# HOW TO PREVENT CLOUD CONFIGURATION SECURITY VULNERABILITIES


When more than 100 million personal data records are exposed, it's bound to spark questions and concerns – rightfully so. In the recent cloud security breach involving Capital One, U.S. Senator Rod Wyden of Oregon is asking, does Amazon Web Services share responsibility for the data exposure? And, if so, is there more that can be done to protect other AWS customers from the same issues? His letter is here.

One may think this issue is not worth Senate attention, but the Senator does point out that if it's one company involved, then it's likely on that organization. However, in this case a well-known "vulnerability" in the way certain AWS services are configured led to the data exposure. The *Wall Street Journal* says it found an additional 800 AWS customers potentially at risk from the same configuration error.

## The Shared Responsibility Model

When it comes to security in the cloud, Amazon tries to limit the gray area via its Shared Responsibility Model. In short, Amazon says it is responsible for security *OF* the cloud – the infrastructure and software that run it. The customer is responsible for securing what's *IN* the cloud.

It states: "This shared model can help relieve the customer's operational burden as AWS operates, manages and controls the components from the host operating system and virtualization layer down to the physical security of the facilities in which the service operates. The customer assumes responsibility and management of the guest operating system (including updates and security patches), other associated application software as well as the configuration of the AWS provided

security group firewall."

## Why Does This Keep Happening?

Simply stated, the cause of the cloud security failure is, once again, a misconfigured cloud service. If it has a **configuration** switch, the enterprise is responsible for configuring that service so that it is secure.

This is somewhat easy and manageable in a simple environment, but how many of today's enterprise applications are simple? This gets more complicated by **velocity of change and scale of change**. Dozens (or more) of scrum teams continuously update their microservices, and with each update comes the risk of a simple misconfiguration. Security and compliance checks must be run with each release, but **manual and/or ad-hoc checks** create bottlenecks and **lack of consistent governance** can introduce risk.

This type of error can happen to ANYONE; the scale and speed of change which the cloud enables leads to just such a risk. Publicly disclosed exposures due to misconfigured cloud resources are up 20 percent year-over-year[1], and misconfigured cloud resources remain the #1 cause of cloud security failures[2]. Either through misunderstanding – 53% of organizations mistakenly believe their CSP is wholly or majority responsible for securing their data[3] – or negligence, these misconfigurations prove time and again that getting cloud security right is a monumental undertaking.

## Why Has This Specific Breach Captured Our Attention?

Consumers are highly protective – rightly so – of their personal and financial data. Whether it's because the brand is well-known, or that this is yet another financial breach, or that instead of a nameless, faceless actor of a hostile foreign power we have the name and face of a suspect who documented her exploits on social media, the Capital One breach, more so than any since Equifax (which was an on-prem breach), has captured the attention of the media.

Far too much has been made of the fact that the hacker was a former AWS employee. True, she was employed there some number of years ago, but her employment history with AWS is entirely irrelevant. More precisely, it is her familiarity and expertise in using AWS cloud services which is key. One does not have to ever have been an employee of Amazon to develop these skills. There is no public evidence available thus far to warrant an insider threat.

So, what *did* happen? Simply stated, the suspect allegedly found and exploited a misconfigured WAF (web application firewall, a type of cloud service) to access Capital One's VPC (virtual private cloud). Once inside, she launched an EC2 instance to trick the AWS Metadata Service into trusting it, thereby receiving credentials to access and decrypt data stored in an S3 bucket. An excellent detail of the hack can be found at Krebs on Security. See Server-Side Request Forgery for an explanation of how an SSRF works. **In lay terms, a misconfigured cloud resource (WAF) caused the cloud security failure**.

## How Do We Prevent This from Happening Over and Over?

Thanks to the suspect having left extensive breadcrumbs on her social media, we have a very clear

view of how she exploited the cloud security of Capital One. Beyond the "how" it happened, what can we – the collective "we," the enterprise "we" - do to prevent recurrence? Fortunately, the answer is plenty.

A solution for effectively managing the enterprise's cloud security posture must include:

- **Automation**. Automation is the key, because manual checks are a fool's errand: the cloud footprint changes too fast. Multiple, self-organizing scrum teams push updates to PROD asynchronously and multiple times daily. With each update comes the risk of a single cloud resource being misconfigured. As such, you cannot solve the problem of cloud security posture management (CSPM) by throwing bodies at it. The speed and scale of change outstrips human ability to keep pace. Automation eliminates human error and delays caused by manpower bottlenecks, making cloud security and governance as agile as the developers who use the cloud. It also enables your high-cost security professionals to rise above the noise of repetitive ditch-digging so that they can focus on higher value work. In sum, you get better staff productivity, higher quality, better security, and higher customer satisfaction.
    - Recommendation: Use automated security checks against policy-based best practices such as those published by the Center for Internet Security.
    - Recommendation: Use automated remediation, working programmatically to reconfigure those cloud resources which violate security and compliance policy. Automation removes the element of human error and dramatically diminishes the window of vulnerability.
    - The difference between automatic and automated remediation sometimes causes confusion. *Automated* action runs a remediation script triggered by some human intervention, such as clicking a button. *Automatic remediation takes the same action, but without any human intervention, so that the moment a vulnerability is found, it is automatically fixed – consistently, securely, and with a documented audit trail according to the organization's change management workflow. Both automated and automatic remediation offer profound advantages above manual security and compliance checks.*
- **Policy-based security, compliance, and governance**. The enterprise's Infosec Team codifies security and compliance mandates into a library of security policies, which are applied uniformly throughout the enterprise. Because security checks are run comparing all cloud resources automatically against these policies, real-time security is achieved because there is no manpower bottleneck at the security checkpoint in the process. These security checks are automated, as the fixes should be as well. Exceptions to any security violations are managed according to whatever process the organization puts in place.
    - Recommendation: use the security frameworks published by the Center for Internet Security as the foundation of your security policy library, to speed your time-to-value and achieve consistent configurations across your cloud footprint. Extend those policies as you see fit.
- **Multi-cloud**. Although the Capital One breach was on AWS, AWS is not the only cloud service provider (CSP). AWS, Azure, and Google Cloud dominate the CSP market, and each have hundreds of similar, but not identical, cloud services, every instance of which must be correctly configured. AWS have a laundry list of security tools available for purchase. The trouble is, (1) naturally they focus on AWS services, (2) do not provide for remediation (which is the customer's responsibility), and (3) numerous tools complicate security and make it difficult for developers to secure their microservices. Most enterprises are multi-cloud, and, as already discussed, it is imperative that the organization consistently apply security and compliance policies across its scrum teams, accounts, and cloud platforms. The flexibility which the cloud

provides is powerful, though it is a razor-sharp, double-edged sword.
    - Recommendation: enable business agility by making security easy on the developers. Choose a cloud security solution which supports the "Big 3" CSPs IaaS and PaaS services, so that you get more cooperation from your internal customers (*cough* developers *cough*).
- **Integrate to incident and change management workflows**. As previously mentioned, the speed of change in the cloud from numerous, self-organizing scrum teams challenges the organization's change management. As security professionals, our goal should be to enable agility, and not hamstring it. Once a violation is identified, we should automatically open an incident and change request at the service desk, and kickoff our change management workflow. Once approved, the configuration change is automatically made, thereby closing the vulnerability and the incident closed. The CMDB is updated and an audit trail documented.

# Conclusion

Capital One is a leader and expert in the use of cloud technology. While this type of cloud security failure could happen to anyone, there is plenty we can do to prevent these types of failures. The cloud itself is not inherently insecure, but we (the collective "we," users of the public cloud) continue to struggle with securing our ever-changing footprint in the public cloud. Just like when AWS changed the default configuration of its S3 (Simplified Storage Service) bucket so that it is private, not public, it is worthwhile to have a rational, non-politicized discussion of what actions AWS and all CSPs can take.

At the same time, all organizations are encouraged to take stock of their current methods, skills, and technology for securing their public cloud estate. Waiting for a regulation to fix the problem might invite more trouble. The better course is to take action on your own to make sure the data you store and the services you offer are secure, regardless of where they reside.

[1] 2019 IBM X-Force Threat Intelligence Index Report

[2] i.b.i.d.

[3] EMA Security Megatrends 2019