ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25mar023

A Review on Data Security and Privacy in Serverless Computing: Key Strategies, Emerging Challenges

Bharath Kumar Reddy Janumpally[1]

[1]Independent Researcher

Publication Date: 2025/03/15

Abstract: Serverless computing has revolutionized cloud-based application development by eliminating the need for developers to manage infrastructure. Cloud providers take care of provisioning, scalability, and security fixes, freeing developers to concentrate entirely on code composition. Serverless architectures, primarily driven by Function-as-a-Service (FaaS) and Backend-as-a-Service (BaaS), offer advantages such as cost efficiency, scalability, and reduced operational overhead. However, these benefits come with significant data security and privacy challenges, including authentication weaknesses, data exposure risks, and vendor lock-in. This paper explores the fundamentals of serverless computing, its key security concerns, and mitigation strategies. Analyzing existing literature and emerging trends provides insights into best practices for enhancing security and privacy in serverless environments. Their study also examines compliance requirements, encryption techniques, and architectural advancements that address these challenges.

Keywords: *Serverless Computing, Data Security, Privacy Protection, Encryption, Cloud Computing Security.*

How to Cite: Bharath Kumar Reddy Janumpally (2025). A Review on Data Security and Privacy in Serverless Computing: Key Strategies, Emerging Challenges. *International Journal of Innovative Science and Research Technology*, 10(3), 118-126. https://doi.org/10.38124/ijisrt/25mar023

I. INTRODUCTION

Traditionally, building and deploying applications required developers to manage the underlying infrastructure, like servers and databases. This involved tasks like provisioning servers, scaling resources, and handling security patches. Serverless computing changes this by letting developers focus solely on the code itself. The most recent trend in cloud computing is serverless computing, which aims to significantly streamline application maintenance and development[1]. Figure 1 shows how serverless computing works. The application logic is broken down into discrete, stateless functions that may communicate with one other and other cloud services. Business logic development remains the programmer's responsibility when cloud providers maintain infrastructure operations in this approach[2]. As an additional feature, serverless computing uses a pay-as-you-go pricing model, ensuring that clients are only paid for the resources that really use. Serverless services are now accessible through the major cloud providers Amazon, Microsoft, Google, IBM and Alibaba due to the expanding popularity of serverless computing. Security concerns linked to serverless computing must be addressed because of their critical importance[3].

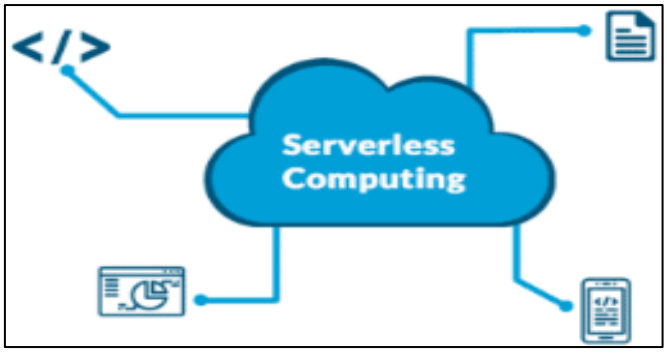

Fig 1 Serverless Computing

The security components within serverless computing platforms consist of authentication and authorization failures together with serverless-exclusive defects data leak threats and encryption maintenance approaches[4]. Serverless application security requires best practices to address emerging risks that are currently affecting these systems. The changing security environment of serverless frameworks can be fully understood through analysis of industry publications along with field observations[5]. Any organization that wants to succeed must prioritize data and information security, as safeguarding sensitive data is essential. The law requires companies to implement security procedures like penetration testing and provide a safe platform for software engineering along with security assurance standards that protect their data [6].

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25mar023

Absolutely remarkable development around serverless architectures from the single purpose services to the complex multi component application[7]. This serverless architecture empowers full-stack development with integration of front-end frameworks and services to deal with backend activities (mainly authentication and data processing)[8]. Google Cloud Run and Amazon Web Services Fargate container services enable you to further expand the serverless concept to the containerized app so that you have more control over the run-time environment, but you don't have to manage the infrastructure[9]. These advancements demonstrate how the serverless ecosystem has grown to meet the demands of a wider range of applications [10].

➢ *Structure of the Paper*

This paper is organized as follows: The foundations of serverless computing are outlined in Section II. Data security and privacy issues in serverless computing are covered in Section III. Section IV addresses key strategies for enhancing security and privacy. Section V presents the emerging challenges in data security and privacy. The literature and case studies are reviewed in Section VI. Findings and suggestions for further study are presented in Section VII.

II. FUNDAMENTALS OF SERVERLESS COMPUTING

Developers may concentrate on executing code instead of providing and managing servers thanks to serverless computing, which isolates infrastructure management. It is based on a pay-per-use paradigm and includes Backend-as-a-Service (BaaS) and Function-as-a-Service (FaaS). Key benefits include scalability, cost-efficiency, and flexibility, though challenges like security, cold starts, and vendor lock-in persist.

A. Architecture of Serverless Computing

The term "serverless computing" sums up the concept: it doesn't involve any server-related thought or care. Only when requests or events are processed do developers have to worry about the nitty-gritty of server administration and scale[11]. Here is their definition of serverless: Bypassing the need for developers to manage server resources, serverless computing allows for on-demand, automatically scalable, and time-based billing of code execution[12]. The three primary tiers of the serverless architecture are the base, compute, and scheduling layers. A look at Figure 2 reveals the Serverless architecture's concept.

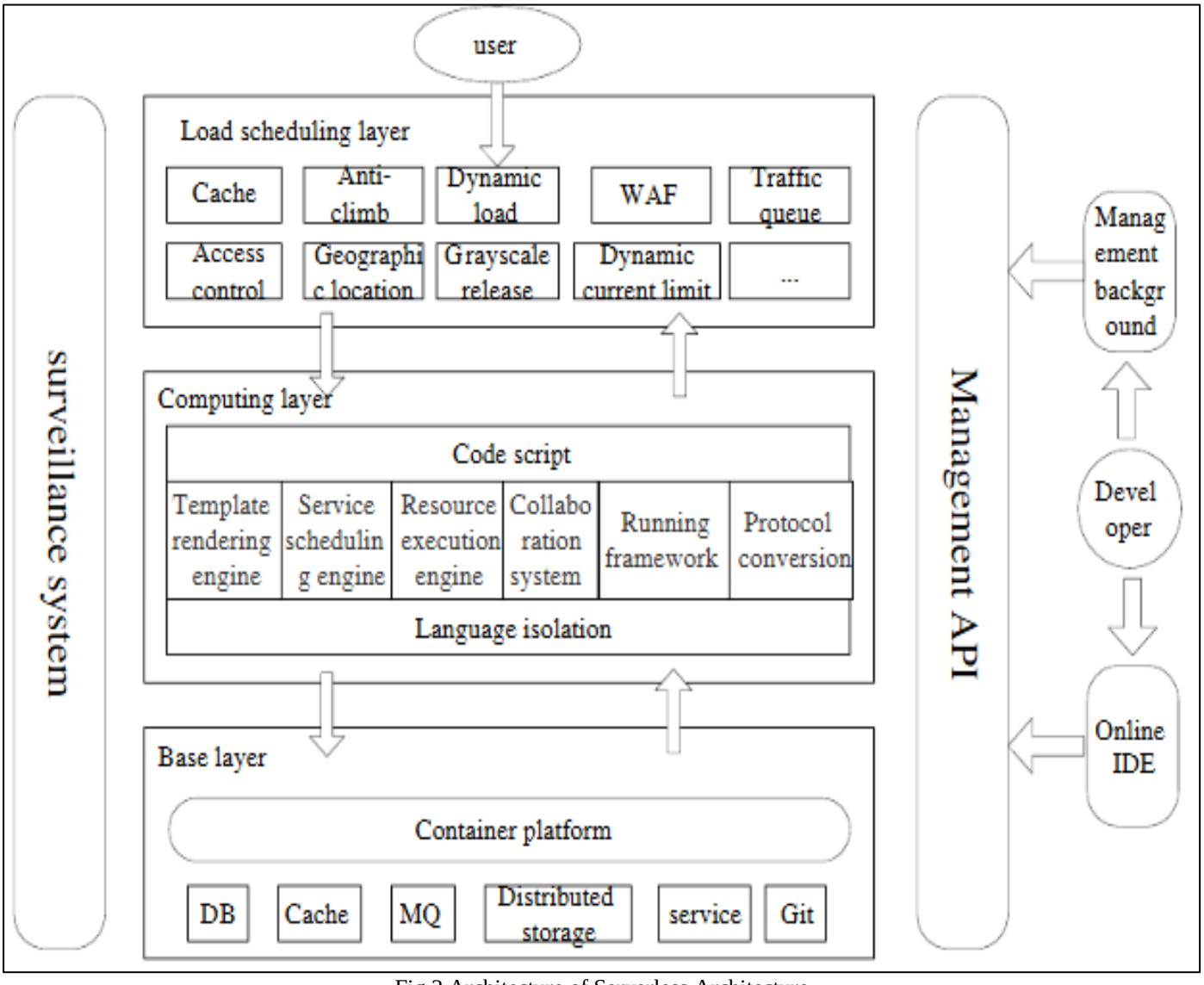

Fig 2 Architecture of Serverless Architecture

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25mar023

➢ *Base Layer*

The Base Layer serves as the foundation of the serverless architecture, providing essential resources and infrastructure for seamless execution[13]. It includes the Container Platform, which enables containerized execution, and a Database (DB) for storing structured and unstructured data.

- The Cache improves performance by storing frequently accessed information, while the Message Queue (MQ) manages communication between distributed services.
- Distributed Storage ensures scalable and resilient data storage, and the Service component handles microservices and backend operations.

➢ *Computing Layer*

The Computing Layer manages code execution and scheduling in a serverless environment while ensuring security and stability[14]. It includes a Template Rendering Engine for processing templates, a Service Scheduling Engine for resource allocation, and a Resource Execution Engine for managing execution.

- The Collaboration System facilitates function communication, while the Running Framework supports serverless execution.
- Additionally, Protocol Conversion ensures seamless interaction between different communication protocols, and Language Isolation maintains isolated execution environments for various code scripts.

➢ *Load Scheduling Layer*

The Load Scheduling Layer manages user requests and balances loads dynamically, ensuring efficient resource allocation. It includes Cache for faster data retrieval, Anti-climb for security, Dynamic Load for adaptive distribution, WAF for cyber threat protection, and Traffic Queue for regulating network flow[15]. Other key components include Access Control for managing permissions, Geographical Location for location-based request processing, Grayscale Release for phased updates, and Dynamic Current Limit for controlling request thresholds[16]. Currently, there are primarily four elements to Server Less's next goal:

- Add more languages.
- Web-based IDE capabilities have improved.
- More skills are configured.
- Incorporate an automated test system.

B. Key Components of Serverless Computing

The term "serverless" refers to a collection of technologies that have both "BaaS" and "FaaS" components.

➢ *Function as a Service (FaaS)*

FaaS is an environment for developing software. Developing serverless apps involves using client-side logic, making remote procedure calls on the cloud, and integrating with third-party services. These applications are cloud-based and event-driven[17]. Figure 3 shows that bare metal servers, virtual machines, and container engines all work together to support containers, which in turn support the FaaS. A database update, an HTTP request, or a message sitting in a queue are all examples of triggers that programmers may use to trigger the release of microservices or standalone apps using FaaS [18][19].

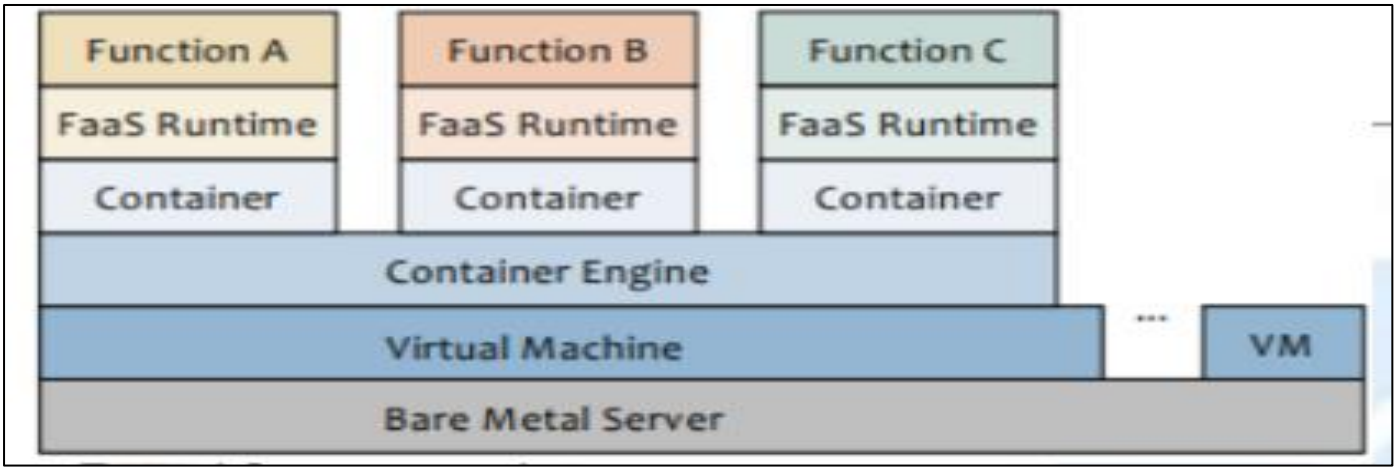

Fig 3 Function as a Service

➢ *Backend as a Service (BaaS)*

BaaS allows for the replacement of server-side components with pre-made services[20]. BaaS allows developers to outsource the backend components of an application, allowing them to construct and maintain all application logic on the front end. Google Firebase is a fully managed database that can be used directly from an application; it is an example of BaaS[21]. By entrusting data storage and other administrative duties to cloud providers, developers may focus on designing the user experience of the application using the BaaS model. In contrast, serverless applications are edge-based and event-driven, which BaaS apps should not necessarily lack[22].

C. Benefits and Challenges of Serverless Models

A web of compelling benefits and noticeable problems emerges from the examination of serverless computing. Serverless architecture has far-reaching effects on cloud computing, and this investigation reveals some of its facets.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25mar023

➢ *Enhanced Scalability and Flexibility:*

Serverless computing simply has the best built-in scalability and flexibility. The characteristic difference with respect to conventional cloud models that have to pre-provision all resources is that serverless systems dynamically allocate resources according to real demand. This flexibility guarantees that programs may effortlessly manage different workloads without any need for human involvement. Serverless computing provides perfect scalability for uncertain use patterns through its automatic scaling features which enable organizations to handle unanticipated traffic or user demand changes.

➢ *Cost-Efficiency and Pay-as-you-Go Model:*

The main attractive feature of serverless computing exists in its cost-effective nature. Organizations pay for their code execution time exclusively through the dedicated pay-as-you-go model of pricing. Applications making infrequent or surprise usage patterns derive the most benefits from this billing system because it produces sizable cost reductions. Serverless computing stands as a more efficient cloud option because it eliminates the requirement for dedicated servers together with their maintenance obligations[23].

➢ *Performance Overhead and Cold Start Latency:*

Serverless computing presents several disadvantages despite its advantages. A major problem is performance overhead, especially when it comes to cold start delay[24]. A significant delay may occur when invoking a serverless function after an inactive time. This delay is caused by the platform initializing the execution environment. The initial system start delay has the potential to damage applications that need quick performance responses[25]. It is important to optimize function initialization and implement techniques to minimize cold starts in order to mitigate this difficulty via rigorous design considerations.

➢ *Vendor Lock-in and Portability:*

Serverless computing faces the serious issue of vendor lock-in, which affects system migration between different providers. Some proprietary features on serverless platforms, such as AWS Lambda and Azure Functions, create challenges in switching between service providers when moving applications between providers[26]. Businesses that depend on a single vendor face tight operating flexibility because it makes them vulnerable to the entire supplier network. Organizations must meticulously assess their long-term strategy and contemplate strategies to increase portability and decrease dependence on a single vendor[23].

III. DATA SECURITY AND PRIVACY CONCERNS IN SERVERLESS COMPUTING

The following are some of the privacy and security issues with serverless computing, along with some examples of these issues in action:

➢ *Insufficient Authentication and Authorization*

Serverless computing has security risks due to insufficient authentication and authorization, which may let unauthorized users access sensitive data or run dangerous operations[27]. Authentication ensures that only legitimate users or systems can access a serverless application, while authorization defines their permissions and access levels.

- Lack of Strong Identity Management: Weak credentials or improper identity controls can expose serverless functions to unauthorized access.
- Overly Permissive Roles & RBAC Misconfiguration Granting excessive privileges increases security risks; implementing least privilege access is crucial.
- Absence of Multi-Factor Authentication (MFA) Without MFA, attackers can exploit compromised credentials to access critical functions.
- Insecure API Gateway Configurations Unprotected APIs may allow unauthorized users to invoke serverless functions.

➢ *Serverless Vulnerability Exploitation*

The term "serverless vulnerabilities" describes the openings that hackers may exploit in serverless architecture to compromise your system. Serverless applications are susceptible to various security threats due to their distributed and event-driven nature[28]. Attackers can exploit vulnerabilities within serverless functions, APIs, and third-party dependencies.

- **Injection Attacks:** Malicious input can manipulate serverless functions, leading to data leaks or unauthorized code execution[29].
- **Insecure Dependencies:** Serverless functions often rely on third-party libraries, which may contain unpatched vulnerabilities that attackers can exploit.
- **Event Injection Attacks:** Since serverless functions trigger events from sources like APIs, databases, and messaging queues, attackers can manipulate event payloads to exploit vulnerabilities.
- **Inadequate Logging and Monitoring:** It is challenging to identify and neutralize assaults in time to prevent harm due to the absence of real-time monitoring.
- **Privilege Escalation:** It is possible to get unauthorized access to sensitive data or services by using overprivileged functionalities[30].
- **Denial of Service (DoS) Attacks:** Attackers can overwhelm serverless functions with excessive requests[31], leading to performance degradation and increased costs.

➢ *Data Exposure Attack*

Data exposure attacks occur when sensitive data is improperly handled, stored, or transmitted in a serverless environment. Since serverless functions interact with various services, unsecured configurations or weak security controls can lead to unauthorized data access or leaks [23].

➢ *Compliance and Regulatory Concerns*

Regulation Compliance (e.g., GDPR, CCPA): Any company that handles the personal information of its customers must follow a number of data protection regulations. Important guidelines comprise:

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25mar023

- **General Data Protection Regulation (GDPR):** European Union regulations regarding the handling, archiving, and sharing of personal data. The document emphasizes the significance of user permission, data protection by design, and data subject rights [32].
- **California Consumer Privacy Act (CCPA):** A California law that grants people access to their personal information, allows them to correct or erase that information and offers them the option not to have it sold[33].

IV. KEY STRATEGIES FOR ENHANCING SECURITY AND PRIVACY

Ensuring security and privacy in serverless computing requires a multi-layered approach to mitigate risks such as unauthorized access, data breaches, and vulnerability exploitation. The following key strategies help strengthen security and safeguard user data.

➢ *Encryption Techniques for Data at Rest and in Transit*

Encryption is a critical security measure to protect sensitive data stored in databases (data at rest) and data transmitted between systems (data in transit). Proper encryption ensures confidentiality, preventing unauthorized access even if data is intercepted or compromised[34]. Data in serverless architectures must be protected at all stages since it may move between different services and storage places.

- **In-transit encryption:** All network communications should be encrypted using robust TLS/SSL protocols.
- **At-rest encryption:** Encrypt information kept in object storage, databases, and other long-term storage.
- **Key management:** Use strong key management procedures, such as frequent key rotation. Despite the fact that cloud providers often provide encryption services, businesses should keep custody of their encryption keys to guarantee data sovereignty and legal compliance.

➢ *Implementing Least Privilege Access*

A cybersecurity best practice known as the Principle of Least Privilege (PoLP) limits access to people, apps, and systems to that which is essential for them to carry out their duties. This lowers the possibility of unintentional setup errors, security breaches, and insider attacks. It entails giving a function only the minimal amount of access required to carry out its intended functions.

- **Granular IAM policies:** Establish functionally distinct roles with clearly specified permissions.
- **Time-bound access:** Use short-term credentials that expire quickly[35].
- **Regular permission reviews:** As functions change, continuously audit and improve access controls. Although it may take some effort to install at first, least privilege greatly lessens the possible effect of a compromised function. Businesses have to consider this an investment in their entire security posture[36].

➢ *Serverless-Specific Threat Detection and Mitigation*

Serverless computing brings security challenges to the table because of its dependency on events while also running brief executions through partnering with other companies. Preventive threat detection methods, along with protection measures, need implementation to protect serverless applications from cyber threats. Traditional security solutions cannot easily be extended due to serverless computing's unique architecture, so there is a need for threat detection and mitigation approaches tailored to serverless computing's unique combination of features.

- **Event-driven security analysis:** Detect anomaly invoking patterns of serverless functions.
- **API Gateway security enforcement:** It should implement API authentication, rate limiting and request validation in order to prevent from doing arbitrary calls to some functions[37].
- **Automated threat intelligence:** Use AI/ML-based models for detecting unusual behaviors and potential attacks[38].
- **Fine-grained logging and alerting:** It would enable detailed logging for function executions as well as generate real time alert for suspicious activities. These strategies must be continuously refined by cloud providers and organizations to better tackle evolving threats that revolve around serverless in particular[39].

V. EMERGING CHALLENGES IN DATA SECURITY AND PRIVACY

The more serverless computing and cloud-based environments evolve so do the challenge regarding security and privacy of the data. The challenges that arise are the serverless workload dynamics, dependence on 3rd party services, and more attack surfaces. There are Advanced security measures, compliance strategies to address these issues and constant monitoring.

➢ *Limited Visibility and Control*

Their work in serverless computing is never without its share of obstacles. The lack of control and visibility is a hurdle. There has to be an update to the typical security method in serverless computing, which does not allow managers direct access to the system. An unauthorized person may get access to the system or data due to the lack of understanding in these areas and the absence of standard networking protections[40]. System monitoring and enhanced security may address these issues. Identifying and reacting to security problems may be rather challenging for an organization without good network visibility[41].

➢ *Cold Start Attacks*

"Cold start" refers to the fact that early functions have not been run lately and is another problem with serverless computing. Security issues arise because of the window of opportunity this little delay gives the attackers to access data. The ineffectiveness of current standard security measures emphasizes the need for innovative alternatives[42]. New methods to shorten the cold start time without sacrificing security must be developed in order to lessen the impact of this difficulty[43]. Serverless apps may have these risks mitigated by using security measures. Solutions to decrease external access during cold start guarantee an improvement in serverless

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25mar023

computing system dependability as the organization overcomes these hurdles[44].

- *Injection Attacks and Isolation*

The constrained runtime system of serverless operations causes isolation problems, which include injection attacks. An attacker may seize an opportunity to acquire unauthorized access due to these obstacles. This emphasizes the need to implement a robust runtime protection system and a solid isolation strategy inside the organization[45]. A serverless architecture faces the main obstacle of maintaining system security among various threats while preserving operational efficiency. During the shared run time environment, the organization must implement robust security measures to identify attackers early on and guarantee the system's integrity. Building a secure and dependable system demands priority attention to these problems since system development increases its importance[46].

- *Resource Exhaustion and Denial-of-Service (DoS)*

Serverless computing faces opposing challenges during scaling up because resource fatigue along with denial-of-service attacks occur. Serverless systems remain exposed to harmful hacking activities because these attacks often cause poor performance alongside service outages[47]. Proactive approaches, such as effective rate limitation and planned use of auto-scaling capabilities, are necessary to address these difficulties[48]. Organizations can enhance their serverless system security through resource optimization and faster system response. This will guarantee robust performance even in the event that hostile attackers try to exploit this dynamic scaling serverless computing paradigm [49].

VI. LITERATURE OF REVIEW

This section presents the findings of earlier studies on the topic of data privacy and security in serverless computing. Table I provides a comparative overview of recent studies in Data Security and Privacy in Serverless Computing, highlighting their focus areas, performance insights, and potential future research directions.

Kumar (2019) gives an in-depth look at the serverless options from several popular cloud providers, including open source, AWS, Azure, and Google Cloud Platform. It provides side-by-side comparisons in important categories, such as computing, storage, databases, messaging, API administration, and tools. The report also includes a comparison of the various serverless architectures that are available, broken down into the most prevalent use cases supported by cloud providers. Additionally, it will highlight the technology's future, potential answers to open challenges, and advantages. In recent times, serverless computing, also known as Function-as-a-Service, has gained significant traction as a viable solution for improving the management of cost, dependability, availability, and scalability[41].

Dey, Reddy and Lavanya (2023) overview of serverless computing, with an emphasis on its underlying architectural concepts, a list of its key challenges, and an evaluation of possible future advancements in this rapidly evolving field. An extensive overview of the fundamental concepts of serverless computing is presented at the beginning of the text. The event-driven architecture, scalability, and shift in focus from infrastructure administration to code-centric development are its primary selling points. This research delves into the serverless computing architecture paradigms, with a particular emphasis on FaaS and BaaS. In addition to outlining current problems and possible solutions, this paper also offers future directions for serverless computing[50].

Ihtesham et al. (2023) makes a significant contribution to the area of serverless cloud security, specifically in relation to searchable encryption, by offering a more sophisticated method of data protection in a serverless computing environment without sacrificing usability. Customers and cloud providers alike have flocked to serverless computing for its adaptability, scalability, and speed of deployment. This study delves into Searchable Encryption as a Service (SEaaS) and presents a novel privacy-preserving Multiple Keyword Searchable Encryption (MKSE) scheme that runs in a serverless cloud setting. It achieves security objectives that have not been achieved before[51].

Govindarajan and Tienne (2023) analyse the intricacies of managing a serverless computing platform, as well as serverless computing and the open-source tools that aid in its implementation. Finally, the article suggests a cloud function scheduler for serverless computing that uses rank to distribute cloud services. Existing literature and early experiments make it clear that the suggested technique would affect throughput and success rate performance metrics for cloud function execution[52].

Wu et al. (2024) propose a privacy-preserving serverless FL scheme for IoT based on secure multiparty computation. They provide a formal security proof that demonstrates the resilience of their scheme against collusion attacks, thereby establishing its effectiveness in achieving robust data privacy. Federated learning (FL) when deployed in an Internet of Things (IoT) ecosystem can facilitate the collaborative training of a global model involving different IoT local systems. They also mitigate the fault tolerance limitation by using secret sharing[53].

Li, Leng and Chen (2023) offers the first comprehensive review of serverless security that takes into account both academic research and industry standards for protection. They provide a summary of the most pressing security issues, review the relevant literature and industry solutions, and catalogue areas where further study may be fruitful. After that, they check the security features of both commercial and open-source serverless systems, as well as industrial and academic solutions, for any gaps. At last, they provide a comprehensive picture of the state of the art in serverless security research. However, new security concerns have emerged due to some aspects of serverless computing, like the disjointed application boundaries[54].

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25mar023

Table 1 Literature on Data Security and Privacy in Serverless Computing

References	Focus on	Performance	Limitations & Future Work
Kumar (2019)	Comparative analysis of serverless offerings from AWS, Azure, GCP, and open-source platforms.	Highlights benefits such as cost efficiency, reliability, and scalability.	Identifies open problems, possible solutions, and future trends in serverless computing.
Dey, Reddy and Lavanya (2023)	Examination of serverless computing architecture, focusing on FaaS and BaaS.	Emphasizes event-driven nature and scalability improvements.	Discusses unresolved issues and potential developments in the field.
Ihtesham et al. (2023)	Security in serverless cloud computing, particularly in searchable encryption.	Introduces a privacy-preserving Multiple Keyword Searchable Encryption (MKSE) scheme.	Addresses previously unmet security goals but requires further optimization.
Govindarajan and Tienne (2023)	Analysis of open-source frameworks supporting serverless computing and complexities in platform management.	Proposes a rank-based cloud function scheduler to enhance throughput and success rates.	Further validation and performance evaluation are necessary.
Wu et al. (2024)	Privacy-preserving Federated Learning (FL) for IoT in serverless environments.	Utilizes secure multiparty computation to ensure data privacy and mitigate collusion attacks.	Addresses fault tolerance with secret sharing, but further testing in real-world scenarios is needed.
Li, Leng and Chen (2023)	Comprehensive survey of serverless security challenges, solutions, and research gaps.	Analyzes security measures from both academic and industrial perspectives.	Highlights fragmented application boundaries as a critical security challenge.

VII. CONCLUSION AND FUTURE WORK

Serverless computing offers scalability and cost efficiency, but its reliance on third-party cloud services and ephemeral execution models introduces new security and privacy concerns. This review identifies key risks, including data leakage, unauthorized access, and compliance issues. Existing security strategies, such as encryption, IAM policies, and runtime monitoring, provide foundational protection but require enhancement to address evolving threats. The research analyzed current security approaches and best practices, which include such solutions as Function-as-a-Service (FaaS) security models, encryption methods, access control frameworks and anomaly detection protocols. The implementation of serverless computing provides operational efficiency yet organizations face essential challenges to maintain safe and privacy-compliant operations when adopting this technology. The operational benefits of serverless computing come with a fundamental requirement for organizations to maintain secure and privacy-compliant settings.

Security of serverless should be future research to create standard security frameworks, enhance threat detection mechanisms and merge AI-driven security models into serverless security. As serverless architecture goes on, research should be done on securing and improving privacy frameworks against new threats. Something that a decent amount of serverless security would consist of would be to use AI and ML for mechanisms like automatic response, predictive threat analysis, and real-time anomaly detection. Overcoming these challenges with the help of interdisciplinary research and collaboration with the industry will be instrumental in the development of the respective secure and scalable serverless systems for further evolution.

REFERENCES

[1]. H. B. Hassan, S. A. Barakat, and Q. I. Sarhan, “Survey on serverless computing,” Journal of Cloud Computing. 2021. doi: 10.1186/s13677-021-00253-7.
[2]. M. Gopalsamy, “Predictive Cyber Attack Detection in Cloud Environments with Machine Learning from the CICIDS 2018 Dataset,” IJSART, vol. 10, no. 10, 2024.
[3]. Mani Shankar, S. Lingolu, and M. K. Dobbala, “A Review Paper on Serverless Computing : A Security,” J. Artif. Intell. Cloud Comput., no. 05, pp. 5872–5878, 2024.
[4]. Suhag Pandya, “Advanced Blockchain-Based Framework for Enhancing Security, Transparency, and Integrity in Decentralised Voting System,” Int. J. Adv. Res. Sci. Commun. Technol., vol. 2, no. 1, pp. 865–876, Aug. 2022, doi: 10.48175/IJARSCT-12467H.
[5]. A. Kushwaha, P. Pathak, and S. Gupta, “Review of optimize load balancing algorithms in cloud,” Int. J. Distrib. Cloud Comput., vol. 4, no. 2, pp. 1–9, 2016.
[6]. Mani Shankar, S. Lingolu, and M. K. Dobbala, “Data Security and Privacy Protection for Cloud Storage: A Survey,” J. Artif. Intell. Cloud Comput., 2024, doi: 10.1109/ACCESS.2020.3009876.
[7]. Rajarshi Tarafdar, “AI-Powered Cybersecurity Threat Detection in Cloud,” Int. J. Comput. Eng. Technol., p. 266, 2025.
[8]. Mani Shankar, S. Lingolu, and M. K. Dobbala, “LEAP Collaboration System,” in Journal of Artificial Intelligence & Cloud Computing, 2024.
[9]. S. Murri, S. Chinta, S. Jain, and T. Adimulam, “Advancing Cloud Data Architectures: A Deep Dive into Scalability, Security, and Intelligent Data Management for Next-Generation Applications,” Well Test. J., vol. 33, no. 2, pp. 619–644, 2024, [Online]. Available:

https://welltestingjournal.com/index.php/WT/article/view/128

[10]. S. Lingolu, M. Shankar, Dobbala, and M. Kumar, "Serverless Architectures and Their Influence on Web Development," J. Artif. Intell. Cloud Comput., vol. 3, no. 2, pp. 1–6, 2024, doi: 10.47363/jaicc/2024(3)297.

[11]. B. Boddu, "Cloud-Based E-CCNN Architecture for Early Heart Disease Detection A Machine Learning Approach," Int. J. Med. Public Heal., vol. 14, no. 4, p. 9, 2024, [Online]. Available: https://www.ijmedph.org/Uploads/Volume14Issue4/73. [1095._IJMEDPH_Jafar] 374-382.pdf

[12]. V. S. Thokala, "Improving Data Security and Privacy in Web Applications : A Study of Serverless Architecture," Int. Res. J., vol. 11, no. 12, pp. 74–82, 2024.

[13]. D. Barcelona-Pons, P. Sutra, M. Sánchez-Artigas, G. París, and P. García-López, "Stateful Serverless Computing with Crucial," in ACM Transactions on Software Engineering and Methodology, 2022. doi: 10.1145/3490386.

[14]. J. Spring, "Monitoring cloud computing by layer, Part 2," IEEE Secur. Priv., 2011, doi: 10.1109/MSP.2011.57.

[15]. L. Jiang, Y. Pei, and J. Zhao, "Overview Of Serverless Architecture Research," J. Phys. Conf. Ser., vol. 1453, p. 12119, 2020, doi: 10.1088/1742-6596/1453/1/012119.

[16]. H. Xi, M. Zhu, K. Y. Lee, and X. Wu, "Multi-timescale and control-perceptive scheduling approach for flexible operation of power plant-carbon capture system," Fuel, 2023, doi: 10.1016/j.fuel.2022.125695.

[17]. J. Scheuner and P. Leitner, "Function-as-a-Service performance evaluation: A multivocal literature review," J. Syst. Softw., 2020, doi: 10.1016/j.jss.2020.110708.

[18]. R. Bishukarma, "Adaptive AI-Based Anomaly Detection Framework for SaaS Platform Security," Int. J. Curr. Eng. Technol., vol. 12, no. 07, pp. 541–548, 2022, doi: https://doi.org/10.14741/ijcet/v.12.6.8.

[19]. R. A. P. Rajan, "A review on serverless architectures-Function as a service (FaaS) in cloud computing," Telkomnika (Telecommunication Comput. Electron. Control., vol. 18, no. 1, pp. 530–537, 2020, doi: 10.12928/TELKOMNIKA.v18i1.12169.

[20]. S. Plangi, "Overview of Backend as a Service Platforms," Univ. Tartu, 2016.

[21]. A. and P. Khare, "Cloud Security Challenges : Implementing Best Practices for Secure SaaS Application Development," Int. J. Curr. Eng. Technol., vol. 11, no. 6, pp. 669–676, 2021, doi: https://doi.org/10.14741/ijcet/v.11.6.11.

[22]. A. Jangda, D. Pinckney, Y. Brun, and A. Guha, "Formal foundations of serverless computing," Proc. ACM Program. Lang., 2019, doi: 10.1145/3360575.

[23]. Er. Venkata Ramanaiah Chintha, "Server Less Computing_ Evaluating the Benefits and Challenges of Server Less Architecture in Cloud Computing." 2024.

[24]. V. S. Thokala, "A Comparative Study of Data Integrity and Redundancy in Distributed Databases for Web Applications," Int. J. Res. Anal. Rev., vol. 8, no. 04, pp. 383–390, 2021.

[25]. V. Kolluri, "A Pioneering Approach To Forensic Insights: Utilization Ai for Cybersecurity Incident Investigations," Int. J. Res. Anal. Rev., vol. 3, no. 3, 2016.

[26]. Suhag Pandya, "Integrating Smart IoT and AI-Enhanced Systems for Predictive Diagnostics Disease in Healthcare," Int. J. Sci. Res. Comput. Sci. Eng. Inf. Technol., vol. 10, no. 6, pp. 2093–2105, Dec. 2023, doi: 10.32628/CSEIT2410612406.

[27]. A. P. A. S. Neepa kumari Gameti, "Innovations in Data Quality Management: Lessons from the Oil & Gas Industry," Int. J. Res. Anal. Rev., vol. 11, no. 3, pp. 889–895, 2024.

[28]. H. S. Chandu, "A Review of IoT-Based Home Security Solutions: Focusing on Arduino Applications," TIJER – Int. Res. J., vol. 11, no. 10, pp. a391–a396, 2024.

[29]. T. Gaber, A. El-Ghamry, and A. E. Hassanien, "Injection attack detection using machine learning for smart IoT applications," Phys. Commun., 2022, doi: 10.1016/j.phycom.2022.101685.

[30]. M. Mehmood, R. Amin, M. M. A. Muslam, J. Xie, and H. Aldabbas, "Privilege Escalation Attack Detection and Mitigation in Cloud Using Machine Learning," IEEE Access, vol. 11, 2023, doi: 10.1109/ACCESS.2023.3273895.

[31]. S. Weisman, "What are Denial of Service (DoS) attacks? DoS attacks explained," NortonLifelock, 2020.

[32]. P. S. N. Patel, D. Parikh, R. K. Eranna, J. Patel, and P. Siddhapura, "Machine Learning Based Security Device For Cloud Computing," 2024

[33]. M. D. Sanjeev Prakash, Jesu Narkarunai Arasu Malaiyappan, Kumaran Thirunavukkarasu, "Achieving Regulatory Compliance in Cloud Computing through ML," Adv. Int. Advert., vol. 2, no. 5, pp. 1–66, 2024.

[34]. A. P. A. Singh and N. Gameti, "Innovative Approaches to Data Relationship Management in Asset Information Systems," vol. 12, no. 6, pp. 575–582, 2022.

[35]. N. Patel, "Secure Access Service Edge(Sase): Evaluating The Impact Of Convereged Network Security Architectures In Cloud Computing," J. Emerg. Technol. Innov. Res., vol. 11, no. 3, pp. e703–e714, 2024.

[36]. Y. Kannan, "Serverless Security: Best Practices for Protecting Functions-as-a-Service," Int. J. Sci. Res., vol. 13, no. 7, pp. 1190–1194, 2024, doi: 10.21275/sr24723103837.

[37]. P. G. Himanshu Kumar et al., "The Journey to Intent-based Networking: Ten Key Principles for Accelerating Adoption," IEEE Access, 2017.

[38]. S. Arora and S. R. Thota, "Automated Data Quality Assessment And Enhancement For Saas Based Data Applications," J. Emerg. Technol. Innov. Res., vol. 11, pp. i207–i218, 2024, doi: 10.6084/m9.jetir.JETIR2406822.

[39]. E. Marin, D. Perino, and R. Di Pietro, "Serverless computing: a security perspective," Journal of Cloud Computing. 2022. doi: 10.1186/s13677-022-00347-w.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/25mar023

[40]. R. Arora, S. Gera, and M. Saxena, "Mitigating Security Risks on Privacy of Sensitive Data used in Cloud-based ERP Applications," in 2021 8th International Conference on Computing for Sustainable Global Development (INDIACom), 2021, pp. 458–463.
[41]. M. Kumar, "Serverless Architectures Review, Future Trend and the Solutions to Open Problems," Am. J. Softw. Eng., 2019, doi: 10.12691/ajse-6-1-1.
[42]. A. Goyal, "Optimising Cloud-Based CI/CD Pipelines: Techniques for Rapid Software Deployment," Tech. Int. J. Eng. Res., vol. 11, no. 11, pp. 896–904, 2024.
[43]. S. R. Thota, S. Arora, and S. Gupta, "Hybrid Machine Learning Models for Predictive Maintenance in Cloud-Based Infrastructure for SaaS Applications," 2024, pp. 1–6. doi: 10.1109/ICDSNS62112.2024.10691295.
[44]. M. Golec, G. K. Walia, M. Kumar, F. Cuadrado, S. S. Gill, and S. Uhlig, "Cold Start Latency in Serverless Computing: A Systematic Review, Taxonomy, and Future Directions," J. ACM, vol. 37, no. 4, 2023, doi: 10.1145/3700875.
[45]. M. R. S. and P. K. Vishwakarma, "The Assessments of Financial Risk Based on Renewable Energy Industry," Int. Res. J. Mod. Eng. Technol. Sci., vol. 06, no. 09, pp. 758–770, 2024.
[46]. J. Xiong, M. Wei, Z. Lu, and Y. Liu, "Warmonger: Inflicting Denial-of-Service via Serverless Functions in the Cloud," in Proceedings of the ACM Conference on Computer and Communications Security, 2021. doi: 10.1145/3460120.3485372.
[47]. B. Boddu, "Securing and Managing Cloud Databases for Business - Critical Applications," J. Eng. Appl. Sci. Technol., 2025.
[48]. S. Murri, "Data Security Environments Challenges and Solutions in Big Data," Int. J. Curr. Eng. Technol., vol. 12, no. 6, pp. 565–574, 2022.
[49]. D. Kelly and F. Glavin, "DoWTS – Denial-of-Wallet Test Simulator: Synthetic data generation for preemptive defence," J. Intell. Inf. Syst., vol. 60, pp. 1–24, 2022, doi: 10.1007/s10844-022-00735-3.
[50]. N. S. Dey, S. P. K. Reddy, and G. Lavanya, "Serverless Computing: Architectural Paradigms, Challenges, and Future Directions in Cloud Technology," in 7th International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud), I-SMAC 2023 - Proceedings, 2023. doi: 10.1109/I-SMAC58438.2023.10290253.
[51]. M. Ihtesham et al., "Privacy Preserving and Serverless Homomorphic-Based Searchable Encryption as a Service (SEaaS)," IEEE Access, 2023, doi: 10.1109/ACCESS.2023.3324817.
[52]. K. Govindarajan and A. De Tienne, "Resource Management in Serverless Computing - Review, Research Challenges, and Prospects," in 12th IEEE International Conference on Advanced Computing, ICoAC 2023, 2023. doi: 10.1109/ICoAC59537.2023.10249574.
[53]. C. Wu, L. Zhang, L. Xu, K. K. R. Choo, and L. Zhong, "Privacy-Preserving Serverless Federated Learning Scheme for Internet of Things," IEEE Internet Things J., vol. 11, no. 12, pp. 22429–22438, 2024, doi: 10.1109/JIOT.2024.3380597.
[54]. X. Li, X. Leng, and Y. Chen, "Securing Serverless Computing: Challenges, Solutions, and Opportunities," IEEE Netw., 2023, doi: 10.1109/MNET.005.2100335.