The Ultimate
AWS Security Checklist

ENGINE ROOM

Inside

The Ultimate AWS Security Checklist

You've likely considered hosting your website in the cloud, on a platform like Amazon Web Services (AWS). When you compare AWS to the rest of the market, it's clear to see that AWS dominates, holding an impressive **33.8% of the global market share** for cloud infrastructures while Microsoft, Oracle, and IBM combined account for only 30.8%.

What else should you know about the widespread prevalence of AWS for cloud hosting?

- AWS has over 1 million active users in 190 countries and five times more deployed infrastructure than the next 14 competitors have collectively.
- One-third of the people who use the internet each day visit sites that are hosted by AWS.
- In 1 day, AWS adds as much cloud infrastructure as they used to run (in total!) in 2012.

There's good reason for this overwhelming consumer vote of confidence in the giant cloud provider — AWS is powerful and has amassed **nearly half of the public cloud infrastructure** market.

With all of this infrastructure to manage, how does AWS handle it all? It's imperative to note that the cloud host has a shared responsibility model, which means even though AWS manages the cloud infrastructure, you're still responsible for securing your website. AWS handles the security of the hardware and data centers, but you're responsible for securing your code and user data.

Since using AWS doesn't mean automatic security from end to end, we've put together some thoughts about ensuring that your website and cloud are both as secure as possible in a five-step AWS security checklist. Are you implementing these essential security practices?

The Benefits of AWS Cloud Computing and a Remote Infrastructure

Cloud computing is an increasingly popular choice for organizations in almost every industry. In fact, **94% of enterprises** already use a cloud service, and 30% of all IT budgets are allocated to cloud computing. What's more? It's projected that the public cloud infrastructure will grow by 35% in 2021.

So, why the increasing adoption of the cloud? *The undeniable benefits.*

With cloud computing, there are **all kinds of advantages** for an organization to take advantage of, including:

- The ability to pay for only the storage you need without having to pay for capital like data centers and servers. There's **no need for over-provisioning** — and no need to pay like you're over-provisioning either. Scale-up and down in a moment to adapt to the changing demands of your organization.
- You no longer have to estimate your infrastructure capabilities. Making capacity decisions before deploying an application is a challenging guessing game that could leave you paying for idle resources or struggling with limited capacity. Cloud computing eliminates these problems.
- Increase speed and agility with sophisticated IT resources and even deploy applications in multiple regions across the globe in minutes — all with lower latency and a better user experience for your customers (and at a much better price!).

Plus? With a cloud computing infrastructure as robust and advanced as AWS, you get top-notch security. **Cloud security** is the highest priority for AWS, and customers benefit from a network architecture that's been built for even the most security-sensitive organizations. The AWS Cloud gives users the power to scale and innovate, all in a secure environment at a lower cost than an on-premises environment.

Additionally, the AWS infrastructure has been designed and managed in alignment with all kinds of compliance guidelines, including:

- SOC 1/ISAE 3402, SOC 2, SOC 3
- FISMA, DIACAP, and FedRAMP
- PCI DSS Level 1
- ISO 9001, ISO 27001, ISO 27017, ISO 27018

Why else does AWS **dominate the cloud market**?

- A diverse customer base
- A wide range of tools to assist with strategic cloud adoption like native cloud applications and e-business hosting
- Easy integration with a wide assortment of software vendors and a broad ecosystem of partners
- Many available partners for app development support
- Countless options for IaaS, PaaS, and SaaS
- Rapid service offerings and high-level solutions expansion

It also is worth mentioning that AWS is affordable. Since its beginning in 2006, AWS has actually cut prices more than 50 times. AWS has made "**aggressive investments**" to expand its network for greater economies of scale. This allows AWS to offer customers lower and lower prices while maintaining full-bodied, enterprise-scale features.

The Need for Your Own Security

If AWS is doing everything it can to keep data secure on their end, why do you need your own security? AWS can be more secure than an on-premises infrastructure and **has been cited as** "the most flexible and secure cloud computing environment available today."

AWS also offers automated encryption at the physical layer and automated security tasks to reduce the risk of human error in configuration. However, even the strongest fortress isn't entirely immune. Misconfigurations can still happen and can leave you entirely vulnerable.

The default setting for Amazon S3 buckets is that they are private by default and can only be accessed by individuals who have been given explicit access to the contents. All too often, organizations fall prey to misconfigurations. With the exception of phishing or social engineering techniques to work into a company's system, an exposed bucket is usually the cause of organizational error or negligence.

Take, for example, **these two serious data leaks** that happened in the UK, where the cause was misconfigured Amazon Simple Storage Service (S3) bucket storage

The first leak affected several UK consulting firms, and researchers were able to uncover information like passport scans, tax documents, background checks, job applications, expense claims, contracts, emails, salary details for thousands of consultants, and much more sensitive data.

While the owner of the unsecured bucket was not immediately clear, the source of the misconfiguration was a "mysterious" financial consulting firm with no public website and no clear confirmation of who actually owned the company. It seems that this consulting agency contained the data from several other consultancy firms, many of which have since stopped their operations and trading on the stock market since the discovery of the misconfiguration. The majority of the compromised data had been collected between 2014 and 2015, but some files dated back as far as 2011.

After notifying AWS and the UK's National Cyber Security Centre (NCSC), the database was finally secured on December 19, 2019.

The second leak centered around a misconfigured AWS bucket belonging to Fresh Film, a production company also based out of the UK. This company specializes in commercials for health and beauty brands. The leak accidentally exposed the data of 40 actors who worked on a commercial in 2017 for Dove and also exposed details about the production team and crew who worked on the campaign.

This misconfigured bucket meant exposing all kinds of personal information, including:

- Names
- Mailing and email addresses
- Phone numbers
- Birth dates
- Bank details
- Passport scans
- National insurance numbers

These are just two examples of the troubles a minor misconfiguration can cause. In the real world, cloud security in AWS is often neglected. As a matter of fact, one cloud security company found among its customers an average of **1,150 misconfigurations in Elastic Cloud Compute** per AWS account.

Without careful attention and thoughtful configuration, organizations can leave all kinds of sensitive data vulnerable, which can be costly to handle and damaging to their reputation.

Organizations have to do everything they can to protect their interests. While Amazon Web Services has entire teams of experts dedicated to keeping their cloud infrastructure secure, it's also up to individual organizations to keep their data protected on the front end. AWS calls this "**shared responsibility.**"

Shared Responsibility

Security and compliance are shared between AWS and its customers. This relieves much of the customer's burden since AWS is in charge of operating, managing, and controlling their host operating system and virtualization layer, and even maintaining the physical security of the facilities hosting the AWS cloud hardware.

For IaaS solutions, this frees up the customer to manage their own guest operating system, including updates and security patches, associated application software, and configuration of the AWS security group firewall.

Essentially what this means is the following:

- **Amazon Web Services** is responsible for protecting the infrastructure that runs the cloud, as well as all of the services offered there. This includes the hardware, software, networking, and facilities of the AWS Cloud.
- **Customers'** responsibility is based on their chosen cloud service model. The amount of configuration work the customer needs to perform as part of their security responsibility is also determined by which cloud service model (IaaS, PaaS, SaaS), software, and applications they use.

AWS secures the server hardware for your instances, but the security of your individual cloud infrastructure is up to you.

With AWS, you get secure cloud infrastructure, but you still have to do everything you can to protect your assets. It's a partnership, and it's up to you to fulfill your obligations to keep your data secure.

Common AWS Security Mistakes — And How to Avoid Them

AWS is fairly intuitive to use, but that doesn't prevent some well-meaning organizations from falling into certain traps that can compromise their entire environment. More and more sensitive data is being stored in the cloud every day, and malicious attackers know this. The way your organization secures your personal overall cloud structure is gaining importance with each passing day.

Here are some of **the most common configuration mistakes** that organizations make all too often:

- Not explicitly putting one person in charge of security or not understanding what role AWS takes in keeping the servers secure versus which responsibilities fall to the users. Customers cannot assume straightaway that the default settings are the proper configurations for their workloads.
- Forgetting to turn on log data with tools like **CloudTrail** to follow API calls, the identity of API callers, as well as the time and IP address for each request. This is a simple yet invaluable tool to pinpoint the source of many potential issues.
- Giving too much access or too many privileges to too many people, or neglecting Identity and Access Management. Just like brick-and-mortar stores are often selective about who gets a key to open their store, and who has an access code to turn off an alarm, only some cloud users should have administrative rights. Yet all too often, administrators give too much access or leave high-privilege access available for terminated users. It's estimated that a whopping **35% of privileged users in AWS** have enough access to bring down the entire customer AWS environment. Instead, admins should set up a variety of user scenarios and tiered access.
- Organization users implement poor passwords, and their admin teams rely too heavily on single-factor authentication. **23 million people** still use " 123456" as their password. Organizations need to have security controls like multi-factor authentication in place to better protect their infrastructure — and encourage employees to choose better passwords.
- Forgetting to **secure root user access**. Every AWS account starts with one user — the "root user," so to speak. This user has access to all the features, services, and resources throughout the account, and AWS recommends not using this account for everyday operations. These accounts should be used thoughtfully and sparingly, as usage needs to be trackable, managed, and protected. Too often, these accounts don't even have multi-factor authentication enabled!
- Putting everything for one organization into a single virtual private cloud (VPC). The more teams or workloads you add to a single account or VPC, the less protected your entire data set is. Instead, organizations should consider isolating workloads and teams into separate regions, VPCs, and even separate accounts.

Use the Principle of Least Privilege: AWS Identity and Access Management

AWS Identity and Access Management (IAM) is a service that allows you to securely control access to AWS resources. The foundation of secure cloud infrastructure is reliant on your IAM implementation.

IAM is free to use at no additional charge for AWS accounts and gives users the power to manage access to all AWS services and resources securely. With IAM, organizations can create and manage users and groups, and oversee the permissions that allow and deny access to various resources stored within the AWS cloud infrastructure.

This invaluable feature allows companies to take charge of their own user permissions for controlled access so that only the people who need access to different segments of the data have it.

AWS Identity and Access Management allow users to check their security status in the IAM console, acting as an audit of the overall security to create a more protected environment.

AWS Identity and Access Management: Best Practices

One of the riskiest things organizations can do is give over too much control and too much access to their users.

Best practices to establish a secure environment include:

- Remove access keys from the root account, and only use the root account to create new users and manage permissions.
- Implement individual users and selective groups to manage separate parts of an AWS account and manage delegation.
- Apply an account-wide password policy to strengthen secure access.
- Enable multi-factor authentication for all users and verify sign-in credentials at every session.
- Check IAM user utilization, following user credentials, multi-factor authentication, and account activity. It is also a wise idea to purge accounts that are no longer in use.

AWS has plenty of **guides on how to use IAM** effectively and **long lists of best practices**. Here are some tips for a quick audit to make sure you've got yours set up correctly.

- Use the principle of least privilege for users and EC2 instances. Give users and groups the minimum permissions they need to do their job, and nothing more.
- Use AWS-managed policies to assign permissions. Amazon provides a predefined set of policies that are completely managed by AWS. These policies serve common use cases while making it easier to enforce access policies than creating policies from scratch.
- Assign permissions at the IAM group or role level rather than the individual IAM user level. For example, create groups, assign permissions to the groups (i.e., administrators, developers).
- PRO TIP: All applications running on the same EC2 instance run with the privileges of the EC2 instance. Don't mix applications that require different levels of permissions on the same server.
- Create individual IAM users to serve as administrators. Give them only the access they need. Don't use the Root user account for basic admin functionality and delete the root account access keys.

Protect Your Secrets

Moving your infrastructure to the cloud gives you a host of secrets to keep. You can't hide behind your physical data center.

Here are the checks you should make to ensure you're protecting your secrets. There are two ways to access AWS you need to protect: console access and programmatic access.

The Differences Between Console Access and Programmatic Access

When creating a new user and enabling them to perform specific tasks, administrators must choose whether they should be given either console access or programmatic access, depending on the type of access the users require.

What should organizations and administrators know about these two kinds of access?

- **Console access** is for users who need to gain access to the AWS management console with a username and password.
- **Programmatic access** is for IAM users who need to make API calls, use the AWS command line interface (CLI), or use other programming tools like Windows PowerShell. They should be given an individual access key ID and a secret access key.

Console Access

Secure your root account password:

- Make it strong
- Don't write it down - use a password manager like LastPass or 1Password, or Dashlane for enterprises
- Use multi-factor authentication (MFA) - in other words use a second source of validation, for example, using a phone or token in a Cisco Duo MFA system
- Create an administrative group and add individual users to the group.

Don't give out the root password. Don't use the root account for everyday administrative tasks.

Programmatic Access

AWS uses an access key and secret key to provide programmatic access to the AWS API. Secure Token Service (STS) can also be used.

- Delete access and secret keys for the root account if they exist. If an attacker can grab these, they'll have full control of your environment.
- If you have to use keys, assign them to individual users (not the root account).
- Rotate keys regularly.
- Password protect PEM files used for SSH access into EC2 instances.
- Don't store access or secret keys in a code repository.
- Use **STS** for programmatic access.
- Use **IAM Role Delegation** to give access to compute resources.

When Is the Right Time to Use Encryption?

When should organizations use encryption, and what are the requirements? Encryption is an essential tool in keeping your data protected, even if it were to get intercepted. The National Institute of Standards and Technology (NIST) best described encryption and cryptographic tools in its **Special Publication 800-57 part 1, rev. 5** as such:

> *"Cryptographic keys play an important part in the operation of cryptography. These keys are analogous to the combination of a safe. If a safe combination is known to an adversary, the strongest safe provides no security against penetration. The proper management of cryptographic keys is essential to the effective use of cryptography for security. Poor key management may easily compromise strong algorithms."*

Having a strong **encryption and encryption key** is an essential tool for data encryption. Encryption key management has four main requirements:

- **Security:** Encryption keys must be protected at all times against internal and external threats.
- **Scalability:** Encryption keys need to be able to manage growing amounts of information
- **Access:** Encryption keys need to quickly and smoothly access the data without much interruption to effectively encrypt it.
- **Compliance:** Your encryption key must fit with all industry compliance standards.

Automatic encryption is one option that makes the most sense for the majority of companies. Manual key management is not only time-consuming, but it's also expensive and prone to user error, especially at scale or for large organizations where there is a massive amount of data to encrypt.

Broadly, **two types of data** should absolutely be encrypted, and beyond that, it's up to the organization what other data should be encrypted:

- **Personally identifiable information** like phone numbers, driver's license numbers, or social security numbers. This kind of information is stored everywhere, including on phones, tablets, and laptops where it needs to be encrypted, but just as important is encrypting this data in the cloud where vulnerable buckets can create security issues.
- **Confidential business information and intellectual property** like plans for a new project or a marketing campaign.

An easy answer is just to encrypt everything. But what if organizations don't have the processing power, staff, money, or expertise to do so? **What should they prioritize**?

- Customer information
- Financial reports
- Product release documents
- Research and development data

Make Wise Use of Encryption

Solid encryption is table stakes for cloud deployments. Enable encryption wherever it's an option. AWS makes it easy, so there's no reason to leave data unencrypted. This means using tools and services like:

- S3 Buckets
- RDS and Aurora databases
- EC2 EBS Volumes

Use AWS Key Management System to Encrypt Data

AWS Key Management Service (KMS) is one reliable way to create and manage cryptographic keys and control them throughout a wide range of AWS services and applications. KWS is both secure and resilient and uses validate security modules to protect cryptographic keys for encryption.

KMS is also integrated into AWS CloudTrail for logs of key usage, which can come in handy when organizations need to meet regulatory and compliance requirements.

What are the **benefits of AWS KMS**?

- **Fully managed:** Administrators control access to encrypted data by specifying their permissions and use keys, and then AWS KMS enforces these permissions for you.
- **Centralized:** KMS is a single control point to manage keys and define policies throughout AWS services and integrated applications. Organizations have the power to create, import, rotate, delete, and manage permissions from the console.
- **Encrypt data in applications:** KMS is easily integrated with AWK Encryption to encrypt locally in each application. APIs make it simple to build encryption and key management into your own applications no matter where they run.
- **Digitally sign data:** KWS gives users the ability to complete digital signing operations and still maintain data integrity. Recipients of digitally signed data can verify signatures even if they don't have an AWS account.
- **Cost-effective encryption key solutions:** Use KWS without commitment or upfront charges. Organizations only pay $1 per month to store any keys, and AWS-managed keys created for you are free to store. Users are only charged for requests when using or managing keys outside of what's included in the free tier.
- **Compliant:** KMS is managed by AWS with security and quality controls in mind, and have been certified through multiple compliance regulations, enabling users to simplify their compliance obligations.
- **Built-in auditing:** KMS integrates with AWS CloudTrail to record API requests, key usage, and key management actions. This empowers users to manage risk, meet compliance requirements, and conduct forensic analysis.

Controlling your cryptographic keys with KMS is easy, but when you use AWS's Key Management System, it's important to know that AWS is controlling the keys. If you're not okay with that, then a solution such as **StrongKey** or **Vault** could be a better choice.

Use HTTPS Everywhere

If you're using CloudFront, you'll need certificates to set up HTTPS connections. Use the **AWS Certificate Manager** to create them.

Why is HTTPS valuable? HTTPS protects both internal communication and customer information. With SSL, a secure sockets layer, almost every website today incorporates HTTPS. Plus? It's good for SEO and Google's search algorithm and makes it easier to get better placement on search results pages.

Seeing as HTTPS is required for accelerated mobile pages, setting up an HTTPS connection is a good idea, and it puts your customers' minds at ease.

Some browser applications like Google Chrome alert users when they are about to enter any HTTP site missing the security of an HTTPS site, especially if it asks users for login or credit card information. It can be a major deterrent for consumers to see this kind of warning come up. Not only is it good to establish your reputation as a credible, secure site, but it's wise to keep your data protected as well. It's also smart to configure your databases to **accept only secure connections**.

Monitor Your Infrastructure

AWS has no shortage of logging options to make it simpler to monitor and audit your infrastructure. Make sure you've configured the following logging services to the greatest effect.

- **CloudTrail Logs:** AWS **CloudTrail** enables governance, compliance, operational auditing, and risk auditing throughout your entire AWS infrastructure. CloudTrail logs, monitors and retains account activity related to your AWS infrastructure and logs event history including actions taken throughout the AWS management console and other AWS services.
- **VPC Flow Logs: These logs** help organizations monitor traffic, tracking the direction of traffic to and from network interfaces. They also can be useful in diagnosing whether an account has too many restrictions in place or overly restrictive security group rules.
- **S3 Access Logging:** These logs are **useful in security audits** and grant or deny permission to deliver access logs.
- **Billing Logs: These logs** cover cost and usage reports, cost explorers, budgets, and more.

AWS has published a **Centralized Logging Implementation Guide**. Check it out to get the most out of your monitoring capabilities.

Keep Checking Yourself

We're programmers at heart here, so we can't leave this list without a bit of recursion. There are a few options from AWS to help you audit your configuration as well. The following tools are worth a look to see if they are right for you.

- **Trusted Advisor**: AWS audits your resources for you.
- **AWS Config**: You can use this to audit your configurations yourself with customizable templates.
- You can write custom scripts using AWS CLI and Bash.
- You may also benefit from other AWS security tools like **Inspector**, **GuardDuty**, **Macie**, and **Shield**.

AWS Well-Architected Framework: Security

AWS Well-Architected is a service that's designed to help cloud architects build an infrastructure for their applications and workloads that is secure, resilient, and high-performing. The framework is based on five pillars:

- Operational excellence
- Security
- Reliability
- Performance efficiency
- Cost optimization

This is a consistent approach for organizations to evaluate their architecture and create designs that are built to scale over time.

Well-Architected's Security pillar emphasizes protecting information and systems alike, focusing on topics like confidentiality and data integrity, as well as managing privileges and access control, implementing system protections, and establishing controls to detect security events.

AWS Security Hub

AWS Security Hub is a paid service with constant monitoring and a comprehensive view of all security alerts across AWS accounts. There are all kinds of security tools available through AWS, like firewalls and endpoint protection and even vulnerability and compliance scanners, but this can be a lot to manage.

Security Hub gives users a single point to manage, organize, and prioritize all of these tools to handle the hundreds or even thousands of security alerts that come in daily. This covers multiple AWS services like:

- Amazon GuardDuty
- Amazon Inspector
- Amazon Macie
- AWS Identity and Access Management Access Analyzer
- AWS Systems Manager
- AWS Firewall Manager
- AWS Partner Network Solutions

Third-Party Assessments

Third-party assessment providers can also be a valuable tool in the fight to keep your AWS infrastructure secure. These providers include reputable vendors like:

- **CloudSploit by Aqua**
- **Coalfire**
- **Nettitude**
- **Threat Stack**

Need some help with your AWS environment? Engine Room Tech is at your service and ready to share what we've learned over the years. Get in touch today to learn more!