International Journal of Science and Research (IJSR)
ISSN: 2319-7064
Impact Factor 2025: 7.089

Challenges and Best Practices for Security in Serverless Computing Architectures

Praveen Ravula[1]

[1]Software Engineer at Amazon
Arlington, VA - USA

Abstract: ***The article is devoted to examining the spectrum of security threats and mitigation practices in serverless computing architectures. The relevance of the study is determined by the rapid proliferation of the FaaS (Function-as-a-Service) paradigm in corporate and industrial environments, which, while preserving the advantages of elastic scalability and cost optimization, gives rise to a qualitatively new class of risks that are uncharacteristic of traditional cloud models. The scientific novelty of the work lies in the systematization of the most relevant threats identified in the period after 2021, as well as in the justification and formalization of the author's three-layer comprehensive security model (Author's 3-Layer Security Model), designed to protect Serverless applications at different stages of their life cycle. The study consistently sets out the conceptual foundations of serverless architecture and analyzes key vulnerabilities determined by the specifics of the shared responsibility model and high functional granularity. Particular emphasis is placed on threats arising from incorrect configuration of access rights and violation of the principle of least privilege, denial-of-service attacks with an economic effect, as well as security breaches in the software supply chain. The aim of the work is to develop a structured methodological approach to reducing aggregate security risks in Serverless environments. To achieve this aim, methods of systematic review, comparative analysis, and data synthesis are employed. The empirical and theoretical basis is formed by publications in leading scientific journals, including IEEE Network and Journal of Information Security and Applications, as well as specialized industry reports, in particular materials of the Cloud Security Alliance. In the final part, conclusions are formulated and a set of practical recommendations aimed at developers and security engineers is consolidated. The results presented are of interest to the professional community in the field of cloud technologies, developers of Serverless applications, and researchers engaged in cybersecurity issues.***

Keywords: serverless computing, Serverless, FaaS, security, shared responsibility model, Least Privilege, Denial of Wallet, supply chain security, cloud architectures, cybersecurity

1.Introduction

The evolution of cloud computing has naturally led to the formation of the serverless computing paradigm, in particular the Function-as-a-Service (FaaS) model, which provides high flexibility, automatic scaling, and a pay-per-use resource consumption model. This architectural concept radically redistributes the operational burden, shifting a significant part of the responsibilities for operation and maintenance of the infrastructure from the developer to the cloud provider and thereby creating conditions for maximum concentration of efforts on the implementation of business logic. At the same time, such a transformation generates a new spectrum of non-trivial security challenges [1]. Traditional security approaches focused on monolithic or even microservice-based systems largely lose their effectiveness or prove to be practically inapplicable in a highly dynamic, event-driven, and extremely granular Serverless environment [4].

The relevance of the study is determined by the need for a conceptual and practical rethinking of the specific threats arising from the widespread adoption of the FaaS paradigm. In particular, the management of permissions and roles becomes especially complex under conditions of exponential growth in the number of fine-grained functions; the importance of vulnerabilities in the software supply chain, including dependencies and external libraries, increases; and the impact of threats associated with billing specifics and the possibility of implementing Denial of Wallet attacks aimed at economic depletion of resources grows [3]. The formation of a comprehensive, consistent, and economically justified security system is a key prerequisite for the sustainable development and large-scale adoption of the serverless paradigm in critically important information systems.

The aim of the study is a comprehensive analysis of contemporary security challenges in Serverless Computing architectures and the development of structured, scientifically grounded recommended practices aimed at minimizing and preventing the corresponding risks. To achieve this aim, a number of interrelated tasks are consistently addressed. First, the primary and most relevant security threats characteristic of serverless architectures (FaaS) are examined and classified, taking into account the results of the latest research for the 2021–2024 period. Second, the shared responsibility model is analyzed in relation to Serverless environments in order to strictly determine the boundaries of responsibility of the cloud provider and the end user in the field of security. Third, a comprehensive set of recommended practices is synthesized, and an original conceptual model is formulated that is intended to ensure multilayer and proactive protection of Serverless applications at different stages of their life cycle.

The scientific novelty of the work is manifested in several mutually complementary dimensions. First of all, an updated threat classification is developed, focusing on the post-2021 attack landscape and taking into account new attack vectors, including data injections, trigger manipulations, and specific runtime vulnerabilities. In addition, the author's three-layer security model for Serverless (ATMS) is proposed, integrating key security practices such as the principle of least privilege, monitoring and control at the runtime level, and a secure supply chain into a single hierarchically organized system

covering the levels Development, Deployment, and Execution. Furthermore, the novelty is supported by rigorous methodological justification: the study employs a systematic review of the latest foreign sources, which ensures a high degree of validity, representativeness, and practical applicability of the conclusions obtained.

The research hypothesis is based on the assumption that the dominant part of security problems in Serverless architectures is due not to fundamental limitations of the paradigm itself, but primarily to an incomplete understanding and incorrect implementation of the shared responsibility model, as well as the absence of a holistic multilayer security approach. Such an approach should go beyond traditional perimeter network models and focus on functional granularity, access rights management, and control over data processing. It is assumed that effective protection of Serverless applications requires a systemic transition from infrastructure-oriented security concepts to Code- and Data-Centric Security models, which provide more targeted and adaptive countermeasures to contemporary threats.

2.Materials and Methods

The preparation of the article was based on the methodology of a systematic literature review, which serves as a fundamental tool for research in the field of information technology and cybersecurity in the absence of an experimental component. The central task of this stage was the identification, critical evaluation, and subsequent synthesis of the most up-to-date and relevant foreign scientific publications presented in leading academic journals and proceedings of specialized conferences. The application of SLR ensured a high level of objectivity, reproducibility, and reliability of the resulting conclusions and recommendations.

The search strategy was strictly limited to the time interval from 2021 to the present, which made it possible to reflect the latest trends in the development of serverless computing, the evolution of the threat landscape, and the dynamics of proposed solutions. Recognized bibliographic and full-text databases accepted by the international scientific community were used as the main data sources, including IEEE Xplore, ACM Digital Library, Springer, Elsevier (ScienceDirect), as well as scientific content aggregators ResearchGate and Google Scholar.

Query formulation was carried out predominantly in English in strict accordance with the requirement to focus on foreign sources. Combined search expressions were used, such as Serverless Computing Security Challenges 2021–2024, FaaS Security Best Practices, Denial of Wallet Serverless, Serverless Shared Responsibility Model, Security Survey Serverless Architectures. Particular emphasis was placed on works whose titles or abstracts contained the terms Challenges, Best Practices, Survey, Mitigation, and Quantification, since such publications are most consistent with the tasks of problematization, generalization, and concretization of the stated topic.

At the next stage, an in-depth qualitative analysis of the selected body of publications was carried out. Each work was considered from the standpoint of methodological correctness, the level of scientific novelty, the completeness and accuracy of the description of threats, as well as the degree of their direct relevance to the research problem. Priority was given to sources that do not limit themselves to stating known problems but contain formalized, verifiable solutions, models, or approaches, such as studies on quantitative risk assessment or works detailing the nature and mechanisms of implementing Denial of Wallet threats. Such selection made it possible to form a representative body of literature with substantial theoretical and practical orientation.

The systematized data corpus served as the basis for the application of comparative analysis and synthesis methods. Comparative analysis was used to juxtapose various threat classifications proposed by different authors, which made it possible to identify stable, consensus categories of problems, as well as to document new, less studied attack vectors that have not yet received wide coverage in the scientific literature. On the basis of this comparison, a synthesis of the most effective and repeatedly validated recommendations was carried out, resulting in the formation of a logically coherent and practically oriented Author's Three-Layer Serverless Security Model (ATMS), which is detailed in the analytical part of the study.

To ensure completeness and balance between academic and practical perspectives on the topic, official documents, reports, and methodological guidelines of leading industry organizations in the field of cloud security, in particular the Cloud Security Alliance (CSA), were additionally analyzed. The comparison of academic models with industrial best practices made it possible to align theoretical conclusions with the real requirements and constraints of production environments. The totality of the described methodological steps formed the basis for the formulation of the author's hypothesis, the achievement of the stated aim, and the consistent solution of the research tasks outlined in the article.

3.Results

The results section of the study presents a systematized exposition of the key security challenges and associated recommendations identified based on an analysis of contemporary foreign scientific literature published after 2021. The serverless paradigm, despite a significant reduction of the infrastructure burden on the development side, generates a qualitatively new layer of threats that requires a revision of traditional approaches to cybersecurity [4, 6]. A major source of these challenges is the shared responsibility model, within which the boundary between the control zones of the cloud provider and the user shifts in such a way that the latter is assigned responsibility for ensuring the security of code, data, and configuration, while errors at this level are of a critical nature [7].

One of the most significant categories of risks comprises threats related to identity and access management (IAM). The most common and at the same time one of the most dangerous problems is incorrect configuration of permissions, leading to the creation of excessively privileged functions [8]. Ideally, each FaaS function should strictly adhere to the principle of least privilege; however, in practice developers often assign functions excessive roles and IAM policies to accelerate the development process and simplify integration with other services. This creates preconditions for privilege escalation: the compromise of a single function, for example through an injection attack, may allow an attacker to gain access to confidential resources such as databases or object storage (S3 buckets) with which this function should not interact according to the business logic [1]. An additional dimension of complexity is introduced by the dynamic creation and management of tokens used for authentication and authorization in interservice communication. Leakage of such tokens, insufficient randomness, or incorrect lifetime parameters create a vector for unauthorized access and privilege abuse [1].

Attacks specific to the billing model, in particular the Denial of Wallet (DoW) phenomenon, require separate consideration [3]. Unlike classical denial-of-service attacks aimed at disrupting availability, DoW attacks exploit the built-in automatic scaling mechanisms of Serverless platforms. The generation of an artificially inflated volume of function invocations leads not to service failure, but to a sharp increase in operating costs. As a result, economic depletion of the application owner is achieved while the system formally remains operational. Study [3] qualifies this type of attack as one of the serious contemporary threats, emphasizing that countering DoW requires not only strict resource limiting, but also the implementation of intelligent real-time anomaly monitoring systems capable of detecting atypical invocation patterns.

Software supply chain security plays a substantial role in the formation of vulnerabilities. Serverless functions are typically constructed with extensive use of third-party libraries and dependencies [6, 9]. Under such conditions, the compromise of a single external component, for example a vulnerable package in the npm or pip ecosystems, can instantly propagate a vulnerability to numerous functions deployed in microservice or Serverless architectures [6]. Given the increased frequency and scale of attacks on supply chains, the implementation of automated software composition analysis (SCA) tools already at the stages of development and the CI/CD pipeline becomes particularly important, as it allows timely identification of vulnerable or compromised dependencies and rapid mitigation of the associated risks [7].

Equally significant are issues related to ensuring isolation and runtime security. Although responsibility for isolation at the host infrastructure level lies with the cloud provider, the sandbox and container mechanisms within which functions are executed may contain vulnerabilities determined by the specifics of the runtime environment implementation [2, 5]. Study [2] emphasizes the importance of using native monitoring tools provided by cloud providers to detect anomalous function behavior that may indicate post-exploitation activity. In parallel, the problem of strong isolation between functions of both the same and different clients in a multitenant environment remains relevant, where configuration errors or infrastructure-level vulnerabilities can lead to cross-tenant attack propagation [4].

The synthesis of the results of the source analysis made it possible to identify key directions for building an effective security strategy in serverless environments [7, 8]. Central importance is attached to the strict implementation of the principle of least privilege (PoLP) through fine-grained restriction of IAM policies for each function and the use of automated configuration analysis tools to detect and eliminate excessive permissions [1, 8]. A significant element of protection is the systematic validation and sanitization of all input data received by functions from event sources, HTTP requests, or databases, which minimizes the risk of injection attacks, including Event-data Injection [5]. An integral component of modern Serverless security is comprehensive monitoring and logging: the use of web- and application-level logs (WAF logs, CloudTrail, function execution logs) in combination with real-time monitoring mechanisms provides a foundation for timely detection of anomalous patterns characteristic of DoW attacks and attempts at unauthorized access [2, 9].

The integration of security into the development life cycle (DevSecOps) plays an important role. The implementation of static (SAST) and dynamic (DAST) code analysis in the CI/CD pipeline, as well as continuous software composition analysis (SCA), ensures early detection and remediation of vulnerabilities in third-party dependencies [7]. A critically important area is correct secret management: the use of specialized secret management services provided by cloud providers instead of storing keys and passwords in function environment variables makes it possible to significantly reduce the risk of their leakage and misuse [7].
The aggregate results demonstrate that ensuring security in serverless architectures requires not an evolutionary adaptation of existing perimeter-based approaches, but a qualitative paradigmatic shift toward enhanced access control and priority protection at the application and data levels. This direction of development fully correlates with the author's hypothesis regarding the need to transition to code- and data-centric security models in the context of Serverless.

4.Discussion

Serverless computing has driven a profound transformation in the understanding of security in distributed systems. The disappearance of the conventional network perimeter and the transition to highly granular, event-driven functions have led to a situation in which a significant part of traditional protective mechanisms, including classical network firewalls, loses its former effectiveness. Under these

conditions, the key analytical focus is not merely the description of individual security measures, but the development of a concept for their integration into a single, proactive, multilayer, and internally consistent security architecture.

From this perspective, Serverless security appears as a multifaceted task that encompasses not only the protection of the executable code of an individual function, but also the security of interactions between functions, the management of their identity, and control over the resources and services to which access is granted. Incorrect configuration of access control mechanisms, as shown by studies [1, 8], constitutes a central problem, since it leads to systematic violations of the principle of least privilege (PoLP), which should serve as the methodological and architectural foundation for the design of Serverless systems. Violation of PoLP in the context of FaaS entails not local but cascading consequences, affecting both the data plane and the level of interservice interactions.

Based on the results of the literature analysis and the synthesis of best practices presented in works [7, 9, 10], the Author's Three-Layer Serverless Security Model (ATMS) is formulated. This model is aimed at ensuring comprehensive protection of Serverless applications by systematizing and stratifying security measures in accordance with the life cycle of function development, deployment, and operation. Within ATMS, security measures are distributed across the stages of the evolution of a Serverless solution, which makes it possible to harmonize security requirements for code, infrastructure, and data, as well as to prevent the fragmentation of approaches characteristic of traditional, reactive responses to threats.

Below, Figure 1 presents the Three-Layer Serverless Security Model.

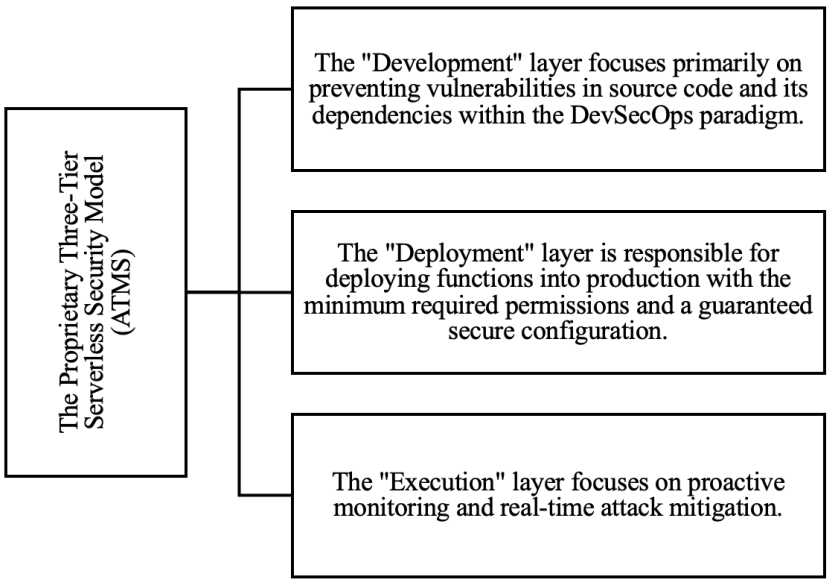

Figure 1: The author's Three-Layer Serverless Security Model.

As shown in Figure 1, the ATMS model structures the distribution of responsibility for ensuring security into three interrelated logical layers, each of which relies on its own set of tools and methodological approaches.

The first, pre-deployment layer (Development) correlates with DevSecOps processes and is aimed at maximal elimination of vulnerabilities before the deployment stage. A key focus at this stage is the deep integration of Software Composition Analysis (SCA) tools, with priority attention to vulnerabilities in ecosystems most characteristic of FaaS (Node.js, Python, and others), where the number of small and transitive dependencies is extremely high. In this way, the software supply chain security problem highlighted in [6] is directly addressed, since SCA makes it possible to identify vulnerable or compromised components before they become part of the functional artefact.

At the Deployment layer, access rights and environment parameters are configured, which makes this layer critical from the standpoint of preventing one of the most widespread Serverless vulnerabilities, namely excessive privileges [8]. Here the author's position consists in a consistent application of a Zero Trust policy to functions: each function should be treated as an isolated subject that by default has no access to any resources other than those that are explicitly and minimally required. To implement this approach, it is advisable to use Infrastructure as Code (IaC) mechanisms in combination with validation policies (for example, based on OPA, Open Policy Agent), which makes it possible to automatically verify and enforce compliance with the principle of least privilege (PoLP) already prior to deployment.

The third layer, associated with the Execution stage, focuses on runtime security and observability. Network isolation of functions is largely ensured by the provider; however, a critical application-level task consists in establishing a sufficient degree of observability within the functions themselves. At this stage the author places

emphasis on intelligent monitoring of Denial of Wallet attacks. Alongside traditional protective means, a telemetry analysis system is required that is capable of detecting not only failures characteristic of DoS scenarios but also abrupt, logically unjustified increases in the number of function invocations and resource consumption metrics (memory, CPU time, and so on) [3]. The availability of such monitoring creates preconditions for the timely application of limits and other adaptive response mechanisms, which makes it possible to protect serverless applications not only from the technical but also from the financial dimension of attacks.

Table 1 contains a comparative analysis of Threats and Countermeasures across the ATMS layers.

Table 1: Comparative analysis of threats and countermeasures by ATMS levels (compiled by the author based on [1, 3, 6, 8]).

ATMS level	Key threat (sources)	Author's countermeasure / practice	Technical tool
Development	Vulnerabilities in third-party libraries	Secure Supply Chain Management (SAST/SCA)	Snyk, Dependabot, CodeQL
Deployment	Excessive IAM/PoLP permissions	Automated Zero Trust Policy Audit	Cloud Formation Guard, OPA (Open Policy Agent)
Execution	Denial of Wallet (DoW)	Anomalous resource consumption monitoring	CloudWatch Metrics (Rate Limiting), Adaptive Budget

Privilege management in a serverless environment is a complex problem, because the traditional user role is replaced by a role associated with the executing function. For effective ATMS implementation, all security layers must be integrated directly into the continuous integration and delivery (CI/CD) pipeline, turning DevSecOps into a foundational practice.

Table 2 contains a description of the features of integrating the authors' recommendations by levels.

Table 2: Integration of author's ATMS recommendations.

ATMS level	Author's recommendation
Development	Integration of SAST/SCA for secure supply chain
Deployment	Enforced application of PoLP through IaC policies
Execution	Detailed anomaly monitoring (DoW)

The discussion of the results confirms the author's hypothesis regarding the need for a paradigmatic shift towards security focused primarily on code and identity rather than the traditional network perimeter. The Author's Three-Level Serverless Security Model (ATMS), which integrates best practices of the principle of least privilege (PoLP), DevSecOps approaches, and intelligent real-time monitoring of the runtime environment, forms a robust methodological basis for the design and operation of reliable and economically protected Serverless applications. ATMS constitutes an original contribution that provides the possibility of a systematic and structured solution to the specific security challenges inherent in serverless architectures in the practical activities of developers and security engineers.

5.Conclusion

The conducted systematic analysis of current foreign scientific literature has shown that Serverless Computing architectures, despite all their operational advantages, face a qualitatively different set of security challenges that requires a rethinking of existing approaches. The modern threat landscape is shifting away from traditional network security toward issues related to improper access management and violations of the principle of least privilege (PoLP), as well as toward vulnerabilities in the software supply chain [1, 6, 8]. Improper permission configuration, which forms the so-called Over-Permissive Functions, remains one of the most critical and systematically reproducible vulnerabilities.

A special significance is acquired by the identified and thoroughly analyzed economic threat of the Denial of Wallet type [3]. Unlike classical denial-of-service attacks, it exploits the automatic scaling mechanisms of serverless platforms to inflict financial damage on the application owner without violating the formal availability of the service. This type of impact requires the implementation of intelligent limiting and monitoring systems that go beyond traditional DoS prevention and are focused on detecting anomalies in resource consumption and call patterns.

The conducted study has simultaneously revealed the need to systematize the disparate best practices present in the literature and industrial recommendations. It is precisely this lack of conceptual integrity that became the prerequisite for the development of the Author's Three-Level Serverless Security Model (ATMS), which offers a structured consolidation of protective measures into a single, logically consistent system.

In the course of the work, all the objectives set were consistently addressed. A study and classification of the basic security threats specific to Serverless architectures was carried out, including issues of PoLP implementation, economic attacks of the DoW type, and supply chain vulnerabilities [3, 4, 6]. An analysis of the Shared Responsibility Model was performed, as a result of which it was clearly indicated that the main area of user responsibility is concentrated in the security of code, data, configuration, and access management (IAM) [7]. On the basis of comparison and critical evaluation of existing approaches, a comprehensive system of recommendations was synthesized and the Author's Three-Level Serverless Security Model (ATMS) was proposed, which

systematizes protective measures at the Development, Deployment, and Execution levels.

The aggregate results of the study confirm the author's hypothesis that effective security in Serverless environments is achievable only under the condition of implementing a multi-level, proactive approach based on granular access control and deep integration of security mechanisms into the development life cycle within the DevSecOps paradigm. The article forms a coherent, systematized body of theoretical and applied material and proposes a practically applicable model that can serve as a benchmark for specialists working with Serverless technologies and addressing the tasks of ensuring their cybersecurity.

References

[1] X. Li, X. Leng, & Y. Chen. "Securing serverless computing: Challenges, solutions, and opportunities". IEEE Network, vol.37(2), pp.166–173, 2023. https://doi.org/10.1109/MNET.005.2100335.
[2] L. Ben-Shimol, D. Lavi, E. Klevansky, O. Brodt, D. Mimran, Y. Elovici, & A. Shabtai. "Detection of compromised functions in a serverless cloud environment". Computers & Security, vol.150, 2025. https://doi.org/10.1016/j.cose.2024.104261
[3] D. Kelly, F.G. Glavin, & E. Barrett. "Denial of wallet—Defining a looming threat to serverless computing". Journal of Information Security and Applications, vol. 60, 2021. https://doi.org/10.1016/j.jisa.2021.102843.
[4] J. Wen, Z. Chen, X. Jin, & X. Liu. "Rise of the planet of serverless computing: A systematic review". ACM Transactions on Software Engineering and Methodology, vol. 32(5), pp. 1–61, 2023. https://doi.org/10.1145/3579643.
[5] E. Marin, D. Perino, & R. Di Pietro. "Serverless computing: A security perspective". Journal of Cloud Computing, vol. 11, 2022. https://doi.org/10.1186/s13677-022-00347-w.
[6] P. Escaleira, V.A. Cunha, J.P. Barraca, D. Gomes, & R.L. "Aguiar. A systematic review on security mechanisms for serverless computing". Cluster Computing, vol. 28, 2025. https://doi.org/10.1007/s10586-025-05371-4 .
[7] R. Ouyang, J. Wang, H. Xu, S. Chen, X. Xiong, A. Tolba, & X. Zhang. "A Microservice and Serverless Architecture for Secure IoT System". Sensors, vol. 23(10), 2023. https://doi.org/10.3390/s23104868.
[8] B. Çinar. "The rise of serverless architectures: Security challenges and best practices". Asian Journal of Research in Computer Science, vol. 16(4), pp. 194–210, 2023. https://doi.org/10.9734/ajrcos/2023/v16i4382.
[9] S. Ahmadi. "Challenges and solutions in network security for serverless computing". International Journal of Current Science Research and Review, vol. 7(1), pp. 218–229, 2024. https://doi.org/10.47191/ijcsrr/V7-i1-23.
[10] K. Ni, S.K. Mondal, H.M.D. Kabir, T. Tan, & H. N. Dai. "Toward security quantification of serverless computing". Journal of Cloud Computing, vol.13, 2024. https://doi.org/10.1186/s13677-024-00703-y.