EBOOK

Executive Guide to Cloud Security

Fugue®

Introduction

Getting security on the cloud right means understanding that the cloud is 100% software via programmable APIs and cloud services are engineered to scale up to thousands of servers. As easy as it is to spin up new infrastructure or scale applications immediately, it is just as easy with a few keystrokes to introduce misconfigurations that lead to vulnerabilities.

Cloud misconfigurations are therefore bound to happen and may lead to customer-impacting data breaches. In the case of Capital One[1], a hacker exploited cloud misconfigurations to extract sensitive data for about 100 million customers in the United States. The hacker used these same exploits to access data from more than 30 other organizations. As more organizations move to the cloud, these types of data breaches will become increasingly common.

> This guide will talk about how the cloud's programmable nature fundamentally changes how one should think about cloud security. It requires a different mindset and understanding of computing.

Everything IS PROGRAMMABLE

The cloud's fundamental nature is that it is defined through software, specifically via Application Programming Interfaces (APIs) from cloud providers such as AWS, Microsoft Azure, and Google Cloud Platform. Every object in the cloud—servers, databases, network, even security policies—is created, modified, and destroyed through API calls. While this provides tremendous flexibility, agility, and power, it also means there is significant risk of software errors—misconfiguration of cloud resources.

In addition, there are simply more resources to track and secure in the cloud than in the data center. Teams operating at scale in the cloud may be managing dozens of environments across multiple regions and accounts, and each may involve tens of thousands of resources that are individually configured and accessible via APIs. The size and complexity of the enterprise environments make it incredibly difficult to know what is running where. It's very easy for a developer or administrator to change the wrong resource configuration setting, assign the wrong set of permissions, or lose track of where critical resources are residing.

Significant risk and potential vulnerabilities stem from software errors or misconfiguration of cloud resources.

Cloud Security IS A SHARED RESPONSIBILITY

According to the Shared Responsibility Model, cloud providers are responsible for the security **of** the cloud, which includes the physical infrastructure that runs all of the services offered in their cloud. The customer is responsible for security **in** the cloud, which is the configuration of cloud resources used by the customer. For example, the customer must correctly configure operating systems and applications running on EC2 instances, or ensure that user permissions or network configurations are not overly broad.

Developers Are in Control OF CLOUD SECURITY POLICIES

Whether you realize this or not, developers are already defining your organization's cloud security policies. Because everything in the cloud is specified via software, it means that developers are defining everything required by an application—its compute and storage resources and critical network and security configurations—and most likely without formal oversight.

Cloud Attacks
ARE BASED ON CLOUD-NATIVE TECHNOLOGIES

In the cloud, the concept of a perimeter no longer exists. Hacking Identity and Access Management (IAM) permissions becomes the new way of laterally moving within the enterprise. If a user has the right IAM permissions, he or she can access any cloud resource regardless of network configuration. Also, traditional network and endpoint security tools are no longer relevant in the cloud. Network traffic monitors can't see sensitive data that is exfiltrated from cloud storage buckets with API calls because the data does not traverse the network. Endpoint security agents cannot infer malicious activity from cloud API calls or CLI commands that are typically used for legitimate purposes.

Traditional network and endpoint security tools are no longer relevant in the cloud.

Management Recommendations FOR PROTECTING CLOUD APPLICATIONS

1 CLARIFY WHO HAS ORGANIZATIONAL RESPONSIBILITY FOR CLOUD SECURITY

In some organizations, cloud security is handled by the head of software development or infrastructure rather than the CISO. Alternatively, in organizations with a well-developed and cloud savvy security engineering team it may be preferable to have the CISO and their team own it.

2 EMBED BOTH SECURITY AND DEVELOPERS INTO THE SAME ORGANIZATION

This helps to ensure that people who understand cloud concepts are also empowered to properly secure cloud infrastructure. You may possibly create a “cloud security engineering team” or “cloud center of excellence” that reports to one or the other of the CIO, the head of DevOps, or the CISO.

3 DEVELOP AN INVENTORY OF CLOUD RESOURCES

The ease with which software developers can deploy new cloud infrastructure means you may have orphaned and unused cloud resources. It is important to eliminate these to prevent cyber criminals from leveraging them for their own nefarious purposes.

Develop a full inventory of your cloud infrastructure to assess your resources and determine whether they are properly configured.

4 IMPLEMENT AUTOMATED REMEDIATION

According to AWS CISO Stephen Schmidt, manual remediation methods are simply too slow. The time taken to identify a cloud misconfiguration, create a ticket, and assign it to an engineer who then researches and ultimately fixes the issue could take hours or even days. The longer a cloud misconfiguration exists, the greater the risk of a major data breach.

The longer a cloud misconfiguration exists, the greater the risk of a major data breach.

Technical Recommendations FOR PROTECTING CLOUD APPLICATIONS

❶ APPLY LEAST PRIVILEGE IN IDENTITY AND ACCESS MANAGEMENT

IAM is the first and best line of defense. Identity is considered the "new firewall"[2] and giving widespread permission to users or service accounts that only need limited permissions to perform their jobs creates potential vulnerabilities.

Apply the principle of least privilege to IAM roles.

- Do not permit the same role to perform both read and write operations.
- Do not allow any cloud resources such as virtual servers in production environments to be able to assume a broader set of permissions. In the language of AWS, do not allow EC2 instances to have IAM roles that permit attaching or replacing role policies.
- Require multi-factor authentication for privileged accounts and ensure that accounts not used for at least 90 days be disabled.

❷ TURN ON LOGGING EVERYWHERE

As in physical data center environments, logging is essential in providing visibility into the security posture of cloud environments. Using a service like AWS CloudTrail to log all API commands will give you visibility into any API commands used by attackers to search for or exfiltrate sensitive data.

Best practices for logging:

- Ensure you are logging all network flow data and access to all storage buckets, especially those holding sensitive data.
- Make sure you are encrypting the log data and storing it in secure storage buckets, as well as enabling log file validation.
- Search log files and create custom metrics for critical activities such as failed logins, changes to network configuration or IAM policy, and all commands executed by the root account.

❸ ENCRYPT EVERYTHING

Encrypt all services that support encryption whether data or traffic is at rest or in motion. This means enabling encryption for services such as databases, load balancers, storage buckets, message queues, and notification topics.

Technical Recommendations FOR PROTECTING CLOUD APPLICATIONS

4 LOCK DOWN YOUR NETWORKS

Make it as difficult as possible for attackers to access your cloud resources via the internet. This means checking for overly permissive AWS Security Groups and NACLs or Azure Network Security Groups, such as those that permit access to port 22 from the world. Although these configurations can be locked down in the provisioning process, it is very easy for a developer to open a "temporary" hole to a production resource to perform some troubleshooting when working from home.

5 PROHIBIT UNNECESSARY CONSOLE ACCESS

Anyone with access to the administration console for your cloud environment can potentially change or delete any resource in the environment. This is a sharp contrast from the physical data center, where only those who could actually enter the facility could access the infrastructure. Because of this, remove console access to your production environment from anyone who does not need to make production changes.

This may be a big policy change, especially if your development team is used to having unfettered access to the production console for ease of deployment and troubleshooting. The best practice for cloud deployments is to have an automated CI/CD pipeline for deploying changes to production infrastructure, where only the CI/CD service account is permitted to make production changes.

6 CLEAN UP ORPHANED RESOURCES

It is very easy for inexperienced developers or interns to create cloud resources for the sake of learning or experimentation, even in production environments. When these employees eventually move on, these resources typically are not cleaned up but instead stay unnoticed for weeks or even months. They can provide a foothold for an attacker into an environment, especially if their permissions or network configurations have not been properly secured.

Technical Recommendations FOR PROTECTING CLOUD APPLICATIONS

Make sure you have an automated means of taking inventory of cloud resources and identifying orphans. One scheme for doing so is to create a process for explicitly tagging resources used in development or production. If a resource does not have a tag then it was not properly provisioned and is a candidate for removal.

7 PERFORM PENETRATION TESTING FOR CLOUD MISCONFIGURATION

Include cloud infrastructure misconfiguration in your penetration testing efforts. Use outside penetration testers and make sure they are knowledgeable about how to find and exploit cloud misconfigurations.

Conclusion

Since everything in the cloud is software defined and programmable, your developers are moving fast and making cloud infrastructure decisions, including those with security ramifications. Empower them with the right governance, approach, and tooling, and your data and applications can be more secure in the cloud than in the data center.

FOOTNOTES:
1. https://www.capitalone.com/facts2019/
2. https://www.fugue.co/blog/what-executives-should-know-about-the-capital-one-breach

Fugue®