Securely moving forward with Serverless Architecture

Colin Monroe
Information Security and Digital Forensics

Dr. Petter Lovaas
Information Security and Digital Forensics

***Abstract*—Serverless Architecture is undoubtedly much more cost efficient than a traditional server setup. When compared to traditional client/server networks serverless presents a wider attack surface creating a greater risk for the business. Configuration and management of credentials are the most important aspect in securing serverless. This indicates that the biggest vulnerability in serverless is fundamentally the employees of the company. As shown in numerous misconfiguration breaches, the current security strategies are not suited for serverless. Furthermore, a business must leave their fate in the hands of a third party company who usually places their own interests first. To manage these risks presented by serverless architecture, businesses need to shift to a holistic Defense in Depth strategy.**

***Keywords*—Serverless Architecture, Serverless, Third-Party, misconfiguration, breaches, training, Defense in Depth**

I. Introduction

If organizations utilize serverless architecture it is undoubtedly more cost efficient than a traditional server setup. Serverless architecture which is also referred to as Function as a service (FAAS) or Back end as a service (BAAS) is used through a third party. A third party such as Amazon Web Services (AWS), provides all of the servers and maintenance required for a fee based on usage [15]. Businesses that only need servers for a very short period of time, like microtransactions and blockchains, are starting to use serverless [7]. However, the question remains; how secure is serverless architecture? What are the risks of using this technology, which requires a third party, and how do we manage them and utilize this technology safely?

This paper will identify the main risks to serverless and show that a holistic approach to the Defense in Depth strategy is the best way to manage them. A holistic approach to Defense in Depth is that the strategy must consider all three categories of Information Security: People, Process and Technology [1]. Even though this objective is the implementation of a technological system it affects and is affected by the entire business. Therefore, businesses cannot manage the risk of serverless by instituting a Defense in Depth strategy based on a siloed approach focused on technology. This would leave the businesses vulnerable to threats that bypass technology like Social Engineering and misconfiguration. Recently there have been numerous AWS data breaches, NOFORN and Pentagon breaches, that have occured due to misconfigured databases by third party vendors [11][17]. While it is good for businesses to have a Defense in Depth strategy for each category, these granular strategies are dependant on each businesses individual risks. This paper will only discuss a higher level view strategy that encompasses risks that every businesses will encounter when using serverless. This paper will use the Amazon Lambda model as a use case to explore these questions..

II. Literature Review

A. Serverless Architecture

Serverless architecture also sometimes known as function as a service (FAAS) or Back end as a service (BAAS) is used mainly for micro transactions by companies [3]. The name “serverless”, is quite misleading as it does not mean that there are no servers involved [22]. For businesses, instead of having to create their own client-server network, which is very costly, they can now hire servers for on demand use from companies such as Amazon. Amazon offers a service called Lambda and charges users by the milliseconds of use. Its pricing is approximately $0.0000002 per millisecond [15]. Usually a business will have an application for a user such as a weather forecast app. When this request is sent to the Application Programming Interface (API) gateway of the Lambda service it will startup a server, execute the code, and return the desired value [22]. Most servers will only exist for around 40 milliseconds with 4.5 seconds being the longest lifetime for the server [15].

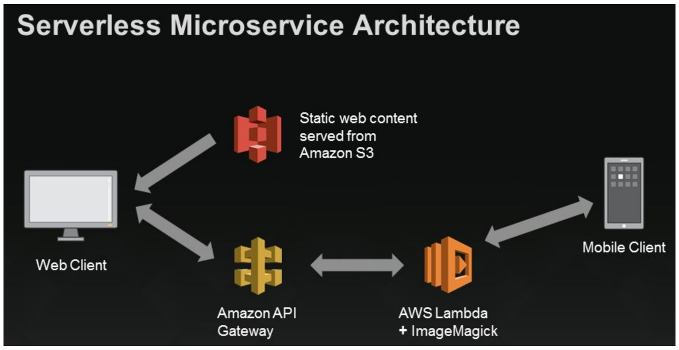
Serverless Microservice Architecture
Static web content served from Amazon S3
Web Client
Mobile Client
Amazon API Gateway
AWS Lambda + ImageMagick

Figure 1. [22]

AWS Identity and Access Management (IAM) allows an organization to set up permissions for groups and individuals [5]. One Lambda function will correspond to one set of credentials [5]. IAM has many features that can be used like multi factor authentication, time of day that server can be accessed, and what IPs are allowed to use AWS [13]. The IAM is configured and maintained by the client not Amazon [15].

An Application Programming Interface gateway acts essentially as a “front door” for all applications and back end services [13]. The Amazon API Gateway acts as a manager for all of the API traffic. All of the API calls that businesses generate including all credential checks from IAM are handled through this gateway [13].

B. Defense in Depth (DiD)

The traditional Defense in Depth principle is based on a layered principle where if one defensive control is defeated than another will be there to protect the network [1]. Most businesses that employ Defense in Depth only utilize it from a technological siloed view [1]. To protect their network, they employ controls like firewalls, Intrusion detection system, and passwords [1]. If one control is defeated then another will be there to protect the network. This traditional approach has drawn criticism of being unrealistic because you are unable to defeat your adversary meaning they can attack you indefinitely [20]. This means that the adversary will eventually be able to find a vulnerability to exploit. Therefore, businesses must not think of creating impenetrable defenses. Businesses must think of a strategy that will cause the resources required to achieve penetration into the system to be too high [20]. Another criticism is that the implementation of strategies that follow strict known standards creates vulnerabilities [20]. This creates vulnerabilities because it allows the attackers to have baseline knowledge of what your strategy is [20]. This is problematic as in any kind of adversarial strategic event the party with the most knowledge of its opponents will win. The traditional Defense in Depth strategy that focuses on technology is unsuited for serverless and its diverse attack surface. Serverless is susceptible to many different risks like: Denial of service, malware, social engineering etc [15]. To combat these different risks, businesses must examine the entire system and identify the risks for People, Process, and Technology and how they affect each other. For example a business may have a good Technology control such as a first class network perimeter security. However, since the business did not consider People, this system is bypassed and defeated by social engineering. So to add breadth to the businesses defense you must also educate the employees which is a proactive strategy for managing the risk of social engineering. It is also important to know which risks pose the greatest threat to the business so that resources are allocated correctly.

C. Penetration Testing

Penetration testing is the best way to find technological vulnerabilities for any system. The degree of a penetration test may vary from a simple scan to a full scale simulated attack on the system. Most companies will prefer to keep the test non-invasive as the risk of damaging systems is greater, which would cost them money. Serverless presents a unique challenge to the penetration testing process as you do not own all of the system that testers need to test [6]. This means that testers need to obtain permission from the company that is providing the serverless functionality [6]. To determine how thoroughly testers can test, the system will be reliant on the cooperation of the third party provider [6].

Some of the known attack vectors that a security testing team, Red Team, would try to use during a penetration test will be examined in the following section. This section will examine and conclude what the currently known technological major threats to serverless are.

The footprinting step is used to map out an organisation to find vulnerabilities to infiltrate. Lambda has both an outer and inner attack surface. The outer attack surfaces main point of vulnerability is the API gateway [15]. To identify where the API gateway is you can first start by looking in the headers of the web requests [15]. If there are requests coming from a service like Cloudfront you have potentially identified an API gateway. It is also possible to use file uploads (s3), or emails (SES) to identify the API gateway [15]. For the inner attack surface you want to examine any other AWS service that is being used with Lambda[15]. This could include queues, database events, streams of information (big data company), and user system [15].

Infiltration: Currently the best known method is to use a black box approach where a tester attacks all the points of exchange and activates all the services possible [15]. Then the output will help determine vulnerabilities that can be infiltrated. A lot of common vulnerabilities still exist like unsanitized input, bugs, flaws, server side script injection, malicious binary files, and most web exploit patterns [15]. However, currently the best and most common way to infiltrate Lambda is through the misconfiguration of the IAM [15].

Exfiltration: Even though there is no direct connection to the internet, testers can utilize tags, meta-information, and the cloud services[15]. Depending on the IAM configuration, testers can look to exfiltrate the information through email or have it transferred to a mobile phone [15]. As Lambda has access to VPC resources, it allows testers to use them once access is gained of the Lambda function. If the functionality of the VPC to securely connect to their corporate network is enabled, it would allow testers to gain access to that network [15].

Even though the lifecycle of servers in Lambda are very short it is possible to create a persistent threat in the system. Lambda has a function called “Alias” which allows it to store old functions in case organizations need to rollback [9]. By installing a backdoor into the function, code and linking it to an old function testers can have malware persist [15].

D. Social Engineering

Social engineering is the act of manipulating people into giving information or allowing access to information. Malicious actors usually accomplish this by taking advantage of human behaviour [21]. Many techniques take advantage of habits, emotions, and people’s preconceived notions [21].

Phishing is the most common type of social engineering and it entails sending out false emails [21]. These emails usually contain a link which sends the user to a webpage asking for credentials [21]. They can also often contain a document that will download malware to the users computer [21]. These emails often try to take advantage of people by instilling fear or offering financial incentive [21]. This is done by pretending to be an entity that is either telling people that their account has been compromised or they have won a prize and to contact them. Then malicious actors will often try to rush people to take advantage of their emotions and not let them take time to question anything.

People's belief in their knowledge of what is safe and secure and what is risky is usually found to be incorrect. A survey asked participants what is more secure a Microsoft Office document or a PDF and why [21]. Most people answered that the PDF was more secure because you cannot edit it, neither are secure [21]. This shows how people's habits can be formed on incorrect knowledge allowing malicious actors to take advantage of them.

E. Third party misconfiguration breaches

In 2017, UpGuard's Chris Vickery discovered a misconfigured Amazon S3 database[11]. On this database they found over a terabyte worth of social media data that the Pentagon had collected over the last ten years [11]. According to UpGuard's O'Sullivan: "A simple permission settings change would have meant the difference between these data repositories being revealed to the wider internet, or remaining secure," [11].

Another misconfigured database that was discovered by UpGuard contained more Pentagon information. This time though it contained NOFORN information which is supposed to be so classified that it is not shared with allies [17]. This misconfiguration occured because the managing third party company merged leaving the database with no oversight [17]).

III. Findings

A. Traditional vs. Serverless Architecture

Most companies now believe that they only have to focus on security in the application layer [14]. One of the top concerns at the app layer is "Function Event Data Injection" which is code injection from any number of events such as IoT, cloud storage etc. [18][19]. In general serverless systems are harder to infiltrate than traditional servers [15]. This is due to there being less common code and malicious actors not being able to use known frameworks [15]. Both the users and functions are isolated and there are no system administrators to escalate too to gain full control [15]. Due to the short lifetime of these servers they no longer have to worry about long standing servers with continual exfiltration of data [15]. Lambda also does not have a persistent connection to the internet and each function has strict permissioning to what it is allowed to access [15].

Security attributes for:	
Traditional server(Client-Server)	**Serverless Architecture**
● Complete granular control of network ● Full penetration testing on system ● All variables known for risk management of system	Short lived server lifespan Less common code No System admin controls to escalate to Hard to exfil data from servers

B. What is the main threat to Serverless Architecture?

Serverless architecture is only as secure as the third party company that you are using. While it is still vulnerable to many common exploits such as unsanitized inputs, flaws and bugs, the real threat to businesses using serverless resides in the people who are configuring and managing it. As shown in the known vulnerabilities from penetration testing the best and easiest way to gain access is through the misconfiguration of the IAM. Recent events show that it is very easy for people to mismanage simple databases like the AWS Pentagon leak [11]. In the case of Lambda if a business has other AWS services this could mean exposing their entire corporate network. Almost every single serverless third party will set you up with insecure defaults [15]. This means that businesses need to have trained staff to configure the system. They need to be aware of how the system operates, not just follow the documentation. As in the case of people using old documentation left on Amazon's forum which led to many vulnerabilities in their systems [15]. This means that the security of the businesses mostly comes down to the competency and training of its employees.

C. Why you need Information Security Professionals

Businesses must not think of serverless as offloading all network security responsibilities as all services are still connected to your company network. They must think of it as just a part of their system which they have no control over. When dealing with a complex environment like AWS that have many different interlinking services. Every AWS service can be a possible threat to every other AWS service which can compromise your company network [13]. If businesses configure just one of these systems incorrectly, it can compromise the whole organization. Businesses need to still have employees that understand the system as a whole, and not just looking at each case in isolation and configuring just that AWS service. Just because each system is secure by itself does not mean it will remain that way when connected together.

When connecting to services outside of your organization's network especially ones that rely on credentials, businesses need to be aware of social engineering. One of the prime attack vectors is the API gateway which controls the flow of information. The Lambda system uses a JSON file to determine the Domain name servers (DNS) of the servers it is supposed to connect to [13]. A social engineering attack that utilizes an attached document that contains malware could manipulate this JSON file. If this occurred theoretically, the attacker could have the company unknowingly connect to a different server. They could then either launch a full scale attack on its network or try to collect data for as long as possible. Also, since the IAM is the main way to compromise a Lambda system, obtaining credentials for this system is desirable. This could be done through phishing: for example a fake email from Amazon asking for someone to enter their credentials on a fake website. It could also be a link or document that installs a trojan or keylogger looking to obtain the IAM credentials.

D. How to educate people to be security conscious

One of the hardest things is not that people cannot recognize a security threat but that they are not in the mindset to do so. It is especially hard to train people to constantly be thinking about security when they have other functions they must focus on. Like a lawyer who is currently focused on their case might click a link and give up their credentials. This is not due to the lawyer not being smart or informed but that they are not consciously considering it [8]. The best way is to start educating people earlier in their life so it become ingrained in them. However, this is not an option for the current situation businesses face. Another difficulty is how to educate people, as everyone has their own style of learning. Mass lectures once every quarter will not teach anyone much. However, it is too inefficient and time consuming for a business to teach people on an individual basis. As Becker, IF; Parkin, S; Sasse, A. 2017 [4], argue for there to be security champions in each business department however it is possible to go further by integrating security members into the workplace. This allows your employees to see them as part of the system and not as a hindrance. This can help in education as people can hopefully see the security staff as teachers, helpers, and colleagues. This also relies on the security staff not blaming the user but instead try to proactively keep the situation away from them. While this is more expensive for businesses, examining the use case of The New York Times the effectiveness of this type of community framework can be observed [16]. The best thing that businesses can do is to facilitate and manage the risk for the employees, essentially taking as much responsibility as possible out of the users hands.

E. Third Party Management

How should a business know or be able to tell if a third party company can be trusted? First a business should do comprehensive research of the company and examine their reputation. They should also find out who developed and is managing the system the company is using [10]. If a third party company was using an encryption program designed by Dr. Adi Shamir, co-creator of RSA, you should be more assured. However, no matter what company or personal reputation businesses should, if possible, try to get an outside evaluation. It's important for the business to keep an open dialogue and oversight of the third party company [10]. Businesses should never get too comfortable and assume everything will be the same. If the third party have key staff changes or communication lines become closed it may be time to seek out a new provider. However, this leads to a problem which currently arises with using serverless: if businesses start using serverless with Amazon and become reliant on it and all of the other AWS services [2]. Even if businesses want to switch the rewriting of code and reconfiguration of the system may prevent the swap due to the cost of downtime [2].

IV. Improvements & Recommendations

Businesses that want to use serverless need to shift to strategies that consider the new widened attack surface. For businesses to manage the risks serverless architecture brings then they need to move to a holistic defense in depth strategy. While resources should be allocated according to threat level no category should be solely focused on. With technological threats like "Function Event Data Injection" being rated the greatest threat by some analysts it is easy to fall back into focusing on Technology [18]. By correctly implementing a holistic Defense in Depth strategy it can raise the resources required for malicious actors to attack your business [1]. This is a good strategy for businesses as malicious actors also consider the economics of attacks and will target the low hanging fruit.

The main focal point of the Defense in Depth strategy must be on People. This is due to not only the fact that it will be the most targeted but it is where businesses can differentiate themselves. Most technology controls are identical in what they do and how strong they are, due to being provided by third party vendors. Therefore, attackers have extensive knowledge of these controls and how they work in systems. If businesses differ on how they educate their employees this can make it more difficult for attackers to attack common vulnerabilities. For the education of employees the first and most important step for businesses is to build a culture of security and trust [16]. To accomplish this, businesses should look towards using people who are teachers and not trainers. Teachers are people who will look to educate, improve, and create a two way street relationship with people. Trainers view relationships as a one way street and simply dump their knowledge on people. Having teachers will increase the likelihood of having a cooperative integrated environment. By having colleagues who are looking to “manage risks” and not be prohibitive, employees will look to learn from them instead of circumventing them [16]. Continuous exposure of integrated security personal will help teach in a repetitious fashion which will ingrain certain principles into employees [21]. This is vitally important, as discussed social engineering is the greatest threat to a business using serverless. Awareness through knowledge is the best proactive strategy against social engineering. Even though there is an emphasis on People, businesses must not ignore the Technology portion. This is evidenced by code injection still being a real threat, as seen by the use of Function Event Data Injection to attack serverless.   It is essential that Information security professionals understand serverless and how it fits into the businesses overall system. Conducting regular penetration tests to find what vulnerabilities are most prevalent should be done regularly. To be able to accomplish this professionals must also be able to understand and oversee the Process category. Since evaluations like penetration tests can only be done in full with cooperation of the third party they must be able to keep an open dialogue with your third party provider [6]. This will also allow businesses to express and solve any risks that they feel need to be addressed. This oversight is important as you should have a “trustless” stance towards third parties [10]. These findings show that without third party management, businesses can not conduct thorough penetration testing. This means that businesses cannot identify the vulnerabilities in their information systems. Which means that employees are vulnerable to hackers looking to gain information from them through phishing attacks, which could lead to important credentials being compromised and leading to a major loss for the business. This is one scenario that can be illustrated from the findings which shows that businesses need to have strategies for People, Process and Technology. Which then needs to be converged into a holistic Defense in Depth strategy that encompasses the entire business.

V. Conclusion

From a business perspective serverless architecture is a vast improvement in cost and efficiency. In terms of security serverless architecture hinges on two very unreliable variables which are people and third parties. With serverless architectures main vulnerability being the configuration of its credential system. This means that people are the main targets of malicious actors. As evidenced by numerous database breaches due to misconfiguration, the training of staff is lacking. Not only the security of your service but potentially the security of your company relies on trusting a third party company. This company’s goals probably do not align with yours and cannot be trusted. Serverless presents a diverse attack surface that gives business a greater exposure to risk. These risks can be managed through using a holistic Defense in Depth strategy that emphasizes people. In conclusion, from a security standpoint serverless architecture can be utilized but only by adapting to this new environment by shifting strategies.

References

[1] Ahmad, A, Maynard, S, Park, S. (2014). Information Security Strategies: Towards an Organizational Multi-Strategy Perspective. Retrieved on 02/12/18 from http://people.eng.unimelb.edu.au/seanbm/research/SecurityStrategy_AhmadMaynardPark_JIM.pdf

[2] Asay, M. (July 11, 2017). The 2 biggest problems with serverless computing. Retrieved on 10/25/17 from https://www.techrepublic.com/article/the-2-biggest-problems-with-serverless-computing/

[3] Baldini, I et al. (10 June, 2017). Serverless Computing: Current Trends and Open Problems. Retrieved on 10/20/17 from https://arxiv.org/pdf/1706.03178.pdf

[4] Becker, IF; Parkin, S; Sasse, A. (April 29, 2017). Finding Security Champions in Blends of Organisational Culture. Retrieved on 01/15/18 from http://discovery.ucl.ac.uk/1554762/

[5] Cui, Y. (October 25, 2017). Many-faced threats to Serverless security. Retrieved on 11/02/17 from https://hackernoon.com/many-faced-threats-to-serverless-security-519e94d19dba
hackernoon.com/many-faced-threats-to-serverless-security-519e94d19dba

[6] Cyberis. (19 May, 2017). Serverless Architectures, Penetration Testing and Authority. Retrieved on 11/05/17 from https://www.cyberis.co.uk/penetration-testing-serverless-architectures.html

[7] Dragonchain. (September, 2017). Dragonchain Business Summary. Retrieved on 01/28/18 from https://dragonchain.com/assets/Dragonchain_Business_Summary.pdf

[8] Kahneman, D. (2011). *Thinking, fast and slow*. New York: Farrar, Straus and Giroux.

[9] Kehoe, B. (January 5, 2017). Serverless Code Security. Retrieved on 10/01/17 from https://serverless.zone/serverless-code-security-d592a3010699

[10] Korolov, M. (December 5, 2017). What is a supply chain attack? Why you should be wary of third-party providers. Retrieved on 12/07/17 from https://www.csoonline.com/article/3191947/data-breach/what-is-a-supply-chain-attack-why-you-should-be-wary-of-third-party-providers.html

[11] Muncaster, P. (November 20, 2017). US Army Exposes Terabytes of Surveillance Data. Retrieved on 11/21/17 from https://www.infosecurity-magazine.com/news/us-army-exposes-terabytes/

[12] Nye, John. (October 17, 2017). DEF CON 25 The Human factor why we are so bad at security. Retrieved on 12/30/2017 from https://www.youtube.com/watch?v=3L3IrAN30a4

[13] Pirtle, J. (July 25, 2017). Security Best Practices for Serverless Applications. Retrieved on 11/03/17 from https://www.youtube.com/watch?v=AV24RTvbgWA

[14] Podjarny, G. (May 17, 2017). Serverless Security: What’s Left to Protect - Guy Podjarny. Retrieved on 10/15/17 from https://www.youtube.com/watch?v=CiyUD_rI8D8

[15] Jones, R. (December 28, 2016). Gone in 60 milliseconds Offensive security in the serverless age (Rich Jones). Retrieved on 10/01/17 from https://www.youtube.com/watch?v=byJBR16xUnc

[16] Sandvik, Rura. (December 12, 2017). Keynote- Building a culture of security at The New York Times - Appsec 2017. Retrieved on 12/29/2017 from https://www.youtube.com/watch?v=_iCLs4jw_yo

[17] Seals, T. (November 28, 2017). Pentagon Exposes Top Secret Classified Info to Public Internet. Retrieved on 11/29/17 from

https://www.infosecurity-magazine.com/news/pentagon-exposes-top-secret/

[18] Segal, O. (January 17, 2018). The First "Serverless Architectures Security Top 10" Guide Released. Retrieved on 01/25/18 from https://www.puresec.io/blog/serverless-top-10-released

[19] Sheridan, K.(January 17, 2018). Where to Find Security Holes in Serverless Architecture. Retrieved on 01/25/18 from https://www.darkreading.com/cloud/where-to-find-security-holes-in-serverless-architecture/d/d-id/1330842

[20] Small, P.(November 14, 2011). Defense in Depth: An Impractical Strategy for a Cyber World. Retrieved on 02/12/18 from https://www.sans.org/reading-room/whitepapers/warfare/defense-depth-impractical-strategy-cyber-world-33896

[21] Vishwanath, Arun. (November 29, 2017). Why Most Cyber Security Training Fails and what we can do about it. Retrieved on 12/30/2017 from https://www.youtube.com/watch?v=3L3IrAN30a4

[22] Wagner, T. (September 5, 2015). Microservices without Servers. Retrieved on 12/01/17 from https://aws.amazon.com/blogs/compute/microservices-without-the-servers/