evident.io

TOP TEN BEST PRACTICES FOR AWS SECURITY

John Martinez - Principal Solutions Architect - Evident.io
John Robel - Principal Solutions Architect - Evident.io

amazon web services

May 2016

CONTENTS

THE STATE OF AWS SECURITY

"In the old days, companies were more concerned about the public cloud not being secure," said Phillips. "Today, most of their security breaches are internal. They now view the cloud, particularly AWS, as more secure than their own facilities."

Since its launch in 2006, Amazon Web Services (AWS) has worked tirelessly to improve the reputation of cloud security. While the cloud was once thought of as unstable and unsafe, developments in cloud technology have changed the way that companies think about storing and securing their information.

In a recent article, Charles Phillips, CEO of the enterprise application provider Infor, stated that his company is working to move clients onto the cloud at an aggressive pace, as the security of off-premises infrastructure is no longer a concern.

The virtual and elastic natures of public cloud infrastructure make security and compliance related tasks easier than ever when compared to operating in a traditional datacenter. Tasks that are often challenging in a traditional environment - such as gathering lists of assets, reconfiguring firewall rules, and copying hard drives for forensic analysis - can be performed with just a few clicks in an Infrastructure-as-a-Service (IaaS) console, or programmatically via scripts and Application Programming Interfaces (APIs).

The IaaS Security Challenge

In an April 2015 report titled, "Best Practices for Securing Workloads in Amazon Web Services," Gartner analysts Neil MacDonald and Greg Young write, "Through 2020, 80% of cloud breaches will be due to customer misconfiguration, mismanaged credentials or insider theft, not cloud provider vulnerabilities."

While extremely powerful from a management and automation perspective, if roles are not properly configured, a malicious insider can employ an IaaS management console or API to hinder operations.

Evident.io's Top Ten AWS Security Best Practices

Fortunately for organizations today, users have come a long way towards securing the infrastructure that they introduce on the cloud.

By using this guide and following the AWS Security best practices mentioned in the following pages, Evident.io shows you how to successfully transition and operate your business on the cloud. Below is our list of the top ten most important security practices that every organization using AWS should consider adopting.

BEST PRACTICE #0: ENABLE AWS CLOUDTRAIL & ENCRYPTION

Moving your architecture to AWS means that your company gets to reap the rewards of the services that are available to those on the cloud. One of these exclusive rewards is having access to AWS CloudTrail, but what is it?

AWS CloudTrail is a tool that allows you to record API log information for security analysis, change tracking and compliance auditing. Much like the story of Hansel and Gretel, with AWS CloudTrail you can create a trail of breadcrumbs that will lead you back to the source of any changes that have been made to your AWS environment.

AWS is an API driven environment so even if you use the AWS console, behind the scenes API calls are still being made on your behalf. When enabled, AWS CloudTrail logs details such as the time of the call, who made the call (even if it is AWS or a third party) where the call was made from (using the Internet Protocol (IP) Address) and whether or not it was successful.

When choosing whether or not to enable AWS CloudTrail, there are a couple of factors to consider. The first is access control, who has access to the logs and what do they want to do with this information? Last year, Evident.io published a blog on the importance of protecting AWS CloudTrail data that highlights the necessity of employing an Amazon Simple Storage Service (Amazon S3) bucket and ensuring that your data is encrypted and secure.

The second factor to consider is the length of time that the logs will be stored. Log files contain little data, so they compress easily and keep storage costs low, but can add up over time if not monitored. While typically kept 30-90 days on Amazon S3, some organizations do chose to keep these files for extended periods of time. To help manage your storage needs and cost, AWS users can employ the Amazon S3

lifecycle policy on the bucket with their AWS CloudTrail data, automatically purging older files.

The third factor to keep in mind is what AWS Regions AWS CloudTrail will be employed in. With its easy-to-use interface, you can enable AWS CloudTrail throughout all of the AWS Regions that you operate in with just a few clicks of your mouse. By enabling AWS CloudTrail as you enter a new AWS Region you can save your organization time by not having to retroactively authorize its use.

The fourth factor to consider is what AWS Region you will log your global API calls in. By employing AWS CloudTrail you can save your organization money on the associated costs for storage by only logging global API calls in one region. Logging global calls in a specific AWS Region prevents you from searching through all of your regional data files for a specific global call log, and optimizes your system.

The final factors to consider are how and when you want to encrypt your data. While we all know that encrypting your data is essential, organizations are often inconsistent when they start encrypting their data. By using AWS CloudTrail as well as Amazon S3 Server Side Encryption and Amazon Key Management System (Amazon KMS), you can encrypt both in-flight and at-rest data in your logs at the start of its activity, eliminating the challenges posed by having to go back and re-encrypt data after the fact.

By following our best practice number zero and enabling AWS CloudTrail, you can save your organization time and money. For more information on how to enable AWS CloudTrail check out these user-friendly how-to guides that outline how to setup your system in just a matter of minutes.

BEST PRACTICE #1:
DISABLE ROOT API ACCESS AND SECRET KEYS

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

With the introduction of AWS Identity and Access Management (IAM) the need for root users who seemingly have unlimited access are over.

A root user is a user with full permission to view and change anything within their environment. Usually root user accounts are created to give access to the system for administrative functions such as gathering information on billing and activity.

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

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

BEST PRACTICE #2: ENABLE MFA TOKENS

Businesses need more than just the single layer of protection provided by usernames and passwords, which have the chance to be cracked, stolen or shared.

By implementing a multi-factor authentication (MFA) token, a security measure that requires that a code be provided in addition to the password, you can assign roles, root accounts, and IAM users securely.

With the option of being able to employ the token in the form of a physical card or as a smartphone application (app), securing your network is both easy and convenient.

BEST PRACTICE #3: REDUCE NUMBER OF IAM USERS WITH ADMIN RIGHTS

By limiting the access of administrators and aligning permission grants to the appropriate level of authority, organizations can minimize the risks caused by allowing too many users with admin-level permissions.

When administrative users enter a company, they often introduce their own processes to security systems. As these administrators leave and time progresses, it can often be hard for new personnel to keep track of all of the procedures put 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 platform.

BEST PRACTICE #4: USE ROLES FOR AMAZON EC2

With advances in IAM, roles can now be granted a temporary security token that will allow it to perform a certain function.

By using advanced technology such as IAM, organizations can help eliminate the risk 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 a temporary security token that will allow it to perform a certain function. With that token, the role can be used in combination with external identity providers, such as Security Assertion Markup Language (SAML), and can also be used to allow third parties to access resources. Since the access key is not static, so there is no need to store it; if a token is compromised, a new one can be issued; and if a token expires, the credential ca rotated.

Through the use of a role, a user with a lower level of access can conduct tasks in in Amazon Elastic Cloud Computer (Amazon EC2) without needing to be granted excessive levels of access. This approach allows very specific access to AWS services and resources, which reduces the number of possible attack surface areas from bad actors.

BEST PRACTICE #5: LEAST PRIVILEGE—LIMIT WHAT IAM ENTITIES CAN DO WITH STRONG POLICIES

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.

By using the 'least privilege' method to determine the lowest levels of access needed by a user, you can eliminate the likelihood of comprising 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 my mind what applications and groups are granted permission to access their networks.

Limiting access based upon the least privilege and IAM policies of your organization allow layers of restrictions to be set in place. When these layers of security are put in place they can act as the foundation of your organization, enabling you to eliminate risks posed to your network.

BEST PRACTICE #6: ROTATE KEYS REGULARLY

By rotating API keys regularly, organizations are able to control the amount of time that a key is considered valid, limiting the impact to business if a compromise occurs.

While roles remove much of the need to manage keys, API keys should still be employed and regularly rotated.

By rotating API keys regularly, organizations are able to control the amount of time that a key is considered valid, limiting the impact to business if a compromise occurs. With Amazon EC2, keys are required for systems running processes outside of it, helping to secure your system.

While the process of rotating keys does require some manual labor, it takes minimal effort for an administrator. By creating 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 #7: APPLY IAM ROLES WITH STS

We are more than half way through the top ten, so let's finish up the IAM discussion before jumping into some of the top AWS configuration areas. This post will wrap up our discussion of individual people controls leveraging the IAM service by leveraging roles to simplify and create a more secure environment.

Earlier in the series, we covered how and why you should use roles for Amazon EC2 instances. The premises for this was to make it easier for your resources to communicate securely and reduce the management burden by leveraging the AWS Security Token Service (AWS STS).

How often is it that you are able to both become more secure and simplify management? While they sound like opposite pairs, they are actually what we strive for.

Any time you can make it easier for users to be more secure, you are more likely to get adoption. Whereas, if you make security too complicated, it may actually result in less security, in practice.

As an example, if you forced all users to use a randomly-generated 24-character password that was impossible to memorize, how many would revert to writing their password down or storing it someplace that may not meet security guidelines?

Sure, the password itself may seem to increase security, but in practice, it could create bad habits and reduce security.

Today, we are going to extend the roles for Amazon EC2 and talk about using roles for your IAM users. Again, this is to make it more secure and easier to maintain that security. In this example, I'll use the Evident.io Demo AWS accounts.

When Evident was just starting out as a company, there was a single AWS account used to demonstrate the Evident Security Platform (ESP) and its ability to create custom validations, security checks and

integrations. A single AWS account with a couple engineers, that should be pretty easy to secure right?

We disabled root API access and ensured there were no secret keys. Then for the two admins, we enabled MFA tokens. For some of the sales folks, we even created IAM users with read-only permissions and specific policies, so they wouldn't get in trouble. Leveraging the built-in ESP IAM Credential Rotation security check, we stayed on top of any keys that needed to be rotated.

Now, ESP was designed to be an enterprise platform and support customers with tens, hundreds and even thousands of AWS accounts (there's a blog about segregation of duties on ESP.)

To extend our demo, we added a handful of AWS accounts and, by this time, the number of people that had access to the demo account was growing. We could go through each and every AWS account and create new users, generate a password, and restrict that user's permissions much like we had done in the beginning, but that seemed like a lot of repetitive, manual steps.

A better option was to leverage the users we already created and secured, and just enable them to have access to the additional accounts. It sounds pretty easy, and in practice, it is.

AWS provides a quick walkthrough to help you get started in delegating access to AWS accounts from IAM users in another account. Now, in many cases, you may even have a master AWS account that has no resources running in it that is just used for administrative control and billing access.

In our case, we extended the one AWS account to allow the engineers and sales folks the same access to the other accounts with a few mouse clicks in the console.

If you have more than one AWS Account, it is worth the time to go through the steps outlined by AWS to get a good feel for what you can accomplish by leveraging roles with your IAM users.

BEST PRACTICE #8:
USE AUTOSCALING TO DAMPEN DDOS EFFECTS

This means your service remains operational during the attack and that business can continue as normal.

In addition to security, there are also many configuration best practices to follow.

As a part of the Shared Security Responsibility model, AWS is committed to the Security of the AWS Cloud and is responsible for the foundation on which applications on AWS are built.

However, as a user you are responsible for securing the product you create or the data you store on AWS, and how applications react to Denial of Service (DoS) and Distributed Denial of Service (DDoS) attacks.

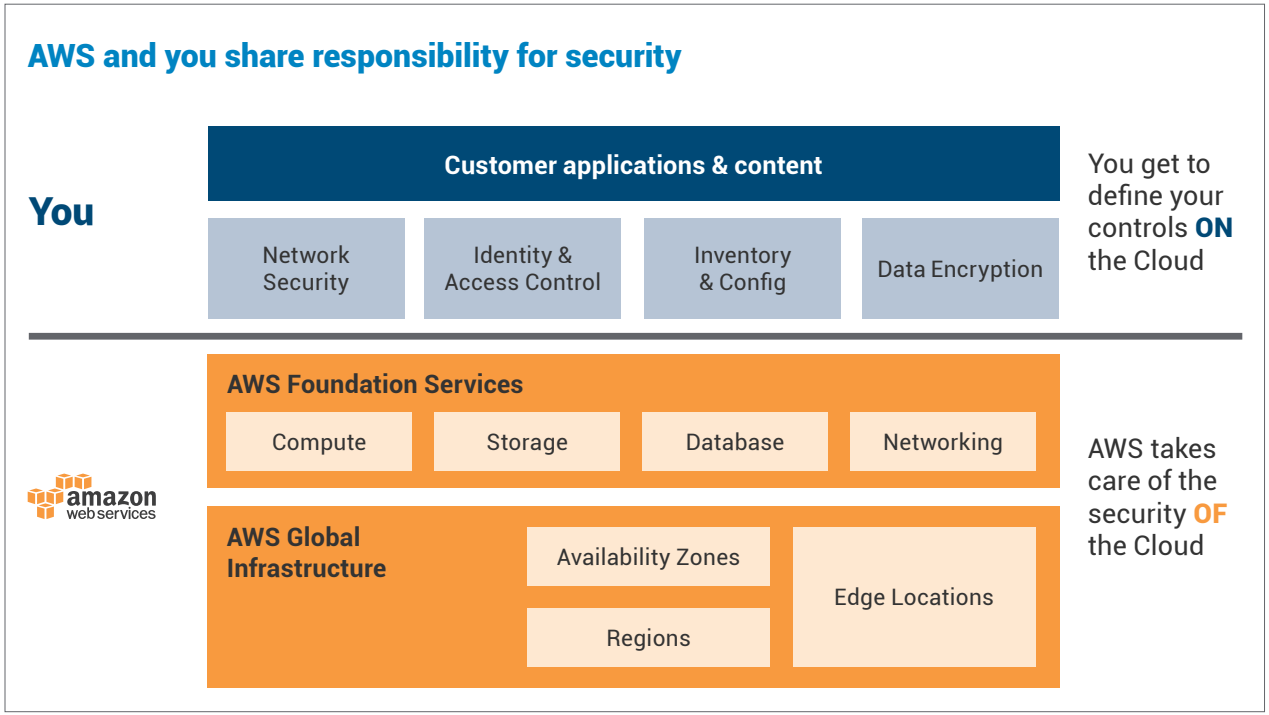


Figure 1: AMAZON WEB SERVICES SHARED RESPONSIBILITY MODEL

DoS and DDoS attacks are defined as attempts at making an application or web site inoperable by means of overwhelming it. DoS attacks are typically conducted by a single attacker while DDoS attacks are characteristically directed by a group of collaborating attackers (DDoS).

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

While some organizations use application firewalls to deny DoS and DDoS attacks, there is a simpler, more cost effective way. The easiest approach to trying to prevent a service interruption is to absorb the attack by Auto Scaling.

Most website traffic is directed by an ELB. As traffic to your website increases, so scales the ELB. Since ELBs only direct Transmission Control Point (TCP) traffic, any attacks that use a protocol other than a TCP will not reach your applications.

When TCP traffic enters your website, the data within the traffic has to be analyzed. As the amount of data increases, you have to be able to scale your ELB efficiently in order to not disrupt service. 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 Scale function.

For example, let's say we 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 come in to 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 that 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.

While Auto Scaling means that the price a company pays for the increase in instance hours will rise, the ability for your business to continue to operate is priceless. Auto Scaling acts an insurance policy that saves your company revenue, making it well-justified in the end.

BEST PRACTICE #9: ENABLING SECURITY MEASURES WHEN AUTO SCALING IS NOT AN OPTION

While Auto Scaling is a best practice for all organizations, it may not be financially feasible for all.

For organizations that do not have 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.

Although they were not built to act as a traditional firewall, 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 that you want to admit into your network, and in the direction that you would like it to flow.

Depending on where the traffic is coming from, by employing an AWS Security Group you are able to prevent malicious traffic from reaching your systems, before they are able to conduct an attack. As an example throughout this best practice, we will use the IP Address number of 0.0.0.0/0 to indicate all users attempting to access your network.

In today's technology-driven world, companies typically deliver their products through web applications. For basic web applications, you may want to allow all types of traffic access as not admitting this commerce could be detrimental to your business' success.

In instances such as these it's recommended that you 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.

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

By only allowing access from the origin IP and port address where you admin the instance from and utilizing it when it is needed you are able to allow customers admin access while maintaining your security posture. By choosing this method, you do eliminate some of the convenience of automation, but ultimately reduce the attack surface area and amount of time that you are exposed to a potential attack. This process can be scripted into your system, for further convenience.

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

Control Lists (NACL) both have inbound and outbound rules that act as defensive layers against bad actors that prevent their traffic from impacting your network.

On the following page is a graphic to help visualize these two similar features 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 from the graphic, 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 the largest amount of traffic furthest away from your application, becoming more granular in the amount of access you grant.

The next time you walk into a building, pay attention to the security measures that are in place. Typically, security is focused inwards. The boundaries in place on the exterior of the building are often times greater than those on the inside. As you proceed through the building, security becomes more granular; while you can access some rooms freely, others are more controlled. Inbound access into your application is similar.

In the context above, where should the IP address 0.0.0.0/0 be configured? If you must allow unlimited access to your application, you will need to configure your network for both a NACL and an AWS Security Group. Oftentimes you will have to create special permissions for both a NACL and the AWS Security Group.

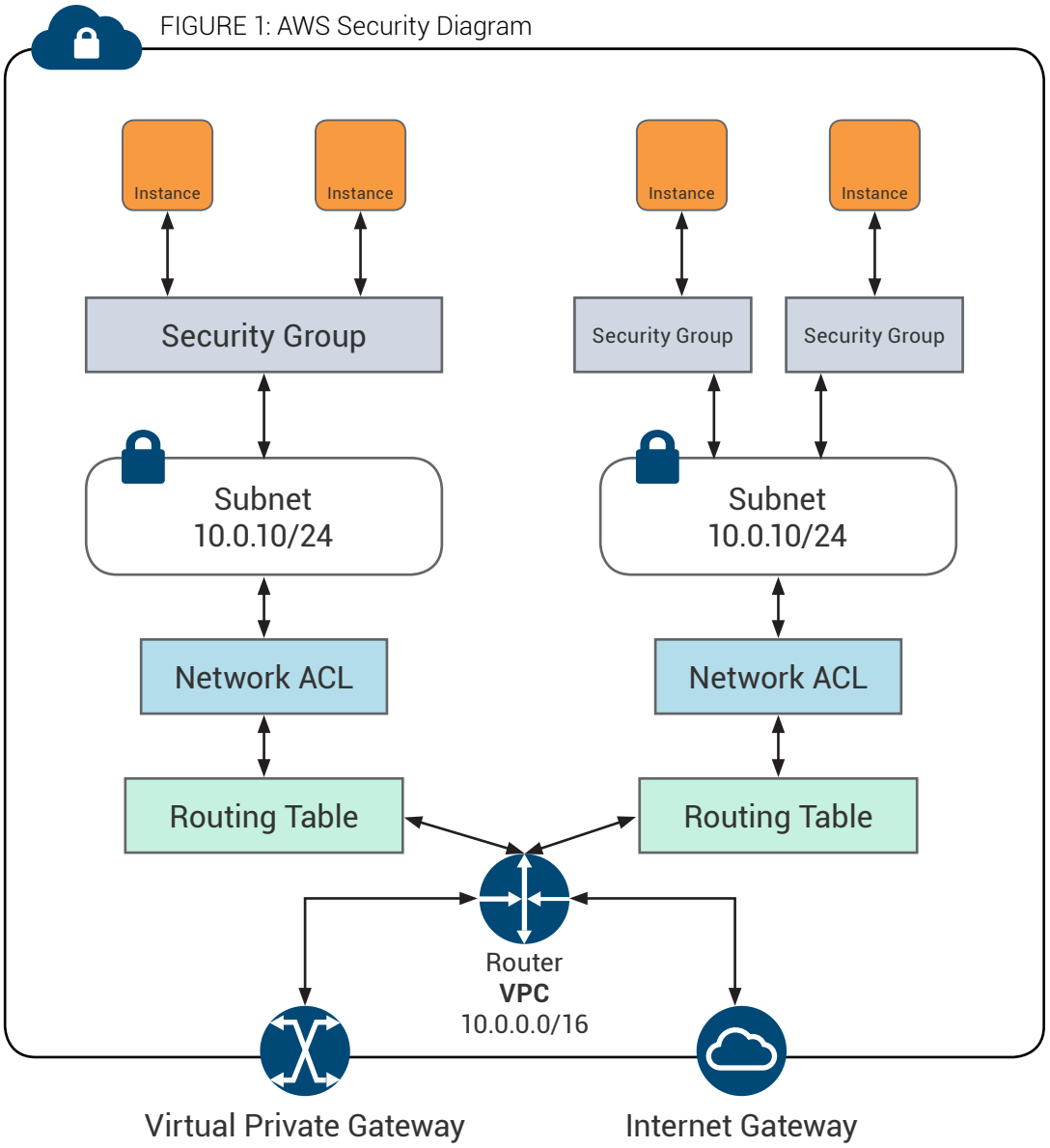
By employing an AWS Security Group, you can limit the ports that the world has access to. If you only need to allow specific network access to your application, you can limit access through the NACL, thus preventing anything you did not define from making it into your AWS Security Group.

The NACL can also be thought of as the perimeter boundary, or your first checkpoint. With a NACL you can configure an explicit deny rule, which denies traffic. AWS Security Groups are very focused on what you allow, denying everything that is not allowed.

With a NACL configured, if you discover that you are receiving a DoS attack and the origin IPs are very specific, you can quickly and easily block it.

While this process isn't quite as seamless for a DDoS attack, if the attack is not evenly distributed the set deny rules will likely be able to hinder part of the attack. When employing either rule, having familiarity

FIGURE 1: AWS Security Diagram



with your logs and partnering with AWS Support will help you identify and repel an attack.

While inbound rules are important, the security of your network also relies on the outbound rules that you have set in place. Configuring outbound allow rules can be done with both AWS Security Groups and NACLs, the process for deciding how to configure them is no different than the process for configuring inbound rules.

When creating rules for outbound traffic both AWS Security Groups and NACLs allow all outbound traffic to pass, allowing you to browse the web and make connections. But should they?

Once an AWS Security Group or NACL is configured, anything not configured will not be denied. Additionally, it is 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 impact inbound sessions, but if you configure outbound rules on a 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 while NACLs are limited to your network. An AWS Security Group can be configured per application while 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 inbound NACL rules being configured to allow connections to web servers from around the world, while the database only allows 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 that sessions are able to be established and the database would limit outbound connections to the web server private subnet IP range.

So, when should the IP address 0.0.0.0/0 be allowed? The answer depends on the customer, but by using AWS services in your Virtual Private Cloud you can tailor your network to your specific needs without breaking your budget.

BEST PRACTICE #10:
WATCH WORLD-READABLE AND LISTABLE AMAZON S3 BUCKET POLICIES

When an organization takes the time to carefully select and maintain a system, security will act as it is expected to.

Although IAM polices are built to provide security, organizations must take careful measures to ensure the stability of their platforms in the long run.

As a company grows in size it often adds access control measures to their network to keep up with the increasing demands placed on it. As the network expands, organizations often layer newer platforms on top of older systems, making it hard to keep track of who or what is being allowed access to their network. A great example of this can be found in the recent article "IAM Policies and Bucket Policies and ACLs! Oh, My! (Controlling Access to Amazon S3 Resources)."

As new personnel come in to the organization, older methods become hard to manage, difficult to visualize, and can present opportunities for administrators to lose track of security checks for older products.

While most security functions are built with a default deny rule, these rules can be circumvented if new systems are built on top of older ones and an organization is unable to keep track of where or what it is allowing.

In order 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.

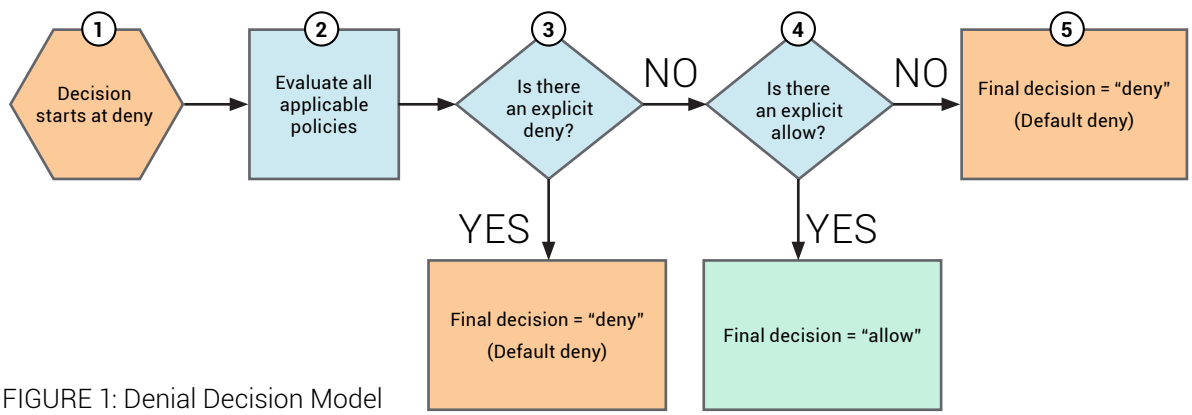


FIGURE 1: Denial Decision Model

Top Ten Best Practices for AWS Security | May 2016

Secure your AWS Cloud Today!

To activate your free 14-day trial, visit us online at www.evident.io or call us at (855) 933-1337 to schedule a demo with one of our Security Solution Architects

About Evident.io

Evident.io is the leader in security and compliance automation for Amazon AWS. The Evident Security Platform (ESP) helps organizations proactively manage security risk and compliance of AWS infrastructure with a single pane of glass view of security threats and compliance across all AWS accounts, services, and regions. ESP manages the complexity and rapid change of public cloud infrastructure by automating threat detection, incident response, and compliance through the continuous monitoring and analysis of configuration and usage data. With ESP security and compliance teams are more productive and efficient while creating a more proactive and adaptive security posture and a process for continuous compliance. Built on Amazon Web Services APIs, ESP is agent-less and can be deployed to even the most complex environments in minutes. Evident.io is a privately held company based in Dublin, CA and backed by Bain Capital Ventures and True Ventures.

About AWS

For 10 years, Amazon Web Services has been the world's most comprehensive and broadly adopted cloud platform. AWS offers over 70 fully featured services for compute, storage, databases, analytics, mobile, Internet of Things (IoT) and enterprise applications from 33 Availability Zones (AZs) across 12 geographic regions in the U.S., Australia, Brazil, China, Germany, Ireland, Japan, Korea, and Singapore. AWS services are trusted by more than a million active customers around the world – including the fastest growing startups, largest enterprises, and leading government agencies – to power their infrastructure, make them more agile, and lower costs. To learn more about AWS, visit http://aws.amazon.com.

evident.io | amazon web services