observable networks

Press your advantage
Know your endpoints

WHITE PAPER

A New Way to Look at AWS Security

By Patrick Crowley
Founder, CTO
Observable Networks

Executive summary

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

However, when viewed from the proper perspective, AWS can be seen to provide an environment for IT security qualitatively superior to any alternatives available to organizations smaller than AWS itself. AWS' built-in foundation of visibility, identity, and policy enforcement enables out-of-the-box detection, and hence, avoidance, of known problems. When combined with Observable Network's ability to detect unknown threats in the behavior of AWS resources, the result is a nimble, scalable, and cost-effective security solution for AWS customers.

Introduction

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

"AWS' built-in foundation of visibility, identity, and policy enforcement enables out-of-the-box detection, and hence, avoidance, of known problems. When combined with Observable Network's ability to detect unknown threats in the behavior of AWS resources, the result is a nimble, scalable, and cost-effective security solution for AWS customers."

However, when viewed from the proper perspective, AWS can be seen to provide an environment for IT security qualitatively superior to any alternatives available to organizations smaller than AWS itself. AWS' built-in foundation of visibility, identity, and policy enforcement enables out-of-the-box detection, and hence, avoidance, of known problems. When combined with Observable Network's ability to detect unknown threats in the behavior of AWS resources, the result is 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 today: most of the difficult legacy information security problems that large enterprises struggle with today 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.

Solved problems in AWS security

Visibility; identity and access management; and policy declaration and enforcement are legacy challenges in on-premise environments, despite years of dedicated and well-funded attempts at comprehensive solutions. In AWS, they are all solved problems.

Consider your on-premise network. Suppose that you had a structured audit trail whenever: a) a new device enters or leaves the network, b) a user authenticates on any device, c) a user makes use of an information service, d) someone makes changes to infrastructure elements like routing tables or firewall rules, e) someone modifies the security permissions or organizational role of a user or device, or f) one of your IT assets has a network interaction.

Suppose further that this information is provided to you, flexibly and programmatically, along with any other application-level logs that you care to monitor and any custom metrics you would like to capture that track activities amongst your users, applications, and IT resources. Well, if you had this for your on-premise network, then you would have more or less solved the visibility problem in IT. Of course, this is classically hard to solve on-premise because it is difficult to generate and aggregate logs from 100% of users and IT resources. Importantly, this is the visibility you have out-of-the-box in AWS. In AWS, visibility and instrumentation are solved problems. Outside of AWS, these problems are costly, difficult to solve in a satisfactory way, and impossible to solve comprehensively.

AWS CloudTrail, delivers a structured feed of all requests to access or modify your AWS footprint.

Amazon CloudWatch reports utilization and status events. It is also an open API that can be used by developers to add log and metric monitoring to custom applications and services.

VPC Flow Logs produces logs that represent an auditable record of all network interactions within your AWS virtual private cloud footprint.

To see why this is possible in AWS, it is necessary to understand both how AWS operates and which services AWS provides to support visibility and instrumentation. AWS operates as a service-oriented architecture (SOA), which means that all actions in your AWS footprint are initiated by authenticated API calls to web service endpoints. That is, when new user accounts are created, firewall rules are changed, and servers are instantiated, specific API calls are made with authenticated user credentials.

All changes to AWS resources are made via 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 them 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 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' built-in services, such as Amazon Elastic Compute Cloud (EC2) (servers), Amazon Relational Database Service (RDS) (databases), and Amazon Elastic MapReduce (EMR) (data analysis), use CloudWatch to report utilization and status events. Beyond built-in monitoring of AWS services, CloudWatch is also an open API that can be used by developers to add log and metric monitoring to custom applications and services.

Additionally, CloudWatch supports alarms and custom events to detect problem states and trigger automatic action. For example, CloudWatch alarms are used to enable auto-scaling groups, in which the number of servers used in an AWS footprint can be dynamically scaled up or down in response to a utilization metric maintained by CloudWatch. With CloudWatch, AWS enables visibility into the operation and activities of specific applications and services.

Second, VPC Flow Logs is a service that provides 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 is made that 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-premise network security will rightly 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, and 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. By contrast, obtaining similar levels of visibility in on-premise environments is infeasible for nearly every organization, regardless of size or resources available.

"AWS provides a built-in service for comprehensive policy declaration and enforcement that is, in practical terms, impossible to reproduce in on-premise environments in a similarly comprehensive way."

The preceding discussion of visibility has also illuminated 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 that is used to provide credentials for all aspects of AWS service interaction, and to declare which user identities exists and what privileges they possess in order to observe and manipulate your AWS footprint. It is possible to use AWS' native IAM service to manage your entire identity and access management workflow, and it is also possible to integrate IAM with a third-party service that you prefer. In any case, it is impossible for your AWS resources (servers, databases, storage, logs, policy objects, etc.) to be viewed or manipulated except via the IAM service. Like visibility, identity and access management is a solved problem in AWS.

Finally, AWS provides a built-in service for comprehensive policy declaration and enforcement that is, in practical terms, impossible to reproduce in on-premise environments in a similarly comprehensive way. AWS Config is a resource inventory and configuration service that provides both ad hoc and continuous auditing of AWS resources and their internal configurations.

Consider a simple example: suppose that you wanted to verify that user passwords were disabled on all of your servers, to ensure that only key-based access was 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 this because a) all changes to AWS resources are managed via authenticated calls to AWS endpoints, and b) 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 in order to automate the detection of policy violations. AWS Config Rules are, in effect, standing, 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, AWS Config makes it possible for users to 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! While these three security areas are legacy challenges that vex on-premise environments, they are by no means comprehensive.

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

Furthermore, AWS creates some new challenges for security that do not generally exist elsewhere. Interestingly, these new challenges have nothing to do with concerns that "I do not own the hardware, hence, I cannot secure it physically," but rather that the shift to AWS involves several other important changes that we will consider later, such as the rate of change in both technology and scale and the changing nature of how software is developed and maintained.

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 have the ability to 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 of your AWS resources is important. If we know the configured state of all of our services, devices, users, and policy objects, then we can reason about whether those states are consistent with our expectations, with best practices, and with respect to known problems and vulnerabilities. Understanding and critically examining configuration state, and demonstrating adherence to policy representations, is 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. Additional services, such as Amazon Inspector, enable you to install an agent on each of your AWS servers in order to regularly verify that the server a) has an internal server configuration that is consistent with best practices, and b) 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, so that corrective action, such as software patches, credential renewal, and repair of misconfiguration, can be taken without relying on direct human effort. In summary, by tracking the configuration of our AWS resources, we can identify and correct known problems and avoid the corresponding security consequences.

"We were looking for a better way to monitor network traffic in virtual private clouds. Dynamic Endpoint Modeling was the only solution that provided visibility into all our devices in our public cloud infrastructure and their network activity."

Taylor Higley
Director of Information Security
AFGE

"What is it doing?" Of course, 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, and can lead to severe security problems. There is an important difference between "what is a resource permitted to do," and "what behaviors has a resource been exhibiting" because, in fact, most security problems can be traced back to a resource behavior that was permitted by its configuration but still proved to be damaging.

In this direction, the extraordinary instrumentation and visibility of the AWS environment make the observation of AWS resource behaviors possible. In fact, much of the value of IT visibility is bound up with how it enables the detection of problems. However, AWS visibility is without question a fire hose of information, and it is up to the consumer of the visibility information to determine when problems arise. And here is where Observable Networks is uniquely positioned to help.

It is precisely for this purpose that Observable Networks provides Dynamic Endpoint Modeling. Our Dynamic Endpoint Modeling solution maintains a software model, i.e., a near-real time simulation, of each of your AWS resources, including 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. Dynamic Endpoint Modeling automatically discovers the role and behavior of your AWS resources, and then tracks that behavior continuously through time in order 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 log-in. Suppose further that a remote log-in did take place on that machine due to a mistaken change in firewall rule policy. Dynamic Endpoint Modeling would spot and report this activity (an "Unusual Remote Access") in near-real time, and would furthermore point out the specific AWS CloudTrail API call (including user name, 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 state? Are you so certain that you wouldn't bother checking for mistakes and mishaps? As much as we would all love to have systems that are secure by construction, the reality is that errors, misunderstandings, and misuse are the most common causes for security incidents, so none of us can afford to operate our IT environments without monitoring for threatening behavior.

Dynamic Endpoint Modeling can automatically detect several important classes of security problems such as: did someone discover a backdoor in a software package we use? Do 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 other unintended resource use? Dynamic Endpoint Modeling is a unique form of security automation that can discover when there is a previously unknown problem with your people, processes, or technology.

The deeper answer

AWS enables qualitatively better security because visibility, identity management, and policy enforcement are comprehensive and present on day one. With technologies like Observable's Dynamic Endpoint Modeling, both known and unknown problems can be found quickly, and can achieve security outcomes that are difficult to match outside of AWS. 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 to how IT operates.

As a primary example, the so-called DevOps trend means that the traditional organization of labor between development, Q&A, and operations has been collapsed to a single organization of developers. This is not a fad; in fact, the development is driven by a fundamentally more sustainable and productive set of incentives. Software developers themselves deal with testing and QA issues as a part of the development task, triaging problems in production operations rotates amongst the development team itself, 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 each person has built-in incentives to do their best at each stage of the process in order to avoid embarrassment and greater stress later. The result? Cloud-based development and IT teams deliver features with greater velocity and fewer operational problems.

At this point in time, it is clear that these differences are not merely superficial. Consider: why is AWS itself growing so dramatically? AWS has seen explosive growth only because the companies that have embraced AWS have, themselves, been exhibiting explosive growth. Why might that be the case? Well, consider, that AWS-based companies are a self-selecting group that has chosen AWS rather than on-premise infrastructure, traditional co-location facilities, or generic virtual computing environments. AWS only makes sense for companies that want to make use of AWS-specific services for purposes of efficiency, scalability, and/or service/feature agility. Anecdotally, most AWS customers pursue all three intentionally. So, why is it that these companies are thriving? It is not unreasonable to assume that these companies thrive because they are more effective at achieving efficiency, scalability, and feature delivery as compared to their competition.

Observable is a cloud-native company, but we serve customers on-premise, in AWS, and in hybrid deployments where both on-premise and AWS footprints are monitored for one organization. And while there are technical reasons that we can deploy in a matter of minutes in AWS as opposed to hours on-premise, the deeper reason we deploy more often and faster in AWS is because *AWS-based organizations are designed to move faster while moving safely* as compared to their on-premise counterparts.

These modern, agile AWS-based companies are the organizations that are winning in the marketplace. So, how does security fit in? Now, we can 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, such as dealing with the consequences of intrusions or breaches, documenting incidents for reporting purposes, and managing 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 Observable Network's Dynamic Endpoint Modeling can be protected from known and unknown security threats in a scalable and cost-effective manner that is difficult to match outside of AWS.

If you think Dynamic Endpoint Modeling might be a good fit for your AWS footprint, or if you have any questions, please do follow up with me at patrick.crowley@observable.net and consider starting a no-commitment free trial at http://observable.net.

Patrick Crowley

Patrick Crowley is founder and CTO of Observable Networks. He is also professor of Computer Science & Engineering at Washington University in St. Louis.

About Observable Networks

Observable Networks, Inc. is an emerging leader of network security technology and advanced threat detection services that identify compromised and misused networked devices currently escaping detection by existing network security tools. Observable's endpoint modeling technology includes a cloud-based service platform incorporating automated security analytics and real-time traffic sensors that continuously model all devices on networks of any size. Observable empowers organizations to understand normal and abnormal device behaviors in their networks, helping them to identify potential threats and facilitate faster remediation. Observable Networks is a privately held company headquartered in St. Louis, MO. For more information, please visit www.observable.net.

observable networks

For further information contact us at
info@observable.net or visit **www.observable.net**