ISSN 2347 - 3983
Volume 8. No. 2, February 2020
International Journal of Emerging Trends in Engineering Research
Available Online at http://www.warse.org/IJETER/static/pdf/file/ijeter43822020.pdf
https://doi.org/10.30534/ijeter/2020/43822020

Security Issues in Serverless Computing Architecture

Sanaa Sharaf
Department of Computer Science, King Abdulaziz University
Jeddah, Saudi Arabia
ssharaf@kau.edu.sa

ABSTRACT

Serverless computing allows a company to run its backend services on an as-used basis. Precisely, could providers are offering business the chance to pay for cloud computing based on their computation needs without the need to reserve a certain number of servers or secure a fixed amount of bandwidth. While offering flexibility through paying for computing power rather than fixed server space and bandwidth, serverless computing presents unique security challenges and issues not characteristic of classic cloud computing. The purpose of this paper is to bring to light these security issues and challenges in addition to developing a proposed framework to solve it.

Key words: Serverless Computing, Cloud Computing, Security Issues, DevOps

1. INTRODUCTION

As the internet and information communication technology has continued to evolve, the idea of computing has evolved with equal measure. During the early days of the internet age, organizations had to own the computer infrastructure needed to support back-end services. A great drawback of the approach is incurring huge IT infrastructure barely utilized to full capacity [1]. As the speed of the internet increased, so did grow the idea of cloud computing. Cloud computing entails an organization renting a fixed amount of server space that could be linked to an organization's back-end IT infrastructure. Cloud computing enabled many organizations to move organizational and employee functions online. This has enabled organizations to allow their employees to work remotely [1]. Secondly, cloud computing reduced IT costs significantly as there was no need to invest heavily in the local area and wide area networks needed to connect different physical servers used initially. Alike, organizations could scale their computing needs more effectively and efficiently.

While cloud computing has delivered a lot of value and opportunities to companies in increasingly interconnected workspaces, they have not eliminated the costs associated with paying for servers that an organization may not use from time to time. While cloud computing eliminated the need to invest in server rooms, associated security and support infrastructure, it has not been able to address the issue of excess server capacity mainly for a seasonal business. Seasonal businesses will have different computing needs at different times [1]. However, current cloud computing models requires a company to rent a fixed amount of server space. This problem has led to the emergence of serverless computing.

Serverless computing allows a company to run its backend services on an as-used basis. Precisely, could providers such as Amazon are offering business the chance to pay for cloud computing based on their computation needs without the need to reserve a certain number of servers or secure a fixed amount of bandwidth [1]. While offering flexibility through paying for computing power rather than fixed server space and bandwidth, serverless computing presents unique security challenges and issues not characteristic of classic cloud computing. The purpose of this paper is to leverage a systematic review of the literature to bring to light these security issues and challenges in addition to developing a model to solve the latter.

2. SERVERLESS COMPUTING

2.1 Background

Serverless computing is an emerging form of cloud computing where the cloud provider leverages dynamic systems to allocate and provision servers. Precisely McGrath and Brenner suggest that serverless computing allows running applications in stateless compute containers that are only triggered when there is an event from the client [2]. Consequently, rather than pre-purchasing computing capacity, serverless computing charges business depending on the number of executions made on the cloud rather than pay for a fixed amount of computing space or bandwidth. McGrath and Brenner add that the core purpose of serverless computing is to eliminate the need for businesses to think about servers, security, implementing, monitoring, and debugging [2]. Rather, the cloud provider can focus on server-side problems and help save businesses a lot of time and cost in IT costs needed to manage physical and cloud servers bought by the company. Consequently, serverless computing helps the business focus on what are the business

goals and objectives the business application running in the cloud is meeting. Adzic and Chatley observe that while serverless computing is increasingly carving its niche in the cloud computing world, it is not the answer to all the needs of the business [3]. Rather, there are some applications where paying for a fixed computing capacity may offer better business value as opposed to paying for the executions. Adzic and Chatley add that serverless computing is increasingly replacing the classic cloud computing for seasonal businesses [3]. Seasonal businesses do not need fixed computing capacity and power since customer numbers vary significantly. During the seasons where the computing need is high, the companies can pay more while saving on cloud infrastructure management, which the cloud provider manages. The serverless approach, consequently, is expected to free up more productive efforts for businesses by allowing them to focus much of the attention to the application themselves as opposed to managing cloud
infrastructure.

2.2 Benefits

Ease of deployment is the main pro of serverless computing apparent in the existing literature. Using the classic cloud computing model, a company would need to determine the computing power required, proceed to determine the best cloud providers for the required capacity, and then make the payments to secure the computing space and bandwidth [4]. The approach made it hard and tedious to deploy business applications. With serverless computing, however, a business does not need to worry about the computing needs of the application. Rather, the business application can be spin up fast in a matter of hours. In [4] the authors suggest that the latter is possible since the company does not have to worry about infrastructure. Rather, an application can be deployed, tested, debugged, and improve by releasing it immediately to the cloud providers. The latter is possible since scalability is automatic in serverless computing eliminating the need for businesses to worry about provisioning needed.

Secondly, low IT costs are a defining benefit of serverless computing. While cloud computing eliminates the need for building a physical server, one needs to build a server in the cloud infrastructure. Building servers enables a company to have different servers for different aspects of the business depending on the computing needs. Serverless computing eliminates the need to create and manage servers and databases. As authors in [4] point out, outsourcing the most recurrent and time-consuming aspects of cloud computing to the cloud providers reduces the human resources costs by freeing up more time to focus on the server-side code. Optimizing the server-side code improves efficiency leading to less computing power. The limited human resources need, and less computing power delivers tremendous cost savings to the business.

Further, serverless computing offers better scalability on than other cloud computing. Precisely, Sewak and Singh observe that serverless computing is very ideal for small and medium enterprises where the need for scalability is difficult to plan [5]. When a business is establishing, the actual server needs are difficult to fathom. If a business chooses to pay for a fixed amount of computing, it could find itself with a lot of wasted computing power if the business fails to grow rapidly. With serverless computing, however, there will be no need to consider server scalability as the cloud provider manages the computing load dynamically [5]. Dynamic allocation of computing power enables businesses to scale operations smoothly if a business grows faster than expected without worrying about the ability to stage changes needed to support a fast-growing business.

Lastly, serverless computing improves organizational flexibility. As noted by Nastic et al., the ease of implementing a business application over a serverless architecture allows a company to see the end results in a short amount of time [6]. In the modern competing world, technology is changing extremely fast requiring organizational ambidexterity to keep up with the pace of innovation. Most of the innovation remains restricted to information technology as most business functions have increasingly moved to over the internet. When a business needs to set up a server over the cloud, it needs to take weeks to months before testing any in-house development efforts. With a serverless computing architecture, however, the organization gets more flexibility in testing applications and systems needed to improve customer purchase process or streamline internal operations. The flexibility comes in handy where situations need pivoting due to the need to restructure either an application or the business [6]. With traditional cloud computing, a business that is restricting has no chance to pivot since the computing power and space is paid for years or months ahead. During the restructuring, the capacity goes to waste leading to restricting costs. With a serverless computing architecture, however, a restructuring business will pay less or nothing at in computing executions since there will be no operations. If the restructuring is on the software to improve efficiency, a business operating on the serverless computing architecture will have more flexibility to test and optimize a software before making the decision whether to use the fixed cloud computing space or choose the serverless computing path.

2.3 Leading Providers of Serverless Computing

Serverless computing providers have emerged over the years. Amazon, Microsoft, and Google are the leading providers in the space. Precisely, Amazon has the **AWS Lambda**. Lambda acts as a computer service requiring a user to embed the service in the code. The user will pay Amazon on the event the code runs as opposed to the classic approach where Amazon requires a company to pay for server space in their

AWS platform. Microsoft delivers serverless computing using the **Microsoft Azure** platform. Azure is designed for cost-sensitive cloud users. It allows for building, testing, deployment, and management of applications through events managed servers. Consequently, a business is able to add subscriptions to their account as the computing power and needs changes. Lastly, Google offers serverless computing through the **Google Cloud Functions,** a service the company launched in 2017. Its subscription is based on triggered events where computing capacity is availed as per the user requests and needs [7].

2.4 Serverless vs Classic Cloud Computing

Serverless cloud computing and the current model of cloud computing differ on execution. However, the two models seek to achieve the same objective. Rather than have all organizations focus on server management and deployment, cloud computing eliminates the need for physical hardware and associated maintenance by allowing companies to rent computing space and bandwidth from a cloud provider. Serverless computers seek to extend the ability to cloud computing to allows businesses to focus on the code only and pay for computing power use only [2]. Precisely, serverless computing allows billing of companies in the cloud depending on the execution of tasks and subsequent computing resources used.

3. SECURITY ISSUES

Challenges affecting serverless computing can be categorized as security issues and organizational challenges. Bureaucracy and legal issues and lack of in-house expertise are the main hindrances to the rapid deployment of serverless computing [8]. Precisely, most organizations have built their IT expertise around managing servers traditionally and through cloud computing. Consequently, supporting and facilitating change through serverless computing must pass through the organizational red tape that is in favor of the current model of renting server space [8]. Secondly, the lack of in-house expertise needed to support serverless computing is a key organizational challenge facing technology. Precisely, serverless computing requires DevOps skills since it seeks to combine software development and information technology operations. Since IT and software development are often independent in most organizations making it harder to appreciate the full potential of serverless computing.

From a security standpoint, three key main security issues affecting serverless computing stand out. First, the possibility of function event data injection is high. Serverless functions operate on event requests form a third-party application. If such an application has an attacker malware, it could affect the whole cloud infrastructure [1][9]. Secondly, insecure serverless deployment configuration could affect the whole system. Cloud providers have a variety of configurations for deployment to allow adaptation to the unique needs of the customer. These different configurations introduce vulnerabilities that may compromise the underlying infrastructure [1]. Lastly, authors in [1] suggest that third-party dependencies increase the security risk for cloud providers in serverless computing industry. Third-party dependencies vulnerabilities are not unique to serverless computing service providers. However, the risks are magnified in serverless systems due to complexity that arises in deploying cloud services on per demand basis. Serverless technology requires dynamic allocation of computer resources. Consequently, the vulnerabilities of a third party are not limited to a certain server or account in a similar manner the current cloud computing model operates. Rather, the distributed nature of the model where each customer is available the free resources based on their needs makes it hard to reduce third party vulnerabilities.

4. METHODOLOGY

The study is based on a systematic review of the literature to find out the best approaches to address the security and organizational challenges affecting serverless computing as identified in the literature review. Precisely, the choice of a systematic review of literature is informed by the limited amount of studies into the serverless computing field. Since the field, if relatively nascent, a theoretical framework needs to be created to aid in further investigation into the subject.

Following the identification of the desired literature, a content analysis approach is used to identify the underlying main themes in the reviewed literature. The main themes are summarized in a table detailing the main theme, associated sub-themes, and several references to the theme in the reviewed literature. The outcome of the analysis is then used to propose a framework of dealing with issues and security challenges identified in the literature review.

5. RESULTS AND DISCSSION

In this study, possible solutions to challenges facing serverless computing are categorized into either organizational solution or security solutions affecting the adoption of the technology. It is apparent from Table 1 that organizational challenges are the biggest barriers to the adoption of serverless technology. This is evident from the number of studies that bring to light a range solution that would support a culture change needed to increase an organization's capability to adapt serverless capability.

Promoting collaboration between development and operations emerges as the most proposed solution to addressing the issue of lack of capability needed to enable serverless computing in an organization. Most of the reviewed studies on the organizational challenges to the adoption and implantation of serverless technology proposed the need for collaboration between IT and software development. This follows the findings in both [1], [8] which

established that lack of DevOps was a key hindrance to the adoption of serverless technology. DevOps entails a set of practices and processes that enable highly interlinked software development and IT team. Authors in [1], [8] asserted that most IT and software development function as independent siloes in an organization. Serverless technology requires a combination of both skills, however, for an organization to realize the full benefits. Collaboration between the IT department and the software development team, which would otherwise operate like independent siloes, helps create the DevOps practices needed to support serverless technology.

Secondly, existing research proposes a parallel implementation of serverless technology alongside existing microservices or monolith-based applications. Upcoming software development companies are leading the way in ascribing to the idea of NoOps as envisioned in serverless computing. NoOps concept views an organizational environment where IT can become so automated and abstracted form an organization to the extent that there is no need to have a dedicated team to manage IT infrastructure in-house. Essentially, new companies have less bureaucracy and fear to inhibit the mover to serverless technology. However, with parallel experimentation, the resistance to relinquishing control of IT infrastructure to cloud providers could reduce as established firms learn the benefits of adopting a serverless approach to computing.

From a security risks standpoint, three key solutions appear prominently in the reviewed literature. First, **controlling permission** to manage is key to reducing the likelihood of an independent functioning containing a hidden security threat. This is in line with the findings in [8] which noted that the numerous independent functions in a serverless architecture pose the threat of certain functions getting more permissions than they ought to due to an underlying malware. Reviewing each function to determine what it needs to do, following the rule of least privilege, and continued scanning of the likelihood of suspicious activity if a function is proposed as potential precautions to address the security risk. Secondly, **applying the security perimeter** to each function emerges as a strong solution proposed in the reviewed literature. The logic behind perimeter security on each function is to reduce the risk of multiple points of vulnerability introduced by the serverless application. Since each function is a point of vulnerability, securing each of them can prevent a widespread infection of the cloud storage due to a security lapse in one of the independent functions. Lastly, **limiting third-party packages** with lots of dependencies is proposed as an approach to reducing third party dependency risks. Since a serverless computing architecture allows a third party to have access to all part of the cloud-based on dynamic allocation, limiting the dependencies in third party packages reduces the complexity of implementing perimeter security to each function. Based on the following findings, the framework shown in Figure 1 is developed to help solve these issues.

6. PROPOSED FRAMEWORK

Figure 1 illustrates the proposed framework to minimize the security issues listed previously.

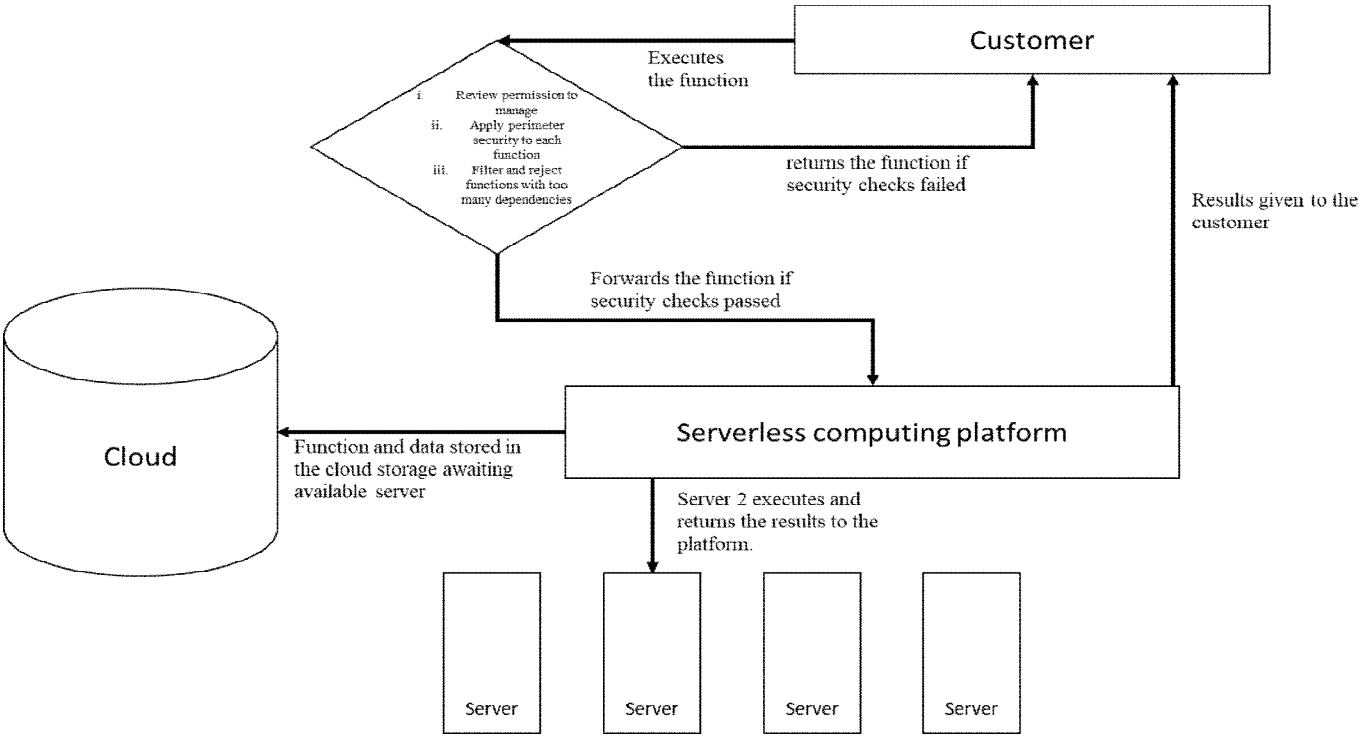

Figure 1: Framework for solving security issues affecting serverless computing

Table 1: Solutions to issues and security challenges facing serverless computing

Organizational Challenges		
Challenges	Solution	Sub-themes
Lack of capability	Collaboration between IT and software development	- Enable knowledge sharing between the two functions - Enable creation of a culture supportive of DevOps - Enable teams to build cross-functional competence
Dedicated team for IT	Running serverless architecture parallel to the existing architecture	- Enables well-established firms to learn of the benefits of serverless technology without risking much - Enables smooth transition to serverless technology
Security challenges		
Hidden security threats	Review each of the multiple functions individually to determine what it wants to do	- minimize roles and permissions for functions - scan each function for suspicious activity
Multiple points of vulnerability	Apply perimeter security to each function	- protects against data breaches - helps profile trusted sources - localizes any vulnerability
Third-party dependency risks	Reduce third-party packages with many dependencies	- use secure links - use automated dependency scanners

7. CONCLUSION

On the basis of the findings, it is apparent that serverless computing is a growing phenomenon in the cloud computing arena. This is evident from a load of challenges facing the adoption of the technology in addition to very limited options. The purpose of the paper was to review the issues and security challenges inhibiting the faster adoption of serverless technology despite the promise of eliminating human resource and maintenance costs associated with running IT infrastructure. Lack of in-house kills DevOps skills emerged as a key organizational challenge to the rapid adoption of serverless technology. Alike, bureaucracy, where the existing management favors current approaches to cloud computing, is also a key hindrance to the rapid deployment of serverless technology. From a security standpoint, more points of vulnerabilities, increase in third-party dependencies, and the likelihood of functions with more permissions than needed pose the greatest security threat to serverless computing approach. Leveraging a systematic review of literature, the research established that promoting a culture of collaboration between software development and IT teams would help create the needed DevOps to support serverless technology. Secondly, running experimental serverless application alongside existing cloud computing infrastructure is proposed a means of helping the rigidly established firms realize the ability and benefits of serverless. Using perimeter security, reducing third party dependence functions, and monitoring each function individually emerged as the most proposed solutions to addressing the security challenges.

REFERENCES

[1] I. Baldini *et al.*, "**Serverless Computing: Current Trends and Open Problems,"** in *Research Advances in Cloud Computing*, S. Chaudhary, G. Somani, and R. Buyya, Eds. Singapore: Springer Singapore, 2017, pp. 1–20.

[2] G. McGrath and P. R. Brenner, **"Serverless Computing: Design, Implementation, and Performance,"** in *2017 IEEE 37th International Conference on Distributed Computing Systems Workshops (ICDCSW)*, 2017, pp. 405–410. https://doi.org/10.1109/ICDCSW.2017.36

[3] G. Adzic and R. Chatley, **"Serverless Computing: Economic and Architectural Impact,"** in *Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering*, 2017, pp. 884–889.

[4] L. Feng, P. Kudva, D. Da Silva, and J. Hu, **"Exploring Serverless Computing for Neural Network Training,"** in *2018 IEEE 11th*

International Conference on Cloud Computing (CLOUD), 2018, pp. 334–341.

[5] M. Sewak and S. Singh, **"Winning in the Era of Serverless Computing and Function as a Service,"** in *2018 3rd International Conference for Convergence in Technology (I2CT)*, 2018, pp. 1–5. https://doi.org/10.1109/I2CT.2018.8529465

[6] S. Nastic *et al.*, **"A Serverless Real-Time Data Analytics Platform for Edge Computing,"** *IEEE Internet Comput.*, vol. 21, no. 4, pp. 64–71, 2017.

[7] T. Lynn, P. Rosati, A. Lejeune, and V. Emeakaroha, **"A Preliminary Review of Enterprise Serverless Cloud Computing (Function-as-a-Service) Platforms,"** in *2017 IEEE International Conference on Cloud Computing Technology and Science (CloudCom)*, 2017, pp. 162–169.

[8] I. Baldini *et al.*, **"The Serverless Trilemma: Function Composition for Serverless Computing,"** in *Proceedings of the 2017 ACM SIGPLAN International Symposium on New Ideas, New Paradigms, and Reflections on Programming and Software*, 2017, pp. 89–103. https://doi.org/10.1145/3133850.3133855

[9] J. M. Hellerstein *et al.*, **"Serverless Computing: One Step Forward, Two Steps Back,"** Dec. 2018.