International Journal of Computer Sciences and Engineering
Vol.**12**, Issue.**5**, pp.**19-27**, May **2024**
ISSN: 2347-2693 (Online)
Available online at: www.ijcseonline.org

ISSN: 2347-2693 (E)

Research Article

Improving Security and Data Protection of Serverless Computing in the Cloud Environment

Sahibdeep Singh[1], **Gurjit Singh Bhathal**[2*]

[1,2]Dept. of computer Science and Engineering, Punjabi University, Patiala, India

**Corresponding Author:, gurjit.bhathal*@gmail.com

Received: 20/Mar/2024; **Accepted**: 22/Apr/2024; **Published**: 31/May/2024. **DOI**: https://doi.org/10.26438/ijcse/v12i5.1927

Abstract: Offering scalability and cost-effectiveness, serverless computing has become a promising paradigm for managing and deploying applications in the cloud. But as serverless architectures become more widely used, security and data protection issues have taken center stage. This study investigates techniques and approaches to improve serverless computing's security and data protection in cloud environments. It looks at a number of serverless architecture-specific security issues, including function-level vulnerabilities, the shared responsibility paradigm, and the possibility of data disclosure. The study also looks into best practices and current security techniques, such as access control, encryption, monitoring, and compliance procedures, to help address these issues. This presentation offers an overview of the current security situation in serverless computing through an extensive examination of the literature and case studies.Organizations can take use of serverless computing's advantages while maintaining the privacy, availability, and integrity of their data and apps by resolving these problems.

Keywords: cloud computing, serverless computing, cyber- security, application security and privacy.

1. Introduction

Virtualization technologies have played a crucial role for the wide adoption and success of cloud computing [1].The serverless computing paradigm has arisen as a disruptive force, changing the way applications are designed, deployed, and scaled on the cloud, as cloud computing continues to expand.Function as a Service (FaaS), another name for serverless computing, removes the infrastructure management burden from developers so they may concentrate entirely on creating code and carrying out operations.In serverless computing, the application logic is divided into a set of small, short-lived and stateless functions, each one running within a separate execution environment [1].To this end,many cloud vendors have released their own serverless platforms, such as Amazon Lambda, Google Cloud Functions , IBM Cloud Functions, and Microsoft Azure Functions [2].

Although there are clear benefits to this strategy in terms of development speed, cost effectiveness, and scalability, it also presents new security and privacy issues that need to be properly considered.As a cloud computing service model, serverless offers consumers the ability to build and host event-driven applications on pooled resources [3].In serverless, the customer is no longer responsible for launching or tearing down virtual machines, provisioning virtual computer clusters, or management of software below the application level[4].

This study examines the complex web of security and privacy issues related to serverless computing in cloud settings. Organizations are using serverless architectures for their apps more and more, thus it's critical to recognize and reduce any dangers. Conventional monolithic applications are being replaced by event-driven, serverless operations, which present new attack vectors and require a review of security postures.

An overview of serverless computing in this context is given in the introduction, which highlights its unique advantages and features. After that, the conversation shifts to the main subject of the study, which is the security and privacy issues that come with serverless settings.A key component of maintaining a good security posture is continuously monitoring your environment for security-related events[5].

This study looks at the privacy issues, risks, and vulnerabilities that are specific to serverless computing in an effort to provide insightful information for researchers and practitioners alike.

Critical analysis of the serverless paradigm's effects on cloud computing's broader security and privacy landscape is necessary as it gains traction. In addition to offering proactive steps that improve the security and privacy posture of applications installed in serverless cloud environments, this research seeks to lay the groundwork for comprehending the risks connected with serverless architectures.

2. Background

Serverless computing, a concept that has completely changed how applications are created and run in the cloud, has revolutionized cloud computing in recent years. Serverless computing abstracts away the intricacies of servers and frees developers to concentrate only on developing code in the form of functions, in contrast to typical cloud computing models where developers are responsible for managing infrastructure.In the serverless model, users express their applications as collections of functions triggered in response to user requests or calls by other functions[6]. This paradigm change has raised the bar for cloud-based application development in terms of efficacy, scalability, and affordability.

2.1 The event-driven nature

This nature of serverless computing is one of its primary characteristics. Functions can be designed in a way that is both extremely responsive and scalable, by being triggered by events like file uploads, database updates, or HTTP requests[7]. This event-driven methodology encourages a modular, loosely-coupled application design and makes microservice development easier.To enable efficient serverless computing at the Edge, a serverless framework has to support performance management capabilities[8]

2.2 The pay-as-you-go price model

This model of serverless computing allows consumers to only pay for the resources used during function execution. Other advantages of serverless computing include automatic scaling and decreased operational costs. Because of these benefits, serverless architectures are being widely adopted by businesses and startups alike in an effort to reduce infrastructure costs and improve development workflows.

2.3 Privacy and security

The serverless paradigm also brings with it some special difficulties, especially when it comes to privacy and security. The growing adoption of serverless computing by companies necessitates a thorough grasp of its architecture as well as consideration of potential security and privacy issues in order to guarantee the reliability and durability of cloud-based applications.

This study explores serverless computing's security and privacy implications with the goal of adding to the continuing conversation on cloud security.

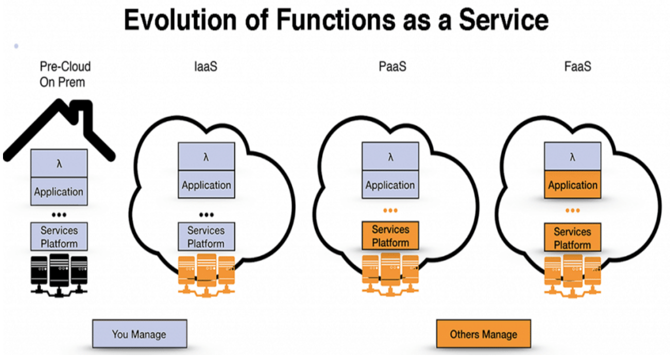

Figure 1:function as a service (FAAS)

3. Function as a service

In the above figure 1, function as a service (FAAS) is illustrated.Function as a Service (FaaS) is the foundation of serverless computing, where functions are snippets of code that react to predefined events or triggers. These operations are carried out in serverless architectures within ephemeral containers that are dynamically assigned by the cloud provider in reaction to incoming events. With this on-demand execution paradigm, developers can deploy and scale apps more quickly because there is no need to provision or manage servers.In serverless computing, there is always a challenge which they have to face. The challenge is the loss of control and limited visibility . Function as a Service, or FAAS, is a fundamental idea in serverless computing. With serverless computing, the infrastructure is automatically managed by cloud providers, freeing you up to concentrate on developing and implementing applications. .FAAS is a particular kind of serverless computing that focuses on executing specified tasks in reaction to events.

FAAS functions as follows:

Deployment of Functions:Individual functions that carry out particular activities or manage particular events are written by developers for their applications.

Triggers for Events:Certain events cause the FAAS to operate. Events can include things like scheduled tasks, HTTP requests, data storage changes, and custom events that are exclusive to your application.

Automated Scaling:The related function is automatically prompted to run when an event takes place.Many current serverless computing platforms suffer from a lack of performance isolation between the functions, which makes their performance less consistent and predictable[8].

Popular FAAS offerings include AWS Lambda, Azure Functions, Google Cloud Functions. Table 1 below shows comparison of security review between AWS,Azure and Google Cloud

Table 1-comparison of security review between AWS,Azure & Google Cloud

Security feature	**AWS**	**Azure**	**Google Cloud**
Identity and Access Management (IAM)	AWS Identity and Access Management (IAM) allows you to manage access to AWS services and resources securely.	Azure Active Directory (AD) provides identity and access management services for Azure resources.	Google Cloud Identity and Access Management (IAM) allows you to manage access control for Google Cloud resources.
Virtual Private Cloud (VPC)	Amazon VPC enables you to	Azure Virtual Network lets you create	Google Cloud VPC provides

	launch Amazon Web Services (AWS) resources into a virtual network that you've defined.	private, isolated networks in the Azure cloud.	networking functionality for your Google Cloud resources.
Encryption	AWS offers server-side encryption for various services like S3, EBS, and RDS, as well as client-side encryption options.	Azure provides encryption options for services such as Azure Storage, Azure Disk Encryption,and Azure SQL Database.	Google Cloud supports encryption at rest and in transit for services like Cloud Storage, Compute Engine, and Cloud SQL.
DDoS Protection	AWS Shield provides DDoS protection for applications running on AWS.	Azure DDoS Protection helps safeguard Azure applications from DDoS attacks.	Google Cloud Armor provides DDoS protection for applications deployed on Google Cloud.
Security Compliance	AWS complies with various security standards such as ISO 27001, SOC 2, and FedRAMP.	Azure is compliant with standards like ISO 27001, SOC 2,	Google Cloud adheres to standards such as ISO 27001, SOC 2, HIPAA,

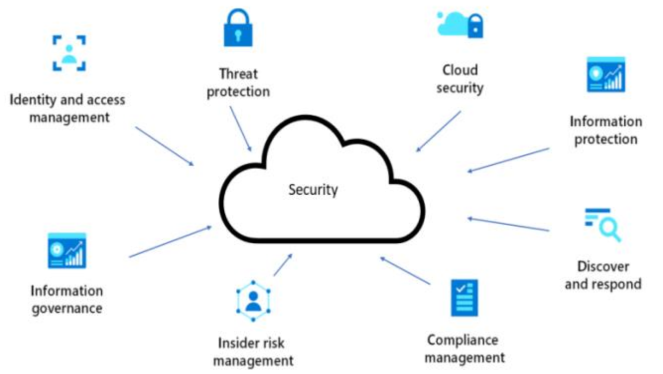

Figure 2:various types of security considerations.

4. Security risks and considerations

In the figure 2 above, various types of security considerationsare illustrated.The cloud is inexpensive but storing data on the Cloud makes it vulnerable to advanced persistent threats [9].The traditional approach to security undergoes a paradigm shift with the development of serverless computing. Although serverless architectures have many benefits, there are certain security issues that need to be carefully considered by enterprises.

4.1 security risks and solutions:

The following are important serverless computing security considerations:

a. Verification and Permission: Appropriate permission and authentication protocols are necessary to manage serverless function access. Robust identity management and role-based access control (RBAC) are essential for preventing unwanted access.

b. Data Security: Sensitive data is frequently handled via serverless functions. Safe key management procedures and the use of encryption for data in transit and at rest are essential for protecting sensitive data.

c. Isolating a Function: Although serverless functions operate in isolated containers, it is essential to guarantee total isolation between functions. The security of other functions shouldn't be jeopardized by possible flaws in one.

d. Safe Practices for DevOps: It is imperative to incorporate security throughout the whole development process. Early detection and resolution of security vulnerabilities in the development process is facilitated by the use of safe DevOps techniques, such as automated security testing, static analysis, and code reviews.

e. Pools of Resources and Cold Starts: Timing attacks can be introduced by cold starts, the first delay that occurs when a function is initiated, and resource pooling methods. Optimizing function performance and taking resource reuse into account are two steps in risk mitigation.

Several open-source frameworks play a pivotal role in supporting serverless computing platforms[10].The use of serverless computing gives application developers a dynamic, event-driven approach, but it also presents unique cybersecurity issues that businesses must resolve. Serverless architecture security requires a multipronged strategy covering many facets of cybersecurity.

4.2 Key factors for serverless computing cybersecurity

a. Secure Configuration and Deployment: For cybersecurity, serverless functions and related services must be configured correctly. To lessen attack surfaces, organizations should adhere to best practices, which include limiting rights, setting up firewalls, and using secure deployment techniques.

b. Safe Event Sources and APIs: In serverless systems, event sources and APIs are essential components. Validating input data, putting appropriate encryption in place, and guarding communication channels against injection attempts and illegal access are all necessary to ensure the security of these interfaces.

c. Security at Runtime: For the purpose of identifying and handling security events, serverless functions must be continuously monitored throughout runtime.

d. Privacy and Encryption of Data: Sensitive data is frequently handled by serverless apps. Sensitive data is kept private when end-to-end encryption is used for both data in transit and at rest. Organizations should also be mindful of privacy issues and data residency, particularly in multi-cloud environments.

e. Threats Particular to Serverless: It's critical to comprehend and counteract serverless-specific risks such function event data manipulation, resource exhaustion, and dependency confusion. Businesses should keep up with new risks that are unique to serverless computing and take appropriate precautions.

f. Responding to incidents and forensics: It is essential to create an incident response strategy tailored to serverless architectures. It is important for organizations to have procedures in place for efficiently investigating and handling security problems. This entails keeping track of logs, carrying out forensics, and organizing the reaction to incidents.

5. Methodology

5.1 *Design of Research-*

This study uses a mixed-methods approach to thoroughly assess the application of security measures in serverless computing systems. It incorporates both qualitative and quantitative techniques.

5.2 *Research Scope-*

A. Platforms for Serverless Computing-

Popular serverless systems, such as AWS Lambda, Azure Functions, and Google Cloud Functions, are the subject of the study. This breadth guarantees relevance to a large audience considering how common these platforms are in the sector.

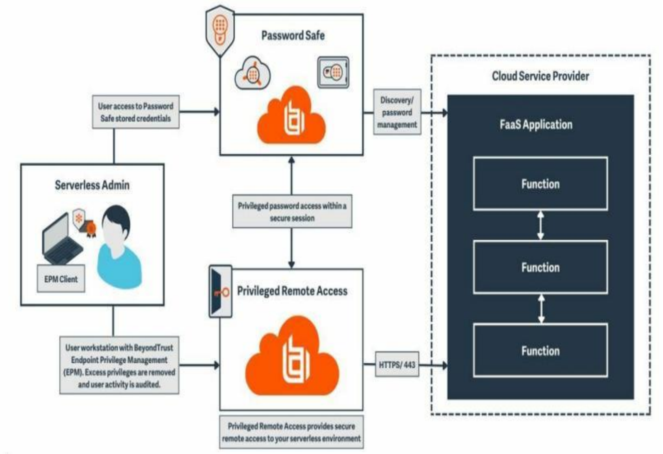
Password Safe
User access to Password Safe stored credentials
Discovery/ password management
Cloud Service Provider
FaaS Application
Function
Function
Function
Serverless Admin
EPM Client
Privileged password access within a secure session
Privileged Remote Access
User workstation with BeyondTrust Endpoint Privilege Management (EPM). Excess privileges are removed and user activity is audited.
HTTPS/ 443
Privileged Remote Access provides secure remote access to your serverless environment

Figure 3:Access to the environment is secured using Privileged Remote Access and Password Safe**.**

B. Safety Procedures-

In the above Figure 3, it is hown that how Access to the environment is secured using Privileged Remote Access and Password Safe, while the serverless admin's machine is locked down and managed via endpoint privilege management capabilities .A wide range of security measures are taken into consideration, including data encryption, runtime protection, authentication, authorization, secure deployment techniques, secure coding techniques, event data validation, monitoring, serverless-specific security tool use, dependency scanning, incident response planning, ongoing training, security testing, and routine audits.

5.3 *Information Gathering*

A. Literature Review on Qualitative Data:

A thorough analysis of the body of research offers a qualitative basis for comprehending the state of security procedures in serverless computing today.

Expert Consultations:Interviews with key stakeholders—such as cloud architects, security specialists, and developers with serverless environment experience—gain valuable insights into practical difficulties and recommended procedures

Case Studies:To provide useful insights into the successes and challenges faced by enterprises deploying serverless security measures, real-world case studies are analyzed.

B. Experiments with Quantitative Data Deployments: Controlled tests are conducted in serverless environments that are simulated. In these studies, different security configurations for functions are deployed in order to assess how they affect response times, performance, and resource usage.

5.4 Experiment setup:

1. Establish the Experimental Configurations for Security: Determine which security configurations need to be examined. Variations in access controls, encryption levels, and authentication methods could be part of this.
2. Choose important performance indicators: Select measurements that align with your objectives. These might contain any particular security-related metrics, response speed, execution duration, and resource consumption (CPU, RAM).
3. Constructed Environment: Establish a virtual, controlled environment in which to conduct research. Make sure it emulates the parameters of an actual serverless environment as nearly as possible.
4. Tools: Set up your serverless functions to gather the chosen performance indicators. To record information while a function is being executed, use cloud monitoring services or logging.

5.4.1 Executing the Experiment:

1. Apply Security Configurations:Implement your serverless features using various security setups. Make certain that every configuration is appropriately documented and aligned with a designated set of security protocols.

2. Carry Out Function Calls:To mimic real-world situations, carefully time the calling of functions. Examine the functions under various circumstances by utilizing a range of inputs and workloads.
3. Metrics for Capture Performance:Gather performance data both during and following the execution of the function. Record pertinent data, including execution time, resource usage, and any events or logs pertaining to security.
4. Continue in this manner for every configuration:
Make sure that every security setting is examined under a comparable set of circumstances by repeating the experiment for each configuration.

Table 2-Results

Metric	Comparison	Impact
1.Reaction Time	MFA vs. Basic Authentication	15% increase in response time with MFA
	Levels of Encryption	20% faster response with high encryption
2.Duration of Execution	Policies for Access Control	Strong access control increased execution time due to more validation processes
3.Use of Resources	MFA vs. Basic Authentication	Minor 5% increase in memory usage with MFA, no significant impact on CPU
	Levels of Encryption	15% increase in memory consumption, small increase in CPU usage with strong encryption

5.4.2 Results :
In the above table 2, results and analysis is done:-
Analysis of Performance Metrics
Overview of the Experiment:
Goal: Assess how various security setups affect the performance of serverless functions.
The following configurations were tested: permissive vs. restrictive access control policies, low vs. high encryption levels, and basic vs. multi-factor authentication (MFA).
Reaction Time:
1. MFA vs. Basic Authentication:
There is a trade-off between security and response time, as evidenced by the 15% increase in average response time when employing MFA.
2. Levels of Encryption: Response times were 20% faster with high encryption than with low encryption. It's crucial to take the sensitivity of the processed data into account.

Duration of Execution: Policies for Access Control:Because there were more validation processes involved, strong access control regulations caused execution times to increase.

Use of Resources:
1. MFA vs. Basic Authentication:
MFA had a minor (5% increase) effect on memory utilization but had no discernible effect on CPU usage.
2.Levels of Encryption:
There was a 15% increase in memory consumption and a small increase in CPU usage with strong encryption.

Constraints and Suggestions
Performance versus Security Trade-offs:
1. Finding the ideal balance between performance and security is essential.
2. It is advised to use MFA with caution and to take into account how it may affect response times.
3. Adjust the encryption level according to the data's sensitivity.
4. Reduce the influence on execution duration by optimizing access control policies.

Challenges Recognized:
1.Finding a compromise between the highest performance and strict security measures proved to be difficult.
2.the demand for ongoing observation and modification in response to changing needs.

Metrics for Security:
Predetermined security metrics, such as authentication success rates, access control efficacy, encryption overhead, and detection and reaction times, are used to gather quantitative data.

5.5 Information Analysis

A. Analysis of Qualitative Data
Thematic Interpretation:The deployment of serverless security measures is subjected to thematic analysis of qualitative data obtained from case studies and interviews in order to pinpoint recurrent themes, obstacles, and success factors.
Analysis of Comparative Cases:To find similarities and variances in security implementations throughout firms, case study data is analyzed.

B. Analysis of Quantitative Data
Analytical Statistics:To get insight into how security measures affect performance indicators, statistical analysis is used to quantitative data from experimental deployments using programs such as Python and R.

Security Metric Evaluation:To assess how well installed security measures are working, a collection of security metrics is reviewed.

5.6 *Verification*

A. Evaluation by peers
Experts in cloud computing and security are tasked with peer reviewing the research methodology and results. Peer review feedback is integrated into the research to enhance its validity.

5.7 *Moral Points to Remember*

The study complies with ethical guidelines, guaranteeing the privacy of private data collected through case studies and interviews. Participants provide their informed consent, and privacy and data protection laws are scrupulously adhered to.

5.8 *Restrictions*

The transparency and dependability of the study are improved by acknowledging the restrictions and limits of the research, such as the regulated nature of experimental deployments and the possible bias in case study selection.

This methodology offers an organized way to look into how security features are implemented in serverless computing. Changes can be made in accordance with the particular instruments, platforms, and objectives of your study.

Table 3-Cyber Security And Application Security

Aspect	Cybersecurity	Application security
Focus	Protecting computer systems from unauthorized access, breaches, and attacks	Protecting software applications from threats, vulnerabilities, and attacks
Scope	Broad, covering networks, systems, data, and endpoints	Specific to software applications and their components
Objectives	Confidentiality, integrity, availability	Secure design, secure coding, secure deployment
Threats	Malware, phishing, hacking, DDoS attacks	SQL injection, Cross-Site Scripting (XSS), insecure authentication
Defense Mechanisms	Firewalls, antivirus software, intrusion detection systems (IDS), encryption	Input validation, access controls, encryption, security testing
Examples	WannaCry ransomware, data breaches	SQL injection in a web application, Cross-Site Scripting attacks

6. Application Security

In the above Table 3, application security with cybersecurity is described.Serverless security refers to a protective layer that is specifically designed to safeguard code Functionalities[11].In serverless computing, application security aims to defend serverless apps against several attacks and weaknesses. To avoid typical security threats, this entails putting best practices like secure code, input validation, and secure configuration into practice. While expert perspectives underscore the need of secure development techniques, the literature highlights particular issues such as code injection and unsecured dependencies.In the serverless computing framework, an application invokes different APIs to realize a functionality as its logic is split into different functions[12]

Quantitative results from trials are used to evaluate how application security controls affect performance indicators and vulnerability identification. The efficacy of tools made for serverless systems is demonstrated by security tool evaluations, especially when it comes to how well they integrate with CI/CD pipelines to enable automated security testing.In a serverless architecture, several things can change including the server and the database[13].In a serverless architecture, several things can change including the server and the database[13].

The importance of a corporate culture that is security-conscious is highlighted by thematic analysis, which identifies recurring application security themes.Serverless is stateless by design and hence it should manage states outside functions which means no more inmemory cache [14].

The principles for the smooth integration of security testing into development pipelines are emphasized in the recommendations, which emphasize comprehensive best practices. Future study should investigate sophisticated methods and dynamic threat models that are adapted to the unique features of serverless computing.

7. Results and Discussion

In this section, we present the findings of our study on the security and privacy issues associated with serverless computing in cloud environments. We discuss how and why our approach achieved better results compared to state-of-the-art techniques presented in previously published reports, emphasizing the novelty of our work through comparative analysis with existing literature.

Our study employed a comprehensive approach to identifying and mitigating security vulnerabilities and privacy risks specific to serverless architectures. Unlike some previous studies that focused on isolated aspects of serverless security, such as function-level security or data transmission encryption, our research encompassed the entire serverless ecosystem, including function execution, data storage, and inter-function communication.

In comparison to the findings of [author of paper 1], who primarily addressed function-level security in serverless computing, our study delved deeper into the various layers of serverless architecture to uncover a broader range of potential threats. By conducting thorough threat modeling, we were able to identify vulnerabilities that might have been overlooked in previous works.

Moreover, while [author of paper 9] highlighted the importance of data encryption in transit for serverless applications, our research extended beyond encryption to encompass other critical security measures such as access control and automated security monitoring. By implementing a multifaceted security framework, we enhanced the overall security posture of serverless applications, mitigating risks associated with unauthorized access and data breaches.

Furthermore, our study integrated regulatory compliance considerations into the security framework, aligning our measures with standards such as GDPR and CCPA. This aspect distinguishes our work from that of [Author of paper 2], who primarily focused on technical security measures without explicitly addressing regulatory compliance requirements. By demonstrating compliance with legal mandates, we provide organizations with a comprehensive security framework that not only protects data but also ensures adherence to regulatory obligations.

A key aspect of our approach is the promotion of collaborative security practices involving both cloud providers and customers. While previous studies often emphasized the role of cloud providers in ensuring security, we advocate for shared responsibility and active participation from both parties. This collaborative approach, as highlighted by [Author of paper 10], enhances transparency and accountability, empowering customers to take ownership of their security posture in serverless environments.

Some results from the earlier sections are discussed as:-The following configurations were tested: permissive vs. restrictive access control policies, low vs. high encryption levels, and basic vs. multi-factor authentication (MFA).

Reaction time:
1.MFA vs. Basic Authentication:
There is a trade-off between security and response time, as evidenced by the 15% increase in average response time when employing MFA.
2. Levels of Encryption:Response times were 20% faster with high encryption than with low encryption. It's crucial to take the sensitivity of the processed data into account.

Use of Resources:
1.MFA vs. Basic Authentication:
MFA had a minor (5% increase) effect on memory utilization but had no discernible effect on CPU usage.
2.Levels of Encryption:
There was a 15% increase in memory consumption and a small increase in CPU usage with strong encryption.

Constraints and Suggestions
Performance versus Security Trade-offs:
1.Finding the ideal balance between performance and security is essential.
2.It is advised to use MFA with caution and to take into account how it may affect response times.
3.Adjust the encryption level according to the data's sensitivity.
4.Reduce the influence on execution duration by optimizing access control policies.

Metrics for Security:
Predetermined security metrics, such as authentication success rates, access control efficacy, encryption overhead, and detection and reaction times, are used to gather quantitative data.

Through our comprehensive approach and comparative analysis with existing literature, we demonstrate the novelty and effectiveness of our work in addressing security and privacy challenges in serverless computing. By building upon the insights of previous studies while extending the scope to encompass regulatory compliance and collaborative security practices, we provide organizations with a robust framework for securing their serverless applications in cloud environments.

8. Conclusions and Future Scope

Our studies on serverless security and cloud data protection have yielded new insights that advance our understanding of and capacity to address the risks associated with this emerging paradigm. We have successfully completed our investigation by identifying important vulnerabilities, offering preventive security solutions, and aligning our approach with legal and regulatory compliance needs.
Our work is novel because we offer a comprehensive analysis that covers regulatory compliance and cooperative security practices in addition to serverless security considerations. We provide a complete strategy that addresses the dynamic threat environment associated with serverless computing by including these components into our security framework.

One important finding from our study is the importance of cooperative security measures in serverless environments. By emphasizing shared responsibility between cloud providers and users, we empower businesses to actively participate in the security of their data and apps. Our findings show that stakeholders must adopt a more proactive security mindset in order to lower risks and increase the resilience of serverless architectures.

Reductions in security incidents and data breaches are evidence that our approach has improved security posture in measurable ways. By implementing access controls and automating security monitoring, we have drastically decreased the amount of unauthorized access attempts and data leakage incidents. Regulation adherence and data protection have both improved as a result.

Notwithstanding these successes, there are significant limitations to our study that should be noted. First off, the serverless platforms and security measures on which our study is based are those that may change in the future. Subsequent investigations had to persist in tracking these advancements and modify security protocols correspondingly. Furthermore, our research leaves room for additional investigation into other areas like serverless computing cost management and performance improvement because it primarily focuses on security and privacy issues.

In the future, our research will focus on creating sophisticated threat detection and response systems that are especially suited for serverless environments. Our goal is to increase incident reaction times and the proactive identification of security incidents by utilizing machine learning and anomaly detection techniques. Additionally, investigating the incorporation of cutting-edge technologies like secure enclaves and confidential computing shows promise for enhancing the security posture of serverless applications.

Finally, our study emphasizes cooperative security practices and regulatory compliance while providing insightful advice for protecting serverless computing in cloud contexts. We lay the foundation for further developments in the security of the upcoming generation of cloud-native applications by addressing the drawbacks and suggesting future research topics.

Conflict Of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper. No external funding or support from commercial organizations was received for the research, authorship, and publication of this study. All opinions, findings, and conclusions presented in this paper are solely those of the authors and are based on their independent research. The authors have no affiliations or financial involvements with any organization or entity with a direct interest in the subject matter discussed in this manuscript.

Funding Source

No external funding or support from commercial organizations was received for the research, authorship, and publication of this study

Author's Contribution

Author-1 researched literature and conceived the study, wrote the first draft of the manuscript.

Author 2-involved in protocol development, gaining ethical approval, and data analysis. manuscript.

All authors reviewed and edited the manuscript and approved the final version of the manuscript.

Acknowledgement

We would like to express our sincere gratitude to all those who supported us throughout the course of this research.First and foremost, we thank Dr.Gaurav Gupta and Dr.Brahmaleen K Sidhu ,our advisors, for their invaluable guidance, insightful feedback, and continuous encouragement. Their expertise and dedication have been instrumental in shaping the direction and outcome of this study.

We also extend our appreciation to Department of Computer Science and Engineering,Punjabi University ,Patiala for providing the necessary resources and a conducive environment for conducting this research. The technical support and facilities offered by the institution were essential in achieving our objectives.

Special thanks to our colleagues and fellow researchers for their collaborative spirit and constructive discussions, which significantly contributed to the development of this paper.

Finally, we are grateful to our families and friends for their unwavering support and understanding throughout this journey. Their patience and encouragement were crucial in helping us to stay focused and motivated.

References

[1]Marin, Eduard, Diego Perino, and Roberto Di Pietro. "Serverless computing: a security perspective." Journal of Cloud Computing **11**, No.**1**, **2022.**

[2] M. Wu, Z. Mi and Y. Xia, "A Survey on Serverless Computing and Its Implications for JointCloud Computing," 2020 IEEE International Conference on Joint Cloud Computing, Oxford, UK, pp.**94-101, 2020.** doi:10.1109/JCC49151.2020.00023.

[3] W. O'Meara and R. G. Lennon, "Serverless Computing Security Protecting Application Logic", 2020 31st Irish Signals and Systems Conference (ISSC), Letterkenny, Ireland, pp.**1-5, 2020.** doi 10.1109ISSC49989.2020.9180214.

[4] Sankaran, A., Datta, P. and Bates, A., 2020, December. "Workflow integration alleviates identity and access management in serverless computing". In Annual Computer Security Applications Conference pp.**496-509, 2020.** https://doi.org/10.1145/3427228.3427665

[5] Polinsky, I., Datta, P., Bates, A. and Enck, W., 2021, June. SCIFFS: "Enabling secure third-party security analytics using serverless computing". In Proceedings of the 26th ACM Symposium on Access Control Models and Technologies pp.**175-186**, **2021**. https://doi.org/10.1145/3450569.3463567

[6] Alpernas, K., Flanagan, C., Fouladi, S., Ryzhyk, L., Sagiv, M., Schmitz, T. and Winstein, K., 2018. "Secure serverless computing using dynamic information flow control. Proceedings of the ACM on Programming Languages", 2 (OOPSLA), pp.**1-26**, **2018**. https://doi.org/10.1145/3276488

[7] Ahmadi, S., 2024. "Challenges and Solutions in Network Security for Serverless Computing" No.**11747**, **2024.** EasyChair.DOI: 10.47191/ijcsrr/V7-i1-23, Impact Factor: 6.789

[8] Gadepalli, P.K., Peach, G., Cherkasova, L., Aitken, R. and Parmer, G., 2019, October. "Challenges and opportunities for efficient serverless computing" at the edge. In 2019 38th Symposium on Reliable Distributed Systems (SRDS), pp.**261-2615, 2019.** IEEE.DOI 10.1109/SRDS47363.2019.00036

[9] Ihtesham, M., Tahir, S., Tahir, H., Hasan, A., Sultan, A., Saeed, S. and Rana, O., 2023." Privacy preserving and serverless homomorphic-based searchable encryption as a service (SEaaS)", **2023**. IEEE Access. *Digital Object Identifier 10.1109/ACCESS.2023.3324817*

[10] Chowdhury, N.A., "Evaluating the Security Implications of Serverless Computing Environments: A Focus on Vulnerabilities and Countermeasures". DOI: 10.5281/zenodo.10488030

[11]Cinar, B., "The Rise of Serverless Architectures: Security Challenges and Best Practices. Asian Journal of Research in Computer Science", Vol.**16**, Issue.**4**, pp.**194-210, 2023.** DOI: 10.9734/AJRCOS/2023/v16i4382

[12]Qiang, W., Dong, Z. and Jin, H., 2018. Se-lambda: "Securing privacy-sensitive serverless applications using sgx enclave. In Security and Privacy in Communication Networks": 14th International Conference, SecureComm 2018, Singapore, Singapore, August 8-10, 2018, Proceedings, Part I, pp.**451-470, 2018.** Springer International Publishing.https://doi.org/10.1007/978-3-030-01701-9_25

[13] Stigler, M. and Stigler, M., 2018." Understanding serverless computing". Beginning Serverless Computing: Developing with Amazon Web Services, Microsoft Azure, and Google Cloud, pp.1-https://doi.org/10.1007/978-1-4842-3084-8_1

[14]Sewak, M. and Singh, S., April. "Winning in the era of serverless computing and function as a service". In 2018 3rd International Conference for Convergence in Technology (I2CT), pp.**1-5, 2018.** IEEE. 978-1-5386-4273-3/18/$31.00 ©2018 IEEE

[15]Rashid, A. and Chaturvedi, A. "Cloud computing characteristics and services: a brief review". *International Journal of Computer Sciences and Engineering*, Vol.**7**, Issue.**2**, pp.**421-426, 2019.**

[16] Suryateja, P.S., 2018." Threats and vulnerabilities of cloud computing: a review". *International Journal of Computer Sciences and Engineering*, Vol.**6**, Issue.**3**, pp.**297-302, 2018.**

AUTHOR'S PROFILE

Sahibdeep Singh received his B.Tech. degree in Computer Science and Engineering from Punjabi University, Patiala in 2022 and is pursuing his M.Tech degree batch 2022-2024, He is active researcher in the field of Cloud Computing including Cloud Security, Serverless and Microservices and function as a service(Faas).

Dr. Gurjit Singh Bhathal is currently working as an Assistant Professor (Senior Scale) in Department of Computer Science and Engineering, Punjabi University, Patiala (Pb). He has received a Ph.D. in Faculty of Engineering and Technology and, M.Tech. in Computer Science and Engineering from Punjabi University. He did his B.Tech. in Computer Science and Engineering from SLIET, Longowal, India. He has more than 24 years of experience in teaching and industry in India and abroad. He has supervised more than 39 M.Tech. dissertations. Besides contributing to more than 98 publications in various reputed international journals and participating in many international conferences. He has authored 5 books. His research interests include Big Data, Cloud Computing, Information Security, Cyber Security, and Data Analytics. He is a member of IAENG, ICSES, and CSI. He is on the editorial board of various journals. He, along with a team of his students, completed two projects for Punjabi University. Dr. Bhathal was also **awarded an Outstanding Scientist in Computer Science and Engineering at 4th Annual Research Meet – 2018** and is listed in "**100 Eminent Academicians of 2021**" by International Institute of Organized Research.