EasyChair Preprint

№ 15891

Mitigating Attack Surfaces in Serverless Architectures: Best Practices for Secure Deployments

Kenny Awkent

March 6, 2025

Mitigating Attack Surfaces in Serverless Architectures: Best Practices for Secure Deployments

Author: Kenny Hawkent *Master of Philosophy, University of Cape Coast, Ghana*

December 2023

Abstract

Serverless computing has transformed the way organizations deploy applications, offering **greater scalability, cost efficiency, and operational agility**. However, the shift to a **function-as-a-service (FaaS)** model introduces **new security risks**, including **misconfigured permissions, insecure dependencies, API vulnerabilities, and difficulties in monitoring short-lived functions**. Unlike traditional monolithic architectures, serverless environments **rely on event-driven execution**, which expands the **attack surface** by increasing the number of exposed interfaces.

This article provides a **comprehensive analysis of security risks in serverless computing** and offers best practices for mitigating attack surfaces. It explores **secure dependency management, strong identity and access controls, API security enhancements, real-time monitoring, and compliance considerations**. By implementing these strategies, organizations can **reduce vulnerabilities, prevent security breaches, and enhance the resilience of their serverless deployments**.

Keywords: Serverless security, cloud computing, identity and access management (IAM), API security, threat detection, secure coding, event-driven security, least privilege, monitoring, cloud-native security.

1. Introduction

Serverless computing enables developers to build and execute applications **without managing infrastructure**, relying on **cloud providers like AWS, Microsoft Azure, and Google Cloud** to handle provisioning, scaling, and execution. The appeal of serverless architectures lies in their **auto-scaling capabilities and cost-effectiveness**, as organizations only pay for function execution time.

However, the **distributed, event-driven nature** of serverless architectures presents **new security challenges**. Traditional security models are built around **long-lived workloads** that operate on dedicated infrastructure, whereas serverless functions are **ephemeral** and may execute for just milliseconds before terminating. This makes it difficult to apply **conventional monitoring, access control, and vulnerability management strategies**.

This article explores the most significant **security risks in serverless architectures** and outlines **best practices for reducing attack surfaces** to enhance cloud-native security.

---

2. Secure Code and Dependency Management

2.1 Why It Matters?

Serverless functions frequently **leverage third-party libraries** to accelerate development. However, **unverified dependencies** introduce the risk of **supply chain attacks, remote code execution (RCE), and dependency confusion attacks**. Attackers can exploit vulnerabilities in open-source libraries to **inject malicious code** into serverless workloads.

2.2 Best Practices

2.2.1 Use Trusted and Maintained Libraries

- Depend on **official repositories** like NPM, PyPI, and Maven.
- Verify **maintainer reputation and update frequency** before using a package.

2.2.2 Perform Regular Dependency Scanning

Automated tools like **Snyk, OWASP Dependency-Check, Trivy, and GitHub Dependabot** should be integrated into CI/CD pipelines to **detect vulnerabilities in dependencies**.

2.2.3 Implement Software Composition Analysis (SCA)

SCA tools like **Black Duck, WhiteSource, and Sonatype Nexus** analyze dependency trees for **security flaws and license compliance risks**.

2.2.4 Enforce Secure Coding Practices

- Use **Static Code Analysis (SCA)** tools like **SonarQube, Bandit, and ESLint**.
- Store sensitive credentials in **secrets management systems** (e.g., AWS Secrets Manager).

---

3. Strong Identity and Access Management (IAM)

3.1 Why It Matters?

Poorly configured IAM policies are a **major source of security breaches** in cloud environments. Overly permissive roles allow **unauthorized access** to sensitive cloud services, increasing the risk of **data exposure and privilege escalation attacks**.

3.2 Best Practices

3.2.1 Implement Least Privilege Access Control

Functions should **only receive the minimum permissions necessary** to perform their tasks. Avoid using broad permissions such as s3:* or dynamodb:*.

3.2.2 Use Role-Based Access Control (RBAC)

Define **pre-configured roles** (e.g., "Read-Only", "Function Executor") rather than assigning permissions directly to users or functions.

3.2.3 Rotate API Keys and Secure Credentials

- **Never hardcode API keys** in function code.
- Use **secrets management tools** for **automated credential rotation**.

3.2.4 Enforce Multi-Factor Authentication (MFA)

Require **MFA for privileged IAM accounts** to mitigate the impact of **credential theft or brute-force attacks**.

---

4. API Security and Event Handling

4.1 Why It Matters?

Serverless applications rely heavily on APIs for **external communication and event processing**. Unsecured APIs can expose **sensitive data**, enable **unauthorized access**, or become targets for **DDoS attacks**.

4.2 Best Practices

4.2.1 Require Strong Authentication for APIs

Use **JWT (JSON Web Token), OAuth 2.0, or API Gateway authentication mechanisms** to enforce access controls.

4.2.2 Use API Gateways for Security Enforcement

AWS API Gateway, Azure API Management, and Google Cloud Endpoints provide built-in **DDoS protection, rate limiting, and request validation**.

4.2.3 Validate API Requests

APIs should sanitize input data to **prevent SQL injection, command injection, and cross-site scripting (XSS) attacks**.

---

5. Monitoring, Logging, and Threat Detection

5.1 Why It Matters?

Since **serverless functions are ephemeral**, traditional logging tools may fail to capture **suspicious activity**. Without **continuous monitoring**, attacks may remain **undetected** until significant damage is done.

5.2 Best Practices

5.2.1 Enable Real-Time Cloud Monitoring

- Use **AWS CloudTrail, Google Cloud Logging, and Azure Monitor** for function monitoring.
- Implement **Security Information and Event Management (SIEM)** solutions like **Splunk and IBM QRadar**.

5.2.2 Implement AI-Based Threat Detection

Machine learning-based **anomaly detection** tools, such as **AWS GuardDuty and Google Chronicle**, help detect **unauthorized function behavior** in real-time.

---

6. Conclusion

Serverless computing introduces **unparalleled efficiency and flexibility**, but its unique architecture also creates **new security challenges**. Organizations must adopt a **proactive, security-by-design approach** to mitigate the **expanded attack surface** in cloud-native environments.

Key **takeaways** from this article include:

1. **Secure dependency management** to prevent **supply chain attacks**.
2. **Enforcing least privilege IAM policies** to reduce unauthorized access risks.
3. **Strengthening API security** by implementing **authentication and rate limiting**.
4. **Using real-time monitoring** to detect and **mitigate security threats** in ephemeral functions.

As **serverless adoption grows**, organizations must **continuously evolve their security strategies** to address emerging threats. Implementing **robust access controls, monitoring frameworks, and**

compliance-driven security practices will ensure that **serverless deployments remain secure, resilient, and future-proof**.

References

1. **Amazon Web Services (AWS).** "Security Best Practices in Serverless Applications," AWS Whitepaper, 2024.
2. **Microsoft Azure Security Center.** "Best Practices for Securing Serverless Applications," 2024.
3. Ahmadi, Sina. "Open AI and its impact on fraud detection in financial industry." *Journal of Knowledge Learning and Science Technology ISSN: 2959-6386 (online)* 2.3 (2023): 263-281.
4. Ahmadi, Sina. "Optimizing data warehousing performance through machine learning algorithms in the cloud." *International Journal of Science and Research* 12.12 (2023): 1859-1867.
5. Ahmadi, Sina. "Next generation ai-based firewalls: a comparative study." *International Journal of Computer (IJC)* 49.1 (2023): 245-262.
6. Ahmadi, Sina. "Elastic data warehousing: Adapting to fluctuating workloads with cloud-native technologies." *Journal of Knowledge Learning and Science Technology ISSN: 2959-6386 (online)* 2.3 (2023): 282-301.
7. **Google Cloud Security Team.** "IAM and API Security in Cloud Functions," 2024.
8. **OWASP Foundation.** "Top 10 Serverless Security Risks," OWASP Report, 2024.
9. **Gartner Research.** "Serverless Security Trends and Threat Intelligence," 2024.
10. **Splunk Security Team.** "SIEM and Real-Time Threat Detection in Serverless Computing," 2024.
11. **IBM X-Force.** "IAM Vulnerabilities in Cloud-Native Environments," 2024.
12. **Snyk Security Report.** "Supply Chain Attacks in Serverless Functions," 2024.
13. **MITRE ATT&CK Framework.** "Tactics for Securing Event-Driven Architectures," 2024.
14. **Cybersecurity & Infrastructure Security Agency (CISA).** "Guidelines for Protecting Cloud-Native Applications," 2024.