12 Best Practices to Enhance the Security of your AWS Configurations

Amazon Web Services (AWS®) has been an advocate for strong cloud security since it launched in 2006. AWS and its customers employ the shared responsibility model, which distributes security roles between the provider and customer. As a cloud vendor, AWS owns the infrastructure, physical network, and hypervisor. The enterprise owns the workload operating system, apps, virtual network, access to their tenant environment/account, and the data.

This model distinguishes between AWS and cloud users in how they manage security. Though the distinction is clear, it doesn't emphasize the complexity of the issue of security management. Especially as organizations move more workloads to AWS, it quickly becomes impossible for human management to cover all the potential risk elements.

> "Through 2022, at least 95% of cloud security failures will be the customer's fault."
>
> —Smarter with Gartner[1]

According to Gartner, "In nearly all cases, it is the user—not the provider—who fails to manage the controls used to protect an organization's data." Jay Heiser, research vice president at Gartner, advises CIOs that they should not be holding back on their cloud initiatives over concerns about cloud security. Instead, they must change their line of questioning from "Is the cloud secure?" to "Am I using the cloud securely?"[2]

To maintain a rigorous security posture across their cloud environments and abide by the shared responsibility model, today's organizations must be disciplined about applying cloud security best practices and accompany their efforts with automated, continuous monitoring.

> "AWS will maintain the strong security and compliance controls across their entire infrastructure platform—data center controls; core network/hardware controls; operational security practices, like change control; and others. Your job, as a customer of AWS, is to manage anything you manipulate on their platform."
>
> —Matt Chiodi, Cloud CTO, Palo Alto Networks

Within the AWS environment, there are all kinds of vulnerabilities that require continuous attention. Misconfigured servers, open Amazon S3 buckets, unsupervised traffic, and a host of other issues must be identified and addressed before they create major risks for an enterprise.

Out-of-the-box AWS configurations can help, but only to a point. The cloud is dynamic, and because it's constantly shifting in response to internal change and customer needs, it demands continuous monitoring and guidelines for responsive remediation.

This guide is intended to help you enhance your AWS environment through 12 critical steps. With these, you can develop a disciplined framework for your team and create a stronger security posture for your data and IT assets.

Best Practice No. 1: Enable AWS CloudTrail

AWS CloudTrail® is a tool that allows you to record API log information for security analysis, change tracking, and compliance auditing. With it, you can create a trail of breadcrumbs to lead you back to the source of any changes made to your AWS environment.

AWS is an API-driven environment, so even if you use the AWS console, API calls are still being made behind the scenes on your behalf. When enabled, AWS CloudTrail logs details such as the time of the call; who made it, even if it is AWS or a third party; where it was made from, using the IP address; and whether it was successful.

When choosing whether to enable AWS CloudTrail, there are a few factors to consider:

- **Access control:** Who has access to the logs, and what do they want to do with this information? It is recommended that access to the AWS CloudTrail API be removed from all users except administrators, and that admin access be protected with multi-factor authentication (MFA). The AWS CloudTrail API should then be ready for administrators to "set it and forget it."
- **Log storage:** For what length of time do you intend to store the logs? Log files contain little data, so they compress easily and keep storage costs low, but they can add up over time if not monitored. Logs are typically kept for 30–90 days on Amazon S3, though some organizations choose to keep them for extended periods. To help manage your storage needs and costs, you can employ the Amazon S3 lifecycle policy on the bucket with your AWS CloudTrail data, automatically purging older files.
- **Regions:** In what AWS regions will you employ AWS CloudTrail? You can enable AWS CloudTrail throughout all AWS regions in which you operate with just a few clicks. By enabling AWS CloudTrail as you enter a new AWS region, you can save your organization time by eliminating the need to retroactively authorize its use.
- **API call logs:** Where will you log API calls? By employing AWS CloudTrail, you can save your organization money on storage by logging global API calls in only one region. Logging global calls in a specific AWS region means you don't need to search through all your regional data for a specific global call log, optimizing your system.
- **Data encryption:** How and when do you want to encrypt your data? We all know encrypting your data is essential, but organizations are often inconsistent when they start doing so. By using AWS CloudTrail as well as Amazon S3 server-side encryption and Amazon Key Management Service, you can encrypt both in-flight and at-rest data in your logs at the start of its activity, eliminating the challenges of going back to re-encrypt data after the fact.

Following this best practice and enabling AWS CloudTrail can save your organization time and money.

Best Practice No. 2: Disable Root API Access and Secret Keys

AWS provides a tool called AWS Identity and Access Management (IAM), which can administer access rights so root users can be assigned limited access but remain equipped to do the work required of their roles.

1. "Is the Cloud Secure?" Gartner, March 27, 2018, https://www.gartner.com/smarterwithgartner/is-the-cloud-secure.
2. Ibid.

A root user is a user with full permission to view and change settings and configurations within an environment. Root user accounts are usually created to give access to the system for administrative functions, such as gathering information on billing and activity. It is often considered standard to grant these users unlimited access, but that broad level of control is not always needed.

With AWS IAM, users must be explicitly granted access to perform functions; no user is granted automatic access to anything. This allows companies to increase agility without incurring additional risk.

Removing root access from the system is a simple action that returns multiple security benefits. In addition to creating a more secure system overall, removing root and granting IAM access improves productivity of DevOps and product teams by letting them operate securely through the convenience and immediate management of their own AWS infrastructure security.

Best Practice No. 3: Enable MFA Tokens

Businesses need more than the single layer of protection provided by usernames and passwords, which can be cracked, stolen, or shared. Keep in mind that AWS IAM controls may provide access to not just the infrastructure, but also the applications installed and the data being used. By implementing MFA—a security measure that requires further proof of identity, such as a code, to be provided in addition to the password—you can assign roles, root accounts, and IAM users securely.

MFA is commonly provided by way of physical or virtual devices separate from users' username/password combinations. These devices generate random values to supplement the basic credentials, helping validate the identity of the person trying to access a cloud environment or some part of it. MFA devices can be separate, physical items or managed as smartphone apps. Aside from standard ID/password alphanumeric values, the biometrics required by some MFA continue to gain popularity, with many devices scanning various aspects that uniquely identify a user, such as retinas, fingerprints, or other unique features.

Given the potential risk, an additional layer of security here makes good business sense. AWS is MFA-friendly, but it's important that organizations use AWS guidelines to ensure compatibility of physical devices with their corresponding cloud infrastructure. When choosing an MFA product, consider both how it will integrate into your workflow and how to recover if, for instance, an MFA token is lost or users replace their mobile devices.

Best Practice No. 4: Reduce the Number of IAM Users with Admin Rights

By limiting administrator access and aligning permission grants to the appropriate level of authority, you can minimize the risks of allowing too many users with administrator-level permissions.

When administrative users enter a company, they often introduce their own processes to security systems. As these administrators leave, it can often be difficult for new personnel to keep track of all the procedures in place, potentially leaving a company exposed to security risks.

By closely monitoring access levels and granting a limited number of users administrative access, you can optimize your security posture.

Best Practice No. 5: Use Roles for Amazon EC2

Using advanced technology such as IAM can help an organization eliminate the risks of security compromises.

Historically, security breaches have often occurred because users store their credentials in insecure locations, such as GitHub® repositories. With advances in IAM, roles can now be granted temporary security tokens that will allow them to perform specific functions. With these tokens, roles can be used in combination with external identity providers, such as Security Assertion Markup Language, as well as to allow third parties to access resources. Since the access keys are not static, there is no need to store them. If a token is compromised, a new one can be issued, and if a token expires, the credential can be rotated.

With roles, users with lower levels of access can conduct tasks in Amazon Elastic Compute Cloud (Amazon EC2®) without needing to be granted excessive levels of access. This approach allows very specific access to AWS services and resources, reducing the possible attack surface area available to malicious actors.

Best Practice No. 6: Employ Least Privilege—Use Strong Policies to Limit What IAM Entities Can Do

By employing IAM, you can limit the access of root users within your organization to the lowest level needed, minimizing the potential security risks to your company. Using the "least privilege" method to determine the lowest levels of access users need, you can eliminate the likelihood of compromising your company's security position if a key is lost or stolen. In addition to the risks posed by users, network administrators must also keep in mind what applications and groups are granted permission to access their networks.

Limiting access based on the least privilege and IAM policies of your organization allows you to put layers of restrictions in place. These layers can act as the foundation of your organization's security, enabling you to eliminate risks posed to your network.

Best Practice No. 7: Ensure Access Logging Is Enabled on the Amazon CloudTrail S3 Bucket

If you use an Amazon S3 bucket to store your CloudTrail logs, you're going to want to maintain records about all activity that touches or affects CloudTrail. With logging settings applied to the relevant Amazon S3 bucket, you'll be able to track access requests as well as maintain a record of those who have access and the frequency with which they're using it.

Since CloudTrail buckets contain sensitive information, these should be protected from unauthorized viewing. With Amazon S3 Server Access Logging enabled for your CloudTrail buckets, you can track any requests made to access the buckets. You can even limit who can alter or delete the access logs to prevent users from covering their tracks.

The corresponding information will include sensitive data and need to be protected. Like any other settings in AWS, this will provide more than a mechanism for insight into activity.

It will also give you a way to modify, remove, or grant access to logs to prevent users from making unauthorized changes to hide their identities or activities.

Best Practice No. 8: Rotate Keys Regularly

With Amazon EC2, systems running processes outside of it require keys, helping keep your system secure. Although roles remove much of the need to manage keys, API keys should still be employed and regularly rotated. By rotating API keys regularly, you can control the amount of time for which a key is considered valid, limiting the impact to business if a compromise occurs.

The process of rotating keys does require some manual labor, but it takes only minimal effort from an administrator. With a system that is easily replicated, the process of alternating your keys every 90 days can be simplified by incorporating encrypted data snippets, such as Chef's encrypted data bags.

Best Practice No. 9: Apply IAM Roles with STS

Using roles for Amazon EC2 instances makes it easier for your resources to communicate securely and helps you reduce management burden by using the AWS Security Token Service (STS).

How often can you simultaneously become more secure and simplify management? These may sound like opposites, but they are what we strive for. Anytime you can make it easier for users to be more secure, you're more likely to get adoption. Conversely, if you make security too complicated, it may reduce security in practice because people are inclined to take shortcuts.

At this point, we want to extend the roles for Amazon EC2 by using roles for your IAM users, making the instance more secure in a way that's easier to maintain. We could go through each individual AWS account to create new users, generate passwords, and restrict permissions—much like we had in the beginning—but that would be a lot of repetitive, manual steps. It's better to leverage the users we've already created and secured, and just grant them access to the additional accounts.

AWS provides a quick walkthrough to help you get started with delegating access to AWS accounts from IAM users in another account. In many cases, you may even have a master AWS account, with resources running in it, that you only use for administrative control and billing access.

If you have more than one AWS account, it's worth the time to go through the steps outlined by AWS to get a good feel for what you can accomplish by making use of roles with your IAM users.

Best Practice No. 10: Use Auto Scaling to Dampen DDoS Effects

In addition to security, there are also many configuration best practices to follow. Per the shared responsibility model, AWS is committed to the security of the AWS Cloud and responsible for the foundation upon which applications on AWS are built.

However, as a user, you're responsible for securing the product you create, the data you store on AWS, and how applications react to denial-of-service (DoS) and distributed denial-of-service (DDoS) attacks.

DoS and DDoS attacks are attempts to make applications or websites inoperable by means of overwhelming them with traffic from multiple sources. DoS attacks are typically conducted by a single attacker, whereas DDoS attacks are characteristically directed by a group of collaborating parties.

The public nature of AWS makes any resource deployed on the system a target for DDoS attacks, so tools such as Elastic Load Balancing and Auto Scaling help prevent DDoS attacks from succeeding. Check out the "AWS Best Practices for DDoS Resiliency" white paper on how to mitigate DDoS attacks.

Although some organizations use application firewalls to deny DoS and DDoS attacks, Auto Scaling is a simpler, more cost-effective way to absorb such attacks and prevent service interruption. Most website traffic is directed by an elastic load balancer (ELB). As traffic to your website increases, the ELB undergoes corresponding scaling. Since ELBs only direct Transmission Control Protocol (TCP) traffic, any attacks that use a protocol other than TCP will not reach your applications.

When TCP traffic enters your website, the data within that traffic must be analyzed. As the amount of data increases, you must be able to scale your ELB efficiently to avoid service interruption. To effectively analyze and accommodate your scaled ELB, you need a system with the capacity to scale with it automatically. Amazon EC2 has the capability to automatically employ its Auto Scaling function.

For example, let's say you set a condition for Auto Scaling to launch two new Amazon EC2 application instances when the amount of network activity crosses a certain threshold. This trigger would already allow your site to scale based on normal, legitimate demand.

However, if abnormal attack traffic were to enter this site, Auto Scaling would also trigger a scale-up event, launching new Amazon EC2 instances to meet demand and process requests. This means your service remains operational during the attack and business can continue as normal.

Once the attack is over, Auto Scaling will automatically scale down the number of Amazon EC2 instances running simultaneously, if configured to do so. Although Auto Scaling means the price you pay for the increase in instance hours will rise, uninterrupted business operation is priceless. Auto Scaling acts an insurance policy that saves your company revenue, which justifies it in the end.

Best Practice No. 11: Enable Security Measures When Auto Scaling Is Not an Option

Auto Scaling is undoubtedly a best practice, but it may not be financially feasible for all organizations. For those without the resources to employ Auto Scaling, AWS Security Groups and Network Access Control Lists (NACLs) provide a way to block hazardous website traffic that would otherwise trigger a need to scale.

Though they were not built to act as traditional firewalls, AWS Security Groups can provide the same functions as firewalls by allowing companies to simultaneously control network traffic accessibility and costs.

Much like a basic firewall, the principle purpose of an AWS Security Group is to allow traffic you want to admit to enter your network and move in the direction you would like it to flow. Depending on where the traffic is coming from, by employing an AWS Security Group, you can prevent malicious traffic from reaching your systems, stopping that traffic before attacks can commence.

Many companies deliver their products through web applications. For basic web applications, you may want to allow access by all types of traffic, since not admitting this commerce could be detrimental to your business's success.

In instances such as these, you should restrict your network's response capabilities. To minimize the risk posed, you can effectively restrict your response to the two most common port numbers: port 80 for unencrypted web traffic and port 443 for encrypted traffic.

Most customers who access your application will never need administrative rights to your network, but a select few may. To continue to secure your network against compromise, you have two options: deploy infrastructure as code and never oversee a box manually; or only allow access from the origin IP and port your instance uses, and employ it when needed.

By only allowing access from the origin IP and port address through which you administer the instance, you can allow customers administrator access while maintaining your security posture. This method eliminates some of the convenience of automation, but you can ultimately reduce the attack surface area as well as the amount of time you're exposed to potential attacks. This process can be scripted into your system for further convenience.

Is it ever acceptable to allow other ports access to your network? For some companies, the answer is yes.

NACLs have both inbound and outbound rules that act as defensive layers against malicious actors, preventing their traffic from affecting your network.

Figure 1 shows Security Groups and NACLs in an Amazon Virtual Private Cloud (Amazon VPC®). Keep in mind that if you're in Classic Amazon EC2 and not Amazon VPC, you only have access to AWS Security Groups.

As you can see, AWS Security Groups are closest to your application while NACLs are closest to your inbound traffic. From a defense-in-depth perspective, you want to limit most traffic the farthest from your application, becoming more granular in the amount of access you grant.

In the context shown in figure 1, where should the IP address 0.0.0.0/0—here representing all users attempting to access your network—be configured?

If you must allow unlimited access to your application, you will need to configure your network for both an NACL and an AWS Security Group.

Often, you'll have to create special permissions for both an NACL and the AWS Security Group. By employing an AWS Security Group, you can limit the ports to which the world has access.

If you only need to allow specific network access to your application, you can limit access through the NACL, preventing anything you do not define from making it into your AWS Security Group.

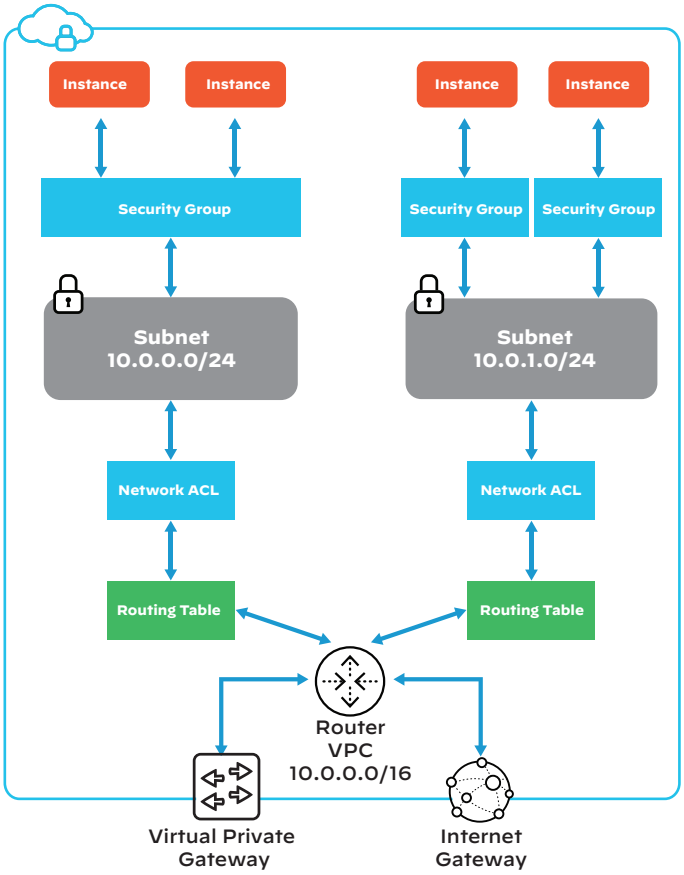

Figure 1: AWS Security Groups and NACLs in an Amazon VPC

The NACL can also be thought of as the perimeter boundary, or your first checkpoint. With an NACL, you can configure an explicit deny rule to deny traffic.

AWS Security Groups are very focused on what you allow, denying everything that is not allowed. With an NACL configured, if you find you are receiving a DoS attack and the origin IPs are very specific, you can quickly and easily block it.

Although this process isn't quite as seamless for a DDoS attack, if the attack is not evenly distributed, the set deny rules will likely still be able to hinder part of the attack. When employing either rule, being familiar with your logs and partnering with AWS Support will help you identify and repel attacks. Inbound rules are important, but the security of your network also relies on the outbound rules you've set in place. Configuring outbound allow rules, the process for which is no different from the process for configuring inbound rules, can be done with both AWS Security Groups and NACLs.

Once an AWS Security Group or NACL has been configured with implicit allow rules, anything not configured will be denied. Additionally, it's important to keep in mind that AWS Security Groups are stateful while NACLs are not. As you configure outbound rules on your AWS Security Groups, they will not affect inbound sessions, but if you configure outbound rules on an NACL, you will need to allow the outbound traffic back to the origin IP to establish a session. It may be easier to configure your system so that AWS Security Groups use ports to access your network but NACLs are limited to

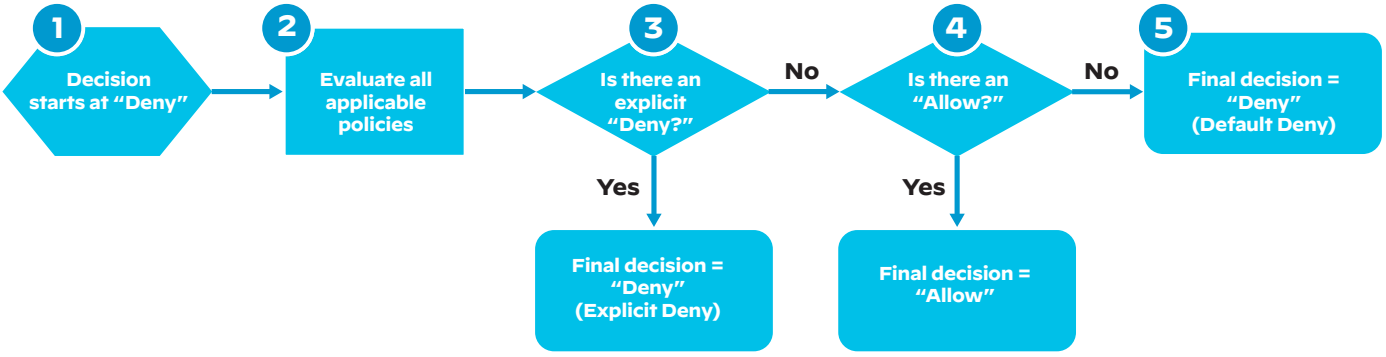

Figure 2: Decision model

your network. An AWS Security Group can be configured per application, whereas NACLs are designed to implement explicit rules for web applications.

For a two-tiered web application with web servers in one group and a database in another, security practices could be implemented by configuring inbound NACL rules to allow connections to web servers from around the world, while the database would only allow inbound connection from an established list of web servers. In this instance, web servers would only allow port 443 connections, while the database would only allow inbound 3306 connections for MySQL.

For outbound connections, you could remove both the web server and the database AWS Security Groups outbound rule to prevent them from initiating connections over the internet. The web servers would allow all outbound traffic out to ensure sessions could be established, and the database would limit outbound connections to the web server's private subnet IP range.

Best Practice No. 12: World-Readable and Listable Amazon S3 Bucket Policies

Although IAM policies are built to provide security, organizations must take careful measures to ensure the stability of their platforms in the long run. As companies grow, they often add access control measures to their networks to keep up with the increasing demands placed on them.

As their networks expand, organizations often layer newer platforms on top of older systems, making it difficult to keep track of who or what is allowed access to their network. A great example of this can be found in the AWS Security Blog post "IAM Policies and Bucket Policies and ACLs! Oh, My! (Controlling Access to Amazon S3 Resources)."

As new stakeholders enter the organization, older methods become challenging to manage and difficult to visualize, sometimes presenting opportunities for administrators to lose track of security checks for older products.

Most security functions are built with default deny rules, but these rules can be circumvented if an organization fails to keep track of where or what it is allowing when new systems are built on top of older ones.

To avoid internal incidents that result from multiple products being used simultaneously, AWS recommends the best practice of choosing and sticking with one product. When an organization takes the time to carefully select and maintain a system, security will act as it is expected to (see figure 2).

Secure Your AWS Cloud Today!

Check out Palo Alto Networks offerings on the AWS Marketplace.

About Palo Alto Networks

Palo Alto Networks, the global cybersecurity leader, is shaping the cloud-centric future with technology that is transforming the way people and organizations operate. Our mission is to be the cybersecurity partner of choice, protecting our digital way of life. We help address the world's greatest security challenges with continuous innovation that seizes the latest breakthroughs in artificial intelligence, analytics, automation, and orchestration. By delivering an integrated platform and empowering a growing ecosystem of partners, we are at the forefront of protecting tens of thousands of organizations across clouds, networks, and mobile devices. Our vision is a world where each day is safer and more secure than the one before. For more information, visit www.paloaltonetworks.com.

paloalto NETWORKS

3000 Tannery Way
Santa Clara, CA 95054

Main: +1.408.753.4000
Sales: +1.866.320.4788
Support: +1.866.898.9087

www.paloaltonetworks.com