Shodh Sari-An International Multidisciplinary Journal

2025, Vol. 04, Issue 02, 129-154
ISSN: 2959-1376
DOI: https://doi.org/10.59231/SARI7814

Serverless Computing for Incident Response in Auditing

Ganapathy, Venkatasubramanian

Faculty In Auditing Department, Southern India Regional Council of The Institute of Chartered Accountants of India (Sirc of Icai), Chennai, Tamil Nadu, Bharat

Abstract

Serverless computing is transforming the landscape of auditing by providing scalable, efficient, and cost-effective solutions for incident response. Traditional auditing methods often rely on static infrastructure, which can be slow and resource-intensive when responding to security breaches or anomalies. Serverless computing, however, enables real-time monitoring, automated alerts, and rapid response mechanisms without the need for dedicated server management. This paper explores the role of serverless computing in enhancing incident response in auditing, focusing on its ability to streamline forensic analysis, improve data integrity, and ensure compliance with regulatory standards. By leveraging cloud-based functions that execute only when needed, organizations can reduce operational costs while maintaining high availability and reliability in audit processes. Key benefits of serverless computing in auditing include automated log analysis, real-time anomaly detection using AI-driven algorithms, and seamless integration with existing audit tools. Additionally, serverless architectures enhance cybersecurity by allowing dynamic scaling during potential security incidents, ensuring swift mitigation and response. Despite its advantages, challenges such as vendor lock-in, latency concerns, and security vulnerabilities must be addressed to maximize its potential in auditing environments. This study highlights best practices for implementing serverless computing in audit workflows, ensuring an adaptive and resilient approach to incident response. By embracing serverless computing, auditors can revolutionize their approach to risk management, enabling more agile, data-driven, and efficient incident response mechanisms.

Keywords: Auditing, cybersecurity, forensic, cost-effective, infrastructure

@2025 International Council for Education Research and Training
ISSN: 2959-1376
2025, Vol. 04, Issue 02, 129-154
DOI: https://doi.org/10.59231/SARI7814

I. INTRODUCTION

Serverless computing is a cloud computing execution model where the cloud provider manages the infrastructure, including server allocation, scaling, and maintenance. Instead of provisioning and maintaining servers, developers write and deploy code as individual functions that are executed on demand.

Incident response (IR) refers to the structured approach taken by an organization to detect, investigate, and mitigate cybersecurity incidents, such as data breaches, malware infections, and denial-of-service (DoS) attacks. The goal is to limit the impact of the incident, restore normal operations as quickly as possible, and prevent similar incidents from occurring in the future.

II. RESEARCH QUESTION

“How can serverless computing be utilized to enhance the efficiency, scalability, and real-time responsiveness of incident detection, response, and reporting in auditing processes?"

III. TARGETED AUDIENCE

1. Academics & Researchers

Who: Professors, students, and researchers in computer science, cybersecurity, cloud computing, and information systems auditing.

Why: They seek to explore novel approaches to automation, serverless architectures, and incident response frameworks for academic study and further research.

2. IT Auditors & Compliance Professionals

Who: IT auditors, financial auditors, and compliance officers working in regulated industries like finance, healthcare, and government sectors.

Why: They are interested in learning how serverless computing can streamline audit processes, reduce response time to incidents, and ensure compliance with standards like **GDPR, ISO 27001, SOC 2**, etc.

3. Cybersecurity Professionals & Incident Responders

Who: Security analysts, incident response teams (IRT), and professionals responsible for Security Information and Event Management (SIEM) tools.

Why: They would benefit from faster, real-time, and cost-effective incident response capabilities provided by serverless architectures. This paper could offer practical insights or tools for enhancing incident response workflows.

4. Cloud Architects & DevOps Engineers

Who: Cloud architects, DevOps, and Site Reliability Engineers (SREs) involved in managing cloud-native infrastructures.

@2025 International Council for Education Research and Training
ISSN: 2959-1376
2025, Vol. 04, Issue 02, 129-154
DOI: https://doi.org/10.59231/SARI7814

Why: They would be interested in **serverless design patterns** for incident response, especially since serverless infrastructure can improve scalability, fault tolerance, and cost-efficiency in auditing and incident detection.

5. IT Managers & Decision Makers (CISOs, CIOs, CTOs)

Who: Senior IT executives, Chief Information Security Officers (CISOs), and technology decision-makers in organizations.

Why: They are responsible for aligning technical decisions with **business goals, cost savings, and compliance requirements**. Understanding the role of serverless computing in incident response could guide strategic investments.

6. Cloud Service Providers & Platform Vendors

Who: Cloud providers (like AWS, Microsoft Azure, Google Cloud) and companies offering serverless platforms or auditing solutions.

Why: They may look for ways to **market new serverless-based incident response tools** or to improve their current cloud offerings. This paper could inform the development of **new features or products**.

7. Regulatory Bodies & Standardization Organizations

Who: Organizations that define and enforce standards (like ISO, NIST, and data protection authorities like the **European Data Protection Board**).

Why: They may seek to understand the impact of serverless computing on incident response and audit trails to **create or refine compliance guidelines**.

IV. OBJECTIVES OF THE STUDY

1. To explore the role of serverless computing in incident response in auditing
2. To develop a conceptual framework for audit-focused incident response
3. To compare serverless computing with audit-based incident response
4. To evaluate the potential benefits and address challenges and limitations of using serverless architecture for incident response in auditing

V. RESEARCH METHODOLOGY AND DATA COLLECTION METHODS

Conceptual Analysis Research Methodology used in this research paper. This paper proposes a conceptual model for integrating serverless computing into incident response workflows for auditing, **Secondary Data** used for the study, collected from e-journals, e-magazines, e-books and the websites of various service providers (Cloud

@2025 International Council for Education Research and Training
ISSN: 2959-1376
2025, Vol. 04, Issue 02, 129-154
DOI: https://doi.org/10.59231/SARI7814

Providers) like AWS, Microsoft Azure, Google cloud, Serverless computing for incident response in auditing domains.

VI REVIEW Of LITERATURE

NO.	Author(s)	Year	Focus of Study	Algorithms/Tools used	Key Findings	Research Gap
1	Eismann et al.	2020	Serverless evolution and architectural changes.	Cloud providers' FaaS platforms.	Serverless introduces scalability, low cost, and modularity.	Incident response integration remains underexplored.
2	Taibi et al.	2020	Serverless vs PaaS for software development.	Serverless development tools.	Serverless has lower development and maintenance costs.	No emphasis on the role of serverless in incident response.
3	Barcelona-Pons et al.	2022	Stateless serverless computing for short-duration tasks	Serverless orchestration tools	Serverless works well for short tasks but struggles with long-running processes	Limited exploration of long-term incident response capabilities

@2025 International Council for Education Research and Training
ISSN: 2959-1376
2025, Vol. 04, Issue 02, 129-154
DOI: https://doi.org/10.59231/SARI7814

4	Sharma & Gupta	2022	Blockchain integration with serverless for auditing	Blockchain, smart contracts	Immutable audit trails for serverless workflows	Need for hybrid serverless-blockchain models for auditing
5	Sreekanth et al.	2023	Incident response automation with serverless	AWS Lambda, Google Cloud Functions	Automated response triggers anomaly detection	No study on multi-cloud serverless incident response
6	IEEE Report,	2024	Application logic protection in serverless.	Data obfuscation, access control	Highlighted vulnerabilities in serverless logic security	Lack of incident-specific logic protection methods
7	IEEE Report	2024	Security challenges in serverless computing	Encryption, access control	Identified data privacy and logic integrity threats.	No clear approach to securing incident response processes.
8	IEEE Report	2024	Architectural paradigms for future serverless technology	Cloud orchestration frameworks	Described emerging serverless paradigms and trends	Need for frameworks that link serverless to audit incident response

@2025 International Council for Education Research and Training
ISSN: 2959-1376
2025, Vol. 04, Issue 02, 129-154
DOI: https://doi.org/10.59231/SARI7814

Key Insights from the Literature

- **Shift from Cloud to Serverless**: Early works (like Mell & Grance, 2011) established the foundation for cloud computing, but it wasn't until 2018–2020 that serverless computing gained traction for dynamic, event-driven applications. Serverless introduced automated scaling, reduced costs, and faster development cycles

- **Incident Response Capabilities**: Research from Eismann et al. (2020) and Barcelona-Pons et al. (2022) reveals that serverless functions are ideal for short-lived, event-driven tasks but struggle with long-running incident response actions due to stateless limitations

- **Security and Privacy**: As serverless adoption grows, so do its security challenges. The 2024 IEEE reports emphasize the need for encryption, secure access control, and logic protection for serverless environments, as incident response often involves sensitive audit data

- **Tools and Algorithms**: Tools like AWS Lambda, Google Cloud Functions, and Azure Functions dominate the serverless landscape. Sreekanth et al. (2023) highlighted the use of anomaly detection algorithms and automated workflows for incident response

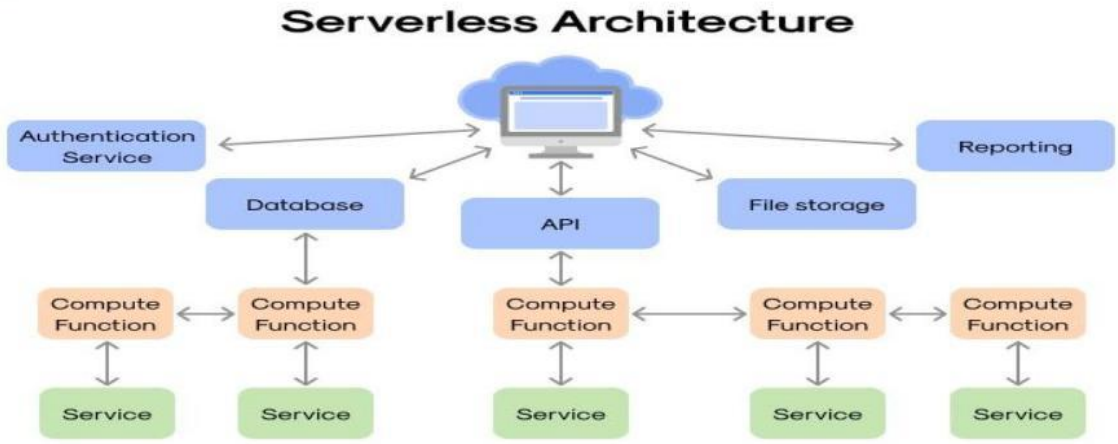
- **Research Gaps**: Key areas for further exploration include developing **long-running incident response workflows** for serverless, designing **hybrid blockchain-serverless systems for immutable audit trails**, and addressing **multi-cloud interoperability** for incident response processes.

I. ROLE OF SERVERLESS COMPUTING IN INCIDENT RESPONSE IN AUDITING

Serverless computing plays a transformative role in enabling **faster, more efficient, and cost-effective incident response** in auditing. By leveraging event-driven, on-demand computing capabilities, serverless functions (like AWS Lambda, Google Cloud Functions, and Azure Functions) automate key aspects of incident detection, response, and reporting.

@2025 International Council for Education Research and Training
ISSN: 2959-1376
2025, Vol. 04, Issue 02, 129-154
DOI: https://doi.org/10.59231/SARI7814

wallarm



1. Real-Time Incident Detection and Response

Role: Serverless functions enable **real-time monitoring** and response to incidents by processing logs, audit trails, and system events as they happen.

How it Works:

- **Event-Driven Triggers**: Serverless platforms can be triggered by system logs, file uploads, API requests, or alerts from monitoring tools like AWS CloudWatch, Azure Monitor, or Google Cloud Logging.
- **On-Demand Execution**: Unlike traditional servers, serverless functions run **only when needed**, significantly reducing latency and enabling a near-instant response to critical incidents.
- **Incident Automation**: When an anomaly is detected in audit logs (e.g., suspicious login attempts or data access), a serverless function can be triggered to take predefined actions like isolating the affected system, alerting the security team, or generating incident reports.

Example Use Case:

- When a **suspicious login attempt** (e.g., multiple failed logins) is detected in AWS CloudTrail logs, an AWS Lambda function is triggered to:
  - **Block the user's IP address** using a security rule update.
  - **Send an alert** to the security operations team (via email, Slack, or a SIEM system).

@2025 International Council for Education Research and Training
ISSN: 2959-1376
2025, Vol. 04, Issue 02, 129-154
DOI: https://doi.org/10.59231/SARI7814

o **Log the event** in a centralized audit trail for compliance purposes.

2. Automation of Auditing and Incident Logging

Role: Serverless functions automate the collection, analysis, and storage of **audit logs** related to security incidents.

How it Works:

- **Data Ingestion**: Serverless platforms collect and aggregate logs from multiple sources (application logs, server logs, network traffic, etc.).
- **Data Transformation**: Serverless functions process, sanitize, and categorize logs for downstream analysis.
- **Log Storage and Management**: After processing, logs are sent to centralized logging solutions like **AWS S3, Azure Blob Storage, or Google Cloud Storage**, ensuring a **tamper-proof audit trail** for compliance.

Example Use Case:

- An AWS Lambda function could be triggered each time a user uploads a file to a **cloud storage bucket**. The function can:

o **Record metadata** (file name, uploader, timestamp, etc.) to the audit log.

o **Check for malware** using a virus scanning service.

o **Alert the security team** if the file is flagged as malicious.

This approach ensures every file upload is automatically logged, scanned, and reviewed, providing a **fully auditable trail** for compliance.

3. Cost-Efficient Incident Response at Scale

Role: Serverless computing provides an **economically efficient, pay-as-you-go model** for running audit-related incident response operations.

How it Works:

- **No Idle Costs**: Unlike dedicated servers, serverless functions only run when triggered, which means no costs are incurred during idle time.
- **Scalability**: When large volumes of log data are ingested (e.g., during a security breach), serverless systems automatically scale to handle the load without manual intervention.
- **Resource Optimization**: Since resources are provisioned on-demand, incident response processes are highly cost-efficient, especially when compared to **always-on** server-based systems.

@2025 International Council for Education Research and Training
ISSN: 2959-1376
2025, Vol. 04, Issue 02, 129-154
DOI: https://doi.org/10.59231/SARI7814

Example Use Case:

- If an audit system experiences a spike in **suspicious user activity** (like multiple unauthorized access attempts), serverless functions are automatically scaled to process **millions of log entries** in real time.
- Traditional infrastructure would require pre-provisioning of resources for peak loads, whereas serverless computing allows for **on-the-fly scaling**.

This cost-efficiency makes it practical for even small and mid-sized organizations to implement **enterprise-grade auditing and incident response**.

4. Incident Mitigation and Threat Containment

Role: Serverless functions help contain and mitigate threats in real time, reducing the potential damage caused by security incidents.

How it Works:

- **Automatic Remediation**: If a threat is detected, serverless functions can automatically block access, quarantine affected files, or update firewall rules.
- **Isolation of Infected Resources**: Serverless functions can temporarily isolate affected systems from the broader network.
- **Role-Based Access Control (RBAC)**: In cases where **unauthorized access** is detected, serverless functions can automatically update user permissions or block accounts.

Example Use Case:

- If a malicious insider tries to export sensitive data from a company's system, a serverless function could:
  - **Disable the user's account** in the Identity and Access Management (IAM) system.
  - **Revoke access to resources** (like S3 buckets) for that user.
  - **Alert the security team** via messaging apps like Slack or Microsoft Teams.

This proactive approach to containment prevents further escalation of the threat.

5. Forensic Analysis and Compliance Reporting

Role: Serverless platforms facilitate the generation of **compliance reports and forensic analysis** for auditing purposes.

How it Works:

- **Data Analysis**: Serverless functions analyze stored logs and security events to

@2025 International Council for Education Research and Training
ISSN: 2959-1376
2025, Vol. 04, Issue 02, 129-154
DOI: https://doi.org/10.59231/SARI7814

generate reports required for audits (like SOC 2, GDPR, and ISO 27001).

- **Report Generation**: Serverless functions can automatically produce summary reports on key security metrics (e.g., how many incidents occurred, how many were mitigated, etc.).
- **Tamper-Proof Log Storage**: By storing audit logs in immutable storage systems like **AWS S3 with Object Lock** or **Azure Immutable Blobs**, serverless functions ensure that evidence is preserved for forensic purposes.

Example Use Case:

- Prior to an annual **SOC 2 audit**, a serverless function is used to generate a compliance report that:
  - **Summarizes system access logs** for the past 12 months.
  - **Lists incident response activities** (e.g., which threats were detected, who responded, and how the threat was contained).
  - **Provides evidence of access control and security protocols** (like user role changes and file access attempts).

This provides auditors with the information they need, while also reducing the burden on internal compliance teams.

6. Security and Privacy Enhancements

Role: Serverless computing enforces better **data security, privacy, and integrity** in auditing and incident response.

How it Works:

- **Secure Storage of Logs**: Logs and event data can be encrypted before storage.
- **Data Masking and Anonymization**: Serverless functions can automatically anonymize personal data to comply with privacy laws (like GDPR) before logs are stored.
- **Access Control**: Serverless functions run with **fine-grained permissions**, following the principle of least privilege, which limits the scope of security breaches.

Example Use Case:

- An AWS Lambda function encrypts every log file uploaded to an S3 bucket with a **unique encryption key**.
- This function ensures that only authorized users (like auditors) can access the encrypted logs, ensuring compliance with **data protection laws (like GDPR and CCPA)**.

This role emphasizes the use of encryption, identity management, and **access control best practices** to maintain data privacy and security

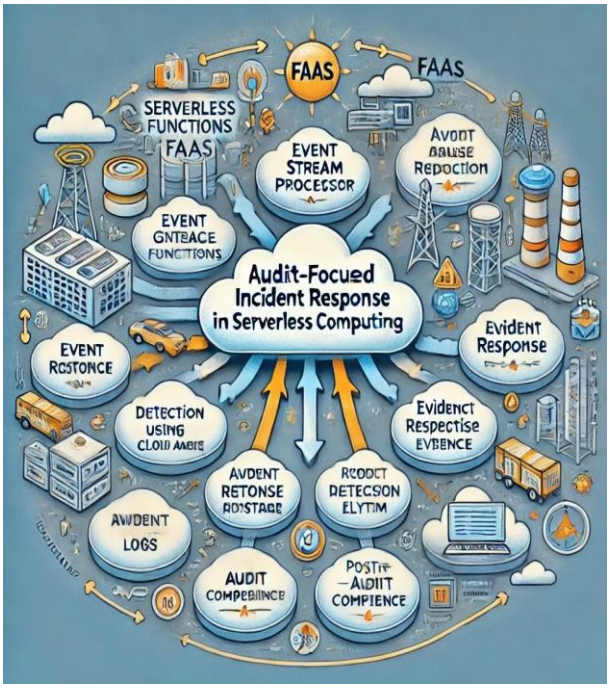
@2025 International Council for Education Research and Training
ISSN: 2959-1376
2025, Vol. 04, Issue 02, 129-154
DOI: https://doi.org/10.59231/SARI7814

Role	Key Features	Benefits
Real-Time Detection	Event triggers, anomaly detection	Faster response to threats
Audit Log Automation	Log ingestion, data storage	Automated audit trails
Cost Efficiency	Pay-per-use model, auto-scaling	Reduced operational costs
Incident Mitigation	Auto-block access, threat isolation	Quicker containment of threats
Forensic Analysis	Report generation, immutable logs	Easy compliance and auditability
Security & Privacy	Encryption, role-based access	Data protection & GDPR compliance

II. DEVELOPMENT OF CONCEPTUAL FRAMEWORK FOR AUDIT-FOCUSED INCIDENT RESPONSE IN SERVERLESS

III. COMPUTING

Visual Representation of Conceptual Framework



Explanation of the Conceptual Framework

The framework consists of four main layers, each with specific responsibilities for ensuring **auditability**, **incident response**, and **compliance** in serverless environments.

Layer 1: Serverless Functions and Event Sources

Objective: Capture events and trigger serverless functions (FaaS) that execute business logic and generate logs for auditing.

Components:

- **Event Sources:**
  - o **API Gateway**: Receives external HTTP/HTTPS requests.

@2025 International Council for Education Research and Training
ISSN: 2959-1376
2025, Vol. 04, Issue 02, 129-154
DOI: https://doi.org/10.59231/SARI7814

o **Cloud Storage**: Event-driven triggers from cloud storage (like AWS S3, Google Cloud Storage).

o **Databases**: Changes in databases (like AWS DynamoDB Streams) trigger serverless functions.

o **Third-Party APIs**: External triggers from external services.

Key Considerations:

- **Auditability**: Each request should have a unique identifier (like a request ID) to trace its origin and execution path.
- **Cold Start Mitigation**: Reduce cold start latency through **provisioned concurrency** or function warmers.
- **Encryption**: Apply end-to-end encryption (TLS) for event transport and data encryption at rest.

Layer 2: Detection and Monitoring

Objective: Monitor serverless function executions, detect unusual behavior using **machine learning (ML)**, and log all events.

Components:

- **Event Stream Processor**: Captures and processes logs and events in real-time from event sources and FaaS.
- **Logging System**: Stores logs generated by serverless functions and captures execution context, input, and output data.
- **Anomaly Detection (ML)**: Machine learning models (like Isolation Forest, Autoencoders) detect abnormal behavior (e.g., unusual function execution times, unexpected API calls, or suspicious IP addresses).
- **Real-Time Alerts**: Alerts are sent to teams via Slack, email, or PagerDuty in response to detected anomalies.

Key Considerations:

- **Auditability**: Capture logs at every stage, tag logs with request IDs, and store them in **tamper-proof, append-only storage** (like AWS S3 with object lock).
- **Machine Learning**: Use supervised learning for pre-classified datasets or unsupervised anomaly detection for zero-day threats.
- **Data Privacy**: Encrypt log data at rest, ensuring compliance with GDPR and other regulations.

Layer 3: Incident Response

Objective: Respond to anomalies and incidents detected during the monitoring phase. Provide real-time response and forensics.

Components:

- **Automated Response**: Trigger actions like revoking permissions, blocking IPs, or isolating serverless functions.

@2025 International Council for Education Research and Training
ISSN: 2959-1376
2025, Vol. 04, Issue 02, 129-154
DOI: https://doi.org/10.59231/SARI7814

- **Root Cause Analysis (RCA)**: Investigate the source of incidents by examining serverless function execution context, event source, and logs.
- **Evidence Collection**: Store evidence for auditability and forensic analysis in tamper-proof storage.

Key Considerations:

- **Auditability**: Ensure a chain of custody for incident evidence. Evidence should be encrypted and stored with integrity checks.
- **Vendor Lock-In**: Consider using multi-cloud forensic tools and **cloud-agnostic monitoring** to avoid reliance on one vendor's proprietary tools.
- **Automation**: Use AWS Step Functions, Azure Logic Apps, or Google Cloud Workflows to automate incident response playbooks.

Layer 4: Audit and Compliance

Objective: Ensure compliance with regulatory frameworks (like GDPR, SOC 2, ISO 27001) and review audit trails post-incident.

Components:

- **Immutable Audit Logs**: Logs are stored in tamper-proof storage, such as AWS S3 with object lock, to prevent deletion.
- **Regulatory Compliance**: Map captured data to specific compliance controls (e.g., GDPR, PCI-DSS, HIPAA).
- **Post-Incident Review**: Conduct a post-mortem, update audit logs, and improve future incident response processes.

Key Considerations:

- **Auditability**: Use **Blockchain-based logging** or append-only logs to support tamper-proof evidence collection.
- **Data Retention**: Ensure log data is retained for an appropriate period (as required by regulations) and establish a **data lifecycle policy**.
- **Reporting**: Generate compliance reports from logs to satisfy auditors and regulators.

Key Design Considerations

1. **Mitigating Vendor Lock-In**
   - o Use abstraction layers like **Cloud Custodian**, **Terraform**, or multi-cloud FaaS frameworks like **Knative** or **OpenFaaS**.
   - o Design serverless functions that can be ported to multiple providers.
2. **Handling Cold Start Latency**
   - o Use **pre-warming strategies** like AWS **Provisioned Concurrency** or **container snapshots** to reduce startup time.

@2025 International Council for Education Research and Training
ISSN: 2959-1376
2025, Vol. 04, Issue 02, 129-154
DOI: https://doi.org/10.59231/SARI7814

o Consider using containerized FaaS (like AWS Lambda SnapStart) for faster startup.

3. **Machine Learning for Detection**

o Use **Unsupervised Anomaly Detection** (like Isolation Forest) to detect new, unseen threats.

o Deploy **Supervised Classification Models** for detecting previously identified attack signatures.

4. **End-to-End Encryption**

o **Data-in-Transit**: Ensure TLS 1.3 for all data transfers.

o **Data-at-Rest**: Use strong encryption (AES-256) for stored log files, evidence, and serverless execution context.

o **Data-in-Memory**: Use homomorphic encryption or confidential computing environments for sensitive in-memory data.

Audit-Ready Logs

o Capture detailed logs of **who, what, where, when, and why** for each function execution.

o Design a schema for **log normalization** to ensure consistency across different FaaS platforms (AWS, Azure, GCP).

Benefits of the Framework

- **Auditability**: Immutable, verifiable logs to support incident investigation.
- **Automation**: ML-based anomaly detection and automated incident response.
- **Compliance**: Demonstrate adherence to GDPR, PCI-DSS, and other frameworks.
- **Cold Start Mitigation**: Reduced cold start delays, improving user experience.
- **Vendor Independence**: Portability across cloud providers, reducing vendor lock-in.

5. Tools and Technologies

- **Serverless Functions**: AWS Lambda, Azure Functions, Google Cloud Functions.
- **Logging & Monitoring**: AWS CloudWatch, Azure Monitor, GCP Cloud Logging, Elastic Stack.
- **Anomaly Detection (ML)**: Python ML libraries (scikit-learn, TensorFlow), AWS Fraud Detector, Google AI.
- **Incident Response Automation**: AWS Step Functions, Azure Logic Apps, Google Cloud Workflows.
- **Forensics and Compliance**: AWS Audit Manager, Azure Security Center, Google Security Command Center.

IV. COMPARATIVE STUDY OF SERVERLESS COMPUTING WITH AUDIT-BASED INCIDENT RESPONSE

@2025 International Council for Education Research and Training
ISSN: 2959-1376
2025, Vol. 04, Issue 02, 129-154
DOI: https://doi.org/10.59231/SARI7814

Serverless computing and audit-focused incident response (IR) are two distinct concepts in cloud computing and cybersecurity. S**erverless computing** focuses on running applications without server management, with security focused on permissions, triggers, and event-driven behavior.

Audit-focused incident response is a reactive cybersecurity approach, driven by log analysis, forensics, and remediation of security incidents.

No	Criteria	Serverless Computing	Audit-focused Incident Response
1	Definition	A cloud-computing model where applications run on demand, without managing server infrastructure.	A cybersecurity process focused on tracking, investigating, and responding to security incidents.
2	Primary Goal	Optimize resource usage, scalability, and cost by running functions only when needed.	Ensure accountability, traceability, and proper response to security breaches.
3	Infrastructure	Managed entirely by a cloud provider (e.g., AWS Lambda, Google Cloud Functions).	Involves on-prem, cloud, and hybrid environments, often requiring visibility across multiple layers.
4	Security Focus	Securing functions, permissions, and event triggers.	Collecting, analyzing, and responding to security incidents using audit logs and forensics.

@2025 International Council for Education Research and Training
ISSN: 2959-1376
2025, Vol. 04, Issue 02, 129-154
DOI: https://doi.org/10.59231/SARI7814

5	Automation	Heavily automated, with event-driven execution.	Automation is common in detection, alerting, and response workflows (e.g., SIEM, SOAR tools).
6	Audit Logs	Cloud providers generate execution logs (e.g., AWS CloudTrail, CloudWatch) for serverless functions.	Audit logs are central to incident response, capturing evidence to support investigations and compliance.
7	Visibility	Limited access to the underlying infrastructure, relying on provider logs.	Full-stack visibility is required to track incidents from network to application layer.
8	Compliance	Compliance depends on the cloud provider's shared responsibility model.	Audit logs play a key role in meeting regulatory requirements like GDPR, HIPAA, and SOC 2.
9	Incident Detection	Relies on event-driven triggers and logs for abnormal execution patterns.	Continuous monitoring via SIEMs, IDS, and EDR tools to detect potential threats.
10	Response Approach	Remediation is typically done by updating the function code or permissions.	Response involves containment, eradication, and recovery of affected systems.

@2025 International Council for Education Research and Training 2025, Vol. 04, Issue 02, 129-154
ISSN: 2959-1376 DOI: https://doi.org/10.59231/SARI7814

11	Cost Model	Pay-per-use model; costs are tied to function invocations and execution time.	Costs are related to labour, tools (like SIEMs), and third-party response services.
12	Role in Cybersecurity	Serverless apps must be secured, but they can also serve as a platform for running security scripts.	Central to cybersecurity operations, ensuring incidents are detected, analyzed, and resolved.
13	Tools & Examples	AWS Lambda, Google Cloud Functions, Azure Functions.	SIEM (e.g., Splunk, QRadar), SOAR (e.g., Cortex XSOAR), forensic analysis tools.

How They Interact

1. **Serverless as Part of Incident Response**: Serverless computing can support incident response processes by running lightweight scripts for data collection, log analysis, or automated responses. For example, AWS Lambda functions might be triggered to analyze and quarantine suspicious files.
2. **Audit Logs for Serverless Functions**: Serverless applications produce audit logs (like AWS CloudTrail) that are critical for incident response. These logs track API calls, permissions, and function executions, helping detect anomalies.
3. **Incident Response for Serverless Attacks**: If an attacker compromises a serverless application, audit-focused incident response will analyze serverless logs, permissions, and API calls to understand and remediate the breach.

These two concepts intersect when audit logs from serverless environments (like AWS Lambda) are used for incident detection and response.

V. BENEFITS AND CHALLENGES OF USING SERVERLESS ARCHITECTURE FOR INCIDENT RESPONSE IN AUDITING

Potential Benefits of Serverless for Incident Response in Auditing

Automation and Speed

- **Event-Driven Automation**: Serverless functions (like AWS Lambda,

@2025 International Council for Education Research and Training
ISSN: 2959-1376
2025, Vol. 04, Issue 02, 129-154
DOI: https://doi.org/10.59231/SARI7814

Azure Functions, and Google Cloud Functions) can automatically trigger actions based on predefined events, such as log anomalies, file uploads, or access to sensitive resources.

- **Faster Response**: Since serverless functions are "always ready" and execute almost instantly, security teams can detect and respond to incidents in real time. For example:
  - **Real-Time Alerts**: Lambda can automatically trigger notifications (e.g., via AWS SNS or Slack) when suspicious activities are detected.
  - **Automated Containment**: When a threat is detected (e.g., malware upload), a serverless function can isolate the infected system or quarantine files.

Cost-Efficiency

- **Pay-per-Use Model**: Serverless computing charges only for execution time. Unlike dedicated security appliances or VMs, serverless functions run only when triggered.
- **Reduced Infrastructure Costs**: No need for dedicated servers or VMs for incident response, leading to significant cost savings.
- **Scalability**: Serverless functions scale automatically with workload, meaning incident response workflows (like large-scale log analysis) can scale during peak incidents.

Improved Auditability and Traceability

- **Detailed Logs and Monitoring**: Serverless providers (AWS, Azure, GCP) generate detailed logs of every function execution, including start times, errors, API calls, and resource access.
- **Comprehensive Audit Trails**: These logs (like AWS CloudTrail) can be ingested into Security Information and Event Management (SIEM) systems, offering clear evidence for regulatory compliance (GDPR, HIPAA, etc.).
- **Immutable Logs**: Since serverless logs are automatically captured and stored (e.g., in S3 or Azure Blob Storage), they are less prone to tampering compared to on-premises log storage.

Flexibility and Adaptability

- **Rapid Deployment**: Incident response scripts can be created and updated as serverless functions, allowing rapid customization and updates in response to new threats.
- **Integration with SIEMs and SOAR**: Serverless functions can act as "connectors" to tools like Splunk, QRadar, and SIEMs to ingest, process, and respond to log data.
- **Use of Cloud-Native Services**: Functions can leverage existing cloud-native services like AWS DynamoDB, AWS Step

@2025 International Council for Education Research and Training
ISSN: 2959-1376
2025, Vol. 04, Issue 02, 129-154
DOI: https://doi.org/10.59231/SARI7814

Functions, and AWS S3 to store, analyze, and manage incident response data.

Advanced Capabilities

- **AI and ML for Incident Response**: Cloud providers offer AI/ML tools (like AWS SageMaker) that can be invoked via serverless functions to analyze anomaly patterns in logs.
- **Orchestration and Workflow Automation**: Using AWS Step Functions or Google Workflows, incident response playbooks can be automated to handle multi-step response processes.

Challenges of Serverless for Incident Response in Auditing

While serverless architecture offers clear benefits for incident response, it also presents unique challenges that can hinder its effectiveness for auditing and incident response.

Limited Visibility and Control

- **Opaque Cloud Infrastructure**: Serverless abstracts the underlying infrastructure, so security teams have no access to the host OS, hypervisor, or network controls.
- **Reduced Access to Logs**: Unlike traditional servers, serverless provides limited logs on infrastructure details (like kernel errors), which could be useful in advanced forensics.
- **Dependency on Provider's Monitoring**: Audit logs (e.g., CloudTrail) come from the cloud provider, making them a "trusted source," but organizations have less control over log content, format, or retention.

Security Concerns

- **Event Injection Attacks**: Malicious actors may manipulate input events (e.g., webhooks or file uploads) to trigger unauthorized serverless functions, leading to data leakage or lateral movement.
- **Misconfigurations**: Permissions and IAM misconfigurations (like an over-permissive role) can allow attackers to escalate privileges and access sensitive logs.
- **Vulnerable Dependencies**: Serverless functions often rely on third-party libraries, which may introduce vulnerabilities into the audit workflow.
- **Data in Transit and Execution**: Data processed in serverless functions is momentary and stored in memory, which could be intercepted if the memory isn't cleared properly.

Complexity in Incident Response Playbooks

- **Orchestration Complexity**: Creating a full "incident response playbook" for serverless workflows may require multiple steps, such as calling external APIs, waiting for

@2025 International Council for Education Research and Training
ISSN: 2959-1376
2025, Vol. 04, Issue 02, 129-154
DOI: https://doi.org/10.59231/SARI7814

certain events, and dynamically controlling execution.

- **Coordination Across Services**: Serverless often depends on multiple cloud services (like S3, CloudWatch, and SNS), and tracking incident response workflows across them can be complex.
- **Cold Start Latency**: When serverless functions are not "warm," they experience a short delay (cold start) before execution begins, which may affect time-sensitive incident responses.

Compliance and Legal Risks

- **Data Residency and Privacy**: Incident response often involves log storage, and cloud providers may store logs in data centers that violate regional privacy regulations (like GDPR's "data sovereignty" rules).
- **Log Integrity and Tamper-Proofing**: Since cloud provider logs (like CloudTrail) are not directly managed by the user, there are questions of log tampering and integrity during forensic investigations.
- **Third-Party Dependencies**: If a cloud provider fails to deliver proper logs (due to a service outage), organizations lose valuable evidence in incident response cases.

Limitations of Serverless for Incident Response in Auditing

While the challenges can often be mitigated, some **inherent limitations** exist when using serverless in incident response and auditing.

Ephemeral Nature of Serverless Functions

- **Short Execution Times**: Serverless functions have a maximum runtime (e.g., AWS Lambda allows 15 minutes). This is sufficient for lightweight auditing tasks but may be too short for complex forensics or extended analysis.
- **No Persistent State**: Since functions run statelessly, it's difficult to maintain long-term memory or context about prior security events. Additional storage (e.g., S3) is required to hold stateful data.
- **Data in Memory**: Since serverless functions do not persist data, evidence may be lost if not stored quickly (e.g., in S3 or an RDS database).

Reliance on Cloud Provider Tools

- **Vendor Lock-In**: Serverless functions are platform-dependent (AWS Lambda, Google Functions, etc.), making it hard to migrate incident response workflows to a multi-cloud or hybrid-cloud strategy.
- **Lack of Advanced Forensics Tools**: Unlike physical servers, serverless environments do not allow direct disk or memory analysis for forensic investigation.

@2025 International Council for Education Research and Training
ISSN: 2959-1376
2025, Vol. 04, Issue 02, 129-154
DOI: https://doi.org/10.59231/SARI7814

- **Limited Debugging and Testing**: Troubleshooting serverless incident response workflows can be difficult due to limited access to "live execution" details. Developers must rely on logs instead of direct observation.

Logging and Evidence Collection

- **Delayed Log Availability**: Some audit logs (like AWS CloudTrail) are not "real-time" and may take several minutes before logs are available, delaying incident response.
- **High Log Volumes**: Serverless environments generate large amounts of audit data, and filtering critical logs for forensic purposes can be overwhelming.
- **Retention and Cost Issues**: Storing logs for auditing and incident response for long periods (for regulatory compliance) can be expensive in serverless environments.

Summary Table

Category	Benefits	Challenges	Limitations
Automation	Fast, real-time execution	Event-driven attack risks	Short execution times (15 min limit)
Cost	Pay only for use	Costs rise with complex playbooks	High log storage costs
Audit Logs	Centralized, immutable logs	Delay in CloudTrail log availability	No direct access to system logs
Security	Integrated IAM, audit trails	Cloud abstraction, limited visibility	Cloud provider log trust issues
Forensics	API-driven investigation	No memory/disk access	Stateless, ephemeral execution

VI. KEY FINDINGS

❖ **Improved Scalability:** How serverless computing can enhance incident response in auditing by providing scalable resources on-demand. This could lead to faster response times during security incidents or audit processes

❖ **Cost-Effectiveness**: A key advantage of serverless computing is its pay-per-use model. The research might conclude that this model can significantly reduce costs associated

@2025 International Council for Education Research and Training
ISSN: 2959-1376
2025, Vol. 04, Issue 02, 129-154
DOI: https://doi.org/10.59231/SARI7814

with maintaining constant infrastructure for incident response and auditing

❖ **Automation Potential:** Given the event-driven nature of serverless computing, the paper may discuss how this can automate certain aspects of incident response and auditing, potentially reducing human error and increasing efficiency

❖ **Challenges in Logging and Tracing:** The research might address the challenges in maintaining comprehensive logs and traces in serverless environments, which are crucial for both incident response and auditing

❖ **Security Considerations:** The unique security challenges posed by serverless architectures in the context of incident response and auditing, such as increased attack surfaces and the need for robust identity and access management. Integration with AI and Machine Learning

❖ **Paradigm Shift in Incident Response:** The findings could lead to a shift in how organizations approach incident response, moving towards more dynamic, event-driven models facilitated by serverless architectures.

❖ **Enhanced Auditing Capabilities**: The research might demonstrate how serverless computing can improve the speed and accuracy of auditing processes, potentially leading to more frequent and comprehensive audits and management.

❖ **Regulatory Considerations:** Influence regulatory bodies to update guidelines and standards for incident response and auditing to account for serverless architectures.

❖ **Education and Training:** The research could highlight the need for updated education and training programs for IT professionals, auditing professionals and those who are interested in this field to effectively manage incident response and auditing in serverless environments.

❖ **Cross-disciplinary Collaboration:** The need for increased collaboration between cloud computing experts, security professionals, and auditors to address the unique challenges of serverless architectures.

VII. RECOMMENDATIONS

➢ **Automation of Incident Response:** Implement automated workflows using serverless functions to respond to security incidents swiftly. This might include automatically quarantining affected resources or shutting down services when a threat is detected.

@2025 International Council for Education Research and Training
ISSN: 2959-1376
2025, Vol. 04, Issue 02, 129-154
DOI: https://doi.org/10.59231/SARI7814

- **Event-Driven Architecture**: Utilize an event-driven architecture to monitor system logs and alerts, triggering responses based on predefined thresholds or security events.
- **Cost Management**: Leverage the cost-effective nature of serverless computing by scaling resources in response to incident loads, ensuring that you have the computational power necessary without incurring high costs when incidents are low.
- **Improved Scalability**: Take advantage of the inherent scalability of serverless solutions to handle unforeseen surges in data or events during an incident.
- **Enhanced Logging and Monitoring:** Integrate comprehensive logging and monitoring capabilities to track incidents in real-time, allowing for better auditing and retrospective analysis.
- **Security Best Practices**: Follow security best practices for serverless architectures, such as minimizing permissions for functions, using environment variables for sensitive information, and employing strong authentication methods.
- **Data Privacy and Compliance**: Ensure that serverless designs comply with relevant regulations and standards, incorporating measures for data security and user privacy in the response architecture.
- **Training and Awareness**: Encourage ongoing training for teams involved in incident response to familiarize them with serverless technologies and their implications.

VIII. FUTURE RESEARCH

- **Integration with AI and Machine Learning:** Investigate how AI and machine learning algorithms can be integrated with serverless architectures to enhance incident detection, classification, and response times. Future studies could focus on developing predictive models that utilize serverless functions to analyze audit logs in real-time.
- **Security and Compliance Frameworks** Examine the implications of serverless computing on security and compliance in auditing contexts. Research could aim to define frameworks that ensure best practices for incident response within various regulatory environments (e.g., GDPR, HIPAA).
- **Performance Metrics and Benchmarks** Establish metrics to evaluate the performance of serverless computing environments in incident response. This could include response times, resource utilization,

@2025 International Council for Education Research and Training
ISSN: 2959-1376
2025, Vol. 04, Issue 02, 129-154
DOI: https://doi.org/10.59231/SARI7814

and scalability during incidents, compared to traditional computing models.

- **Cost-Benefit Analysis** Conduct cost-benefit analyses focusing on the financial implications of using serverless architectures for incident response in auditing. Studies could explore factors such as operational costs, efficiency improvements, and ROI for organizations adopting these technologies.
- **Case Studies and Real-World Applications** Encourage the development of case studies that document the implementation of serverless computing for incident response in various industries. Analyzing success stories and challenges could provide valuable insights for practitioners.
- **Governance and Best Practices** Research governance models specific to serverless architectures in incident response. Identifying best practices for managing serverless functions, including monitoring, logging, and auditing procedures, would be beneficial for organizations.
- **Interoperability with Existing Systems**

Investigate the interoperability of serverless functions with existing auditing frameworks and tools. Understanding how serverless computing can effectively integrate into current workflows is crucial for adoption.

- **User Training and Awareness** Focus on how to effectively prepare auditors and incident responders for leveraging serverless computing. Research could assess training programs or materials that promote awareness of serverless technologies in the context of incident response.
- **Longitudinal Studies on Impact** Conduct longitudinal studies to track the long-term impacts of serverless computing on incident response capabilities in auditing tasks. This can provide insights into trends, challenges, and improvements over time.

IX. CONCLUSION

Serverless computing is revolutionizing auditing for incident response. By automating evidence collection, containment, compliance checks, and incident tracking, serverless solutions offer real-time, cost-effective, and scalable responses to threats. These systems support auditability, regulatory compliance, and incident transparency, ensuring a defensible position in the face of regulatory reviews or legal proceedings. As serverless architectures continue to evolve, they will play a pivotal role in securing cloud-native infrastructures.

@2025 International Council for Education Research and Training
ISSN: 2959-1376
2025, Vol. 04, Issue 02, 129-154
DOI: https://doi.org/10.59231/SARI7814

REFERENCE

1. Serverless on AWS (Amazon Web Service). https://aws.amazon.com/serverless/
2. AWS security incident response user guide. https://docs.aws.amazon.com/security-ir/latest/userguide/security-incident-response-guide.html
3. Youngs, P., & King, M. B. (2002). Principal leadership for professional development to build school capacity. *Educational Administration Quarterly*, *38*(5), 643–670. https://doi.org/10.1177/0013161x02239642
4. Ganapathy, V. (2024). AI-Based risk assessments in Forensic Auditing: benefits, challenges and future implications. *Shodh Sari-An International Multidisciplinary Journal*, *03*(04), 100–128. https://doi.org/10.59231/sari7750
5. Serverless computing in Microsoft. *Azure*. https://azure.microsoft.com/en-in/resources/cloud-computing-dictionary/what-is-serverless-computing.
6. Google cloud serverless computing. Google. https://cloud.google.com/discover/what-is-serverless-computing
7. Youngs, P., & King, M. B. (2002). Principal leadership for professional development to build school capacity. *Educational Administration Quarterly*, *38*(5), 643–670. https://doi.org/10.1177/0013161x02239642
8. Gupta, T. (2025). Redefining Employment Relations: Understanding Recent Advances and Innovative ideas Guiding the New Era of human resource development. *Shodh Sari-An International Multidisciplinary Journal*, *04*(01), 95–110. https://doi.org/10.59231/sari7781
9. Fadare, A. A. (2024). Christian Ethical response to the challenges of digital financial borrowing in Nigeria. *Shodh Sari-An International Multidisciplinary Journal*, *03*(04), 194–207. https://doi.org/10.59231/sari7756
10. *Serverless computing – AWS lambda*, *4422*(3)!651612776783!e!!g!!aws%20lambda!1982822969?!143940519541. https://aws.amazon.com/pm/lambda/?gclid=CjwKCAiAmrS7BhBJEiwAei59i0MSJhZwws4SZn_vMr_46xGyjuvqVsl5Zo5

@2025 International Council for Education Research and Training
ISSN: 2959-1376
2025, Vol. 04, Issue 02, 129-154
DOI: https://doi.org/10.59231/SARI7814

XVzy7V4NrlF4jCzaxJhoCWqUQAvD_BwE&trk=5cc83e4b-8a6e-4976-92ff-7a6198f2fe76&sc_channel=ps&ef_id=CjwKCAiAmrS7BhBJEiwAei59i0MSJhZwws4SZn_vMr_46xGyjuvqVsl5Zo5XVzy7V4NrlF4jCzaxJhoCWqUQAvD_BwE:G:s&s_kwcid=AL

11. *What is amazon, S3*. https://docs.aws.amazon.com/AmazonS3/latest/userguide/Welcome.html
12. Tiwari, A. K. (2024). The art of writing a research paper that is internationally acceptable, according to a reviewer. *Shodh Sari-An International Multidisciplinary Journal*, *03*(03), 379–392. https://doi.org/10.59231/sari7740
13. Kaur, M., & Sharma, J. (2023). The role of digital literacy to promote the gender equality. *Shodh Sari-An International Multidisciplinary Journal*, *02*(04), 315–327. https://doi.org/10.59231/sari7642

Received on Dec 26, 2024
Accepted on Feb 09, 2025
Published on April 01, 2025