Turbulence in the Cloud

Cloud computing technologies enable organizations to evolve their data centers from a hardware-centric architecture where applications run on dedicated servers to a dynamic and automated environment where pools of computing resources are available on demand, to support application workloads that can be accessed anywhere, anytime, and from any device.

However, many organizations have been forced into significant compromises regarding their public and private cloud environments – trading function, visibility, and security for simplicity, efficiency, and agility. If an application hosted in the cloud isn't available or responsive, network security controls, which all too often introduce delays and outages, are typically "streamlined" out of the cloud design. Cloud security trade-offs often include

Simplicity *or* function

Efficiency *or* visibility

Agility *or* security

Many of the features that make cloud computing attractive to organizations also run contrary to network security best practices. For example:

Cloud computing doesn't mitigate existing network security risks. The security risks that threaten your network today don't go away when you move to the cloud. The shared responsibility model defines who (customer and/or provider) is responsible for what (related to security) in the public cloud. In general terms, the cloud provider is responsible for security *of* the cloud, including the physical security of the cloud data centers, and for foundational networking, storage, compute, and virtualization services. The cloud customer is responsible for security *in* the cloud, which is further delineated by the cloud service model. For example, in an infrastructure-as-a-service (IaaS) model, the cloud customer is responsible for the security of the operating systems, middleware, runtime, applications, and data. In a platform-as-a-service (PaaS) model, the cloud customer is responsible for the security of the applications and data – the cloud provider is responsible for the security of the operating systems, middleware, and runtime. In a SaaS model, the cloud customer is responsible only for the security of the data, and the cloud provider is responsible for the full stack, from the physical security of the cloud data centers to the application.

Separation and segmentation are fundamental to security; the cloud relies on shared resources. Security best practices dictate that mission-critical applications and data be separated in secure segments on the network, based on Zero Trust principles. On a physical network, Zero Trust is relatively straightforward, using firewalls and policies based on application and user identity. In a cloud environment, direct communication between virtual machines (VMs) within a server host occurs constantly – in some cases, across varied levels of trust, thus making segmentation a real challenge. Mixed levels of trust, combined with a lack of intra-host traffic visibility by virtualized port-based security offerings, may weaken your security posture.

Security deployments are process-oriented; cloud computing environments are dynamic. The creation or modification of your cloud workloads can often be done in minutes, yet the security configuration for this workload may take hours, days, or weeks. Security delays aren't designed to be burdensome; they're the result of a process that is designed to maintain a strong security posture. Policy changes need to be approved, the appropriate firewalls need to be identified, and the relevant policy updates need to be determined. In contrast, the cloud is a highly dynamic environment, with workloads being added, removed, and changed rapidly and constantly. The result is a disconnect between security policy and cloud workload deployments, which leads to a weakened security posture. Thus, security technologies and processes must be able to auto scale to take advantage of the elasticity of the cloud while maintaining a strong security posture.

Infrastructure as code automates the ability to rapidly scale secure configurations – and misconfigurations. Organizations are rapidly adopting *infrastructure as code* (IaC) as they attempt to automate more of their build processes in the cloud. IaC has become popular as it enables immutable infrastructure. This is the ability to standardize and freeze many parts of cloud infrastructure, so results are consistent and predictable when running code every time. For example, if you know that every node in your cloud has the exact same virtual networking configuration, your chances of having networking-related app problems decreases significantly. And while IaC offers security teams a predictable way to enforce security standards, this powerful capability remains largely unharnessed. The challenge for organizations is ensuring that IaC configurations are consistently enforced across multiple public cloud accounts, providers, and software development pipelines.

Data can be quickly and easily consumed by applications and users in the cloud. However, more sophisticated threats and new privacy regulations have raised the stakes on data security everywhere – including in the cloud. Data loss prevention (DLP) provides visibility across all sensitive information, everywhere and at all times, enabling strong protective actions to safeguard data from threats and violations of corporate policies. But legacy standalone DLP technologies are not efficient for today's cloud-driven world. Built on old core engines specifically for on-premises environments, the technology has not changed significantly in the last decade. To adjust to cloud initiatives, legacy DLP providers are simply extending their existing solutions to cloud environments, which creates a gap in visibility and management and minimizes policy control. Organizations that have spent enormous amounts of time and money to build a custom DLP architecture to fit their network environments are now struggling with complexity and poor usability as they try to "add in" their cloud apps, data, and public cloud instances. Additionally, security teams face the challenge of using effective but complex DLP technologies while balancing the constant work that comes with them – from ongoing policy tuning to exhausting incident triage cycles and incident response decisions. These teams are drowning in too too many alerts – most of which turn out to be false positives – and often respond to a data incident too late.

Key Terms

Infrastructure as code (IaC) is a *DevOps* process in which developers or IT operations teams can programmatically provision and manage the infrastructure stack (such as virtual machines, networks, and connectivity) for an application in software.

DevOps is the culture and practice of improved collaboration between application development and IT operations teams.

SaaS application risks

Data is located everywhere in today's enterprise networks, including in many locations that are not under the organization's control. New data security challenges emerge for organizations that permit SaaS use in their networks.

With SaaS applications, data is often stored where the application resides – in the cloud. Thus, the data is no longer under the organization's control, and visibility is often lost. SaaS vendors do their best to protect the data in their applications, but it is ultimately not their responsibility. Just as in any other part of the network, the IT team is responsible for protecting and controlling the data, regardless of its location.

Because of the nature of SaaS applications, their use is very difficult to control – or have visibility into – after the data leaves the network perimeter. This lack of control presents a significant security challenge: End users are now acting as their own "shadow" IT department, with control over the SaaS applications they use and how they use them. But they have little or no understanding of the inherent data exposure and threat insertion risks of SaaS, including:

Malicious outsiders. The most common source of breaches for networks overall is also a critical concern for SaaS security. The SaaS application becomes a new threat vector and distribution point for malware used by external adversaries. Some malware will even target the SaaS applications themselves, for example, by changing their shares to "public" so that the data can be retrieved by anyone.

Accidental data exposure. Well-intentioned end users are often untrained and unaware of the risks their actions pose in SaaS environments. Because SaaS applications are designed to facilitate easy sharing, it's understandable that data often becomes unintentionally exposed. Accidental data exposure by end users is surprisingly common and includes:

Accidental share. A share meant for a particular person is accidentally sent to the wrong person or group. Accidental shares are common when a name auto fills, or is mistyped, which may cause an old email address or the wrong name, group, or even an external user, to have access to the share.

Promiscuous share. A legitimate share is created for a user, but that user then shares with other people who shouldn't have access. Promiscuous shares often result in the data being publicly shared because it can go well beyond the control of the original owner.

Ghost (or stale) share. A share remains active for an employee or vendor that is no longer working with the company, or should no longer have access. Without visibility and control of the shares, the tracking and fixing of shares to ensure that they are still valid is very difficult.

Malicious insiders. The least common but real SaaS application risk is the internal user who maliciously shares data for theft or revenge purposes. For example, an employee who is leaving the company might set a folder's share permissions to "public" or share it with an external email address to later steal the data from a remote location.

The average employee uses at least eight applications,[1] but as employees add and use more SaaS apps that connect to the corporate network, the risk of sensitive data being stolen, exposed, or compromised increases. It is important to consider the security of the apps, what data they have access to, and how employees are using them. Here are several best practices for securing sensitive data in SaaS apps:

Discover employee use of unvetted SaaS applications. As SaaS adoption rapidly expands, manual discovery of SaaS usage in the enterprise becomes increasingly untenable. Instead, to quickly identify risk – and extend appropriate security controls – your organization needs an automated way to continuously discover all SaaS applications in use by employees.

Protect sensitive data in SaaS applications. Implement advanced DLP capabilities using an *application programming interface* (API)-based approach to scan for sensitive information stored within SaaS applications. Compared to inline, an API-based approach provides deeper context and allows for automatic remediation of data-risk violations.

Secure your weakest link – SaaS users. Start with user training and interactive coaching to identify and help change risky behavior. Then, give your security team tools to help them monitor and govern SaaS application permissions. Look for a solution with robust access controls, including:

Multi-factor authentication (MFA)

Role-based access control (RBAC)

Protection for administrative accounts

User access monitoring that can detect malicious or risky behavior

[1] "2019 SaaS Trends." Blissfully. 2019. https://blissfully.com/saas-trends/2019-annual/.

Enforce compliance requirements in the cloud. Create and enforce a consistent, granular security policy for compliance that covers all SaaS applications used by your organization. This includes automating compliance and reporting for all relevant regulatory requirements across your SaaS applications.

Reduce risk from unmanaged devices. Deploy a security product that differentiates access between managed and unmanaged devices to protect against the increased security risks inherent with personal devices. For instance, you could allow downloads to managed devices but block them for unmanaged ones while enabling access to core functionality.

Control data sharing from SaaS applications. Use an inline approach to gain visibility into sensitive data flowing into high-risk, unsanctioned applications. Create and enforce DLP policies that control data-sharing activities in the SaaS applications employees use.

Stop SaaS-borne malware threats. Implement threat prevention technology that works with your SaaS security to block malware and stop threats from spreading through SaaS applications, eliminating a new insertion point for malware.

Key Terms

An *application programming interface* (API) is a set of routines, protocols, and tools for building software applications and integrations.

Multi-factor authentication (MFA) refers to any authentication mechanism that requires two or more of the following factors: something you know, something you have, something you are.

Role-based access control (RBAC) is a method for implementing discretionary access controls in which access decisions are based on group membership, according to organizational or functional roles.