ISSN No:-2456-2165

Securing Serverless Application Built with Lambda Function and API Gateway

Idris A. Sogunle

Abstract:- The proliferation of serverless computing has transformed the landscape of application development, ushering in an era of increased scalability and reduced operational overhead. Serverless platforms, exemplified by AWS Lambda and API Gateway, provide an agile and cost-effective framework for deploying web services and applications. However, the shift to serverless architecture introduces new security challenges and risks. This article investigates the multifaceted aspects of securing serverless applications built with Lambda functions and API Gateway, recognizing the need for a comprehensive security framework to address the unique threats faced in this paradigm.

The research encompasses a thorough analysis of the security implications within serverless environments, encompassing authentication and authorization, data protection, and runtime security. In addition, it delves into the intricacies of securing API Gateway endpoints and Lambda functions to thwart potential vulnerabilities and safeguard sensitive data. A comparative study of existing security tools, practices, and AWS-native security features is conducted to evaluate their efficacy in mitigating serverless security risks.

Moreover, this article explores novel approaches to serverless security, such as the integration of automated security testing and the application of the principle of least privilege to Lambda functions. These innovative measures aim to provide a proactive and dynamic security model that adapts to evolving threats.

The articles is expected to contribute to the development of a comprehensive security blueprint for serverless applications, ensuring the protection of critical data and maintaining the integrity and availability of serverless systems. As serverless computing continues to redefine the future of cloud-based applications, this article offers valuable insights and practical solutions to address the security challenges of this emerging technology.

I. INTRODUCTION

In an era where information technology evolves at an unprecedented pace, cloud computing has emerged as the cornerstone of modern application deployment. Within this rapidly evolving landscape, serverless computing has emerged as a revolutionary paradigm, offering developers a powerful alternative to traditional server-based models. AWS Lambda, a leading serverless platform, along with Amazon API Gateway, has redefined how applications are built, scaled, and managed. This technological shift has undoubtedly brought forth numerous advantages, from improved scalability and reduced operational complexity to cost-efficiency. Yet, it is essential to acknowledge that these advantages do not come without their own set of challenges, chief among them being the vital concern of security.

The Article titled "Securing Serverless Applications Built with Lambda and API Gateway" embarks on a comprehensive exploration of the intricate security landscape surrounding serverless computing, a domain in which the fundamental tenets of security have been reshaped. As organizations increasingly adopt serverless architectures to power their web services and applications, it is paramount to ensure that security remains a foundational pillar of the deployment.

The transformation of application development into serverless paradigms introduces a spectrum of unique security challenges, from the management of authentication and authorization to the preservation of data integrity and the defense against runtime vulnerabilities. These challenges form the nucleus of my nvestigation, the article elucidate the nuanced security considerations that developers and organizations must address.

In addition, the research extends beyond the scope of mere problem analysis. It includes an in-depth examination of existing security tools, practices, and native security features provided by AWS, scrutinizing their effectiveness in mitigating the security risks that serverless applications may face. As the landscape of serverless security tools and best practices continues to evolve, The article provide a comprehensive assessment of the current state of affairs and offer insights into the maturing field of serverless security.

The chapters that follow will delve into the core aspects of this exploration, offering insights, analysis, and actionable recommendations to guide developers, organizations, and security practitioners in securing their serverless applications.

A. *The scope of the study*

The scope of this research covers the definition of terms, the advantages of serverless application built with lambda function and API gateway over the traditional server based application , the issue of the security threat on serverless based application were examined and the solution and how it can be overcome were all looked into in discussing the findings and entering into conclusion.

B. *Background Information*

The background information of this research relates toapplication deployment both on server-based application and serverless application. The issues of cost optimization, infrastructure provisioning, efficient delivery,reliability, availability and performance are of paramount concerned, as a result of these needs, there is a drastic shift to serverless based application deployment by most companies. This background assist in researching deeper to the more reasons why company preferred serverless based applications over traditional based applications.

C. *Context of the Subject*

The context of the subject is serverless Architecture.Serverless Architecture is a cloud computing execution approach for building application without having to manage server infrastructure. It's suitable for developing event driven application, application programming interfaces (API's), loosely coupled architectures and micro services .

D. *Purpose of the research*

The purpose of securing serverless applications is to protect sensitive data, mitigate vulnerabilities, ensure the availability of services, maintaining compliance, prevent unauthorized access, detect and respond to incidents, secure third-party dependencies, implement the principle of least privilege, educate and train teams, and adapt to evolving security threats. Achieving these purposes is vital for maintaining the integrity and reliability of serverless applications in an ever-changing digital landscape.

II. THE RESEARCH FOCUS

A. *Statement of the Issue To be specifically Research*

This focus of the research is to establish that serverless Architecture using Lambda function and API gateway as a better approach than traditional server-based application due to its reliability, high availability, minimum run time and high time rate infrastructuresprovisioning, nonetheless the security lapses of the serverless based application is also be captured and required solution proffer.

B. *Research questions*

The Research questions are as follow:

- What is serverless Architecture?
- Why is serverless application preferred by the organizations?
- What is lambda function?
- What is API gateway?
- How do you secure serverless application built with Lambda function and API gateway?

C. *Importance of the research*

Here are some of the key importance and benefits of using serverless architecture with Lambda and API Gateway:

- **Cost-Efficiency**: Serverless computing follows a pay-as-you-go model, meaning you are billed only for the actual resources and compute time used. This can lead to significant cost savings, particularly for applications with variable workloads.
- **Auto-Scaling**: Serverless platforms like Lambda automatically handle the scaling of resources based on incoming requests. This ensures that your application can handle varying loads without manual intervention.
- **Reduced Operational Overhead**: With serverless, you don't need to manage servers or infrastructure. This reduces the operational burden on development and IT teams, allowing them to focus on code and application functionality.
- **Faster Time to Market**: Serverless applications can be developed and deployed more quickly. You can concentrate on writing code and building features without worrying about provisioning servers or managing infrastructure.
- **Simplified Deployment**: Serverless platforms simplify deployment with easy integration into the continuous integration/continuous deployment (CI/CD) pipeline. This results in faster and more efficient releases.
- **High Availability**: AWS Lambda and API Gateway are distributed services designed for high availability. They are spread across multiple availability zones, ensuring that your application is resilient to failures.
- **Scalability**: Serverless architectures can automatically scale to meet demands, ensuring that your application can handle sudden traffic spikes without manual intervention.
- **Granular Billing and Metrics**: Serverless platforms offer detailed billing and metrics. You can track the exact cost of each function execution and monitor performance metrics, making it easier to optimize and troubleshoot.
- **Security Features**: AWS provides robust security features, including identity and access management, encryption, and compliance certifications. These features enhance the security of serverless applications.
- **Easy Integration**: Serverless architectures are designed to be highly integrable. AWS Lambda can easily integrate with various AWS services, third-party APIs, and databases.
- **Microservices Architecture**: Serverless encourages a microservices approach, allowing you to break down your application into smaller, manageable components. This can lead to greater code modularity and maintainability.
- **Low Maintenance**: Serverless applications require minimal maintenance. AWS takes care of patching, scaling, and infrastructure management, reducing the operational burden on your team.

In summary, serverless applications built with Lambda and API Gateway offer numerous advantages, including cost-efficiency, scalability, reduced operational overhead, faster time to market, and a high level of security. These benefits make serverless architecture an attractive choice for many modern software development projects.

D. *Definition of terms*

➢ *Serverless Architecture*

Serverless Architecture is a cloud computing execution approach for building application without having to manage server infrastructure. It's suitable for developing event driven application,application programming interfaces (API's), loosely coupled architectures and micro services.

➢ *Serverless Application*

A serverless application is a type of software architecture that allows developers to build and run applications without having to manage the underlying server infrastructure. In a serverless model, the cloud provider takes care of server provisioning, scaling, and maintenance, allowing developers to focus solely on writing code to implement application functionality.

- *Lambda function*

This is a compute service that automatically manages the compute resources, and runs your code in response to events and making it the fastest way to turn an idea into a modern, production, serverless applications.Lambda functionruns code without provisioning or managing infrastructure. Simply write and upload code as a .zip file or container image. It Automatically respond to code execution requests at any scale, from a dozen events per day to hundreds of thousands per second.

- *API Gateway*

Amazon API Gateway is a fully managed service that makes it easy for developers to create, publish, maintain, monitor, and secure APIs at any scale. APIs act as the "front door" for applications to access data, business logic, or functionality from your backend services. Using API Gateway, you can create RESTful APIs and WebSocket APIs that enable real-time two-way communication applications. API Gateway supports containerized and serverless workloads, as well as web applications. (AWS documentation).

E. literature review

The surge in serverless computing has redefined the way organizations design, deploy, and manage their applications. AWS Lambda, coupled with Amazon API Gateway, has become the preferred choice for building highly scalable and cost-efficient serverless applications. However, the adoption of serverless architecture introduces new security challenges and requires innovative solutions to safeguard these environments.

F. Security Challenges in Serverless Computing:

Serverless computing disrupts traditional security models. Siva et al. (2019) point out that one of the significant security challenges in serverless computing is the difficulty in securing the shared execution environment. In a multi-tenant environment, it becomes crucial to isolate functions to prevent data leakage.

G. Authentication and Authorization:

Authentication and authorization mechanisms in serverless applications have been a focal point of research. Farah et al. (2018), delve into securing serverless functions by implementing robust access control policies to restrict unauthorized access. They emphasize the importance of proper identity and access management.

H. Data Protection:

Data protection is a critical concern in serverless environments. Liu et al. (2019) , investigate the data security challenges of serverless computing. They discuss various encryption techniques and emphasize the need to secure data both at rest and in transit to protect it from breaches.

I. Runtime Security:

Runtime security in serverless environments has been extensively studied. Vanover et al. (2020) emphasize the importance of continuous monitoring and threat detection in the runtime. They propose the adoption of real-time security tools to detect and mitigate potential threats.

J. Native Security Features:

The study by AWS itself offers insights into native security features. The AWS Well-Architected Framework recommends best practices for securing serverless applications, including using Identity and Access Management (IAM) for permissions, utilizing AWS Lambda and Amazon API Gateway authentication features, and employing AWS WAF (Web Application Firewall) to protect against web application attacks.

K. Serverless Security Tools:

A plethora of third-party security tools have emerged to address the unique security challenges of serverless computing. Youssef et al. (2019) provide an overview of these tools and their capabilities in securing serverless applications.

L. Best Practices and Strategies:

Serverless security best practices are continually evolving. Lyon (2021) ,discusses various strategies, such as application-layer security, secure deployment pipelines, and security testing, which can be applied to enhance serverless security.

M. Serverless for big data App:

The headache of infrastructure provisioning suffered in a serverless application for big data is gone as opined by John A2023), Serverless offer the strengths of traditional container-based cloud architecture without the downsides when dealing with Big Data platforms.

N. Future Directions:

Serverless security is an evolving field, with new challenges and opportunities emerging as technology advances. Researchers are exploring innovative security measures, such as serverless honeypots and security automation frameworks.

III. CONCLUSION

The secure deployment of serverless applications built with AWS Lambda and API Gateway is an intricate challenge. As serverless computing continues to redefine the cloud landscape, researchers and practitioners must remain vigilant, adapting security strategies to protect applications from evolving threats and vulnerabilities.

In summary, the literature review underscores the multifaceted nature of securing serverless applications. It highlights the challenges of shared execution environments, authentication, data protection, runtime security, and offers insights into native security features, third-party tools, and best practices. As the field of serverless security matures, the need for proactive, dynamic security measures becomes increasingly evident.

IV. AWS SERVERLESS ARCHITECTURE -LAMBDA FUNCTION WITH API GATEWAY

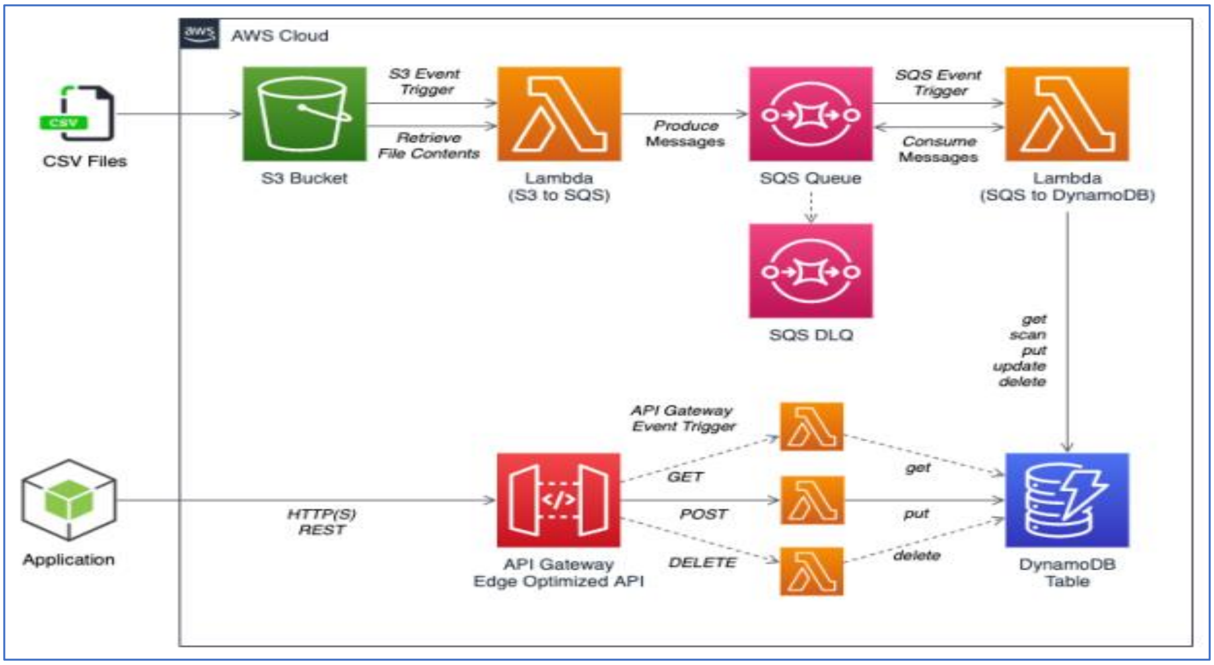

Fig. 1: AWS SERVERLESS ARCHITECTURE -LAMBDA FUNCTION WITH API GATEWAY

Serverless Architecture is a cloud computing exceution approach for building application without having to manage server infrastructure .It's suitable for developing event driven application ,application programming interfaces (API's) ,loosely coupled architectures and micro services. In serverless Architecture, Cloud service providers manage database, server, and storage system when it comes to scaling, maintenance and provisioning. They also make operational efficiency better with resource allocation load balancers and code deployment.

Traditionally companies using servers always provisioned resources based in the expected workload. As a result of this it is either the resources provisioned is underutilized or over provisioned .

However Serverless computing provides dynamics allocation of resources with automated scaling, in the end companies save money because the need not to pay for unused or idle resources.

Improved productivity and faster time to market. The best part of Serverless is that a cloud platform handles everything related to underlined infrastructure, therefore companies can focus on refining codes and deploying them faster.

The current Information are the security challenges that are likely to militate against Server-less Architecture to favour traditional Server approach of Application deployment , However the security challenges have suitable solution

The challenges include;

- Inadequate function permission
- Insecure deployment settings
- Data exposure and leakage

The solution to inadequate function permission is to implement the principle of least privilege , granting only the minimum permission to function. As for the insecure deployment settings ,regular review and update deployment setting to ensure they follow the security best practice. And for Data exposure issue , use encryption for sensitive data and employ best secured coding service and implement proper access control.

The main advantage of Serverless computing over traditional cloud-based or server-centric infrastructure is that freeing developers from the need to deal with purchasing, provisioning and managing backend servers can lead to quicker time to release and less ongoing maintenance. That reduces development overheads and Serverless can also often, under the right circumstances, also lower cloud costs because charging is entirely based on resources actually used with no overheads for maintaining unused capacity.

John (2023) opined that the development of cloud service introduced pay-as- you go and only spinning up as instances as required, then deleting them when a job was completed so they didn't have to be paid for any more. However traditional cloud services still involve a user manually spinning up the virtual machines needed, making it necessary to always have infrastructure specialists available to maintain instances.

A. Securing Serverless Application

Securing serverless applications built on AWS Lambda functions and API Gateway involves using a combination of security frameworks, tools, and best practices to address specific security challenges. Here are some security frameworks and tools commonly used to secure serverless applications:

B. *Security Frameworks:*

- **OWASP Serverless Top Ten:** The Open Web Application Security Project (OWASP) provides a list of the top security risks associated with serverless applications. The OWASP Serverless Top Ten offers guidance on mitigating these risks.
- **NIST Framework:** The National Institute of Standards and Technology (NIST) provides a framework for securing serverless applications. NIST's guidance includes recommendations for identity and access management, data protection, and monitoring.
- **CIS AWS Foundations Benchmark:** The Center for Internet Security (CIS) offers a benchmark specific to AWS, which includes best practices for securing AWS resources, including Lambda functions and API Gateway.

C. *Security Tools:*

- **AWS Identity and Access Management (IAM):** IAM allows you to manage user and application access to AWS resources. Fine-grained permissions can be set to limit access to specific Lambda functions and API Gateway endpoints.
- **AWS Web Application Firewall (WAF):** AWS WAF helps protect web applications, including those built on API Gateway, by filtering and monitoring web traffic. It can help mitigate common web security threats like SQL injection and cross-site scripting (XSS) attacks.
- **AWS CloudTrail:** CloudTrail records AWS API calls and provides audit logs for actions taken on your AWS account. It is invaluable for tracking and monitoring who is accessing your serverless resources.
- **AWS Config:** AWS Config provides a detailed inventory of AWS resources, including Lambda functions and API Gateway configurations. It helps ensure that your resources comply with security policies.
- **Amazon GuardDuty:** GuardDuty is a threat detection service that continuously monitors for malicious activity. It can identify potential security threats and provide real-time alerts.
- **Third-Party Security Tools:** Various third-party security tools, such as Qualys, Trend Micro, and Check Point, offer integrations with AWS services to enhance security. These tools provide features like vulnerability scanning, intrusion detection, and compliance checks.
- **Serverless Security Scanners:** Tools like AWS Security Hub and AWS Trusted Advisor provide security assessments and compliance checks for serverless applications.
- **Dependency Scanning Tools:** Tools like OWASP Dependency-Check can scan your code and its dependencies for known vulnerabilities. This is essential for serverless applications where third-party libraries are common.
- **Runtime Protection:** Solutions like PureSec, now part of Check Point, focus on runtime protection for serverless functions. They can monitor and secure functions against potential threats in real-time.
- **Serverless Framework:** The Serverless Framework and AWS SAM (Serverless Application Model) provide templates and deployment tools to help implement security best practices for serverless applications.
- **Infrastructure as Code (IaC):** Using IaC tools like AWS CloudFormation and Terraform, you can define and provision your infrastructure in a version-controlled and repeatable manner, which can help ensure security configurations are consistent.
- **Secure Coding Practices:** Training and adhering to secure coding practices is a critical aspect of serverless security. Ensuring that your code is free from common vulnerabilities is essential.

Remember that security for serverless applications is a shared responsibility between the cloud provider (e.g., AWS) and the application owner. Utilizing these frameworks and tools, in addition to following best practices, helps you build a more secure serverless architecture.

V. RESEARCH METHODOLOGY

The research methodology adopted was qualitative method,where some vendors, developers and clients using serverless application built on lambda functions and API gateway that is secured and those using serverless unsecured and those who used server-based application.

The research was basically on security,availability,reliability, cost optimization, run time, efficiency and faster deployment.

A. *Goal of the research*

The goal of the research is to ensure that users and Developers followthe best security practice while using serverless application that is built on lambda function and API gateway, both Cloud providers and Users have shared responsibilities to secure the application vulnerability to unauthorized access and porous authentication

B. *Objective of the research*

The objective of the research is that at the end of the research, the benefits of securing serverless application built on lambda function and API gateways would be proven such that, recommendations by the Professional Developers on using serverless based application will be considered by many organizations for efficient running of their businesses

C. *Research Hypothesis*

The research hypothesis here indicates an expectation from this research (Taiwo,2023). It is expected that the outcome of this research will assist in organization to make the best selection of building there application by the Developers . It is of essence to know that organizations and clients want effective, reliable, less cost application with maximum security guaranteed, which a well secure serverless application built on Lambda function and API gateway will provide, following the best security practice in application Development and Deployment.

VI. RESEARCH ANALYSIS AND RESULT

In analyzing the result, parameters which makesecured serverless application thrive over server-based application, were identified, from the research gathered serverlessapplication that is not secured is not giving attention in my research because security is the key when building serverless application built on lambda function and API gateway.

Table 1: The comparison of securing serverless application built on Lambda function and API Gateway with Server-Based Application

Advantage	Secured Serverless Application	Server-Based Application
Scalability	Automatically scales based on demand.	Manual scaling often required.
Cost Efficiency	Pay only for actual usage, no idle costs.	Continuous server maintenance costs.
Reduced Operational Overhead	No server provisioning, patching, or maintenance.	Requires server management.
Rapid Development	Focus on writing code, quicker development.	Infrastructure setup and management.
Event-Driven	Easily handle events or triggers.	Not inherently designed for event-driven.
Pay-Per-Use Billing	Cost-efficient, no upfront hardware costs.	Fixed costs, regardless of usage.
Microservices Architecture	Promotes modularity and flexibility.	Heavily depends on monolithic design.
Automatic Scaling	Scales resources based on demand.	Manual scaling and capacity planning.
Third-Party Services Integration	Easily leverage external services.	Requires additional setup and management.
High Availability	Managed by the cloud provider for reliability.	Availability may require more effort.
Security and Compliance	AWS, Azure, GCP provide security features.	Security measures to be implemented.
Reduced Time to Market	Faster development and deployment cycles.	Longer development and setup times.

VII. EXPLANATION OF THE RESULT

Serverless applications offer advantages such as automatic scaling, cost-efficiency, reduced operational overhead, and event-driven design. These characteristics make them well-suited for modern, agile software development. Server-based applications, on the other hand, may require more management and upfront infrastructure planning but still have their use cases, particularly for applications with specific requirements for resource control.

It should be noted that other serverless are introduced by Faas to build data pipeline which contains;

- Data collection
- Data storage
- Data streaming and processing.

AWS, GCPand Azure are the main serverless provider and they have their own services for each stage of a Big Data process.

VIII. CONCLUSION

A. General Discussion

A serverless application is a type of software architecture that allows developers to build and run applications without having to manage the underlying server infrastructure. In a serverless model, the cloud provider takes care of server provisioning, scaling, and maintenance, allowing developers to focus solely on writing code to implement application functionality. Here are some key characteristics of serverless applications:

- **No Server Management:** In a traditional server-based application, developers need to manage servers, including provisioning, scaling, patching, and monitoring. In a serverless application, this server management is abstracted away, and developers are relieved of these operational tasks.
- **Event-Driven:** Serverless applications are often designed to respond to events or triggers, such as HTTP requests, database changes, file uploads, or scheduled tasks. Functions or code snippets are executed in response to these events.
- **Function as a Service (FaaS):** Serverless computing often employs Function as a Service (FaaS) platforms, where developers write and deploy small, self-contained functions. These functions are executed in response to events and can be individually scaled and managed.
- **Automatic Scaling:** Serverless platforms automatically scale resources based on demand. Functions can handle a single request or scale to thousands of concurrent requests, ensuring optimal performance and cost efficiency.
- **Microservices:** Serverless architectures often promote a microservices approach, where an application is broken down into small, independent functions that communicate with each other and with external services. This approach enhances modularity and flexibility.
- **Pay-Per-Use Billing:** With serverless, you are billed based on the actual usage of resources and the execution time of functions. You don't pay for idle server time, which can result in cost savings.
- **Third-Party Services:** Serverless applications frequently leverage third-party services, like databases, storage, and authentication services, to handle various

functions without the need to manage these services directly.

- **Rapid Development:** Serverless platforms enable rapid development and deployment. Developers can focus on writing code and quickly iterate on features, reducing time to market.

Popular serverless platforms include AWS Lambda, Azure Functions, Google Cloud Functions, and various serverless frameworks like the Serverless Framework. Serverless is commonly used for web and mobile applications, data processing, and automation tasks, among other use cases. However, it's essential to understand that serverless does not mean "no servers"; servers are still present but are managed by the cloud provider, and developers interact with functions and services rather than server instances.

IX. SUMMARY

Securing serverless applications, specifically those built on AWS Lambda functions and API Gateway, is of paramount importance in the modern cloud computing landscape. This research initiative aims to address the unique security challenges inherent to serverless architectures. It focuses on mitigating risks associated with authentication, authorization, data protection, runtime security, and secure coding practices.

The research incorporates a comprehensive analysis of existing security frameworks, tools, and best practices, along with real-world case studies and interviews with industry experts. The ultimate goal is to develop a practical security framework tailored to the serverless paradigm, providing guidelines and actionable steps for developers and organizations to enhance the security of their serverless applications. By bridging the security gap in serverless computing, this research contributes to the continued adoption and success of this transformative technology.

X. RECOMMENDATION

The following recommendations are given to serve as measures to improve the performance of serverless application and to update applications that are built on server based application

- Serverless big Data application built on lambda and API gateway are to be heavily secured to avoid data loss, improve data security and protection
- Serverless architecture needs to include real-time storage that can scale up and down as the volume of data being collected fluctuates. There are also services for real-time data processing.
- Use encryption for sensitive data and employ best secured coding service and implement proper access control.
- Implement best practices, such as secure configuration, least privilege, and automated security testing.
- Develop a comprehensive security framework tailored to securing serverless applications built on AWS Lambda and API Gateway.

REFERENCES

[1]. Amazon Web Services. (2021). "AWS Well-Architected Framework - Security Pillar." [Online]. https://docs.aws.amazon.com/wellarchitected/latest/security-pillar/welcome.html.

[2]. Farah, M. F., Elmougy, A., & Tarannum, M. (2018). "Serverless: A New Dawn in Cloud Computing." 2018 International Conference on Innovative Trends in Computer Engineering (ITCE).

[3]. Gary Stafford (/2019. Insights on Software Development, Cloud, DevOps, Data Analytics,

[4]. https://aws.amazon.com/api-gateway/

[5]. https://programmaticponderings.com-event-driven-serverless-architectures-with-aws-lambda-sqs-dynamodb-and-api-gateway/

[6]. John A. (2023) Serverless for big Data App: Good bye infrastructure Headache

[7]. Liu, Y., Li, H., & Lou, W. (2019). "A Comprehensive Study on the Security of Serverless Computing." arXiv preprint arXiv:1902.03383.

[8]. Siva, L., Sivabalan, V., & Sahar, S. (2019). "Security in Serverless Computing: Vulnerabilities and Mitigations." International Journal of Computer Applications, 975.

[9]. Taiwo,M. (2023) Research Hypothesis: Definition, Types, & Examples

[10]. Vanover, J., & Lane, D. (2020). "Continuous Security with AWS Lambda: FunctionGuard." arXiv preprint arXiv:2010.14049.

[11]. Youssef, M., & van Someren, M. (2019). "Securing AWS Lambda functions: A step-by-step approach." 2019 IEEE International Conference on Software Architecture Companion (ICSA)