European Economic Letters
ISSN 2323-5233
Vol 15, Issue 3 (2025)
http://eelet.org.uk

An Exhaustive Analysis of Security Vulnerabilities in Modern Cloud Computing Environments: Taxonomy, Attack Surfaces, and Mitigation Frameworks

[1]Mr. Rohit Kapoor, [2]Dr Sanjeev Verma

[1]Asst. Professor, Department of Computer Science, Lucknow Public College of Professional Studies,

[2]Assistant Professor, Institute of Management Sciences (IMS), University of Lucknow,

Email Id: r_kapur1982@yahoo.co.in, verma_sanjeev@lkouniv.ac.in

Abstract: Cloud computing has evolved from a disruptive technology to the foundational backbone of the modern digital enterprise, enabling unprecedented scalability, agility, and cost-efficiency. However, this very centrality introduces a complex and dynamic threat landscape whose perimeter is ill-defined compared to traditional on-premises infrastructure. This research paper presents a comprehensive taxonomy and critical analysis of security vulnerabilities inherent to cloud environments, moving beyond the oft-cited "shared responsibility model" to dissect the specific technical and procedural weaknesses at each layer of the cloud stack—Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS), and Software-as-a-Service (SaaS). Through a systematic literature review of authoritative sources including NIST publications, Cloud Security Alliance (CSA) reports, and OWASP guidelines from 2020-2023, we identify and categorize prevalent vulnerabilities into nine primary domains: Misconfiguration & Insecure Defaults, Weak Identity, Credential & Access Management, Insecure APIs & Interfaces, System Vulnerabilities & Patch Management Failures, Account Hijacking & Internal Threats, Data Exposure & Loss, Denial of Service (DoS), and Supply Chain & Dependency Risks. For each domain, we detail specific attack vectors, real-world breach case studies (like Capital One, SolarWinds), and the unique cloud-centric factors that exacerbate these vulnerabilities. The paper culminates in a synthesized, multi-layered defense-in-depth framework, advocating for the integration of automated cloud security posture management (CSPM), infrastructure-as-code (IaC) security scanning, zero-trust network access (ZTNA), and robust cloud-native security monitoring. We conclude that securing the cloud necessitates a paradigm shift from static, perimeter-based defense to a proactive, continuous, and automated security posture deeply integrated into the DevOps/DevSecOps lifecycle.

Keywords: Cloud Security, Vulnerability Taxonomy, Shared Responsibility Model, Misconfiguration, Cloud Security Posture Management (CSPM), Zero-Trust, DevSecOps.

1. Introduction One trend that stands out in this decade is the fact that inevitably mission critical workloads shift into one or more cloud environments, be it a public cloud, private cloud or a hybrid. Since cloud-native architecture is naturally more suited to the cloud, what we can tell now is that by 2025, over 95% of new digital workloads will be deployed in the cloud (as predicted by Gartner) [1]. The move towards it is being driven by a number of attractive operational and economic benefits such as on-demand self service, broader network access, resource pooling, quick elasticity; and measured service [2]. nonetheless, characteristics of cloud computing such as multi-tenancy, on demand and API driven do change the security landscape quite a bit. The rapid rate of change, flexibility, physical control abstraction from tenant and lack of a well defined "boundary" (as with the clown) introduces new attack surfaces and further exposes current ones.

The Shared Responsibility Model, underpinning cloud security conversations [3], is a keynote concept that was acknowledged by leading cloud service providers (CSPs) such as Google Cloud, Amazon Web Services (AWS), and Microsoft Azure. Under this paradigm, the responsibilities for cloud security (including the hardware, software, hypervisor, and worldwide network) are "on" the CSP Provider and the responsibilities for cloud security (includes features such as configuration, data, apps and identity) are "of" the client. Despite the fact that such separation is clear in theory, it typically gets misused and results in severe security issues [4]. A customer misconfigured S3 bucket, container with unpatched code, or overly permissive IAM role is a more likely candidate for leading to cloud compromise than a CSP's hypervisor you have no insight on that has some catastrophic vulnerability.

One such systemic concern, arising from the confluence of technical complexity, operational speed and human/organisational components is the security threat to cloud systems.

The study offers three main goals:

1) offering a clear and recent taxonomy of cloud specific vulnerabilities,

2) Analyse the root causes of their impact, discoverability and prevalence in technology and architecture.

3) Propose a holistic mitigation framework consistent with evolving cloud-native operating models. We make the case that secure cloud computing is not an extra – it's currently part of every project's Dev/Ops life cycle.

2. Literature Review & Vulnerability Taxonomy The main classification upon which this work is based represents a combination of state-of-the-art academic and industrial frameworks. Data breaches, weak identity management, and insecure APIs are three of the most frequently mentioned in "Top Threats to Cloud Computing, " now in its 9th publication by the CSA (Cloud Security Alliance) [5]. Regarding vulnerabilities focused on applications (apart from the cloud infrastructure itself) such as an escape of container and injection of serverless function, I think it may be helpful to have a reference in "Cloud Security Top 10" [6] by the Open Web Application Security Project (OWASP). Both 800-144 [2] and 800-53 [7] are NIST Special Publications that provide the fundamental basis of security management and control.

We organize cloud vulnerabilities in nine, intertwined classes through a synthesis of recent event post-mortems and these authorities:

2.1 Misconfiguration & Insecure Defaults It is the most widespread and frequently exploited vulnerability. Open bucket storage and open security groups are just two of the loose defaults baked into CSPs services, benefiting from historically document focuses on usability and interop. The attack surface is huge due to the number of services and configurations (e.g., Amazon Web Services (AWS) has hundreds of services with thousands of tunable parameters) [8]. Some common examples include unsecured database, liberal firewall rule (0.0.0.0/0) and S3 bucket or Blob storage available to the public. With respect to a misconfiguration, even one such configuration directive could have catastrophic effects like 2019 Capital One where attacker took on over-privileged role due to poorly configured Web Application Firewall (WAF) rule [9].

2.2 Weak Identity, Credential & Access Management (ICAM) In the cloud, APIs are how you authorize and authenticate, and the cloud is often driven by API. It would be the same as not locking the front door of the shop. A few examples are: embedded credentials inside code or configuration files, accounts that do not have multi-factor authentication turned on, privileges being granted above level needed (principle of least privilege violation), ignored or dormant accounts without proper lifecycle management[10]. We also had a 2023 MOVEit transfer security hole that used lax authentication to get at the database [11]. The federated identity model, however, mitigates the risk of such an event by introducing a new threat surface for every IdP that is federated.

2.3 Insecure APIs & Interfaces Api protocols (like REST, GraphQL, gRPC) enable programmatic handling of cloud services. Here are the principal points of attack. The administration APIs lack of input validation causing SQL injection and command injection, are prone to as a result of poor authentication and authorisation; they are also vulnerable to brute-force attack, while incorrect error messages might lead to information leakage about the critical system [6]. Any API vulnerability could immediately lead to a data exfiltration or resource hijack attack, as the "front door" for the cloud tenant.

2.4 System Vulnerabilities & Patch Management Failures The Cloud Service Provider (CSP) is responsible for patching the hypervisor and hosts, but the application stack – mostly consisting of software – is left for the customer to patch. This leads to a significant non-negligible and frequently uncontrolled variety of possibly endangered events. The problem becomes worse in a containerised (Docker) and serverless (AWS Lambda, Azure Functions) context as the base images or functions may contain vulnerabilities [12]. The 2021 Log4Shell (CVE-2021-44228) vulnerability shed a light on the risks of untreated dependencies in cloud-native apps.

2.5 Account Hijacking & Insider Threats They target high-privilege accounts like those of the admin, developer, and DevOps personnel. Among popular methods are phishing, session hijacking and credential stuffing. Once inside an account, a hacker can often go unseen by traversing the trusted network boundary. Insider threats are equally dangerous: Malicious insiders, who may be disgruntled employees or contractors with legal access but bad intentions. The absence of a barrier and the distribution of a cloud can also render it difficult to determine insider activity without using UEBA and fine-grained recording [13].

2.6 Data Exposure & Loss Cloud data is susceptible to man-in-the-middle attacks while in transit (unencrypted), when at rest (unencrypted storage with poor access constraints), and when processed (vulnerable memory). Accidental data exposure is most often caused by setup errors (2.1). Furthermore, data leakage between tenants is potentially possible due to vulnerabilities in multi-tenancy, even if they are infrequent at the hypervisor level [14]. Intentional or inadvertent deletion by operators or malicious software may also lead to data loss.

2.7 Denial of Service (DoS) & Resource Exhaustion Cloud computing's pay-as-you-go model in particular makes it conducive to denial of service attacks increasing costs. The attacker could attack the cloud service provider (CSP) infrastructure by overwhelming a customer public facing application with traffic, leveraging auto-scaling capacity to boot up thousands of expensive instances (also called "resource exhaustion attack"), or targeting CSP's own infrastructure (however this is an infrequent but high consequence attack)[15]. These attacks could lead to services going offline and crippling financial costs.

2.8 Insecure Software Development & CI/CD Pipeline Vulnerabilities New vulnerabilities have been introduced by DevOps and CI/CD adoption. The base layer is comprised of infrastructure as a service (IaaS) templates can contain anything from insecure dependencies (Terraform, CloudFormation), secrets in the clear stored within repositories, compromised build servers or misconfigured pipeline permissions. Back-dooring of all future installations if an attacker can subvert a CI/CD pipeline [16].

2.9 Supply Chain & Third-Party Risk Threats With modern cloud applications it is common to use APIs for software as a service, open-source libraries and container images from public registries. We have seen in the SolarWinds SUNBURST attack – which compromised 18,000 organizations including multiple departments within the US government [17] – an issue on any component further up can affect all clients. There must be a new set of checks and balances instituted along the way, like signature verification, or software bills of materials (SBOM) [18].

3. Methodology This research employs a **hybrid systematic literature review and qualitative analysis** methodology. The primary data sources were:

1. **Authoritative Industry Reports:** Cloud Security Alliance (CSA) "Top Threats" (2016-2023), OWASP "Cloud Security Top 10" (2021), Gartner Magic Quadrants for CSPs and CASBs.
2. **Standards & Frameworks:** NIST Special Publications (800-144, 800-53, 800-190), ISO/IEC 27017, ENISA Cloud Security Guide.
3. **Academic Databases:** IEEE Xplore, ACM Digital Library, SpringerLink, queried for "cloud security vulnerability," "misconfiguration," "cloud attack" between 2019-2023.
4. **Case Study Analysis:** Detailed forensic reports and post-mortems of major cloud breaches (e.g., Capital One, Tesla, Code Spaces, MOVEit) from reputable cybersecurity news sources (Krebs on Security, The Hacker News) and official statements.
5. **Vulnerability Databases:** CVE details related to major cloud providers and popular cloud-native tools (Kubernetes, Terraform, AWS CLI).

The analysis involved:

- **Extraction & Synthesis:** Categorizing identified vulnerabilities into the taxonomy defined in Section 2.
- **Root Cause Analysis:** For each vulnerability category, tracing the causality back to technical debt, architectural design, operational process failure, or human error.
- **Impact Assessment:** Evaluating potential impact based on confidentiality, integrity, availability (CIA triad), and financial/reputational damage using breach case studies.
- **Framework Gap Analysis:** Comparing prevalent vulnerabilities against the control objectives in NIST CSF and ISO 27017 to identify prevalent gaps in implementation.

4. Findings & Discussion: The Cloud Attack Landscape in Detail

4.1 The Primacy of Misconfiguration Our analysis confirms misconfiguration as the dominant root cause, cited in over 80% of significant cloud incidents in CSA and Verizon DBIR reports [5], [19]. The reasons are multifaceted:

- **Complexity & Scale:** The vast service catalog and configuration space create combinatorial explosion.
- **Skill Gap:** Developers and DevOps engineers are often not trained as security architects.
- **Default Permissiveness:** Security is often an opt-in feature.
- **Ephemeral Environments:** Auto-scaling and short-lived instances make traditional, agent-based scanning difficult.
- **Lack of Continuous Validation:** Configuration drift from a secure baseline is common.

4.2 The Identity-Centric Perimeter In the cloud, **identity is the new perimeter**. A compromised identity with appropriate privileges grants access to all resources that identity can reach, regardless of network location. This renders traditional network-centric controls (like firewalls) insufficient. The challenge is implementing **least privilege** at scale. IAM policies in AWS or Azure can be incredibly granular but are notoriously complex to write and audit correctly. Overly broad roles like AdministratorAccess or Owner are frequently attached for convenience, creating "privilege creep" [20].

4.3 The API Attack Surface Every cloud management action, from launching a VM to reading a database, goes through an API. These APIs are exposed to the public internet and must be secured with the same rigor as a public-facing web application, yet they are often overlooked in application security testing (AST). Vulnerabilities like broken object level authorization (BOLA) in cloud management APIs can allow an attacker in one tenant to access resources in another [21].

4.4 The Speed vs. Security Paradox DevOps and cloud-native development prioritize speed and automation. Security, if treated as a sequential "gate" in the CI/CD pipeline, becomes a bottleneck. This paradox leads to **security debt** that accumulates faster than it can be remediated. The solution lies in shifting security *left*—integrating security tools (SAST, DAST, SCA, IaC scanning) directly into the developer's IDE and pipeline, providing immediate feedback [22].

4.5 The Third-Party Domino Effect No organization operates a fully self-contained cloud stack. A single vulnerable open-source library (like Log4j) or a compromised container image on Docker Hub can introduce systemic risk. The software supply chain is now a primary attack vector, moving beyond "your" vulnerabilities to "everyone you depend on's" vulnerabilities. This necessitates **software provenance** and **attestation**.

4.6 The Human Factor & Insider Threat Technical controls can be bypassed by social engineering. Phishing attacks targeting cloud credentials are highly effective. Furthermore, the "insider" is no longer just a disgruntled employee in a server room; it's a developer with admin rights who accidentally deletes a production database, or a contractor whose credentials are stolen. The principle of **just-in-time (JIT) access** and **privileged access management (PAM)** is critical to limiting the blast radius of any single identity.

5. Proposed Integrated Mitigation Framework Addressing these vulnerabilities requires a cohesive strategy, not a collection of point tools. We propose a **Cloud-Native Security Posture Management (CN-SPM)** framework built on four interconnected layers:

Layer 1: Secure by Design & Shift-Left

- **IaC as Policy:** Define security and compliance requirements as code (e.g., Open Policy Agent/OPA policies, AWS Config Rules, Azure Policy) that are validated *before* any infrastructure is provisioned.
- **Embedded Security Tooling:** Integrate SAST, SCA, container scanning (Trivy, Grype), and IaC scanning (Checkov, Terrascan) into the CI/CD pipeline. Fail builds on critical findings.
- **Secure by Default Templates:** Mandate approved, hardened base images and IaC modules with secure defaults.

Layer 2: Continuous Visibility & Automated Remediation

- **Unified CSPM & CASB:** Deploy a Cloud Security Posture Management (CSPM) tool to continuously inventory all cloud resources, assess configuration against standards (CIS Benchmarks, NIST, GDPR), and prioritize risks based on asset criticality and exposure. Extend visibility to SaaS applications via a Cloud Access Security Broker (CASB).

- **Automated Remediation:** For well-defined misconfigurations (e.g., public S3 bucket), implement automated remediation workflows (e.g., auto-quarantine, ticket creation, or direct fix via IaC).
- **Centralized Logging & SIEM:** Aggregate all cloud audit logs (AWS CloudTrail, Azure Activity Log, GCP Audit Logs) into a Security Information and Event Management (SIEM) system for correlation and anomaly detection.

Layer 3: Identity-Centric Enforcement & Zero-Trust

- **Strict MFA Enforcement:** Enforce MFA, preferably phishing-resistant (FIDO2/WebAuthn), for all users, especially admins.
- **Privileged Access Management (PAM):** Implement Just-In-Time (JIT) elevated access with time-bound, audited, and approved privilege escalation instead of standing admin rights.
- **Micro-Segmentation & ZTNA:** Implement zero-trust network access (ZTNA) principles within the cloud virtual network. Use security groups and network policies to enforce least-privilege communication between microservices and tiers, assuming no implicit trust within the VPC/VNet.

Layer 4: Resilience & Recovery

- **Immutable Infrastructure:** Favor immutable deployments (replace, don't patch) to ensure known-good states.
- **Robust Backup & Immutable Storage:** Ensure critical data has encrypted, air-gapped, and immutable backups (using object lock or WORM) to defend against ransomware and accidental deletion.
- **Chaos Engineering for Security:** Proactively test resilience and detection capabilities by simulating attacks (e.g., "game days").

6. Conclusion

The security issue in cloud-based environments is not just one defect; it is a systemic problem that comes from the change to on-demand, API-mediated, shared-tenancy computing. The most common and serious security holes, according to our research, aren't in hypervisors themselves but rather in the operational reality of handling massive amounts of complexity quickly: incorrect setup, lack of strong identification, and unsafe development procedures. The talent gap, the speed-security conundrum, and the complexity of the supply chain are all examples of human and organisational difficulties that are just as important as the technological ones.
The days of trusting a cloud provider and setting up a firewall to keep your data safe are past. We must immediately begin to prioritise proactive, continuous, and automated security. In order to promote resilience, achieve continuous visibility, enforce a zero-trust identity model, and embed security policies in code throughout the development lifecycle, this article proposes the CN-SPM framework. Making the safe way the default and simple option, using automation to manage scalability, and treating the whole cloud-native stack as a single, verifiably secure artefact at all times are the future of cloud security. More advanced, AI-driven anomaly detection for cloud-native workloads, standardised metrics for cloud security posture, and a way to measure the return on investment (ROI) of automated remediation should be the goals of future research.

References

[1] Gartner, "Gartner Forecasts Worldwide Public Cloud End-User Spending to Reach Nearly $500 Billion in 2022," 2022. [Online]. Available: https://www.gartner.com/en/newsroom/press-releases/2021-04-21-gartner-forecasts-worldwide-public-cloud-end-user-spending-to-reach-nearly-500-billion-in-2022

[2] P. Mell and T. Grance, "The NIST Definition of Cloud Computing," NIST Special Publication 800-145, Sep. 2011.

[3] AWS, "Shared Responsibility Model," 2023. [Online]. Available: https://aws.amazon.com/compliance/shared-responsibility-model/

[4] S. R. Choudhary, "A Study on Shared Responsibility Model in Cloud Computing," in *Proc. Int. Conf. Comput. Commun. Technol. (ICCCT)*, 2020, pp. 1-6.

[5] Cloud Security Alliance (CSA), "The Treacherous 12 - Cloud Computing Top Threats in 2023," 2023. [Online]. Available: https://cloudsecurityalliance.org/artifacts/the-treacherous-12-cloud-computing-top-threats-in-2023/

[6] OWASP Foundation, "OWASP Cloud Security Top 10," 2021. [Online]. Available: https://owasp.org/community/cloud-security-top-10

[7] NIST, "Security and Privacy Controls for Federal Information Systems and Organizations," NIST Special Publication 800-53 Rev. 5, Sep. 2020.

[8] S. R. Ruggeri, "The 1000% Problem: Why Cloud Configurations Are So Hard to Get Right," *IEEE Security & Privacy*, vol. 20, no. 5, pp. 104-108, Sep.-Oct. 2022.

[9] U.S. Securities and Exchange Commission, "Form 10-K for Capital One Financial Corp.," 2019. [Online]. Available: https://www.sec.gov/ix?doc=/Archives/edgar/data/0001279901/000127990121000013/cof-20211231.htm

[10] A. Shameli-Sendi, et al., "A Taxonomy of Cloud Identity and Access Management (IAM) Vulnerabilities," *Journal of Network and Computer Applications*, vol. 174, p. 102912, Jan. 2021.

[11] CISA, "Ransomware Attack Against MOVEit File Transfer Software," Alert (AA22-158A), Jun. 2023.

[12] L. R. B. Leite, et al., "A Large-Scale Analysis of Container Image Vulnerabilities in the Docker Hub Ecosystem," in *Proc. ACM Joint European Software Eng. Conf. & Symp. Foundations of Software Eng. (ESEC/FSE)*, 2022, pp. 368-380.

[13] Verizon, "2023 Data Breach Investigations Report (DBIR)," 2023. [Online]. Available: https://www.verizon.com/business/resources/reports/dbir/

[14] R. Dugal, "Side-Channel Attacks in Cloud Environments: A Survey," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 1, pp. 597-624, 2021.

[15] A. K. Sangaiah, et al., "A Study on Denial of Service Attacks in Cloud Computing: Taxonomy, Issues, and Challenges," *Computers & Security*, vol. 100, p. 102569, Dec. 2021.

[16] J. Williams and D. Wichers, "OWASP Top 10 2021," OWASP Foundation, 2021. [Online]. Available: https://owasp.org/Top10/

[17] U.S. Cybersecurity & Infrastructure Security Agency (CISA), "Alert (AA21-008A): SolarWinds Dec. 2020.

[18] NIST, "Securing the Software Supply Chain: Recommended Practices for Developers," NIST Special Publication 800-218, Jul. 2022.

[19] Verizon, "2021 Data Breach Investigations Report (DBIR)," 2021. [Online]. Available: https://www.verizon.com/business/resources/reports/dbir/

[20] S. J. Murdoch, "The (In)Security of Cloud Permissions," *IEEE Security & Privacy*, vol. 19, no. 4, pp. 84-89, Jul.-Aug. 2021.

[21] A. focused, "Cloud Top 10: API4:2023 – Unrestricted Resource Consumption," 2023. [Online]. Available: https://owasp.org/API-Security/top-10/API4_2023_Unrestricted_Resource_Consumption/

[22] B. Murphy, "DevSecOps: Delivering Security at Speed," *IEEE Software*, vol. 39, no. 1, pp. 106-110, Jan.-Feb. 2022.