C-Level Guidance to Securing Serverless Architectures

cloud security alliance®

The permanent and official location for Cloud Security Alliance Serverless Computing research is https://cloudsecurityalliance.org/research/working-groups/serverless/

Acknowledgments

Editors / Working Group Co-Chairs

Aradhna Chetal
Vishwas Manral

Contributing Authors

Marina Bregkou
Shahna Campbell
Ricardo Ferreira
Vani Murthy
John Wrobel

CSA Analyst

Marina Bregkou

Reviewers

Joseph Arcelo
Shawn Clark
Tuhin Goswami
Tim Sedlack
Peter van Heijk
Paul Willy

CSA Global Staff

Claire Lehnert

Table of Contents

Acknowledgments 3
1. Introduction - Executive Summary 5
Purpose and Scope 5
Audience 6
2. Business benefit 6
2.1 Innovation 6
2.2 Agility 7
2.3 CapEx/OpEx 7
2.4 Speed to Market 7
2.5 Automation 8
3. Security and risk management for serverless 9
3.1 Elevation of inherited security with Serverless 9
3.2 Serverless on the CIA security Triad 10
3.3 Reshaping the CIO and CISO relationship 10
3.4 Threat Model and Serverless Best Practices 11
4. Conclusion 14
5. References 15
Appendix: Acronyms 16
Appendix 2: Glossary 17

1. Introduction - Executive Summary

Serverless computing enables developers to develop and deploy faster, allowing a more effective way to move to Cloud-native services without managing infrastructures like container clusters or virtual machines. As businesses work to bring technology value to market faster, serverless platforms are gaining adoption with developers.

Like any emerging technology, Serverless brings with it a variety of cyber risks. This paper in its first part covers the business benefits of Serverless architecture like agility, cost, speed to market etc. which are all correlated. The second part of the paper focuses on security for serverless applications, describing the industry-wide best practices and recommendations. In its conclusion, it summarizes how executive management should look at serverless architectures and what factors they should consider when adopting them.

The information is intended for readers who are getting introduced to serverless computing and need to understand its business and security implications.

Purpose and Scope

The purpose of this document is to provide a high level business overview of Serverless computing, along with the risks and the security concerns when implementing a secure serverless computing solution.

Two players are involved in a Serverless service:

- The Service/Platform Provider - the provider of the serverless platform on which serverless applications are built.
- The Application Owner - the user of the serverless solution/service whose applications run on the platform.

Under a serverless solution/service (or serverless platform), a service provider offers compute resources that are elastically auto-allocated to serve the needs of different customers. This way, customers are charged based on usage and not on a fixed amount of bandwidth or number of servers.

The designation 'serverless' reflects the fact that although physical servers are still used, a customer does not need to be in charge or aware of them, as they are only the provider's responsibility. In other words, although there are technical executions on server hardware, only the cloud service provider is responsible for the compilation of the functions or the successful delivery of the service provided.

Examples of serverless services include Functions as a Service and Database as a Service.

The scope of this document is limited at the perspective of implemented workloads on top of Serverless platforms offered by platform providers.

The primary goal is to present and promote serverless computing as a secure cloud computing execution model.

For a more detailed overview of Serverless Computing, the paper "How to design a Secure Serverless architecture"[1] by Cloud Security Alliance is recommended.

Audience

The intended audience of this document are Chief Information Security Officers (CISOs), Chief Information Officers (CIOs), Security Professionals, application and Security Engineers, risk management professionals, and other security business functions like product managers, marketing managers and business development managers, interested in serverless computing and its security.

Owing to the constantly changing nature of technologies in the serverless space, readers are encouraged to take advantage of the other resources, including those listed in this document, for current and more detailed information.

2. Business benefit

Serverless computing offers several business benefits over traditional cloud-based or server-centric infrastructure. A cloud-native, serverless architecture benefits consumers in the following areas:

2.1 Innovation

One of the aspects organization leaders are pursuing is the speed of innovation that serverless and Cloud platforms bring to their existing products and services. The latest report from PwC [PwC, 2021] shows the CIO measuring ROI by faster innovation and delivery of services and applications brought by cloud platforms and their capabilities.

Serverless supports the migration and modernization of workloads in a Cloud-native way, but more importantly, it allows the transformation of an organization into a modular paradigm. This organizational shift is required as technology innovation requires cross-collaboration, and serverless enforces the self-service modular approach.

For example, an organization can rearrange different business units for efficient, reusable, and innovative outcomes. Serverless is a way an organization can achieve modularity, cost reduction, and reduced complexity of creating, scaling, and managing cloud infrastructures. It helps developers deploy applications rapidly, accelerating the time-to-market of new products and introducing features (including security). It's a new way of application design that organizations should use to increase competitiveness.

1 How to design a Secure Serverless architecture: https://cloudsecurityalliance.org/artifacts/serverless-computing-security-in-2021/

A real-world example is PingAn [MITSloan, 2021], which reorganized itself to provide remote tools to its workforce during the pandemic and afterward quickly reshaped to use those tools for customers only made possible by its innovative cross-collaboration and modular paradigm very much aligned to how microservices and modular approaches work. [Mix, 2011].

2.2 Agility

An aspect of serverless is that it enables a modular organization, making it easier for organizations to adapt to market events and being quicker to adapt and/or respond to unforeseen circumstances.

Since Serverless is built on Cloud-Native paradigms, Application programming interfaces (APIs) are key to those architectures. Using them internally and externally makes the organizations shift very quickly from API consumers to producers generating revenue and providing niche services and applications to the market.

Today, organizations fueled by the post-pandemic digitization are investing in key areas that would allow them to become more competitive through technology to achieve business objectives. One key aspect is using serverless to modernize the digitization of business processes as they foster activities such as DevOps and cross team collaboration which enables the organization to break silos and drive transformation throughout the organization.

2.3 CapEx/OpEx

One key aspect of cloud computing and transformation was the shift from Capital Expenditure (CapEx) to Operational Expenditure (OpEx). With serverless it provides further cost saving benefits by removing the traditional costs for running containers and managing containerised infrastructure.

The nature of sub second billing of serverless coupled with the event driven architectures allows organizations to optimize spending as a function of scalability - only requiring execution on demand and paying for operations that are necessary only when they run (or "Pay as you go/ pay per transaction").

As with any architecture, organization mileage may vary, and it is important to understand the trade-offs. For example, BBVA labs did an assessment on the economics of serverless that shows the sweet spot of the high cost effectiveness accordingly [BBVA, 2020]. Additionally, researchers from UC Berkeley [Berkeley EECS, 2019] also did a study quantifying the value of serverless to customers and reached the conclusion that serverless is still under-adopted and that many cloud customers today would benefit from a serverless model.

2.4 Speed to Market

Organizations or Cloud Service Customers (CSC) leveraging serverless architecture can create new applications and operate them without having to procure, harden and maintain the underlying infrastructure and platform.

The Cloud Service Provider (CSP) and CSC follow a shared responsibility model where the CSP has the responsibility for security, the infrastructure and platform and the CSC has the responsibility of their application and the data.

By utilizing fully managed serverless services, enhanced security and faster development are both possible, allowing for faster development and deployment cycles therefore enabling CSC to benefit from accelerated "Speed to Market".

Serverless computing increases scalability, agility and speed of delivery in a few different ways, as provided below:

1. Underlying hardware infrastructure is invisible
2. Fewer and shorter downtimes
3. Faster releases
4. Cost reductions

1. **Underlying infrastructure is invisible**
   The underlying infrastructure is abstracted and therefore invisible to the end user since it is managed by the cloud service provider (CSP). The CSP also takes care of provisioning, maintaining and the management of the servers. Thus, the user can focus on development and execution of applications.
2. **Fewer and shorter downtimes**
   There are shorter and fewer downtimes because Cloud Service Provider (CSP) can isolate failures due to improved availability and redundancy across multiple zones. The CSP also handles the patching and maintenance.
3. **Faster releases**
   The microservice architecture and DevOps pipeline enable consumers to deliver releases faster.
4. **Cost reductions**
   These organizations can spin up or down, causing applications to scale as quickly as needed based on utilization and demand, with no delays due to limited resources, therefore enabling "Speed to Market"
   The Serverless functions can be instantiated on an as needed basis, based on utilization and demand, alleviating the need for Cloud Service Customer (CSC) to manage applications or the resources. CSCs rely solely on the Cloud Service Provider (CSP) to deliver the capacity at scale. The total cost of development, maintenance, and infrastructure is shared between both CSP and CSC with CSC ending up with significant cost savings when compared to the cost if they were solely responsible.

2.5 Automation

For several years now, among successful enterprises, it has been a common understanding that automation and orchestration are key tools in improving cost effectiveness, speed to market, and security. Serverless computing services are one such tool that can help organizations achieve both operational and security benefits by automating certain workloads. This is especially true when serverless computing tools are used in conjunction with other automation tools, for example.

i. **Operational uses**
Serverless computing services can be used for operational benefits by themselves, or by supporting an automated workflow. E.g.:
   - Provision new infrastructure
     - As part of a self-service tool's workflow
     - Automatically based on certain conditions (i.e. scale an application up during a period of high load, or down during low load)
   - Integrating third party services
   - Power backend APIs (especially if they are infrequently used)
   - Performing one-time tasks such as replication of data for a migration or scheduled backups
   - Support service-based application architectures
   - Event ingestion and integration.

ii. **Security uses**
Serverless computing models also provide some benefits for security via automation, both directly and indirectly. Here are a few examples of this:
   - Remove the potential for human error in sensitive, administrative tasks (i.e. securely configuring services as they are deployed).
   - Gather information for logging purposes on a set schedule
   - Perform regular configuration audits.
   - Improved visibility into application behavior via more transparency around data flow between microservices and improved ability to debug specific application functions.

3. Security and risk management for serverless

3.1 Elevation of inherited security with Serverless

A major advantage of serverless architecture is that the organization and the Cloud Service Provider(CSP) share the responsibility for handling security and compliance. The CSP handles security of the cloud which can include the serverless architecture and infrastructural computing components such as hardware, software, networking and facilities. The customer handles securing the data they put in the cloud. This allows organizations to focus more time and resources on advancing their applications versus managing infrastructure, while also decreasing overhead costs of managing and maintaining servers.

Responsibility of the customer and CSP vary based on their deployment. There is Infrastructure-as-a-Service(IaaS), Platform-as-a-Service(PaaS), and Software-as-a-Service(SaaS) with SaaS placing the most responsibility on the CSP where they manage infrastructure and applications while the customer manages securing data and access.

The common responsibilities of the customers are:

- User identity management and access controls of service systems
- Data security
- Security management and control or hardware, software, application systems, and devices

[CSA, 2019]

By managing the underlying infrastructure CSP providers handle securing the underlying:

- Hardware
- Software
- Operating System
- Runtime Security
- Patch Management

Concluding, by implementing a serverless model and allowing the CSP to secure underlying infrastructure, organizations are able to increase agility, reduce costs and decrease operational overhead.

3.2 Serverless on the CIA security Triad

There are four main principles on how to secure the serverless environment in order to achieve confidentiality, integrity and availability (CIA), otherwise known as the CIA security triad.

1. Apply the principle of least privilege:
   Allocate only the minimum privileges required per occasion.
2. Minimisation of attack surface:
   Minimisation of the attack surface is achieved when only the functions that use the HyperText Transfer Protocol (HTTP/s) or API requests are exposed and nothing else.
3. Layering security measures to achieve defense in depth:
   Security measures need to be layered in order to achieve defense in depth so that, if the first layer of defense fails, the subsequent layers will protect the system. Network policies, application event analysis, and security standards exist for this purpose.
4. Encrypt everything:
   Encrypt everything, even environment variables at rest, by default. [TeamForm, 2018]

3.3 Reshaping the CIO and CISO relationship

The roles of the CIOs and CISOs may change in many organizations based on the organizational needs and cultural requirements.

The CIO is no longer an operational executive but an orchestration executive, as nowadays, there is no business strategy in organizations that does not involve technology [CIO DIVE, 2021].

CISOs on the other hand, except from technology specialists have also become business enablers.

Security is becoming more of a shared business responsibility and many aspects of IT management now reside outside the CIO's and CISO's reporting structure. In these cases, the CISO and CIO should collaborate to ensure that security is a part of the "checklist" of best practices and metrics tracking across the organization.

Collaboration between CIOs and CISOs is becoming critical to the business outcomes, especially because of the need to:

> - Develop security measures collaboratively
> - Build an integrated security architecture
> - Automate workflows and threat intelligence
> - Stay on top of business managed IT.

[Fortinet, 2020]

The Shared Responsibility model that characterizes the cloud, simplifies infrastructure maintenance but also affects the security approaches. Thus, its implications need to be recognized by CIO's and security experts in order to incorporate stringent measures to ensure that the organization's IT infrastructure is safeguarded at all levels.

Depending on the organization's cloud infrastructure, the responsibility of security lies with either the "Customer", which is the organization using the services, or the "Cloud Service Provider (CSP)", which is the entity that provides these services.

The answer however isn't always clear-cut, so the job of CIOs is to make sure there are no security blind spots, where one party thinks that it is the other party taking care of security.
A customer's IT landscape is only as secure as its least guarded endpoint.
The best practice would be to schedule regular security audits across all areas, in order to plug vulnerabilities before any mishap. [Cymune, 2021]

Serverless functions are one of the tools better fitted for a present and a future of resiliency, in a world of ransomware and other possible threats. Threats and challenges that the fragility-fighting CIO and the resilient-striving CISO [TheNewStack, 2020] are called to encounter and manage through the strategy architecture they build for the organization.

3.4 Threat Model and Serverless Best Practices

Threat Model of Serverless

Due to their innovative design, serverless applications present unique security challenges. The evolution of technology is inevitably followed by evolution of threat actors to find and exploit vulnerabilities in new technologies. For this reason, new technologies need to be adopted carefully and proper diligence is required. Serverless applications often cross many traditional trust boundaries that legacy security controls are designed around. As a result, these traditional security approaches are sometimes ineffective at securing serverless applications on their own. This is often compounded by the shifting of control of some functionality from developers and support teams,

to Cloud Service Providers that is often inherent in the use of serverless cloud technologies. This often means enterprises have less opportunities to embed security controls within an application's execution flow.

We identify three key threat areas for serverless applications:

- Application Owner Setup Phase Threats
  - Threats that stem from actions taken by application owners when setting up infrastructure to host an application.
- Application Owner Deployment Phase Threats
  - Threats that stem from actions taken by application owners during the process of deploying their applications.
- Service Provider Conduct Threats
  - Threats that stem from actions taken by the entity providing the service and/or infrastructure to application owners.

Each of these categories contains both threats that are unique to serverless, and threats that are not unique to serverless but are aggravated by the architecture of serverless applications. Please see CSA's *How to Design a Secure Serverless Architecture*[2] for a complete threat model with additional details.

Security design, controls, and best practices

When used properly, serverless capabilities can provide some security benefits when compared to traditional applications. These benefits include, but are not limited to: stateless and ephemeral components, inherent data compartmentalization, and simplified patching in some cases.

The major design challenges and corresponding best practices for addressing organizational security requirements include:

Challenge: Serverless functions inherently have public-facing egress access
Best Practices:

- Apply network policies to limit outbound connectivity from functions using Virtual Private Cloud services.
- Apply service or resource policies to limit the endpoints that can access your function, and thus reduce exfiltration paths.

Challenge: Inadequate function logging and monitoring
Best Practices:

- Use a structured logging format to make logs easier to parse by other tools.
- Integrate logs into a centralized logging and monitoring solution.
- Discover and monitor all APIs or endpoints that are exposed.
- Implement runtime detection and response at Platform layer for policy violations etc.[3]

2 Document can be downloaded free: https://cloudsecurityalliance.org/artifacts/serverless-computing-security-in-2021/
3 'How to Design a Secure Serverless Architecture', Section 6.2 - Controls for FaaS

Challenge: Insecure serverless deployment configuration

Best Practices:

- Security settings and policies need to be defined and reliance on default configurations should be minimized.
- Configurations should undergo security testing prior to production deployment

Challenge: Insecurely storing application secrets

Best Practices:

- Have your serverless code deployed to a confidential computing instance to protect secrets while in use.
- Determine if default or managed encryption options from the provider meet your requirements. If not, implement application-layer encryption as a compensating control.

Challenge: Insecure management of third-party dependencies

Best Practices:

- Perform source composition analysis of any third party libraries used in your application (tools are available to do this in an automated fashion).
- Use monitoring solutions to identify libraries at run time that are either vulnerable, or not used, and remove them if possible.

When dealing specifically with applications based on Function as a Service(FaaS) technologies, it is especially important to address the following control categories:

- Platform service provider API and management controls & integrations.
- CI/CD Pipeline security controls (In the 'How to Design a Secure Serverless Architecture' paper are included controls for FaaS also).
- Identity and Access Management,
- Platform and Runtime layer detection controls.[4]

4 For further details on mitigations please consult the 'How to Design a Secure Serverless Architecture' paper by CSA, sections 5 and 6, page 18 - 61.

4. Conclusion

Serverless computing changes security responsibilities, While security of serverless infrastructure is the responsibility of CSPs, application and data security is still the responsibility of cloud customers. Serverless computing however, does also have the risks inherent in both application services and multi-tenant resource sharing in a cloud environment.

Cloud functions also need fine-grained configuration like access to private keys, storage objects, and access to other resources. Hence, it is important that security policies are converted from existing applications while secure API integrations are adapted for dynamic use in cloud functions.

A function may have to delegate security privileges to another cloud function or other cloud services. Hence, context based access control mechanisms are important in serverless architectures.

Distributed management of security capabilities is exacerbated in a cloud functions implementation. At the system level, fine-grained security isolation for each function is critical. There can be sensitive applications which may require protection against co-residency attacks, e.g. Spectre. Meltdown also makes reserving a whole core or even a whole physical machine, attractive for users. Cloud providers may offer a premium option for customers to launch functions on physical hosts dedicated exclusively to their use.

Cloud functions are widely distributed, the network transmissions can leak more sensitive information to a network attacker in the cloud (e.g. an employee), even if the payload is encrypted end-to-end. The distributed nature of serverless applications exacerbates this security exposure.

Finally, serverless adoption is bound to grow and become mainstream due to the ease of improved developer efficiencies and the reduced management of infrastructure and other dependencies. As the use of serverless computing increases, executives need awareness of the opportunities and challenges inherent to these technologies. Organizations move to serverless in order to take advantage of abstraction, providing high level programming and the fine-grained isolation of cloud functions.

5. References

[TeamForm, 2018]	TeamForm. (2018). Steps to secure AWS Serverless — Lambda (part 1). https://medium.com/orchestrated/steps-to-secure-aws-serverless-lambda-part-1-a6e5d1b05f45
[Fortinet, 2020]	FORTINET. (2020). CIO - CISO Relationship Must Change Per New Report. https://www.fortinet.com/blog/ciso-collective/as-the-cio-role-shifts-cio-ciso-relationship-must-also-change-per-new-report
[BBVA, 2020]	BBVA. (2020). Rodriguez, A., BBVA Labs, Alvarez, F. Diaz Lopez, G., Evgeniev, M., Horillo, P. Economics of 'Serverless'. https://www.bbva.com/en/economics-of-serverless/
[Berkeley EECS, 2019]	UC Berkeley. EECS Department. (2019). Technical Report No. UCB/EECS-2019-3. Cloud Programming Simplified: A Berkeley View on Serverless Computing. https://www2.eecs.berkeley.edu/Pubs/TechRpts/2019/EECS-2019-3.html
[CIO DIVE, 2021]	CIO DIVE. (2021). Security disconnect: Why the CISO role is evolving. https://www.ciodive.com/news/gartner-ciso-role-evolution-security-leader/610583/
[CSA, 2019]	Cloud Security Alliance. (2019). How to Share the Security Responsibility Between the CSP and Customer. https://cloudsecurityalliance.org/blog/2019/09/05/how-to-share-the-security-responsibility-between-the-csp-and-customer/
[CSA, 2021]	Cloud Security Alliance. (2021). How to Design a Secure Serverless Architecture. https://cloudsecurityalliance.org/artifacts/serverless-computing-security-in-2021/
[Cymune, 2021]	Cymune. (2021). Shared Responsibility on Cloud. https://www.cymune.com/blog-details/Shared-Responsibility-on-Cloud
[GoogleCloud, 2018]	Google Cloud. (2018). Linton, M., O'Connor, M. Answering your questions about "Meltdown" and "Spectre". https://cloud.google.com/blog/topics/inside-google-cloud/answering-your-questions-about-meltdown-and-spectre
[Mix, 2011]	Management Innovation eXchange. (2011). Hashim, E. Hack: Modular Organization 1.1; Enterprising With The Flow [updated]. https://www.managementexchange.com/hack/modular-organization
[MITSloan, 2021]	MITSloan. (2021). Greven M., Yu H., Shan J. Why Companies Must Embrace Microservices and Modular Thinking. https://sloanreview.mit.edu/article/why-companies-must-embrace-microservices-and-modular-thinking/

[PwC, 2021]	PwC. (2021). CIOs and technology leaders. https://www.pwc.com/us/en/tech-effect/cloud/cloud-business-survey/cio-technology-leaders.html
[TheNewStack, 2020]	THENEWSTACK. (2020). KubeCon EU: Cloud Native Security Tools for the Next Decade Will Focus on Recovery. https://thenewstack.io/kubecon-eu-cloud-native-security-tools-for-the-next-decade-will-focus-on-recovery/

Appendix: Acronyms

Acronyms and abbreviations used in this paper are defined below.

API	Application programming interface
CapEx	Capital expenditure
CIA	Confidentiality, Integrity, Availability
CI/CD	Continuous integration/ continuous delivery
CIO	Chief Information Officer
CISO	Chief Information Security Officer
CSP	Cloud Service Provider
FaaS	Function as a Service
HTTP/s	HyperText Transfer Protocol (Secure)
IT	Information Technology
OpEx	Operational Expenditure
PWC	PricewaterhouseCoopers
ROI	Return on Investment
UC	University of California

Appendix 2: Glossary

CI/CD
CI/CD refers to continuous delivery and/or continuous deployment.

This is a process, often visualized as a pipeline, that involves adding a high degree of ongoing automation and continuous monitoring to app development.

Continuous delivery usually means a developer's changes to an application are automatically bug tested and uploaded to a repository (like GitHub or a container registry), where they can then be deployed to a live production environment by the operations team. This helps with the problem of poor visibility and communication between dev and business teams. As such, continuous delivery ensures that it takes minimal effort to deploy new code.

Continuous deployment (CD) can refer to automatically releasing a developer's changes from the repository to production, where it is usable by customers. It addresses the problem of overloading operations teams with manual processes that slow down app delivery. It builds on the benefits of continuous delivery by automating the next stage in the pipeline.

Meltdown
Security flaws caused by "speculative execution," a technique used by most modern processors (CPUs) to optimize performance. Most vendors refer to the terms 'Meltdown' and 'Spectre' by Common Vulnerabilities and Exposures aka "CVE" labels, which are an industry standard way of identifying vulnerabilities. [Google Cloud, 2018]

Serverless Architecture
Serverless architecture (also known as *serverless computing or function as a service*, **FaaS**) is a software design pattern where applications are hosted by a third-party service, eliminating the need for server software and hardware management by the developer. Applications are broken up into individual functions that can be invoked and scaled individually.

Serverless Platform
Serverless platforms allow business owners to regulate the number of used computing resources. If the application handles small workloads, the organization doesn't have to overpay for excessive server space. As soon as there is a need for an increase in computing power, the platform will provide resources.

Serverless platforms take full responsibility for server performance.