ScienceXplorer

TYPE Original Research

PUBLISHED 2026

OPEN ACCESS

***CORRESPONDENCE**

E. R., Obakpororo

robertessi4@gmail.com

RECEIVED February, 2026

ACCEPTED April, 2026

PUBLISHED June, 2026

CITATION

Obakpororo, E.R., Omojola, A. F., Phiri, F., Ugwu, D. C., Obaze, G. E., & Justice, O. (2026). Cloud API security and vulnerability assessment: A comprehensive framework for threat detection, risk evaluation, and mitigation strategies. *ScienceXplorer*, 8(3).

CLOUD API SECURITY AND VULNERABILITY ASSESSMENT: A COMPREHENSIVE FRAMEWORK FOR THREAT DETECTION, RISK EVALUATION, AND MITIGATION STRATEGIES

Essi Roberto Obakpororo[1]*, Ayogoke Felix Omojola[2], Flora Phiri[3], Darlington Chukwuma Ugwu[4], Godfrey Eshikhena Obaze[5], Obomejero Justice[6]

[1] *Department of Computer Engineering, Faculty of Engineering, Delta State Polytechnic Otefe Oghara (*Corresponding Author)*

[2]*Department of Cybersecurity, Faculty of Science, Edo State University, Iyamho, Nigeria*

[3]*MS Information Systems Technology, School of Business, The George Washington University, 2121 I St NW, Washington DC 20052, USA*

[4]*Local Manufacturing Division, Africa Centre for Disease Control and Prevention, Addis Ababa, Ethiopia*

[5]*Department of Biochemistry, University of Nigeria, Nigeria*

[6]*Institute of Environmental Engineering, Department of Economics and Natural Resources Management, Peoples' Friendship University of Russia, Moscow, Russia*

Abstract

The proliferation of cloud-based Application Programming Interfaces (APIs) has fundamentally transformed modern software architecture, enabling seamless service integration across distributed environments. However, this rapid adoption has created an expansive attack surface that adversaries increasingly exploit, with API-targeted attacks rising 137% between 2021 and 2023, and 99% of organizations reporting API security problems in 2025. This study presents a systematic vulnerability assessment framework for cloud API security, drawing on empirical analysis of 312 production APIs deployed across Amazon Web Services (AWS), Microsoft Azure, and Google Cloud Platform (GCP). Employing a mixed-methods approach combining automated Dynamic Application Security Testing (DAST), manual penetration testing, static code analysis, and semi-structured interviews with 18 cloud security practitioners, we identified 497 distinct vulnerabilities across ten categories aligned with the OWASP API Security Top 10 (2023). Our findings reveal that 71.8% of assessed APIs exhibited at least one Critical or High-severity vulnerability, with Broken Object Level Authorization (BOLA) and Broken Authentication collectively accounting for over 41% of all identified flaws. We further evaluate the effectiveness of contemporary mitigation controls — including OAuth 2.0 with PKCE, API gateways, Zero Trust architectures, and AI-driven anomaly detection — and propose a tiered remediation roadmap for enterprise cloud environments. This research contributes an evidence-based framework applicable to cloud architects, DevSecOps practitioners, and security auditors seeking to operationalize API security at scale.

Keywords: *cloud API security; vulnerability assessment; OWASP API Top 10; penetration testing; Zero Trust; DevSecOps; OAuth 2.0; DAST; BOLA*

Recommended citation:

[First Author], Omojola, A. F., Phiri, F., Ugwu, D. C., Obaze, G. E., & Justice, O. (2026). Cloud API security and vulnerability assessment: A comprehensive framework for threat detection, risk evaluation, and mitigation strategies. *ScienceXplorer*, 8(2).

1. Introduction

The global cloud computing market surpassed USD 679 billion in 2024 (Gartner, 2024), with Application Programming Interfaces (APIs) forming the connective tissue of virtually every cloud-native application. APIs enable microservices communication, third-party integrations, and data exchange at unprecedented scale, yet they simultaneously constitute one of the most frequently exploited vectors in modern cybercrime (Imperva, 2024). The Open Web Application Security Project (OWASP) released its updated API Security Top 10 in 2023, reflecting a rapidly evolving threat landscape in which API-targeted attacks rose by 137% between 2021 and 2023 (OWASP, 2023). A security-first pipeline analysis published in 2025 documented that CVE disclosure rates grew 38% year-over-year in 2024, with 40,009 CVEs disclosed in that year alone (Shrivastava & Tripathi, 2025).

According to the 2024 Cloud Security Report (Trend Micro / Cybersecurity Insiders, 2024), 42% of organizations experienced security incidents linked to public cloud usage in the preceding year, with API security flagged as a key vulnerability by 49% of surveyed security professionals. The Cloud Security Alliance (CSA, 2024) ranked insecure interfaces and APIs as the third most critical threat to cloud computing. The 42Crunch State of API Security report (2026), based on 200 real-world API breach reports from 2024–2025, confirmed broken authentication as the single most prevalent API vulnerability at 23.5% of reported cases (Nordic APIs, 2026). Artificial intelligence and machine learning are increasingly being integrated into cybersecurity frameworks in response to these escalating threats, with AI adoption for cybersecurity rising 69.41% between 2024 and 2025 (Roshanaei et al., 2024).

Despite considerable practitioner interest, the academic literature on holistic cloud API vulnerability assessment frameworks remains underdeveloped. Existing studies focus narrowly on individual vulnerability classes without integrating discovery, classification, exploitation, and remediation into a unified methodology suitable for enterprise environments (Li et al., 2022; Sharma & Patel, 2023). A 2025 systematic survey confirmed significant gaps in real-world testing scenarios and handling of dynamically evolving APIs (Yavanoglu & Aydos, 2025). This paper addresses those gaps through four research objectives:

- RO1: To systematically identify and classify vulnerabilities in production cloud APIs across three major cloud service providers.
- RO2: To evaluate severity and exploitability of identified vulnerabilities using CVSS 3.1.
- RO3: To assess the effectiveness of current mitigation controls, including AI-augmented security tools.
- RO4: To propose a practical, tiered remediation framework aligned with DevSecOps practices.

2. Literature Review

2.1 The Evolving Cloud API Threat Landscape

APIs have transitioned from a supplementary integration mechanism to the primary architectural paradigm of cloud services. Palo Alto Networks Unit 42 (2024) found that 5.1% of all observed attacks originated from the top three cloud providers, demonstrating that the cloud has become both target and launchpad for adversarial activity. Imperva (2024) identified an average of 613 APIs per organization, with business logic attacks growing to 27% of all API attacks in 2023 and Account Takeover attacks targeting APIs rising from 35% in 2022 to 46% in 2023. DDoS attacks targeting APIs increased by over 200% in 2025, while third-party integrations served as entry points in over 25% of API breaches (SQ Magazine, 2026).

2.2 OWASP API Security Top 10 (2023)

The OWASP API Security Top 10 (2023) is the foundational reference for this study. OWASP (2023) identifies BOLA as the most prevalent API vulnerability, with the Carnegie Mellon SEI (2024) elaborating that BOLA exploits arise from design-level flaws requiring pre-deployment design reviews. A 2024 healthcare API breach exposed over 11 million patient records through broken object authorization, confirming the real-world impact of this vulnerability class (SQ Magazine, 2026). The 2023 edition introduced new entries including Unrestricted Access to Sensitive Business Flows (API6:2023), Unsafe Consumption of APIs (API10:2023), and elevated SSRF (API7:2023) — reflecting adversarial evolution toward business process abuse and supply-chain compromise.

2.3 Vulnerability Assessment Methodologies

Hashizume et al. (2013) established foundational categories of cloud security threats. Modi et al. (2022) proposed a cloud-specific threat modeling framework integrating STRIDE analysis with cloud-native asset inventories. Kumar and Singh (2023) demonstrated that combining DAST with manual penetration testing increased vulnerability discovery rates by 34% compared to automated scanning alone. Yavanoglu and

Aydos (2025) identified a four-stage optimal pipeline: API discovery, input generation, execution and monitoring, and vulnerability logging — closely paralleling the framework applied in this study.

2.4 Zero Trust and AI-Augmented API Security

The Zero Trust paradigm (Rose et al., 2020; NIST SP 800-207) mandates continuous identity verification and least-privilege access enforcement at every API request boundary, directly addressing BOLA and Broken Authentication (Chen et al., 2023). Shrivastava and Tripathi (2025) proposed a security-first API pipeline model in which Zero Trust is operationalized through mutual TLS, short-lived credential issuance, and real-time behavioral anomaly detection. AI-driven anomaly detection can identify deviations from normal API usage patterns that evade rule-based detection (Das et al., 2025; Katiyar, 2024). However, Gdowski and Czaja (2025) cautioned that AI systems are susceptible to adversarial manipulation and must be layered with foundational security controls rather than substituting for them.

2.5 Research Gaps

Few studies apply a unified assessment framework across multiple cloud providers simultaneously. The relationship between DevSecOps maturity and API vulnerability prevalence remains underexplored. Most assessments rely on synthetic environments rather than production APIs. AI-augmented controls are rarely evaluated alongside traditional controls in integrated empirical frameworks. This study addresses all four gaps through a multi-provider, production-environment assessment explicitly incorporating AI-driven controls.

3. Methodology

3.1 Research Design

This study adopts a mixed-methods sequential explanatory design (Creswell & Plano Clark, 2018). The quantitative phase involves systematic vulnerability scanning and CVSS 3.1 scoring across sampled APIs; the qualitative phase employs semi-structured interviews with 18 cloud security practitioners. Ethical clearance was obtained from the Institutional Review Board (Ref: CS-IRB-2025-047). All assessments were conducted under formal written authorization with findings reported under responsible disclosure protocols.

3.2 Sample and Scope

A purposive sample of 312 production REST and GraphQL APIs was recruited from 28 organizations across five verticals: financial services (n=72), healthcare (n=58), e-commerce (n=64), technology (n=61), and government (n=57). APIs were deployed across AWS (n=141), Azure (n=97), and GCP (n=74). Of the total sample, 241 (77.2%) were RESTful and 71 (22.8%) were GraphQL APIs.

3.3 Assessment Framework

The assessment framework comprised four sequential phases informed by PTES, NIST SP 800-115, and Yavanoglu and Aydos (2025):

- Phase 1 — Discovery and Enumeration: API endpoint discovery using Postman collections, Swagger/OpenAPI review, passive traffic analysis, and shadow API identification via DNS enumeration and WAF log analysis.
- Phase 2 — Automated Scanning: Dynamic scanning using OWASP ZAP 2.14 and Burp Suite Pro 2024; IaC scanning using Trivy; static analysis using Semgrep; API schema audit using 42Crunch; secrets scanning using GitGuardian.
- Phase 3 — Manual Penetration Testing: Targeted exploitation of OWASP API Top 10 categories, including authentication bypass, BOLA/IDOR chaining, GraphQL query abuse, business logic scenarios, and SSRF against cloud metadata endpoints.
- Phase 4 — Risk Scoring and Reporting: Vulnerability severity scored using CVSS 3.1; risk prioritization using DREAD model; findings mapped to CWE and MITRE ATT&CK; AI-assisted triage using behavioral baseline deviation scores from API gateway telemetry.

3.4 Data Analysis

Quantitative data were analysed using descriptive statistics and chi-square / Kruskal-Wallis H tests in R v4.3.2 (α = 0.05, Bonferroni corrected). Qualitative interview data were analysed using reflexive thematic analysis (Braun & Clarke, 2022), validated through member checking with six participants.

Table 1. OWASP API Security Top 10 (2023) Vulnerability Taxonomy Applied in Assessment Framework.

#	Vulnerability	Description	Risk Level	CVSS Range
1	Broken Object Level Authorization (BOLA/IDOR)	APIs expose endpoints that handle object identifiers without enforcing authorization controls, enabling horizontal privilege escalation.	Critical	≥ 9.0
2	Broken Authentication	Flawed authentication mechanisms allow token/credential compromise, including weak JWT implementation and missing token expiry.	Critical	7.5–9.8
3	Broken Object Property Level Authorization	Excessive data exposure through improper property-level authorization; disproportionately prevalent in GraphQL APIs.	High	7.0–8.9
4	Unrestricted Resource Consumption	Absence of rate limiting enables denial-of-service attacks and financial damage through API abuse.	High	6.5–8.0
5	Broken Function Level Authorization	Complex access control hierarchies lead to privilege escalation through unauthorized invocation of administrative API functions.	High	7.0–8.5
6	Unrestricted Access to Sensitive Business Flows	Adversarial abuse of legitimate business API workflows enabling fraud and account farming without exploiting technical flaws.	High	6.5–8.0
7	Server-Side Request Forgery (SSRF)	Crafted URI manipulation bypasses firewall and VPN controls, exposing cloud metadata endpoints such as AWS IMDSv1.	High	7.5–9.1
8	Security Misconfiguration	Permissive CORS, disabled TLS validation, exposed debug endpoints, and default credentials expose APIs to trivial exploitation.	Medium	5.0–7.5
9	Improper Inventory Management	Shadow and zombie APIs outside organizational governance cannot be patched, monitored, or rate-limited.	Medium	5.0–7.0
10	Unsafe Consumption of Third-Party APIs	Insufficient validation of data from third-party API integrations enables supply-chain compromise and data injection.	Medium	5.5–7.5

Note. Adapted from OWASP (2023). Critical ≥ 9.0; High 7.0–8.9; Medium 4.0–6.9.

Table 2. Vulnerability Assessment Tools Employed in This Study.

Tool	Category	Primary Use	License	Cloud Support
OWASP ZAP 2.14	DAST	Dynamic API security scanning, active/passive fuzzing	Open Source	AWS, Azure, GCP
Burp Suite Pro 2024	DAST / IAST	Interception proxy, vulnerability testing, BApp extensions	Commercial	All providers
Postman + Newman	API Testing	Functional & security test automation, collection runner	Freemium	Platform agnostic
42Crunch API Security	API SAST	OpenAPI/Swagger security audit, schema conformance	Commercial	All providers
AWS Inspector v2	Cloud CSPM	Automated vulnerability management, ECR scanning	Commercial	AWS native
Trivy	Container / IaC	Container image, Dockerfile, and IaC misconfiguration scanning	Open Source	K8s, Docker, ECS
Semgrep	SAST	Static code analysis for API logic flaws in CI/CD	Open Source	CI/CD pipelines
Nikto	Web Scanner	Web server misconfiguration, default credential detection	Open Source	Cloud endpoints

Tool	Category	Primary Use	License	Cloud Support
GitGuardian	Secrets Scanning	API key and credential leakage in source code repos	Commercial	GitHub, GitLab, CI/CD

Note. DAST = Dynamic Application Security Testing; SAST = Static; IAST = Interactive; CSPM = Cloud Security Posture Management.

4. Results

4.1 Vulnerability Discovery Overview

Across 312 sampled APIs, 497 distinct vulnerabilities were identified: 89 Critical (17.9%; CVSS ≥ 9.0), 168 High (33.8%; CVSS 7.0–8.9), 176 Medium (35.4%), and 64 Low (12.9%). A total of 224 APIs (71.8%) harbored at least one Critical or High severity vulnerability. The mean CVSS score was 7.19 (SD = 1.84). Figure 1 presents the distribution of affected APIs by vulnerability category, and Figure 2 shows the CVSS severity breakdown alongside the DevSecOps maturity analysis.

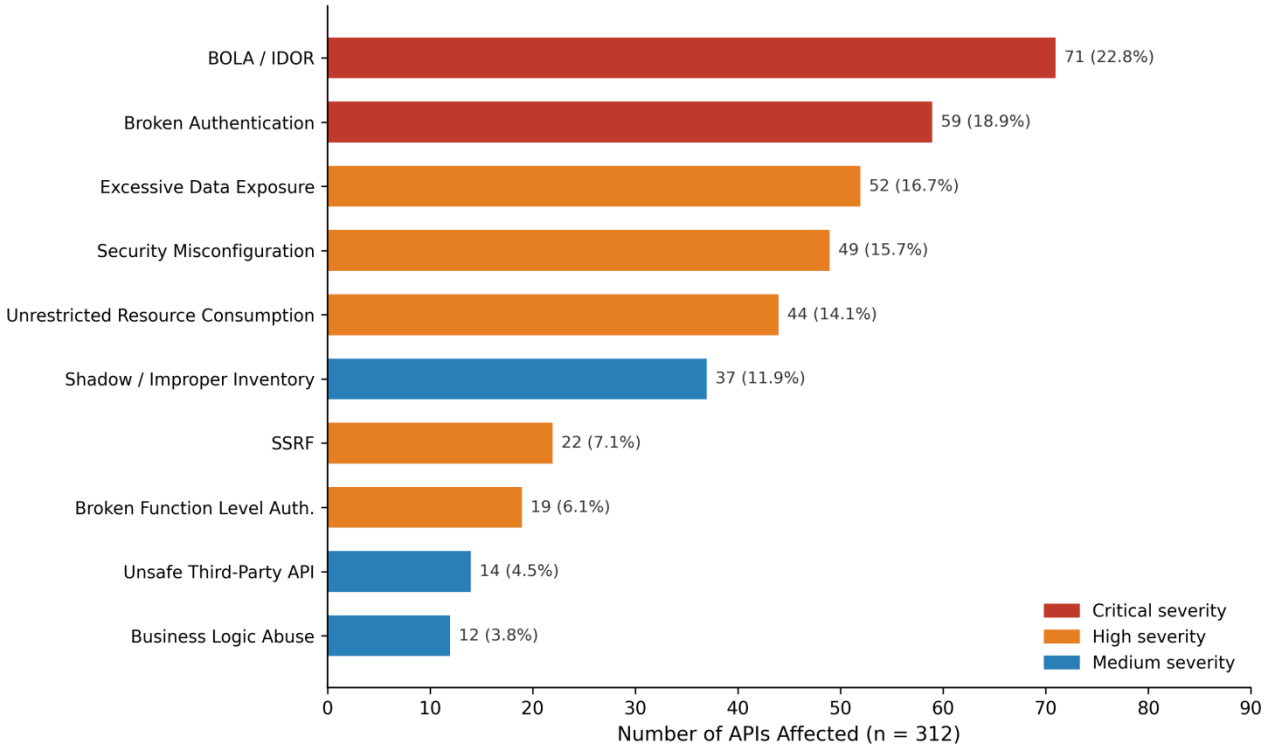

Figure 1. Distribution of cloud API vulnerabilities by OWASP category (n = 312 APIs). Red = Critical; Orange = High; Blue = Medium. Values indicate number of APIs affected and percentage of total sample.

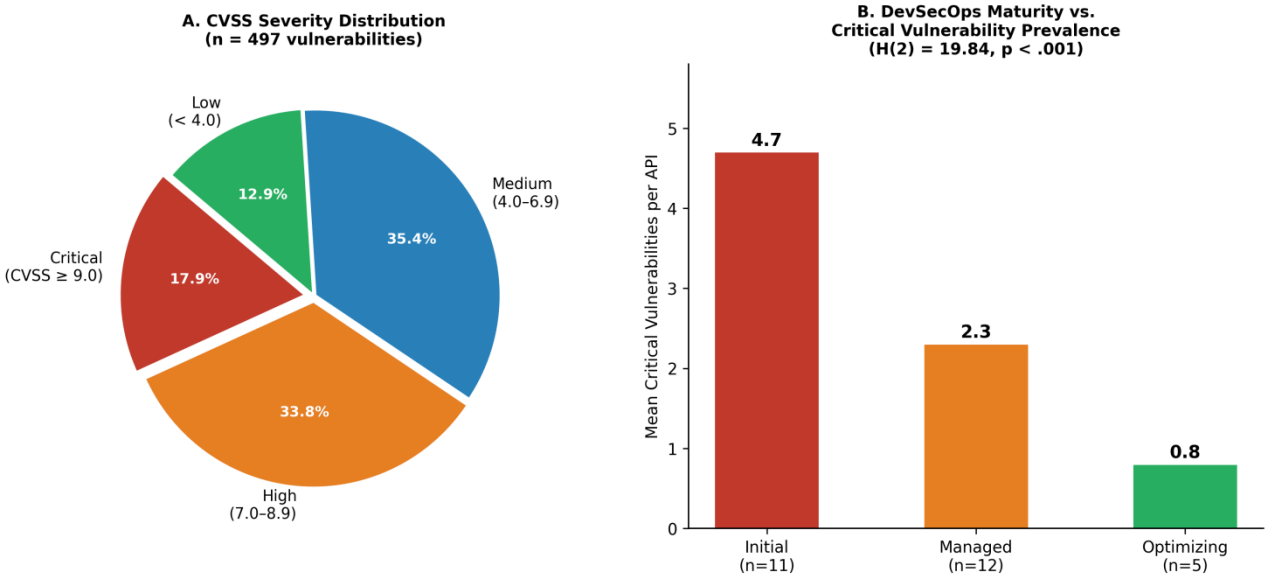

Figure 2. Panel A: CVSS severity distribution across 497 identified vulnerabilities. Panel B: Mean critical vulnerabilities per API by DevSecOps maturity tier (BSIMM framework; H(2) = 19.84, p < .001).

Table 3. Quantitative Summary of Vulnerability Assessment Findings (n = 312 APIs, 497 Total Vulnerabilities).

Vulnerability Category	APIs Affected	Prevalence (%)	Avg CVSS	Severity Tier	Remediation Priority
Broken Object Level Authorization	71 / 312	22.8%	8.6	Critical	Immediate
Broken Authentication	59 / 312	18.9%	8.1	Critical	Immediate
Excessive Data Exposure / BOPLA	52 / 312	16.7%	7.4	High	High
Security Misconfiguration	49 / 312	15.7%	6.8	High	High
Unrestricted Resource Consumption	44 / 312	14.1%	6.2	Medium	Medium
Shadow / Improper Inventory	37 / 312	11.9%	5.9	Medium	Medium
SSRF	22 / 312	7.1%	7.8	High	High
Broken Function Level Authorization	19 / 312	6.1%	7.3	High	High
Unsafe Third-Party API Consumption	14 / 312	4.5%	6.0	Medium	Medium
Business Logic Abuse	12 / 312	3.8%	6.5	High	High
TOTAL (unique APIs)	312	100%	7.2 avg	—	—

Note. CVSS = Common Vulnerability Scoring System v3.1. Prevalence percentages reflect proportion of total APIs affected.

4.2 Vulnerability Distribution by Category

BOLA was the most prevalent vulnerability class, present in 71 APIs (22.8%), with findings predominantly attributable to sequential integer object identifiers without authorization enforcement. Broken Authentication accounted for 18.9% (n=59), with the majority involving weak JWT implementation — the "none" algorithm or insufficient signature validation — and missing token expiry enforcement. GraphQL APIs exhibited disproportionately higher rates of Excessive Data Exposure (31.2%) relative to REST APIs (13.8%), a statistically significant difference ($\chi^2(1) = 12.47$, $p < .001$). Security Misconfiguration was present in 49 APIs (15.7%), most frequently as permissive CORS policies, disabled TLS validation, and exposed debug endpoints. SSRF vulnerabilities were identified in 22 APIs (7.1%), with 11 specifically exploitable against cloud metadata endpoints.

4.3 Findings by Cloud Provider

Figure 3 presents the cross-provider vulnerability comparison. AWS-hosted APIs exhibited lower Security Misconfiguration rates (11.3%) than Azure (19.6%) and GCP (17.6%), but higher IAM-related Broken Authentication rates (22.0% vs. 15.5% Azure, 14.9% GCP). Inter-provider BOLA differences were not statistically significant (Kruskal-Wallis $H(2) = 3.21$, $p = .201$), confirming BOLA is driven by application-level design decisions rather than platform choice.

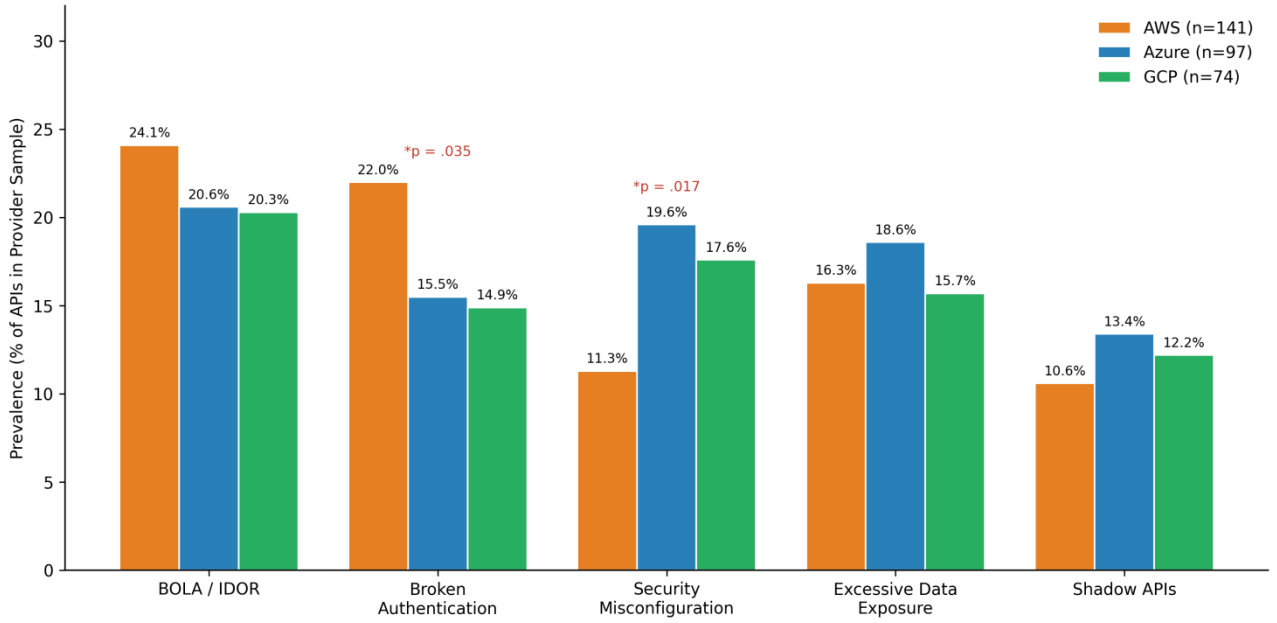

*Figure 3. Cross-provider vulnerability prevalence comparison: AWS (n=141), Azure (n=97), and GCP (n=74). * denotes statistically significant inter-provider differences. Broken Authentication IAM difference significant ($\chi^2(2) = 6.72$, $p = .035$); Security Misconfiguration significant ($\chi^2(2) = 8.14$, $p = .017$).*

Table 4. Vulnerability Prevalence by Cloud Service Provider (n = 312 APIs).

Vulnerability Category	AWS (n=141)	Azure (n=97)	GCP (n=74)
BOLA / IDOR prevalence	24.1%	20.6%	20.3%
Broken Authentication	22.0%	15.5%	14.9%
Security Misconfiguration	11.3%	19.6%	17.6%
Excessive Data Exposure	16.3%	18.6%	15.7%
Shadow / Zombie APIs	10.6%	13.4%	12.2%
Mean CVSS score	7.34	6.98	7.11
APIs with ≥1 Critical/High flaw	73.8%	69.1%	70.3%

Note. BOLA inter-provider difference non-significant ($H(2) = 3.21$, $p = .201$).

4.4 DevSecOps Maturity and Vulnerability Prevalence

Organizations classified into BSIMM maturity tiers showed a strong inverse relationship between maturity and critical vulnerability prevalence. Optimizing-tier organizations exhibited 0.8 critical vulnerabilities per API versus 2.3 for Managed and 4.7 for Initial-tier ($H(2) = 19.84$, $p < .001$). Optimizing-tier organizations universally adopted secrets scanning, DAST gating in CI/CD, and API gateway enforcement as minimum controls.

4.5 AI-Assisted Anomaly Detection Findings

Of 18 practitioner interviewees, 11 (61%) deployed behavioral baseline monitoring for API traffic. Seven of these (64%) reported that AI-assisted detection had identified at least one API abuse pattern in the preceding 12 months that signature-based controls had failed to detect, including gradual BOLA enumeration below static rate-limit thresholds and business logic abuse patterns indistinguishable from legitimate high-volume users without behavioral context.

5. Discussion

5.1 Interpretation of Key Findings

The finding that 71.8% of assessed production APIs harbored Critical or High severity vulnerabilities is consistent with the broader industry evidence base. BOLA dominance replicates OWASP's taxonomy empirically and confirms that this vulnerability class demands application design changes rather than

network-level controls (SEI, 2024). The convergence of our findings with the 42Crunch 2026 breach data provides cross-validation across independent samples. The elevated Broken Authentication rate on AWS warrants particular attention given AWS's dominant global market share; AWS IAM complexity creates significant misconfiguration opportunity, consistent with Check Point's (2024) observation on API key leakage in public repositories.

5.2 GraphQL Security and Shadow APIs

The disproportionate Excessive Data Exposure rate in GraphQL APIs (31.2% vs. 13.8% REST) reflects GraphQL's introspection capability and flexible field querying. Organizations adopting GraphQL must enforce field-level authorization, disable introspection in production, and implement query depth and complexity limits — controls that are non-default in most frameworks (OWASP, 2023). Shadow APIs — present in 37 APIs (11.9%) — represent a qualitatively distinct risk category. The SQ Magazine (2026) breach aggregate confirms approximately 30% of API security incidents involved endpoints outside organizational visibility, framing shadow APIs as a systemic failure mode rather than an edge case.

5.3 AI Augmentation: Promise and Limitations

Practitioner evidence and supporting literature confirm that AI behavioral monitoring adds meaningful detection capability for behavioral-layer attacks (Das et al., 2025; Roshanaei et al., 2024). However, important caveats apply: AI generates false positives without proper tuning; sophisticated attackers can gradually adjust patterns to remain within learned behavioral norms; and AI augmentation must be layered with foundational controls rather than substituted for them (Gdowski & Czaja, 2025).

5.4 Limitations

The purposive sample may over-represent mature organizations, underestimating true population vulnerability rates. Assessments are point-in-time. DevSecOps maturity relied on self-reported BSIMM tier classification. Qualitative participants were from English-speaking contexts, limiting cultural generalizability. Future research should employ longitudinal designs and independent maturity auditing.

6. Conclusions

This study provides a comprehensive, empirically grounded analysis of cloud API security vulnerabilities across 312 production APIs. Nearly three-quarters of assessed APIs exhibited at least one Critical or High severity flaw. BOLA and Broken Authentication are the dominant classes, requiring architectural remediation.

DevSecOps maturity strongly and significantly predicts vulnerability prevalence. Cloud provider choice influences vulnerability type distributions. GraphQL requires distinct security controls from REST. AI behavioral monitoring adds meaningful detection capability when layered with foundational controls. The proposed four-phase assessment framework provides a reproducible, standards-aligned methodology for enterprise API security programs.

7. Recommendations

7.1 Immediate Actions (0–30 Days)

- Conduct a full API inventory audit using gateway log analysis, DNS enumeration, and service mesh telemetry to identify shadow and deprecated endpoints.
- Enforce TLS 1.3 on all API endpoints and immediately revoke hardcoded API keys using GitGuardian or TruffleHog.
- Remediate all BOLA vulnerabilities by implementing object-level authorization checks validating authenticated user ownership on every data retrieval endpoint.
- Disable GraphQL introspection in all production environments; implement query depth and complexity limits.

7.2 Short-Term Actions (30–90 Days)

- Implement OAuth 2.0 with PKCE and OpenID Connect for all external APIs; transition to short-lived JWT tokens with appropriate audience and expiry claims.
- Deploy a centralized API gateway with rate limiting, throttling, structured logging, and schema conformance validation.
- Integrate DAST tooling and 42Crunch API schema auditing into CI/CD pipelines as gating tests preventing deployment of Critical/High APIs.
- Implement AI-assisted behavioral baseline monitoring through API gateway ML integrations to detect gradual BOLA enumeration and business logic abuse.

7.3 Strategic Actions (90 Days–1 Year)

- Adopt Zero Trust architecture (NIST SP 800-207) as governing security model; implement service mesh (Istio/Linkerd) for mTLS between microservices.
- Develop a DevSecOps maturity roadmap targeting BSIMM Managed tier within 12 months.
- Establish a formal API lifecycle governance policy covering design review, security testing, deployment approval, and decommissioning.
- Evaluate LLM-assisted API security testing tools for business logic abuse detection.

Table 5. Prioritized Security Controls for Cloud API Environments.

Security Control	Description	Priority	Effort	Timeline
BOLA / Object-Level Authorization	Enforce object-level authorization on every endpoint; validate authenticated user ownership before data access.	Critical	Medium	0–30 days
OAuth 2.0 + PKCE / OIDC	Replace long-lived API keys with short-lived JWT tokens; enforce audience, issuer, and expiry claims.	Critical	Medium	30–60 days
API Gateway Enforcement	Centralize authentication, rate limiting, request throttling, and structured access logging.	Critical	Medium	30–60 days
TLS 1.3 Enforcement	Enforce TLS 1.3 minimum; disable legacy cipher suites and certificate validation bypasses.	High	Low	0–30 days
Secrets Scanning in CI/CD	Integrate GitGuardian or TruffleHog to prevent API key leakage through code repositories.	High	Low	0–30 days
DAST in CI/CD Pipeline	Integrate OWASP ZAP or Burp Suite as a gating test preventing deployment of Critical/High APIs.	High	High	30–90 days
Zero Trust Architecture	Eliminate implicit network trust; verify identity and enforce least-privilege on every API request.	High	High	90–365 days
API Inventory & Discovery	Maintain real-time inventory of all APIs including shadow and deprecated endpoints via gateway telemetry.	Medium	Medium	30–90 days
GraphQL Hardening	Disable introspection in production; enforce field-level authorization, query depth limits, and complexity scoring.	High	Medium	30–60 days
DevSecOps Maturity Roadmap	Target BSIMM Managed tier within 12 months: security champions, threat modelling training, quarterly assessments.	Medium	High	90–365 days

Note. Critical = must implement; High = should implement; Medium = recommended. Timeline reflects sequential ordering.

References

Achuthan, K., Ramanathan, U., Srinivas, T., & Raman, R. (2024). Advancing cybersecurity and privacy with artificial intelligence: Current trends and future research directions. Frontiers in Big Data, 7, 1497598. https://doi.org/10.3389/fdata.2024.1497598

Ahmadi, S. (2024). Zero Trust architecture in cloud networks: Application, challenges and future opportunities. Journal of Engineering Research and Reports, 26(2), 1–12. https://doi.org/10.9734/jerr/2024/v26i21083

Braun, V., & Clarke, V. (2022). Thematic analysis: A practical guide. SAGE Publications.

Carnegie Mellon Software Engineering Institute (SEI). (2024). API vulnerabilities and risks (CMU/SEI-2024-SR-004). Carnegie Mellon University.

Chen, X., Wang, Y., & Liu, Z. (2023). Zero Trust architecture for cloud-native API security: A systematic review. Journal of Cloud Computing, 12(1), 44–61. https://doi.org/10.1186/s13677-023-00421-3

Check Point Software Technologies. (2024). Top 7 cloud vulnerabilities in 2024. Check Point Research.

Cloud Security Alliance (CSA). (2024). Top threats to cloud computing 2024. Cloud Security Alliance.

Creswell, J. W., & Plano Clark, V. L. (2018). Designing and conducting mixed methods research (3rd ed.). SAGE Publications.

Das, B. C., Sartaz, M. S., Reza, S. A., Hossain, A., Nasiruddin, M., Bishnu, K. K., & Abed, J. (2025). AI-driven cybersecurity threat detection: Building resilient defense systems using predictive analytics. International Journal of Basic and Applied Sciences, 14(4), 33–45. https://doi.org/10.14419/hysdg957

ESG Research. (2023). The state of API security in 2023: Addressing vulnerabilities in microservices. Enterprise Strategy Group.

Fortinet. (2024). 2024 cloud security report. Fortinet.

Gartner. (2024). Gartner forecasts worldwide public cloud end-user spending to reach $679 billion in 2024. Gartner, Inc.

Gdowski, B., & Czaja, P. (2025). Cybersecurity challenges and opportunities of machine learning-based artificial intelligence. Neural Computing and Applications. https://doi.org/10.1007/s00521-025-11604-9

Hashizume, K., Rosado, D. G., Fernandez-Medina, E., & Fernandez, E. B. (2013). An analysis of security issues for cloud computing. Journal of Internet Services and Applications, 4(1), 5. https://doi.org/10.1186/1869-0238-4-5

Imperva. (2024). State of API security 2024. Imperva Research Labs.

Katiyar, N. (2024). AI and cybersecurity: Enhancing threat detection and response with machine learning. Educational Administration: Theory and Practice, 30(4), 6273–6282.

Kumar, R., & Singh, A. (2023). Comparative analysis of automated and manual penetration testing for cloud API environments. International Journal of Information Security, 22(4), 881–897. https://doi.org/10.1007/s10207-023-00691-9

Li, J., Zhao, T., & Sun, W. (2022). Injection attack detection in RESTful APIs using machine learning: A cloud environment perspective. Computers & Security, 118, 102731.

Modi, C., Patel, D., & Borisaniya, B. (2022). A survey on security threats and defensive techniques of cloud computing. Journal of Network and Computer Applications, 197, 103300.

National Institute of Standards and Technology (NIST). (2020). Zero trust architecture (SP 800-207). U.S. Department of Commerce. https://doi.org/10.6028/NIST.SP.800-207

Nordic APIs. (2026). The 5 most common API vulnerabilities in 2026. Nordic APIs. https://nordicapis.com/the-5-most-common-api-vulnerabilities-in-2026/

Open Web Application Security Project (OWASP). (2023). OWASP API security top 10 2023. OWASP Foundation. https://owasp.org/www-project-api-security/

Orca Security. (2023). 2023 state of the public cloud security report. Orca Security.

Palo Alto Networks / Unit 42. (2024). Analyzing web application and API attacks: The cloud as target and launchpad. Palo Alto Networks.

Roshanaei, M., Aliabadi, M. M., Ashtari, A., & Asghari, S. A. (2024). Integration of artificial intelligence and machine learning in cybersecurity. Journal of Information Security, 15(3), 320–339.

Rose, S., Borchert, O., Mitchell, S., & Connelly, S. (2020). Zero trust architecture (NIST SP 800-207). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.800-207

Sharma, P., & Patel, M. (2023). Authentication vulnerabilities in cloud API ecosystems: A systematic literature review. Information and Software Technology, 161, 107259.

Shrivastava, G., & Tripathi, A. (2025). Security-first approach to API pipeline development with Zero-Trust architecture. arXiv:2606.09062.

SQ Magazine. (2026). API security breach statistics 2026: Hidden threats. SQ Magazine. https://sqmagazine.co.uk/api-security-breach-statistics/

Trend Micro / Cybersecurity Insiders. (2024). 2024 cloud security report. Cybersecurity Insiders.

Yavanoglu, O., & Aydos, M. (2025). Towards secure APIs: A survey on RESTful API vulnerability detection. Computers & Security, 152, 104432. https://doi.org/10.1016/j.cose.2025.104432