2022 30th Telecommunications Forum (TELFOR) | 978-1-6654-7273-9/22/$31.00 ©2022 IEEE | DOI: 10.1109/TELFOR56187.2022.9983732

Distributed Denial of Wallet Attack on Serverless Pay-as-you-go Model

Dimitar Mileski, Hristina Mihajloska
Faculty of Computer Science and Engineering, Ss. Cyril and Methodius University in Skopje, North Macedonia
dimitar.mileski@ieee.org, hristina.mihajloska@finki.ukim.mk

***Abstract*—The serverless pay-as-you-go model in the cloud enables payment of services during execution and resources used at the smallest, most granular level, as was the initial idea when setting the foundations and concepts of the pay-as-you-go model in the cloud. The disadvantage of this method of payment during execution and the resources used is that it is subject to financial damage if we have an attack on serverless services. This paper defines notions for three types of attacks that can cause financial damage to the serverless pay-as-you-go model and are experimentally validated. The first attack is Blast DDoW - Distributed Denial of Wallet, the second attack is Continual Inconspicuous DDoW, and the third one is Background Chained DDoW. We discussed financial damages and the consequences of each type of attack.**

***Index Terms*—distributed denial of wallet, serverless, FaaS, DDoS, pay as you go model**

I. Introduction

The Pay-as-you-go model allows us to pay only for the services we use. However, this term is general and covers many services that are in the cloud. Each of these services defines the payment model differently. Some of the dependencies on the payment model can be which services are from IaaS, PaaS, FaaS, or SaaS [1]. If we use a virtual machine service pay-as-you-go model, it will mean that we pay for what we use in terms of virtualization and do not pay for hardware management and operations. However, for the system deployed inside the virtual machine, we will also pay for the time when the application is not used, i.e., we do not have requests that arrive at the system. The disadvantage of this method is that we will pay for the time for which our system does not work effectively. The advantage of such a system, which is deployed in a virtual machine and is not paid during the execution of the request and the resources it uses, as in certain serverless [2] services, is that if we have an attack, the financial damage that will be inflicted is independent of the number of requests coming to the system. Serverless services have a pay-as-you-go model that depends on the execution time of an individual workload and the resources used, which means it is subject to attacks such as DoS, DDoS [3] or tests - Load Testing, Stress Testing, which will cause financial damage. The advantage of serverless services is that many of them are flexible and automatically scalable, which causes an increase or decrease in the resources needed to meet a specific workload. If we have an attack, the resilience of serverless services will cause an increase in resources to satisfy the workload, which will have even more significant financial damage. Suppose we have a flexible system that can satisfy a large workload. In that case, a Distributed Denial of Service Attach type attack can turn into a Distributed Denial of Wallet (DDoW) [4] attachment on a serverless pay-as-you-go model. This paper defines three types of attacks on the serverless pay-as-you-go model. The first one is Blast DDoW, which is Distributed Denial of Service attachment that generates a large workload in a short period and maximizes the use of configured service resources. The second type of attack is the Continual Inconspicuous DDoW which generates a small workload over a long period, making it much more difficult to detect than the Blast DDoW, causing less financial damage. The third type of attack is Background Chained DDoW which refers to a group of serverless services that are somewhere in the system's background. However, we do not know their endpoint and have no access to that service, but we take advantage of the fact that these services can be triggered by actions taken on our part. An example of a Background Chained DDoW would be automated browsers like Selenium [5] to create several users requests that in the background would trigger a chained reaction to one or more serverless services. The paper is further organized as follows. An overview of similar attacks on serverless services is given in Section II. The system is described in Section III with details on methods - experimental architecture, experiments, and evaluation methodology. Section IV addresses results from the experiments, Section V discusses the results, and Section VI concludes with a perspective on future work.

II. Related Work

With the appearance of serverless technology, the security vulnerabilities and challenges of the technology are revealed. Some papers define and explain the potential security problems of serverless technology.

A. DDoS in Cloud computing

Survey on DDoS Attacks and Defense Mechanisms in Cloud and Fog Computing [6] categorizes DDoS attack strategies, and exhausting victim resources is one of them. This category of DDoS attacks is the one that corresponds to the three types of attacks that are defined in this paper, consuming resources and thus causing significant financial damage. Agrawal and Tapaswi in [7] discuss how the attackers exploit the cloud computing features to launch various DDoS attacks. This survey considers the high-rate and all the possible variants of low-rate DDoS attacks in a cloud computing environment.

978-1-6654-7273-9/22/$31.00 ©2022 IEEE

Blast DDoW and Continual Inconspicuous DDoW defined in this paper correspond to the categorization of high-rate and low-rate DDoS attacks, respectively.

B. *DDoS and DDoW - Serverless*

Denial of Wallet, forced financial exhaustion is defined In [4]. This work defines and identifies the threat of Denial of Wallet. The serverless platform used to execute functions is OpenFaaS, a mock application that will trigger the functions on OpenFaaS runs on Apache Server 2.1. They presented Theoretical damage analysis with the official cost guides for each platform they use; AWS Lambda, Google Cloud Functions, Microsoft Azure Functions, and IBM Cloud Functions. Linear increase of costs incurred with Google Cloud functions resulting in the most significant charges followed by AWS Lambda, IBM Cloud Functions, and finally Azure Functions. With 1000 nodes, a slow rate attack of 2000 requests per hour will cost an application owner roughly $40,000 after one month and between $400,000 and $500,000 if left unchecked for a year. Ten thousand nodes will do the same damage in one month that 1000 nodes would do in a year [4]. OWASP released a report [8] about the most common attacks and risks associated with serverless. Vulnerabilities can be exploited to initiate Blast DDoW and Continual Inconspicuous DDoW. OWASP stated that logging and monitoring enable cyberattacks to go unnoticed as consumers cannot identify that their applications have been compromised or that their services are used for illicit purposes. We exploit this flaw in our paper on the Continual Inconspicuous DDoW attack, executed for an extended period, to cause long-term financial damages and to be unnoticed precisely because of the flaws in logging and monitoring. In [9] a new framework for securing serverless applications was proposed called SecLambda. Using SecLambda, three security functions are developed for modeling and monitoring application behaviors, obfuscating credentials in requests, and rate-limiting, in order to prevent flow injection attacks, data leakage, and DoS attacks [9].

III. Methods

The solution architecture of the three types of attacks on serverless services defined in this paper is shown in Fig. 1.

A. *Solution architecture*

Blast DDoW and Continual Inconspicuous DDoW are attacks on an already known public service address. Background Chained DDoW can be Blast DDoW or Continual Inconspicuous DDoW, but the difference is that we do not know the URL address of the serverless service, and we are not authorized to access the service. To generate workload to serverless services in Blast DDoW and Continual Inconspicuous DDoW we use Workload Generator which generates and sends workload to DDoW target service. The implementation of Workload Generator is serverless and uses multiple serverless services in the cloud to be able to generate a large number of requests per second, but the architecture and implementation of this generator is not discussed in this paper. DDoW target service in the experiments was Google Cloud Function which has 1000ms execution time, 256MB memory allocated, CPU 400MHz, minimum instances 0, and maximum instances 3000. Background Chained DDoW attack can also be Blast DDoW, to generate a large number of requests, or Continual Inconspicuous DDoW to generate a small number of requests to the serverless service but in large time periods of days, weeks, months, and even years, which will cause long-term financial damages. Since we do not know the URL of the service or we are not authorized in this type of attack, browser automation can be used. The browser automation via the user interface will cause the triggering of one or more chained serverless services in the cloud. In this paper, Blast DDoW and Continual Inconspicuous DDoW will be evaluated experimentally because Background Chained DDoW can be of both types and is a combination of multiple serverless services that are triggered in the background.

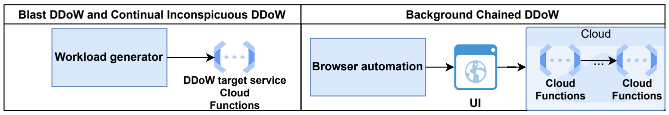

Fig. 1. Solution architecture

B. *Experiments*

1) *Blast DDoW :* This experiment ran for 30 minutes. The Workload Generator is configured with the number of requests per second, target URL of the serverless service, setting for how long the Workload Generator will work, and other settings. The implementation of Workload Generator itself has a certain delay when the serverless instances that will generate the calls are raised. The goal of this experiment is to generate as many invocations per second as possible. The value we will target is 3000 invocations per second. The Workload Generator is capable of generating more invocations per second, but for this experiment we will focus on 3000. This experiment should generate a large number of requests to the service in a short period of time that would cause large financial damages and Denial of Service until the attack is discovered.

2) *Continual Inconspicuous DDoW:* The experiment ran for 24 hours. The goal of this experiment is to generate a small number of invocations per second but over a long period. The experiment ran for 24 hours and is supposed to model the financial damage if such attacks were carried out for days, months, or years without being noticed.

3) *Background Chained DDoW:* This type of attack can be one of the previous Blast DDoW or Continual Inconspicuous DDoW. Also, it can be a combination of several serverless services that are chained and are the target of the attack without knowing their URLs. However, they are not public, and we are not authorized to access them, so we use browser automation to trigger a chained reaction that will trigger serverless services in the background. In this paper, we will not evaluate Background Chained DDoW because it depends

on several factors, such as which and how many serverless services are used and how they are connected. Since we do not know the public URL of the service and are not authorized to access it when we carry out the attack directly, we do not know exactly which services we are using, so we do not even know how much the financial damage would be.

C. Evaluation methodology

In both experiments, we will evaluate the financial damage from the attacks. Price per day for the number of invocations per second and active instances will be considered evaluation metrics. The invocations per second and active instances will help evaluate how serverless services' resilience will affect the attacks' financial damage.

Price per day for number of invocations per second - what will be the financial damage for one day if we have a number of invocations per second all day. Abbreviations: SPD – Seconds per day, IPS – invocations per second, TOI – Total number of invocations. $24 \times 60 \times 60 = 86400SPD$, SPD - 86400, $TOI = SPD \times IPS$. Invocations: TOI, Minimum number of instances: 0, Maximum number of instances: 3000. Price estimate per day is calculated using Google Cloud Pricing Calculator [10] for Google Cloud Function [11].
Active instances – with the increased number of IPS, a greater number of active instances will be launched, so we will be able to evaluate how the increased number of IPS and the increased number of active instances affect the financial damage.

IV. Results

For both serverless pay as you go model attacks, results were obtained for price per day for number of invocations/s and active instances.

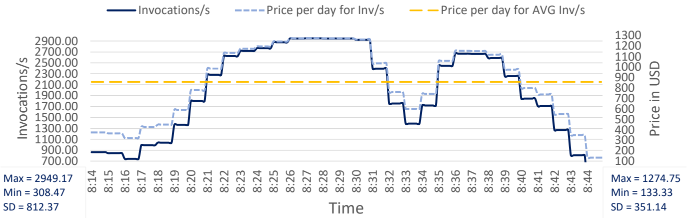

Fig. 2. Price per day for number of IPS - Blast DDoW

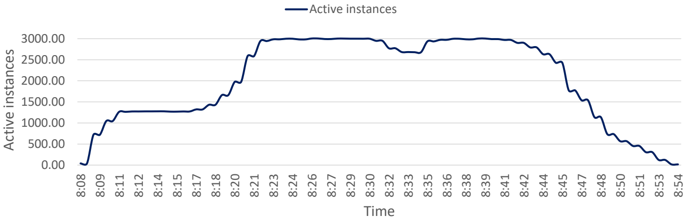

Fig. 3. Active FaaS instances - Blast DDoW

A. Blast DDoW

Fig. 2 presents the invocations/s for Blast DDoW and how they affect the price that will have to be paid to the cloud provider due to the attack. For the average number of invocations/s 1982.10, the price per day is 856.75 USD. For the Min number of invocations/s 308, the financial damage from the attack is 133 USD per day. For Max number of invocations/s 2949, the financial damage from the attack is 1274 USD per day. The Blast DDoW type of attack can cause significant financial damages in a short period if our serverless services are not protected. Fig. 3 shows active instances of Google Cloud Function for the Blast DDoW attack type. We can see that the maximum number of active instances launched during the attack is 3000. This means that this attack caused the launch of the maximum number of theoretically possible active instances of Google Cloud Functions.

B. Continual Inconspicuous DDoW

Fig. 4 presents the invocations/s for Continual Inconspicuous DDoW and how they affect the price that will have to be paid to the cloud provider as a result of the attack. For the average number of invocations/s 155,347, the price per day is 63.65 USD. For Min number of invocations/s 8.55, the financial damage from the attack is 3.50 USD per day. For Max number of invocations/s 160, the financial damage from the attack is 65.79 USD per day. This type of attack can cause minor but long-term financial damages. The results shown in Fig. 4 are from a 24h attack with an average of 155 invocations/s. Fig. 5 shows active instances of the Google Cloud Function for the Continual Inconspicuous DDoW attack type. We can see that the average number of active instances launched during the attack is 304. The implementation of

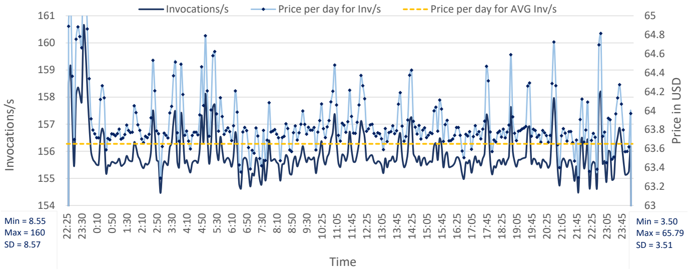

Fig. 4. Price per day for number of Invocations/s - Continual Inconspicuous DDoW

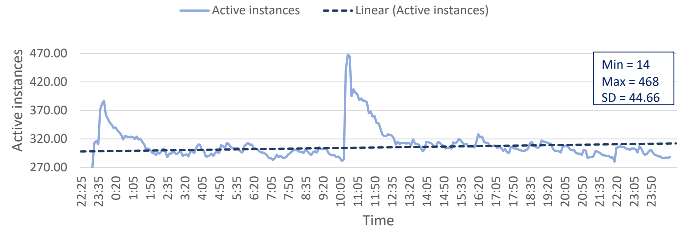

Fig. 5. Active FaaS instances - Continual Inconspicuous DDoW

the Workload generator allows for the generation of a linear number of invocations/s with a standard deviation of 8 invocations/s for 24h as the Continual Inconspicuous DDoW attack was executed. The execution time of the DDoW target service for both Blast and Continual Inconspicuous DDoW is a linear time that corresponds to the set value for executing the function, which is 1000 ms. DDoW target service is scalable. With the increase of the workload, the throughput increases, and response time is nearly constant (Blast DDOW SD = 0.1s, Continual Inconspicuous DDoW SD = 0.004s).

V. Discussion

The three types of attacks on the serverless pay-as-you-go model defined in this paper can cause sizeable financial damage. Smaller companies, startups, and even large companies can suffer significant damage from this attack, which can be both DDoS and DDoW simultaneously. The first type of Blast DDoW attack sends many requests per second, which can be detected because monitoring metrics can be observed. Blast DDoW tends to cause significant financial damage in a short period, which is why it is possible to have a denial of service if the serverless service does not scale well or there are other services after the serverless service that cannot handle the increased workload. The second type of attack, Continual Inconspicuous DDoW, is intended to be executed over a long period and at a lower intensity, making it inconspicuous but causing financial damage in the long run. The third type of Background Chained DDoW attack can be Blast DDoW or Continual Inconspicuous DDoW in that we do not have direct access to call the serverless service, we are not authorized, or we do not know precisely how many and what kind of serverless services exist in the background. With this type of attack, with the help of browser automation, and automation of user interface actions, we want to cause the triggering of a chained reaction that will cause the execution of serverless services. One example is if we assume that a client application for user registration in the background uses a serverless service. With the help of browser automation like Selenium [5] or its alternatives, we can create users. We can use one of the services that create temporary email addresses. Such automated creation of users in the form of either Blast DDoW or Continual Inconspicuous DDoW attacks in the background will cause the serverless service to run. For example, this service can apply some algorithm to the image attached during creation by the user and save it in Cloud Storage. Then there can be a serverless service that processes the entered user registration data and saves it.

1) Example - Blast DDoW: The average number of invocations/s from the Blast DDoW experiment is 1982.10 invocations/s. If we execute the attack for 2 days, the financial damage will be $2 \times 856.75 = 1713.5 USD$.

2) Example - Continual Inconspicuous DDoW: The average number of invocations/s from the Continual Inconspicuous DDoW experiment is 155,347 invocatons/s. If we execute the attack for 1 month, the financial damage will be $30 \times 63.65 = 1909.5 USD$.

These are rough calculations and estimate the financial damage that may occur. The financial damage from the attacks is in US dollars, and the effective date for the price of cloud provider services is 2022-06-29.

VI. Conclusion

This paper presented three serverless pay-as-you-go model attacks and experimentally verified the financial damage. The pay-as-you-go model, together with the flexibility of serverless services, allows payment during execution and the resources used at the level of request or event-driven. However, this method is subject to attacks that will exploit the elasticity and thus the increase or decrease of resources and the payment method during execution, and the resources used. In this paper, three types of attacks were defined: Blast DDoW, Continual Inconspicuous DDoW, and Background Chained DDoW. Some recommendations to protect yourself from attacks are not to set up public services but to use IAM Identity and Access Management [12] services from the cloud. Set alarms that will notify via mail or Pub/Sub messages, if the price for a service exceeds a certain threshold, allowing us to detect the attack. Configuration of serverless services and protection mechanisms implemented inside the code can protect against such attacks.

References

[1] P. Castro, V. Ishakian, V. Muthusamy, and A. Slominski, "The rise of serverless computing," *Communications of the ACM*, vol. 62, no. 12, pp. 44–54, 2019.

[2] E. Jonas, J. Schleier-Smith, V. Sreekanti, C.-C. Tsai, A. Khandelwal, Q. Pu, V. Shankar, J. Carreira, K. Krauth, N. Yadwadkar *et al.*, "Cloud programming simplified: A berkeley view on serverless computing," *arXiv preprint arXiv:1902.03383*, 2019.

[3] F. S. d. Lima Filho, F. A. Silveira, A. de Medeiros Brito Junior, G. Vargas-Solar, and L. F. Silveira, "Smart detection: an online approach for dos/ddos attack detection using machine learning," *Security and Communication Networks*, vol. 2019, 2019.

[4] D. Kelly, F. G. Glavin, and E. Barrett, "Denial of wallet—defining a looming threat to serverless computing," *Journal of Information Security and Applications*, vol. 60, p. 102843, 2021.

[5] S. Gojare, R. Joshi, and D. Gaigaware, "Analysis and design of selenium webdriver automation testing framework," *Procedia Computer Science*, vol. 50, pp. 341–346, 2015.

[6] D. Chaudhary, K. Bhushan, and B. B. Gupta, "Survey on ddos attacks and defense mechanisms in cloud and fog computing," *International Journal of E-Services and Mobile Applications (IJESMA)*, vol. 10, no. 3, pp. 61–83, 2018.

[7] N. Agrawal and S. Tapaswi, "Defense mechanisms against ddos attacks in a cloud computing environment: State-of-the-art and research challenges," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 4, pp. 3769–3795, 2019.

[8] OWASP, "Owasp top 10 (2017) interpretation for serverless," 2017. [Online]. Available: https://owasp.org/www-pdf-archive/OWASP-Top-10-Serverless-Interpretation-en.pdf

[9] D. S. Jegan, L. Wang, S. Bhagat, T. Ristenpart, and M. Swift, "Guarding serverless applications with seclambda," *arXiv preprint arXiv:2011.05322*, 2020.

[10] Google, "Google cloud pricing calculator," 2022. [Online]. Available: https://cloud.google.com/products/calculator/

[11] M. Malawski, A. Gajek, A. Zima, B. Balis, and K. Figiela, "Serverless execution of scientific workflows: Experiments with hyperflow, aws lambda and google cloud functions," *Future Generation Computer Systems*, vol. 110, pp. 502–514, 2020.

[12] I. A. Mohammed, "Cloud identity and access management–a model proposal," *International Journal of Innovations in Engineering Research and Technology*, vol. 6, no. 10, pp. 1–8, 2019.