A New Way to Look at AWS Security

aws partner network

Advanced
Technology Partner

SaaS Partner

Providing an ideal IT security environment

Organizations large and small are shifting IT resources to Amazon Web Services (AWS) on a historic scale, driven by demands for greater capital efficiency, agility, and scalability. Any new and dynamic environment like AWS should be approached cautiously by those mindful of information security concerns.

However, AWS can be seen to provide an environment for IT security that is qualitatively superior to most alternatives available to organizations smaller than AWS itself. AWS's built-in foundation of visibility, identity, and policy enforcement enables out-of-the-box detection and, hence, avoidance of known problems. Combining the capabilities of AWS and Cisco® Stealthwatch Cloud—which can detect unknown threats in the behavior of AWS resources—results in a nimble, scalable, and cost-effective security solution for AWS customers.

Why does AWS represent an ideal environment for security? The short answer is best viewed through the near-term lens that all security professionals can appreciate: most of the difficult legacy information security challenges that large enterprises struggle with are solved on day one in an AWS deployment. The deeper answer—which involves achieving forward-looking, steady-state security for modern firms whose data analysis and computing approaches are cloud native—is the ultimately more compelling answer, but we'll consider the short answer first.

Contents

Solved problems in AWS security

The legacy challenges in on-premises environments include visibility; identity and access management; and policy declaration and enforcement. In AWS, they are all solved problems.

Consider your on-premises network. Suppose that you have a structured audit trail whenever:

- A new device enters or leaves the network
- A user authenticates on any device
- A user makes use of an information service
- Someone makes changes to infrastructure elements like routing tables or firewall rules
- Someone modifies the security permissions or organizational role of a user or device
- One of your IT assets has a network interaction

Suppose further that this information is provided to you, flexibly and programmatically. Suppose that you also get any other application-level logs that you care to monitor and any custom metrics you would like to capture that track activities among your users, applications, and IT resources. This is the visibility you have out of the box in AWS. In AWS, visibility and instrumentation are solved problems.

To support this visibility and instrumentation, AWS operates as a Service-Oriented Architecture (SOA). All the actions in your AWS footprint are initiated by authenticated API calls to web service endpoints. That is, specific API calls are made with authenticated user credentials when new user accounts are created, firewall rules are changed, and servers are instantiated.

All changes to AWS resources are made through authenticated API calls, and all of these calls are logged and made available to the account owner. This audit trail is not a superficial aspect of AWS. Rather, it is the intrinsic AWS feature that allows it to bill you for your usage. So you don't have to worry about this going away anytime soon. Of course, this API visibility is not just available in principle. It is provided in AWS by a service called AWS CloudTrail, which delivers a structured feed of all the requests to access or modify your AWS footprint.

Two other AWS services provide important built-in visibility. First, Amazon CloudWatch is a monitoring service for AWS resources and applications. All of AWS's built-in services, such as Amazon Elastic Compute Cloud (servers), Amazon Relational Database Service (databases), and Amazon Elastic MapReduce (data analysis), use Amazon CloudWatch to report utilization and status events. Developers can use Amazon CloudWatch's open API to add log and metric monitoring to custom applications and services.

Additionally, Amazon CloudWatch supports alarms and custom events to detect problem states and to trigger automatic actions. For example, you can use Amazon CloudWatch alarms to manage auto-scaling groups, so the number of servers used in an AWS footprint can be dynamically scaled up or down in response to a utilization metric that you maintain. With Amazon CloudWatch, you gain visibility into the operation and activities of specific applications and services.

Second, the VPC Flow Logs service gives you visibility into the network traffic that your AWS servers send or receive. When any of your AWS VPC resources have a network interaction, a VPC Flow Log entry records the details of the network conversation, including the source and destination network interface and IP addresses, ports, protocol, byte count, and packet count seen. Those with experience in on-premises network security will recognize this as an analog to the NetFlow logs that can be produced by enterprise-grade switches, routers, and firewalls. These logs are significant because they represent an auditable record of all network interactions within your AWS virtual private cloud footprint.

So these three AWS services—AWS CloudTrail, Amazon CloudWatch, and VPC Flow Logs—together represent a comprehensive visibility layer for your AWS footprint. They provide out-of-the-box visibility into your account usage, user behavior, infrastructure management, application and service activity, and network activity. Importantly, AWS users obtain the benefit of these services without having to bear the maintenance or capital costs required to provide them.

This discussion of visibility has also illuminated the importance of Identity and Access Management (IAM). In fact, it is impossible to use AWS without structured, audited IAM credentials. AWS is built with a fully integrated IAM service provides credentials for all aspects of AWS service interaction. It declares which user identities exist and what privileges they possess so you can observe and manipulate your AWS footprint. You can use AWS's native IAM service to manage your entire identity and access management workflow, and you can also integrate IAM with a third-party service. In any case, your AWS resources (servers, databases, storage, logs, policy objects, etc.) cannot be viewed or manipulated except through the IAM service. Like visibility, identity and access management is a solved problem in AWS.

Finally, AWS has a built-in service for comprehensive policy declaration and enforcement: AWS Config. This service provides both ad hoc and continuous audits of your AWS resources and their internal configurations.

Consider a simple example: suppose that you want to verify that user passwords are disabled on all your servers, so only key-based access is possible in your AWS footprint. AWS Config makes it easy to run that report for all of your servers. Consider more sophisticated examples: "No servers can use port 22," "Only administrators can change firewall rules," or "Only user Betsy can create new user accounts, and she can only do so on Tuesdays." AWS Config can do all that because:

- All changes to AWS resources are managed through authenticated calls to AWS endpoints
- All policies governing AWS resources and their usage are expressed and enforced in code

In the summer of 2016, the AWS Config Rules service was fully released to automate the detection of policy violations. AWS Config Rules are, in effect, continuous configuration queries that produce event notifications when they are violated. For example, rather than running AWS Config queries periodically to verify that all server disks are encrypted, AWS Config Rules can be used to continuously scrutinize server disks for this condition. In this way, you can automatically produce continuous compliance reports that accurately represent the configured state of all AWS assets.

Does AWS solve all security problems?

If visibility, identity and access management, and policy enforcement are solved in AWS, does that mean that all security problems are solved? Of course not.

The shared responsibility model of AWS security illustrates this. AWS is a flexible platform for computing, and it provides ample flexibility to make mistakes. From avoiding the use of software with known vulnerabilities to maintaining proper user credentials, the AWS environment inevitably demands that users take care to secure the resources they initiate in their footprint.

Furthermore, AWS creates challenges for security that do not generally exist elsewhere. Interestingly, these new challenges have nothing to do with such concerns as "I do not own the hardware; hence, I cannot secure it physically." Rather, the challenges that arise involve developments such as the rate of change in both technology and scale and the changing nature of how software is developed and maintained. We will consider these later.

A simple way to think about security in AWS

There are two important questions to ask when securing your AWS footprint: How is it configured? and, What is it doing? If you can ask and answer these questions clearly, then you can trust that you are keeping up your end of the shared responsibility for security in AWS.

How is it configured? Knowing the configuration state of all your AWS resources is crucial. If you know the configured state of all your services, devices, users, and policy objects, then you can determine whether those states are consistent with your expectations, with best practices, and with respect to known problems and vulnerabilities. Understanding and critically examining configuration states, and demonstrating adherence to policy representations, make up the intellectual core of most forms of regulatory compliance for security and risk governance.

As discussed previously, AWS Config makes it easy to articulate and enforce adherence to policies governing asset creation, access, and use in a user-annotated way. Other services, such as Amazon Inspector, enable you to install an agent on each of your AWS servers so you can regularly verify that the server:

- Has an internal server configuration that is consistent with best practices
- Does not include software that exhibits a known vulnerability (as documented in the CVE archive)

The assessment capabilities of AWS Config and Amazon Inspector effectively automate your ability to track the configuration state of your AWS resources. Corrective action, such as software patches, credential renewal, and repairs of misconfiguration, can be taken without direct human effort. In summary, by tracking the configuration of your AWS resources, you can identify and correct known problems and avoid the corresponding security consequences.

What is it doing? Not all problems are known in advance. Unknown software vulnerabilities, stolen credentials, user misbehavior, and the unintended consequences of policy choices are all examples of circumstances that cannot be detected through configuration management and assessment. There is an important difference between what a resource is permitted to do and what behaviors a resource has been exhibiting. Most security problems can be traced to a resource behavior that was permitted by its configuration but still proved to be damaging.

Observing AWS resource behaviors is made possible by the extraordinary instrumentation and visibility of the AWS environment. Much of the value of such IT visibility is bound up with how it helps you detect problems. However, AWS visibility is without question a fire hose of information, and it is up to the consumer of the visibility information to identify any problems. And here is where Cisco Stealthwatch Cloud is uniquely positioned to help.

Stealthwatch Cloud offers entity modeling. Our entity modeling solution maintains a software model (that is, a near-real time simulation) of each of your AWS resources. These can include servers and users, along with AWS-specific resource types like security groups and auto-scaling groups. These models take as input the structured data feeds provided by AWS services, including VPC Flow Logs, AWS CloudTrail, Amazon CloudWatch, AWS Config, and Amazon Inspector. Entity modeling automatically discovers the role and behavior of your AWS resources. It then tracks that behavior continuously to detect when risky or threatening behaviors occur.

For example, suppose a server instance within a VPC should, according to policy intention, never be the destination for a remote login. Suppose further that a remote login did take place on that machine due to a mistaken change in firewall rule policy. Entity modeling would spot and report this activity (an "Unusual Remote Access") in near-real time and would point out the specific AWS CloudTrail API call (including username, date, and time, among other details) that triggered the change in the firewall rule.

Consider: How certain are you that there are no errors or unintended consequences in your software, policies, and configuration states? Are you so certain that you don't bother checking for mistakes and mishaps? Much as we would all love to have systems that are secure by virtue of their construction, the reality is that errors, misunderstandings, and misuse are the most common causes for security incidents. None of us can afford to operate our IT environments without monitoring them for threatening behavior.

Entity modeling can automatically detect several important classes of security problems such as, Did someone discover a backdoor in a software package we use? Does any third-party software or appliance in our footprint dial home? Is an authorized user abusing privileges? Has a configuration mistake been made, enabling remote access or another unintended use of resources? Entity modeling is a unique form of security automation that can discover a previously unknown problem with your people, processes, or technology.

The deeper answer

AWS enables effective security because visibility, identity management, and policy enforcement are comprehensive and present on day one. With technologies like Stealthwatch Cloud's entity modeling, you can find both known and unknown problems quickly, and you can achieve effective security outcomes. That's the short answer.

The deeper answer can be seen in the nature of cloud computing itself. The transition to the cloud is not a simple "move your servers to someone else's environment." New software architectures, and new habits and processes for organizing the activity of software developers, are creating substantive changes in how IT operates.

As a primary example, the so-called DevOps trend means that the traditional division of labor among development, Q&A, and operations has been collapsed into a single organization of developers. This is not a fad. The consolidation of labor is driven by a more sustainable and productive set of incentives. Software developers themselves now deal with testing and QA issues. They rotate the responsibility of triaging problems in production operations, and debugging and bug fixes are routed back to the developer who originated the code. Up and down the process, there is nowhere to hide, and everyone does their best at each stage of the process to avoid embarrassment and greater stress later. The result? Cloud-based development and IT teams deliver features faster and with fewer operational problems.

It is clear that these differences are not merely superficial. Consider why AWS itself is growing so dramatically. AWS has seen explosive growth because the companies that have embraced AWS have themselves been exhibiting explosive growth. AWS makes sense for companies that want to make use of AWS-specific services for purposes of efficiency, scalability, and service and feature agility. Anecdotally, most AWS customers pursue all three intentionally. It is not unreasonable to assume that these companies thrive because they are effective at achieving efficiency, scalability, and feature agility.

These modern, agile AWS-based companies are winning in the marketplace. We can now illuminate the deeper motivation for the qualitatively superior security in AWS: all aspects of AWS security must be designed with efficiency, scalability, and agility in mind. AWS-based companies demand it. In fact, in most AWS-based DevOps organizations, security and incident response activities are supported as an operations problem. Notifications, whether they represent security, operations, or software-correctness problems, are all generally triaged back to the DevOps engineer responsible for last changing the resource. Information security, specifically, has some downstream effort that in large organizations is handled by specialists. They deal, among other things, with the consequences of intrusions or breaches; they document incidents for reporting purposes; and they manage changes to policies and processes based on lessons learned.

But the point is that, increasingly, security incident response is being blended into DevOps in a manner analogous to what has happened to QA and standalone operations and for more or less the same reasons. No one is better positioned to understand, diagnose, and fix security problems than the developer who controls the function.

Conclusion

In summary, AWS environments that make use of AWS-native services and Cisco Stealthwatch Cloud's entity modeling can be protected from known and unknown security threats in a scalable and cost-effective manner.

If you think entity modeling might be a good fit for your AWS footprint, or if you have any questions, please consider starting a no-commitment free trial at: www.cisco.com/c/en/us/products/security/stealthwatch/stealthwatch-cloud-free-offer.html.

C11-739663-00 09/17