist in isolation; misalignment in one domain weakens the posture of others. Traditional frameworks rarely account for such dependencies, often assessing controls in silos.

4.2. Incident Typologies and Framework Mapping

A closer inspection of incident narratives reveals recurring breach archetypes that challenge the enforceability of traditional frameworks:

1. IAM Breach Archetype: A financial SaaS provider suffered lateral privilege escalation after exposing AWS credentials in a CI pipeline. Though ISO A.9 mandates access control, the real enforcement failed because static IAM policy definitions were not tied to automation events.
2. Configuration Drift Incident: A technology firm deploying a hybrid cloud infrastructure encountered firewall misconfigurations due to unsynchronised Terraform modules. Despite recent ISO audit compliance, actual runtime states had diverged, reflecting a lack of drift detection and reconciliation enforcement.
3. Runtime Blindspot Scenario: An e-commerce company suffered a data breach when container escapes went undetected due to suppressed alerting during performance tests. Although frameworks require continuous monitoring, there was no guardrail ensuring the logging configuration persisted across environments.
4. Supply Chain Propagation Case: A popular open-source package distribution was hijacked through a compromised third-party CI provider. This breach propagated through dozens of consumer systems despite all vendors appearing "approved" under ISO A.15 risk classifications.
5. Automation Misuse Event: An internal deployment script reused insecure default configurations across environments, accidentally exposing staging credentials to the internet. While COBIT BAI03 mandates change validation, the controls failed to accommodate rapid automation cycles.

These examples illustrate how documented controls are regularly bypassed, not because they are absent, but because they lack operational binding. Audit-driven declarations (e.g., “enable MFA,” “encrypt data at rest”) are often non-verifiable in real time unless implemented via code.

Frameworks like ISO 27001, NIST CSF, and COBIT provide comprehensive catalogues of what should be secured, but rarely how to do so in automated, ephemeral environments. For example, ISO A.12 references “control of software installation” yet does not guide container image validation or runtime signature enforcement. Similarly, COBIT's DSS05 emphasizes monitoring but does not offer architectural blueprints for implementing observability in distributed microservices.

CSA's Cloud Controls Matrix offers better alignment with cloud-native concepts, yet adoption remains uneven. Fragmentation across frameworks further complicates governance: duplicated objectives with differing terminologies increase audit fatigue and reduce clarity. Organisations must often choose between multiple overlapping frameworks, none of which directly translate to policy-as-code enforcement without manual interpretation.

4.3. Implications for Enforceable Governance Design

The patterns identified in breach incidents reveal a central issue: while preventive IT General Controls (ITGCs) are well-represented in standards and frameworks, their implementation in cloud-native environments often lacks enforceability. These environments are characterised by rapid changes, ephemeral resources, and decentralised deployment pipelines, factors that undermine traditional governance models grounded in periodic reviews and static documentation.

In such settings, preventive controls must be continuously validated and context-aware. Frameworks like ISO 27001, COBIT, and NIST CSF offer valuable guidance, but their abstract control objectives are insufficient unless translated into mechanisms that can adapt to real-time infrastructure states. For instance, access control policies are ineffective if they do not account for automated role creation within CI/CD pipelines, and encryption mandates lose significance if data lifecycle monitoring is absent across microservice boundaries.

To remain relevant, governance mechanisms must shift from descriptive to executable. This means encoding control policies directly into system workflows, such that compliance is evaluated not periodically but at each point of infrastructure change. Known as Governance-as-Code, this approach leverages policy engines and integration hooks to apply guardrails automatically during development, deployment, and runtime phases. Examples include validating infrastructure-as-code against compliance baselines, embedding access policy checks into Git workflows, or enforcing logging standards through deployment configurations.

Crucially, this approach reframes governance not as a separate layer or audit deliverable, but as an operational function embedded in the software lifecycle. It allows organisations to operationalise the intent of existing frameworks without relying solely on manual oversight or post-incident reviews. In doing so, Governance-as-Code addresses the implementation gap that has led many preventive ITGCs to fail when applied to fast-moving, cloud-native architectures.

This shift also challenges the traditional perception of ITGCs as static benchmarks. Instead, it redefines them as dynamic, executable artefacts, interwoven into infrastructure behaviour and enforced continuously. Bridging this enforcement gap is critical to modernising governance for distributed, automated, and scalable systems.

5. Conclusion

This research has examined the persistent failure of preventive IT general controls (ITGC) in the context of cloud native systems, highlighting how traditional security governance approaches have not adapted adequately to dynamic infrastructure, DevOps practices, and declarative system design.

The meta-synthetic analysis reveals that despite the implementation of widely recognised frameworks such as ISO 27001, NIST CSF, and COBIT, security incidents continue to emerge in the form of misconfigurations, identity misuse, and runtime control failures. A key insight is that preventive controls are often misaligned with the automation-centric and distributed nature of modern systems, resulting in governance becoming more performative than functional.

Significantly, the findings demonstrate that while cloud native environments offer scalable, flexible architectures, they simultaneously introduce complexity that undermines static control models. Governance as Code (GaC) and Policy as Code (PaC) approaches represent promising advancements but face challenges in adoption, tooling fragmentation, and runtime verification [19][20][46]. The need for real-time, infrastructure-aware control logic continues to grow, especially as the threat landscape becomes more cloud native itself [54][39].

Moreover, the mismatch between DevSecOps practices and IT governance structures creates an accountability vacuum. Control failures are no longer solely technical, but also structural and procedural, a reflection of the friction between agile development lifecycles and rigid governance enforcement [42][10].

In sum, the paper concludes that the failure of preventive controls is not only a result of implementation gaps but stems from a fundamental disconnect between governance design and modern system behaviour.

References

[1] S. Garg, S. Bawa, and J. Singh, "Cloud computing security: Attacks, threats, and solutions," Future Generation Computer Systems, vol. 117, pp. 579–598, 2021. https://doi.org/10.1016/j.future.2020.12.001

[2] P. Patel, A. Ramachandran, and S. N. Srirama, "A Cloud-Native Approach for Scalable Multi-Tenant Applications Using Kubernetes," IEEE Access, vol. 9, pp. 11722–11736, 2021. https://doi.org/10.1109/ACCESS.2021.3051463

[3] N. Gruschka et al., "Privacy Issues and Data Protection in Big Data: A Case Study Analysis under GDPR," Computer Law & Security Review, vol. 37, p. 105405, 2020. https://doi.org/10.1016/j.clsr.2020.105405

[4] M. Ali, M. S. Akbar, and M. Usman, "Security in Cloud of Things: Integrating Cloud and IoT securely," Journal of Network and Computer Applications, vol. 168, p. 102761, 2020. https://doi.org/10.1016/j.jnca.2020.102761

[5] International Organization for Standardization, ISO/IEC 27001:2022 - Information Security, Cybersecurity and Privacy Protection. Geneva: ISO, 2022.

[6] National Institute of Standards and Technology, Framework for Improving Critical Infrastructure Cybersecurity, Version 2.0, 2024. https://www.nist.gov/cyberframework

[7] ISACA, COBIT 2019 Framework: Governance and Management Objectives, Rolling Meadows, IL: ISACA, 2019.

[8] ENISA, Threat Landscape 2023 – Cloud Threats, European Union Agency for Cybersecurity, 2023. https://www.enisa.europa.eu

[9] IBM Security, Cost of a Data Breach Report 2023, IBM Corp., 2023. https://www.ibm.com/reports/data-breach

[10] A. R. Sampaio, L. M. Silva, and H. Madeira, "Security Challenges and Opportunities of DevSecOps: A Systematic Literature Review," Journal of Systems and Software, vol. 195, p. 111555, 2023. https://doi.org/10.1016/j.jss.2022.111555

[11] H. Kazim and V. Zhu, "Security and Privacy in DevOps: A Multivocal Literature Review," Information and Software Technology, vol. 143, p. 106751, 2022. https://doi.org/10.1016/j.infsof.2021.106751

[12] S. Youssef and N. Abouzakhar, “Security Governance in Cloud Computing: A Literature Review,” Procedia Computer Science, vol. 177, pp. 325–331, 2020. https://doi.org/10.1016/j.procs.2020.10.045

[13] H. Takabi, “Modern Security Governance for Cloud-Native Systems,” IEEE Cloud Computing, vol. 10, no. 1, pp. 65–74, Jan. 2023. https://doi.org/10.1109/MCC.2023.3238984

[14] Center for Internet Security, Cloud Security Configuration Guide v2.0, CIS, 2023. https://www.cisecurity.org

[15] Cloud Security Alliance, Top Threats to Cloud Computing: Pandemic Eleven, CSA, 2022. https://cloudsecurityalliance.org

[16] Palo Alto Networks Unit42, Cloud Threat Report 2H 2023, 2023. https://unit42.paloaltonetworks.com

[17] D. Chatterjee, S. Ghosh, and A. N. Mitra, “Secure Cloud Governance in Agile IT Landscapes,” ACM Computing Surveys, vol. 55, no. 3, pp. 1–36, 2023. https://doi.org/10.1145/3512766

[18] F. Hussain, S. A. Hussain, and A. Hassan, “Cybersecurity in Cloud Systems: A Governance and Compliance Perspective,” Computers & Security, vol. 125, p. 102959, 2023. https://doi.org/10.1016/j.cose.2022.102959

[19] R. Widyastuti and M. A. Putra, “Governance-as-Code for Compliance Enforcement in Cloud-Native Systems,” International Journal of Information Management Data Insights, vol. 2, no. 2, p. 100120, 2022. https://doi.org/10.1016/j.jjimei.2022.100120

[20] S. Purdy and C. Madden, “Runtime Governance of Cloud-Native Architectures Using Policy-as-Code,” Journal of Cloud Computing, vol. 12, no. 1, pp. 1–15, 2023. https://doi.org/10.1186/s13677-023-00412-z

[21] M. A. Rahman and F. Alotaibi, “A Comparative Study of IT Governance Frameworks in the Era of Digital Transformation,” International Journal of Information Management, vol. 61, p. 102435, 2021. https://doi.org/10.1016/j.ijinfomgt.2021.102435

[22] A. De Haes, W. Van Grembergen, and R. S. Debreceny, “COBIT 5 and Enterprise Governance of Information Technology: Building Blocks and Research Opportunities,” Journal of Information Systems, vol. 34, no. 2, pp. 233–259, 2020. https://doi.org/10.2308/isys-52650

[23] M. Sookhak et al., “Cloud-native security: State-of-the-art and research directions,” Journal of Systems Architecture, vol. 112, p. 101836, 2021. https://doi.org/10.1016/j.sysarc.2020.101836

[24] R. Ismail and T. Almunawar, “Revisiting COBIT in Cloud-Based IT Environments,” Procedia Computer Science, vol. 179, pp. 673–680, 2021. https://doi.org/10.1016/j.procs.2021.01.058

[25] N. A. Rizal and N. H. Zakaria, “Cloud Service Governance: Integration of COBIT and ISO Standards,” Journal of King Saud University - Computer and Information Sciences, vol. 35, no. 3, pp. 341–348, 2023. https://doi.org/10.1016/j.jksuci.2021.03.001

[26] M. Chamola et al., “Security and Privacy Issues in Modern Cyber-Physical Systems: Challenges and Solutions,” IEEE Access, vol. 9, pp. 29230–29265, 2021. https://doi.org/10.1109/ACCESS.2021.3058533

[27] Cloud Security Alliance, Cloud Controls Matrix v4.0, 2021. https://cloudsecurityalliance.org

[28] H. Assal and S. Chiasson, “Security in the Cloud: A User-Centric Threat Taxonomy,” ACM Transactions on Internet Technology, vol. 21, no. 3, pp. 1–23, 2021. https://doi.org/10.1145/3446282

[29] N. U. Hassan and S. A. Madani, “Cloud Misconfiguration: Origins, Detection Techniques, and Future Research Directions,” Future Generation Computer Systems, vol. 128, pp. 239–253, 2022. https://doi.org/10.1016/j.future.2021.10.015

[30] Palo Alto Networks Unit 42, Cloud Threat Report: Misconfiguration Risks and Real-World Exploits, 2023. https://unit42.paloaltonetworks.com

[31] A. Ferrari, E. Russo, and M. Mori, “Authentication Pitfalls in Cloud APIs: A Systematic Review,” Journal of Cloud Computing, vol. 11, no. 1, pp. 1–19, 2022. https://doi.org/10.1186/s13677-022-00291-x

[32] ENISA, Threat Landscape for Supply Chain Attacks, European Union Agency for Cybersecurity, 2022. https://www.enisa.europa.eu

[33] M. Owaida, “API Security in Cloud-Native Applications: A Systematic Mapping Study,” Journal of Systems and Software, vol. 190, p. 111363, 2022. https://doi.org/10.1016/j.jss.2022.111363

[34] L. Garcia, T. L. Alves, and E. B. Marques, “Cloud Monitoring and Forensics: Challenges and Emerging Directions,” Digital Investigation, vol. 38, p. 30121, 2021. https://doi.org/10.1016/j.diin.2021.301121

[35] S. Pearson and M. Sebastian, “Bridging the Governance Gap in Cloud Security,” Computer Law & Security Review, vol. 40, p. 105618, 2021. https://doi.org/10.1016/j.clsr.2021.105618

[36] P. K. Sharma and J. H. Park, “Blockchain-based Distributed Framework for Secure and Trustworthy Data Governance in Cloud Environments,” IEEE Access, vol. 9, pp. 134090–134103, 2021. https://doi.org/10.1109/ACCESS.2021.3115904

[37] H. Liu, Y. Wang, and Q. Li, “Governance-as-Code: Automating Policy Enforcement in DevOps Pipelines,” Future Generation Computer Systems, vol. 127, pp. 345–358, 2022. https://doi.org/10.1016/j.future.2021.09.003

[38] M. Sharif, A. Khan, and I. Khan, “Compliance Enforcement in Cloud Deployments: Policy, Tools, and Challenges,” Journal of Cloud Computing, vol. 10, no. 1, p. 33, 2021. https://doi.org/10.1186/s13677-021-00250-z

[39] Cloud Security Alliance, Cloud Threats and Incidents Report 2023, 2023. https://cloudsecurityalliance.org

[40] IBM Security, X-Force Threat Intelligence Index 2024, IBM Corp., 2024. https://www.ibm.com/reports/xforce

[41] K. Almutairi, H. B. Hashem, and A. Almogren, “Operationalizing Cloud Security Controls: From Frameworks to Code,” Computers & Security, vol. 123, p. 102916, 2023. https://doi.org/10.1016/j.cose.2022.102916

[42] G. Smith and R. Kumar, “DevSecOps Misalignment with Governance Frameworks: A Case Study,” Journal of Software: Evolution and Process, vol. 34, no. 2, p. e2367, 2022. https://doi.org/10.1002/smr.2367

[43] A. L. Pinto and J. Rosado, “Multi-Cloud Governance Challenges: A Systematic Literature Review,” Information and Software Technology, vol. 139, p. 106698, 2021. https://doi.org/10.1016/j.infsof.2021.106698

[44] K. Hashmi and N. Mavridis, “Security Policy Enforcement in DevOps: Toward Governance-as-Code,” Computers & Security, vol. 114, p. 102577, 2022. https://doi.org/10.1016/j.cose.2021.102577

[45] S. B. Hill and T. Mahoney, “Security Automation in the Age of DevSecOps: A Survey of Runtime Governance Models,” Journal of Cloud Computing, vol. 10, p. 44, 2021. https://doi.org/10.1186/s13677-021-00263-8

[46] E. Adegbite, H. Takabi, and A. X. Liu, “Governance-as-Code: Automating Information Security Governance in Cloud Infrastructure,” Journal of Information Security and Applications, vol. 63, p. 103045, 2022. https://doi.org/10.1016/j.jisa.2021.103045

[47] N. Xiong and A. R. Butt, “Policy-as-Code: Security Policy Enforcement for Cloud Deployments Using OPA,” IEEE Transactions on Cloud Computing, early access, 2023. https://doi.org/10.1109/TCC.2023.3248755

[48] R. Kalman and M. Morillo, “Infrastructure Compliance via Declarative Policies: Empirical Evidence from Kubernetes,” Future Generation Computer Systems, vol. 136, pp. 142–153, 2023. https://doi.org/10.1016/j.future.2022.06.012

[49] A. Meidan et al., “Behavioral Governance in Cloud Environments: From Monitoring to Enforcement,” ACM Transactions on Cyber-Physical Systems, vol. 6, no. 3, pp. 1–24, 2022. https://doi.org/10.1145/3435781

[50] C. Rong, Z. Y. Tan, and H. Jin, “Semantic Governance for Multi-Cloud Architectures,” IEEE Transactions on Services Computing, vol. 16, no. 1, pp. 158–169, 2023. https://doi.org/10.1109/TSC.2021.3061374

[51] L. H. Vu and J. Kim, “Governance-Aware Service Mesh: Toward Secure Microservice Communication,” Journal of Systems Architecture, vol. 128, p. 102482, 2022. https://doi.org/10.1016/j.sysarc.2022.102482

[52] Cloud Security Alliance, Top Threats to Cloud Computing: Navigating the Era of Cloud Complexity, CSA, 2023. https://cloudsecurityalliance.org

[53] ENISA, Threat Landscape 2023 – Cloud Infrastructure Incidents, European Union Agency for Cybersecurity, 2023. https://www.enisa.europa.eu

[54] Palo Alto Networks Unit42, Cloud Threat Report 1H 2024, 2024. https://unit42.paloaltonetworks.com

[55] A. Martin and P. Jamal, “Mapping Preventive Control Failures in Cloud Security Incidents,” Journal of Cybersecurity and Privacy, vol. 3, no. 1, pp. 24–38, 2023. https://doi.org/10.3390/jcp3010003