A view on serverless computing and cybersecurity:Approach to prevent social engineering attacks

S. Mathangi,
Department of Computer and Communication Engineering, Panimalar Engineering College.

N. Lakshmi Priya,
Department of Computer and Communication Engineering, Panimalar Engineering College

G. Pavithra,
Department of Computer and Communication Engineering, Panimalar Engineering College.

V.Rubashree,
Department of Computer and Communication Engineering, Panimalar Engineering College.

C.Keerthika,
Department of Computer and Communication Engineering, Panimalar Engineering College.

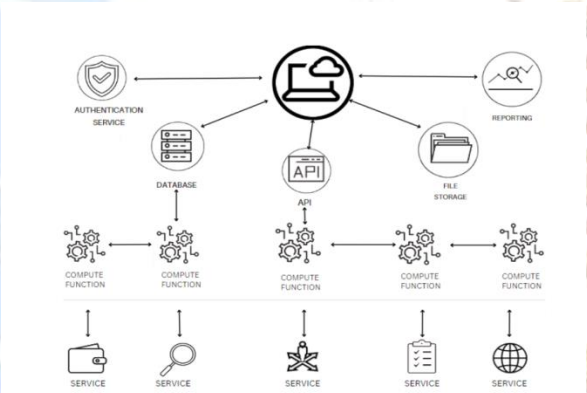
***Abstract*—The popularity of serverless computing is rising as a result of its portability and ease of administration. Serverless specifically enables users to concentrate entirely on the function in question while delegating additional time-consuming management and scheduling concerns to the platform provider, who is in charge of finding a balance between high-performance scheduling and cheap resource cost. Serverless computing enables organizations to avail of the inherent and unlimited flexibility and scalability that serverless provides, without having to consider the underlying infrastructure[1].We discuss the advantages and problems of serverless computing for scientific applications. Based on the analysis of existing solutions and approaches, we propose a science-oriented architecture for a serverless computing framework based on the existing designs. This paper focuses on several types of cybercrime and discusses some recent social engineering-based cybercrime assaults that have been defeated through cloud computing. Finally, we provide an outlook for current trends and future directions. We hope that our work in this paper can inspire those researchers and practitioners who were engaged in related fields to appreciate serverless computing, thereby setting foot in this promising area and making better contributions to its developments.**

Keywords—Serverless computing,cloud computing, Cyber security,social engineering attacks.

I. INTRODUCTION

Serverless computing is a cloud computing execution model in which the cloud provider allocates machine resources on demand, taking care of the servers on behalf of their customers. "Serverless" is a misnomer in the sense that servers are still used by cloud service providers to execute code for developers. However, developers of serverless applications are not concerned with capacity planning, configuration, management, maintenance, fault tolerance, or scaling of containers, VMs, or physical servers. Serverless computing does not hold resources in volatile memory; computing is rather done in short bursts with the results persisted to storage. When an app is not in use, there are no computing resources allocated to the app. Pricing is based on the actual number of resources consumed by an application[1]Serverless computing frees developers from having to handle infrastructure, allowing them to create apps more quickly. In serverless applications, the infrastructure needed to run the code is automatically provisioned, scaled, and managed by the cloud service provider. IT firms are converting to the serverless computing paradigm as a result of the explosion in cloud-based platforms because it promises to be a more affordable way to create and manage cloud. Popular serverless computing services include IBM Open Whisk, Cloudflare Workers, Google Cloud Functions, Microsoft's Azure Functions, and others. The application of conventional techniques like setting up and configuring firewalls, utilizing IPS technologies, or even utilizing server-based security measures. In order to secure the code functionalities, serverless security is the extra layer of prevention that is directly implemented to the applications. This gives developers compliance and security posture over their apps. The easiest way to reduce privileges in independent functions is to isolate them from one another and restrict interactions between them by granting IAM roles based on their rights..

II. SERVERLESS COMPUTING ARCHITECTURE:



Serverless architecture is largely based on a Functions as a Service (FaaS) model that allows cloud platforms to execute code without the need for fully provisioned infrastructure instances. FaaS, also known as Compute as a Service (CaaS), are stateless, server-side functions that are event-driven, scalable, and fully managed by cloud providers. DevOps teams write code that focuses on business logic and then define an event that triggers the function to be executed, such as an HTTP request. The cloud provider then executes the code and sends the results to the web application for users to review.Serverless computing is a cloud computing model in which cloud providers manage the underlying infrastructure and automatically allocate computing resources as needed to execute and scale applications. The mechanism of serverless computing involves several key components:

1. Function: A function is a small, independent piece of code that performs a specific task or handles a specific event. In serverless computing, applications are broken down into small, independent functions that can be executed in response to specific triggers.
2. Event Trigger: An event trigger is a specific event that causes a function to execute. For example, a user request, data update, or scheduled event can all be event triggers.

3. Function Container: A function container is a lightweight runtime environment that executes a function.

4. Cloud Provider: The cloud provider is in charge of overseeing the underlying infrastructure and distributing computer resources automatically.

5. Pay-per-use: With serverless computing, developers only pay for the computer resources they really utilise. This enables cost-effective scaling and can reduce the cost of infrastructure.

The cloud provider oversees the underlying infrastructure and automatically assigns computing resources based on demand, allowing for maximum scalability and cost-effectiveness.

A. Cloud computing and Serverless computing:

Clients operate an application using cloud computing, which is more automated than Serverless computing, in a contemporary data centre. Customers must manage everything in this scenario. Customers who use serverless architecture are solely concerned with the code. It aids in the development of many kinds of enterprises. As a result, more and more businesses are curious about how to put such solutions into practise and save money. Cloud computing and serverless architecture both have advantages and disadvantages.

1. Infrastructure Management: In cloud computing, the infrastructure's underpinnings are managed by the cloud provider, leaving developers free to build and deploy functionalities.

2. Scalability: In cloud computing, scaling is frequently accomplished by acquiring additional virtual machines or other infrastructure resources. It is more scalable than conventional cloud computing

3. Cost: Cloud computing requires the developer to pay for the computing resources used

4. Flexibility: Cloud computing is a more flexible model than serverless computing, as developers have more control over the infrastructure and can deploy a wider range of applications.

b. *Technologies used in serverless computing:*

In the serverless computing paradigm, the cloud service provider dynamically controls the distribution of computer resources to run code on the client's behalf. Some instances are;

1. Function-as-a-Service (FaaS) platforms: Serverless applications are deployed and managed using FaaS platforms. AWS Lambda, Google Cloud Functions, and Microsoft Azure Functions are a few examples of FaaS systems. 2. Cloud Storage: Static assets and application data are stored in the cloud using services like Amazon S3 or Azure Blob Storage.

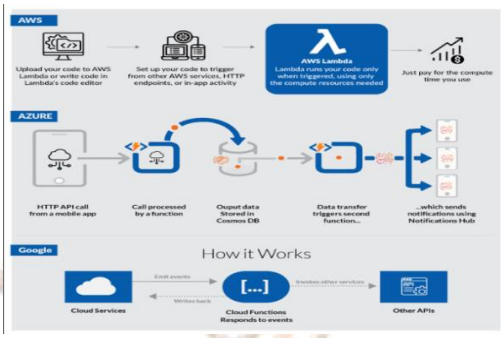
2. API Gateway: To create a public API for your serverless application, utilise an API Gateway. Examples of API Gateway services are AWS API Gateway and Azure API Management.

3. Event Triggers: Event triggers are used to invoke serverless functions in response to events. Events can be generated by other cloud services or external systems. For example, AWS S3 can trigger a Lambda function when a new object is uploaded.

4. Infrastructure-as-Code (IaC): IaC tools like Terraform and CloudFormation are used to define and deploy the infrastructure for serverless applications.

5. Serverless Frameworks: Serverless frameworks like Serverless Framework and AWS SAM provide a higher-level abstraction for deploying and managing serverless applications.

AWS Lambda, Microsoft Azure Functions, Google Cloud Functions and IBM Open Whiskare the examples of cloud providers.



Google Cloud Functions:Reducing development time and server costs while simplifying the build process are goals that universally appeal to business teams and IT teams. HomeAway relied on Google Cloud Functions to develop an app that allowed users to search and comment on the recommendations of travelers in real time, even in areas without an internet connection.

AWS Lambda function: Users may run code for practically any kind of application or backend service with AWS Lambda, a serverless, event-driven computing service, without creating or managing servers. Coca-Cola, a global leader in soft drinks, has eagerly embraced serverless after seeing how much money it can save when used in vending machines. The payment gateway calls the Amazon API Gateway whenever a beverage is purchased. The ability to pay per request rather than operating at full capacity had a significant influence on cutting expenses because vending machines must connect with headquarters for inventory and marketing purposes. To transmit notifications from Amazon Web Services (AWS) to DevOps teams through Slack, a popular cloud-based business communication channel, marbot employs a serverless application

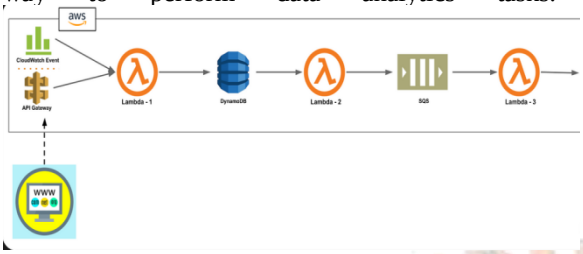
III. APPLICATIONS OF SERVERLESS COMPUTING:

Here are some examples of applications of serverless computing:

- Web and mobile backends: Serverless computing can be used to build scalable web and mobile applications. Developers can use services like AWS Lambda, Google Cloud Functions.
- Big data processing: Serverless computing can be used to process and analyze large volumes of data. Services like AWS Lambda, Google Cloud Functions trigger data processing workflows in response to events.
- IoT applications: Serverless computing can be used to build IoT applications that collect, process, and act on sensor data.
- Event-driven applications: Serverless computing can be used to build event-driven applications. Services like AWS Lambda, Google Cloud Functions, or Azure Functions can be used to trigger the application logic in response to the events.

Serverless functions can be used for;[2]

- Serverless computing can be used in **data analytics** to perform real-time data processing, machine learning, and data visualization tasks and to create data processing pipelines. Serverless computing can provide a cost-effective and scalable way to perform data analytics tasks.



create node js file;

```
mkdir serverless-sns-ses && cd serverless-sns-ses
npm init -y
touch aa.js
```

open the file and insert following;

```
exports.handler = async () => {
const responseBody = "heyyy!";
const response = {
"statusCode": 200,
"body": JSON.stringify(responseBody)};
return response;};
```

now it returns heyyy!

setting up serverless framework;

```
npm install -g serverless
```

run the terminal;

```
serverless -v
```

create a yml file and deploy;

```
service: serverless-deploy
provider:
  name: aws
  runtime: nodejs14.x
  stage: dev
  region: us-east-1
functions:
   fun:
     handler: aa.handler
     events:
      - http:
         path: fun
         method: get
```

now created one function, fun which uses the exported handler from our
configured with some HTTP triggers.

```
serverless deploy --aws-profile serverless-sns
service: serverless-deploy
provider:
  name: aws
  runtime: nodejs12.x
  stage: dev
  region: us-east-1
functions:
   hello:
     handler: index.handler
     events:
      - http:
         path: /hello
         method: get
   dispatcher:
     handler: dispatch.handler
     events:
       - sns: dispatch
exports.handler = async (event) => {
const response = {
"statusCode": 200,
"body": JSON.stringify(event)};
return response;};
```

This handler would receive the SNS publish event.

another js file create to invoke sns then Adding SES;

```
const { SESClient, SendEmailCommand } = require("@aws-sdk/client-ses");
const REGION = "us-east-1";
const sesClient = new SESClient({ region: REGION });
exports.handler = async (event) => {
const promises = event.Records.map((record) => {
const message = JSON.parse(record.Sns.Message);
const params = {
Destination: {
ToAddresses: [
message.to,
],
},
Message: {
Body: {
Html: {
Charset: "UTF-8",
Data: message.content,
},
Text: {
Charset: "UTF-8",
Data: "TEXT_FORMAT_BODY",
},
},
Subject: {
Charset: "UTF-8",
Data: "Sign up",
},
},
Source: "verifiedsesemail@test.com"
};
sesClient.send(new SendEmailCommand(params)).then(
function(data) {
const response = {
"statusCode": 200,
"body": JSON.stringify(data)
};
return response;
}).catch(
function(err) {
console.error(err, err.stack);
});
});
await Promise.all(promises);
};
```

thus bulk email sent using SES.

IV. SECURITY CONSIDERATIONS:

Here are some key security considerations in serverless computing:

Function Security: Functions must be essential to ensure that they are secure. This includes protecting against injection attacks, data leakage, and unauthorized access. Developers should ensure that functions are designed to handle security threats and that they are tested thoroughly for vulnerabilities.

API Security: APIs are used to trigger functions in serverless computing, so it is essential to ensure that APIs are secure. This includes protecting against denial-of-service attacks, API key exposure, and unauthorized access.

Data Security: Serverless applications often rely on third-party services for data storage and processing.

Users must make sure data are secure, adhere to all legal requirements, and install efficient monitoring and logging tools in order to make the apps secure.

A. Social engineering attacks in serverless computing:

Social engineering attacks in serverless computing can be particularly dangerous because serverless applications typically rely on third-party services and APIs. Here are some common social engineering attacks in serverless computing:

Phishing: Phishing attacks are a common social engineering tactic that involves tricking users into clicking on a malicious link. In serverless computing, phishing attacks can target developers, system administrators, or end-users, and can be used to steal API keys or other sensitive information.

Social Engineering through APIs: Social engineering attacks can also be carried out through APIs. Attackers might create a fake API or impersonate a legitimate API in order to gain access to sensitive data or execute unauthorized functions.

Denial-of-service (DoS) attacks: DoS attacks occur when an attacker floods a serverless function with requests, causing it to become unresponsive.

Man-in-the-middle (MITM) attacks: MITM attacks occur when an attacker intercepts communication between a client and a serverless function, allowing them to view and modify the data being transmitted.

For example, considering a case in Mexico; where More than 93 million voter registration records in Mexico were hacked in 2016. The cause was a database that was badly set up and unlawfully hosted on an Amazon cloud server outside of Mexico.After an incident in 2018, the identities and emails of 21 million users of the Time hop application were stolen. To get access to the application's cloud environment, the attacker exploited Time hop admin credentials.In 2019, the information of over 100 million Capital One customers was hacked. Customers' social security information, bank account numbers, and credit card applications were accessible to a hostile actor thanks to a cloud misconfiguration.

V. SERVERLESS COMPUTING AND CYBERSECURITY:

This paper examines the implications of serverless computing on cybersecurity, including the benefits and challenges it poses.

A. *Benefits of Serverless Computing for Cybersecurity*:

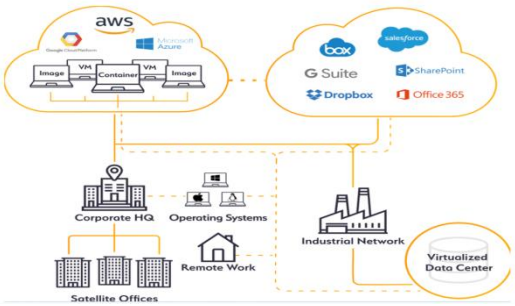
- One of the benefits of serverless computing is that it reduces the attack surface area. Since the cloud provider is responsible for managing the infrastructure, the developer does not need to worry about patching servers, managing firewalls, or configuring load balancers. Another advantage is that serverless computing provides automatic scaling. Cloud providers offer various security features such as identity and access management, encryption, and monitoring, which can be easily integrated into serverless applications.

B. *Challenges of Serverless Computing for Cybersecurity:*

Despite its benefits, serverless computing also presents some challenges for cybersecurity. One of the main challenges is the lack of visibility and control over the infrastructure. Serverless applications typically use many third-party services, which can introduce vulnerabilities.

C. *ROLE OF AI:*

AI (Artificial Intelligence) is increasingly being used in cloud security to enhance threat detection, response, and prevention capabilities.AI can help organizations to better protect their cloud environments by improving threat detection, response, and prevention capabilities. These are some companies offer various AI-based solutions for cloud security, including threat detection and response, behavioural analysis, vulnerability assessment, and fraud detection. They use different AI technologies, such as machine learning, natural language processing, and deep learning, to enhance their cloud security offerings, they are ;
Darktrace ,Vectra AI ,Cynet ,Zscaler ,Fortinet, Trend Micro, McAfee .



D. *Mechanism to Protect Data in Serverless Computing:*

There are several mechanisms that can be used to protect data in serverless computing. Here are some examples:

1. Key management services: Serverless computing providers such as AWS, Google Cloud, and Azure offer key management services (KMS) that can be used to generate, store, and manage cryptographic keys for encrypting data.
2. Encryption: Encryption can be used to protect data both at rest and in transit. Data can be encrypted before being stored in databases.
3. Auditing and Logging: Auditing and logging can be used to track and monitor access to serverless functions and other resources, providing visibility into who has accessed data.
4. Data loss prevention (DLP) tools: DLP tools can be used to monitor and prevent the exfiltration of sensitive data from serverless functions
5. Secure Coding Practices: Secure coding practices can be used to ensure that serverless functions are developed with security in mind. This includes following security best practices such as input validation, output encoding, and error handling.

Thus,serverless computing offers several benefits for cybersecurity, including a reduced attack surface, automatic scaling, and the promotion of best practices for security services.

VI. ALGORITHMS USED IN SERVERLESS COMPUTING SERVERLESS COMPUTING :

There are a variety of algorithms used in serverless computing, Here are a few examples:

1. Event-driven programming: Serverless computing is built around event-driven programming, where functions are triggered by events such as a user uploading a file or a message arriving in a queue.
2. Resource allocation algorithms: Serverless computing platforms automatically allocate resources to functions based on their workload. Load balancing algorithms: Serverless computing platforms may use load balancing algorithms to distribute traffic across multiple functions. These algorithms typically involve monitoring the workload .

There are several security algorithms used in serverless computing to ensure the security of applications

1. Access control: Serverless computing platforms typically use access control algorithms .
2. Encryption: Serverless computing platforms may use encryption algorithms such as AES or RSA to encrypt data.
3. Key management: Key management algorithms are used to securely manage encryption keys.

4. DDoS protection: Distributed denial-of-service (DDoS) protection algorithms are used to prevent DDoS attacks, which can overwhelm serverless applications with traffic.
5. Security information and event management (SIEM): SIEM algorithms are used to collect and analyse security data from serverless applications.
6. Machine learning algorithms: It performs variety of tasks, such as predicting resource usage,detecting anomalies in network traffic.

VII. CONCLUSION:

The future is with serverless. The issue of allocating cloud virtual machines was resolved by the invention of serverless computing. Serverless was developed as a solution to this issue by introducing automation that does away with the requirement for users Because some of these components might contain unknown communication formats that the conventional application layer protection might not adequately assess. Due automatic resource scaling based on demand, it is simple to construct, manage, and is cost-effective. It creates fresh possibilities in a variety of IOT applications, such as systems for monitoring supplies and the land.

VIII. REFERENCES

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

[2] Miller, Ron (24 Nov 2015). "AWS Lambda Makes Serverless Applications A Reality". TechCrunch. Retrieved 10 July 2016.

[3] "What Is Serverless Computing?". ITPro Today. 2021-12-13. Retrieved 2022-03-23

[4] Taylor, H., 2019. 2020 CYBERSECURITY PREDICTIONS FOR SOFTWARE DEVELOPMENT AND ENTERPRISE ARCHITECTURE.

[5] Lynn,T,Rosati,P,.Lejune A Emeakaroha,V,2017.A Preliminary review of Enterprise serverless Cloud Computing(Function as a Service)Platforms 2017 IEEE International Conference on Cloud Computing Technology and Science(Cloudcom).IEEE.p162.