WHITE PAPER

McAfee
Together is power.

Definitive Guide to Amazon Web Services Security

Table of Contents

WHITE PAPER

Definitive Guide to Amazon Web Services Security

While popular out-of-the-box software-as-a-service (SaaS) products like Salesforce, Box, Dropbox, and Office 365 are becoming ubiquitous in the workplace, many enterprises have business needs that require tailor-made applications.

Introduction

In the past, organizations relied on custom, in-house-developed applications hosted in their own data centers. Having recognized the advantages of cloud computing, over the last 10 years, these applications have slowly migrated to the public cloud, private cloud, or a hybrid of both. Today, more than half of all custom applications (60.9%) are still being hosted in private datac enters, according to a recent Cloud Security Alliance report.[1] However, cloud usage has reached a tipping point, and deployment of test and production application workloads in the public cloud is accelerating at the expense of enterprise datacenters. Not only are enterprises increasingly developing new custom applications on infrastructure-as-a-service (IaaS) platforms like Amazon Web Services (AWS), but enterprises are also migrating their existing custom applications to the public cloud. Collectively, these two trends are expected to drive the percentage of custom applications running in the data center to an all-time low of 46.2% in the next 12 months, plunging 14 percentage points.

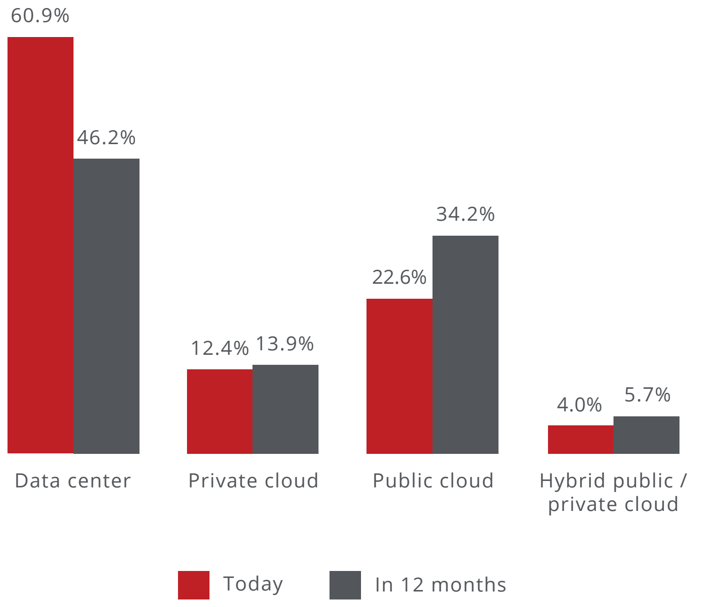


Figure 1. Application workloads—percentage deployed by infrastructure type

Connect With Us

While the number of custom applications at an enterprise varies, the average enterprise has 465 custom applications deployed. Larger enterprises tend to have more applications—organizations with more than 50,000 employees have an average of 788 custom applications. Enterprises increasingly rely on these applications to handle business-critical functions. Most organizations today have at least one custom application that, if it experienced several hours of downtime, could have a significant impact on the business. As a result, these applications and the infrastructure they run on are targets of cyberattacks.

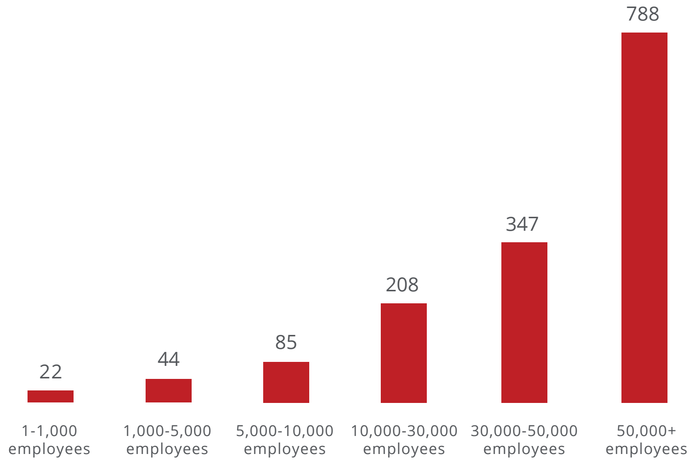


Figure 2. Number of custom applications—by company size

In 2014, a cybercriminal launched an attack against the AWS account Code Spaces used to deploy their commercial code-hosting service. The attacker gained access to their control panel and demanded money.

When the company refused, the attacker began to systematically delete Code Spaces' resources hosted on AWS, including all elastic block store (EBS) snapshots, Simple Storage Service (S3) buckets, Amazon Machine Images (AMIs), some EBS instances, and a number of machine instances. While Code Spaces maintained backups of their resources, those too were controlled from the same panel and were permanently erased. The attack was so devastating it forced Code Spaces, a thriving company, to shut down for good.

The threat landscape is evolving rapidly, but with the right preparation, any company can implement security practices that significantly reduce the potential impact of a cyberattack. In this eBook, we will discuss the current state of AWS adoption, Amazon's model for AWS security, security challenges and threats to applications and data in AWS, and AWS infrastructure security best practices, as well as security best practices for applications built on AWS. Lastly, we will explore how a cloud access security broker (CASB) can help enterprises secure their AWS environments and the custom applications deployed in them.

The worst-case scenario can be far worse than downtime.

AWS Adoption Trends

According to Gartner,[2] IaaS was a $16.2-billion market in 2015, and is estimated to grow 38% to reach $22.4 billion in 2016, making it the fastest-growing segment in the public cloud market. A report by Structure Research[3] showed that the vast majority of the $22.4-billion market will be dominated by a handful of IaaS providers, with AWS leading the pack at nearly $10 billion in sales.

Today, an impressive 41.5% of custom applications in the public cloud are deployed in AWS, giving it a larger market share than Microsoft Azure, Google Cloud Platform, IBM Softlayer, and Rackspace combined. What's even more remarkable is that, according to the 2016 IaaS Magic Quadrant report[4] released by Gartner, AWS's computing capacity is 10 times larger than the next 14 IaaS providers combined. Amazon's massive scale has positioned the company to respond to the rapidly growing demand for cloud computing infrastructure.

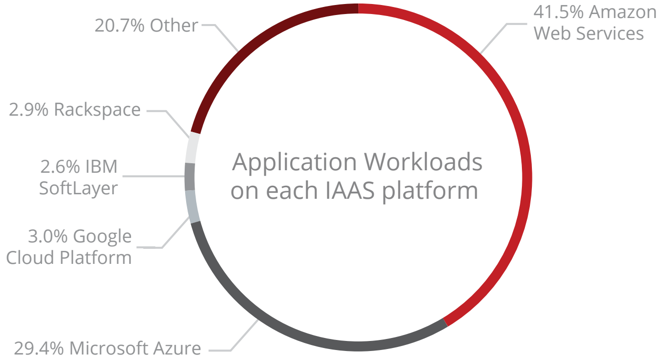


Figure 3. IaaS platform adoption—percentage of applications deployed

The overall percentage of custom application workloads deployed in the public cloud is expected to surge from 22.6% to 34.2% in the next twelve months, and it's likely many of these applications will be deployed in AWS. Underneath these numbers, enterprises have more widely deployed test and QA workloads in the public cloud. Today, 23.6% of the test/QA workloads are in the public cloud, which is slightly higher than the percentage of production workloads. However, in the next 12 months, test/QA workloads will migrate to the public cloud at an even faster rate, to 38.2% of workloads. Production workloads are migrating at a slightly slower pace, with 32.9% expected to be in the public cloud 12 months from now.

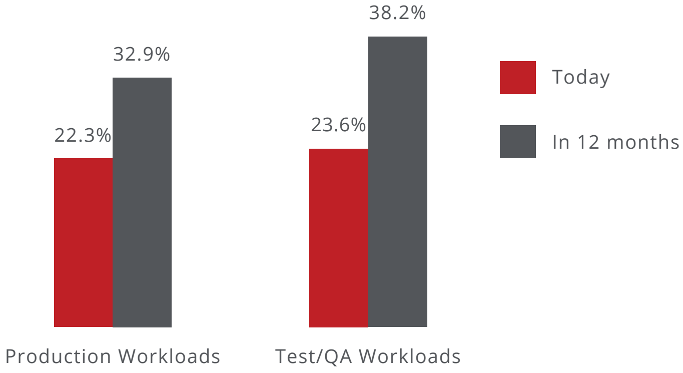


Figure 4. Application workloads—percentage of workloads deployed in the public cloud

As it stands today, the IaaS market consists of three dominant players: Amazon, Microsoft, and Google.

AWS Security Challenges

Threats to applications and data in AWS

Enterprises can't afford to have their AWS environment—or the custom applications running in AWS—compromised, because a sizable majority (72.2%) have business-critical applications—defined as an application that, if it experienced downtime, would greatly impact the organization's ability to operate. As an example, an airline cannot operate if their flight path application goes down, much the same as a rental car company cannot take reservations over the phone if their call center application goes down. Moreover, enterprises store sensitive data such as payment card numbers, Social Security numbers, and other data in custom applications.

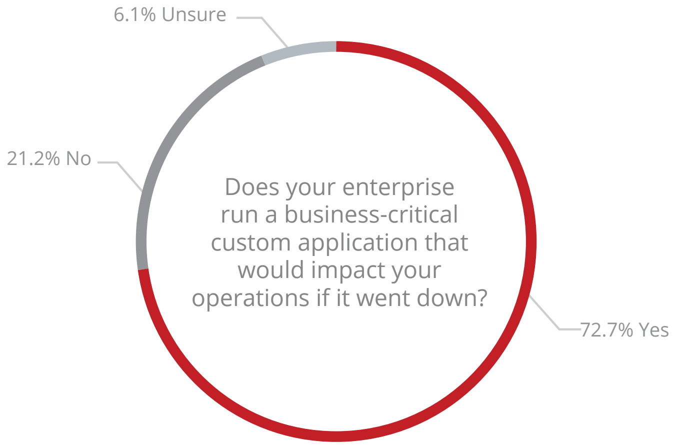


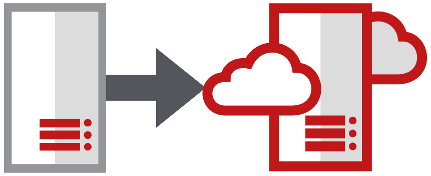
Figure 5. Business-critical applications—percentage of enterprises with at least one

Threats to applications running on AWS and the data stored within them can take many forms:

- **Compromise of Amazon's platform**—Amazon has made significant investments in security to protect its platform from intrusion. However, the small possibility remains that an attacker could compromise an element in the AWS platform and either gain access to data, take an application running on the platform offline, or permanently destroy data.
- **Denial of Service (DoS) attack on an application**—Amazon has developed sophisticated DoS prevention capabilities delivered in AWS Shield for all customers. However, it's possible a large attack could overwhelm Amazon's defenses and take an application running on the platform offline for a period of time until the attack is remediated.
- **Insider threats and privileged user threats**—The average enterprise experiences 10.9 insider threats each month and 3.3 privileged user threats each month. These incidents can include both malicious and negligent behavior— ranging from taking actions that unintentionally expose data to risk, to employees stealing data before quitting to join a competitor.
- **Third-party account compromise**—According to the Verizon Data Breach Investigations Report,[5] 63% of data breaches, including the breach that sunk Code Spaces, were due to a compromised account where the hacker exploited a weak, default, or stolen password. Misconfigured security settings or accounts that have excessive identity and access management (IAM) permissions can increase the potential damage.

- **Sensitive data uploaded against policy/regulation**—Many organizations have industry-specific or regional regulations, or internal policies, that prohibit certain types of data from being uploaded to the cloud. In some cases, data can be safely stored in the cloud, but only in certain geographic locations (for example, a data center in Ireland but not in the United States).
- **Software development lacks security input**—Unfortunately, IT security isn't always involved in the development or security of custom applications. IT security professionals are only aware of 38.6% of the custom of the custom applications in use in their organizations. This means when it comes to custom application development, IT security is often circumvented, making the task of securing these applications more difficult.

According to Gartner, from now through 2020, 95% of security incidents in the cloud will be the fault of the customer, not the cloud provider. As enterprises continue to migrate to or build their custom applications in AWS, the threats they face will no longer be isolated to on-premises applications and endpoint devices. While the move to the cloud transfers some responsibility for security from the enterprise to the cloud provider, as we will see in the next section, preventing many of these threats falls on the shoulders of the AWS customer.



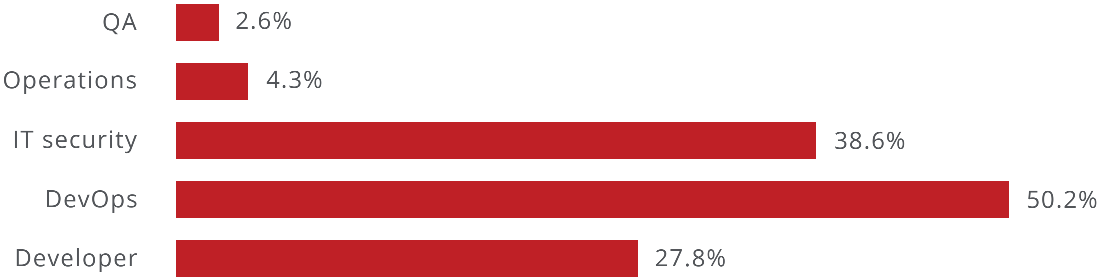


Figure 6. Awareness of custom applications by job role as a percentage of total custom applications

Shared responsibility model

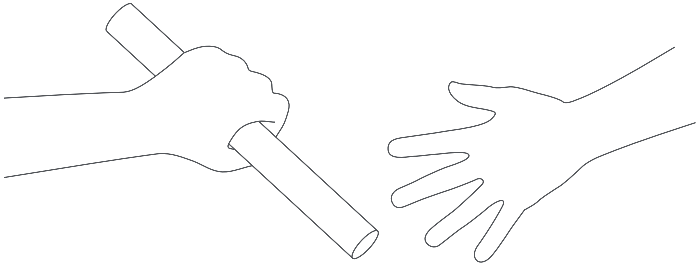
Like most cloud providers, Amazon operates under a shared responsibility model. Amazon takes responsibility for the security of its infrastructure, and has made platform security a priority in order to protect customers' critical information and applications. Amazon detects fraud and abuse, and responds to incidents by notifying customers. However, the customer is responsible for ensuring their AWS environment is configured securely, data is not shared with someone it shouldn't be shared with inside or outside the company, identifying when a user misuses AWS, and enforcing compliance and governance policies.

Amazon's responsibility

Since it has little control over how AWS is used by its customers, Amazon has focused on the security of AWS infrastructure, including protecting its computing, storage, networking, and database services against intrusions. Amazon is responsible for the security of the software, hardware, and the physical facilities that host AWS services. Amazon also takes responsibility for the security configuration of its managed services such as Amazon DynamoDB, RDS, Redshift, Elastic MapReduce, WorkSpaces, and others.

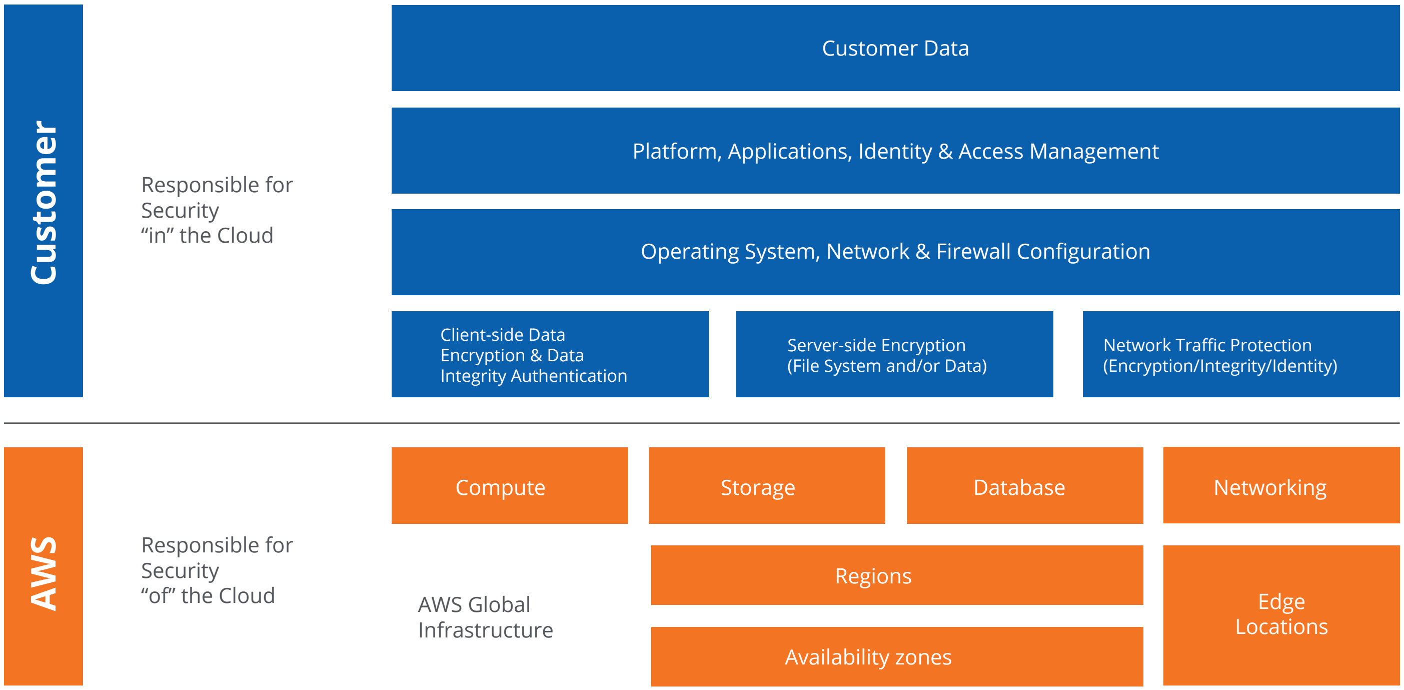
Customer's responsibility

AWS customers are responsible for secure usage of AWS services that are considered unmanaged. For example, while Amazon has built several layers of security features to prevent unauthorized access to AWS, including multifactor authentication, it is the responsibility of the customer to make sure multifactor authentication is turned on for users, particularly for those with the most extensive IAM permissions in AWS.



> "Through 2020, 95% of cloud security failures will be the customer's fault."
>
> —Gartner, "Top Predictions for IT Organizations and Users for 2016 and Beyond"

While there are a number of services in AWS, Amazon's primary IaaS services consist of Elastic Compute Cloud (EC2), Amazon Virtual Private Cloud (VPC), and Amazon S3. The customer is fully responsible for configuring the security controls of these services. Updating or applying security patches to guest operating systems for EC2, for example, falls under the customer's responsibility as well. The customer is also responsible for configuring the security groups (Amazon's firewall), ensuring that individual user accounts are set up with Amazon Identity and Access Management (IAM), and enabling activity logging with AWS CloudTrail.


Customer
Responsible for Security "in" the Cloud
Customer Data
Platform, Applications, Identity & Access Management
Operating System, Network & Firewall Configuration
Client-side Data Encryption & Data Integrity Authentication
Server-side Encryption (File System and/or Data)
Network Traffic Protection (Encryption/Integrity/Identity)
AWS
Responsible for Security "of" the Cloud
Compute
Storage
Database
Networking
AWS Global Infrastructure
Regions
Availability zones
Edge Locations

	Customer	AWS
Preventing or detecting when an AWS account has been compromised	●	
Preventing or detecting a privileged or regular AWS user behaving in an insecure manner	●	
Preventing sensitive data from being uploaded to or shared from applications in an inappropriate manner	●	
Configuring AWS services (except AWS Managed Services) in a secure manner	●	
Restricting access to AWS services or custom applications to only those users who require it	●	
Updating guest operating systems and applying security patches	●	
Ensuring AWS and custom applications are being used in a manner compliant with internal and external policies	●	●
Ensuring network security (DoS, man-in-the-middle (MITM), port scanning)	●	●
Configuring AWS Managed Services in a secure manner		●
Providing physical access control to hardware/software		●
Providing environmental security assurance against things like mass power outages, earthquakes, floods, and other natural disasters		●
Database patching		●
Protecting against AWS zero-day exploits and other vulnerabilities		●
Business continuity management (availability, incident response)		●

Table 1. Shared responsibility model at a glance

Data breach fallout

While Code Spaces is perhaps the worst-case scenario of what could happen when a hacker successfully attacks an organization's AWS environment, an incident that results in downtime of even a few hours could have a sizable impact. For example, in August 2016, a six-hour application outage at Delta Airlines delayed flights for hundreds of thousands of passengers, and is estimated to have cost the company tens of millions of dollars.

With stakes this high, a data breach will likely lead to people getting fired. Both the CEO and CIO of Target were fired after a breach in 2014 that compromised payment card numbers for upwards of 40 million customers. In a 2017 survey of IT security leaders, 50.3% of respondents said the IT security personnel responsible for securing AWS would likely get fired in the case of a breach. However, responsibility extended all the way to the CIO—29.1% said the top IT leader would be let go following a damaging and costly data breach.

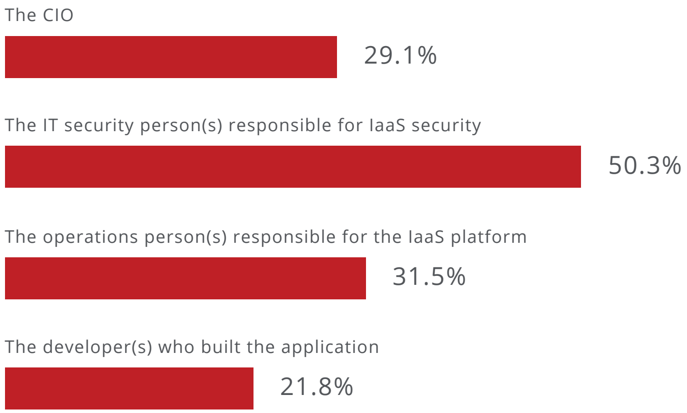


Figure 7. Who is fired after a breach?—percentage of respondents

AWS Security Best Practices

Amazon has invested heavily in building a powerful set of security controls for its customers to use across all AWS services. With CloudTrail and CloudWatch, for example, customers can monitor and track both the health and security of their AWS resources. The Identity and Access Management service gives AWS customers granular control over managing users and enforcing access control policies. It is therefore incumbent on the AWS customers to configure AWS's security controls appropriately to tighten their security posture.

Security monitoring

1. **Ensure CloudTrail is enabled across all AWS.**

   By enabling global CloudTrail logging, it will be able to generate logs for all AWS services including those that are not region specific, such as IAM, CloudFront, and others.

AWS customers need to consider the best practices within five key areas in securing their AWS environment:

1. Security monitoring
2. Secure authentication
3. Secure configuration
4. Inactive entities
5. Access restrictions

2. **Turn on CloudTrail log file validation.**

   When log file validation is turned on, any changes made to the log file itself after it has been delivered to the S3 bucket will be identifiable. This capability provides an additional layer of protection for the integrity of the log files.

3. **Enable CloudTrail multi-region logging.**

   The AWS API call history provided by CloudTrail allows security analysts to track resource changes, audit compliance, investigate incidents, and ensure that security best practices are followed. By having CloudTrail enabled in all regions, organizations will be able to detect unexpected activity in otherwise unused regions.

4. **Integrate CloudTrail with CloudWatch.**

   CloudWatch can be used to monitor, store, and access log files from EC2 instances, CloudTrail, and other sources. With this integration, real-time and historic activity logging based on user, API, resource, and IP address is facilitated. It also supports setting up alarms and notifications for anomalous or sensitive account activity.

5. **Enable access logging for CloudTrail S3 buckets.**

   CloudTrail S3 buckets contain the log data that is captured by CloudTrail, supporting activity monitoring and forensic investigations. By enabling access logging for CloudTrail S3 buckets, customers can track access requests and identify potentially unauthorized or unwarranted access attempts.

6. **Enable access logging for Elastic Load Balancer (ELB).**

   Enabling ELB access logging will allow the ELB to record and save information about each TCP and HTTP request made. The access logging data can be extremely useful for security audits and troubleshooting sessions. For example, your ELB logging data can be used to analyze traffic patterns that may be indicative of different types of attacks, which can then inform custom protection plans that should be implemented.

7. **Enable Redshift audit logging.**

   Amazon Redshift is an AWS service that logs details about user activities such as queries and connections made in the database. By enabling it, you can perform audits and support post-incident forensic investigations for a given database.

8. **Enable Virtual Private Cloud (VPC) flow logging.**

   VPC flow logs provide visibility into network traffic that traverses the VPC and can be used to detect anomalous traffic and provide insight during security workflows. It is one of AWS's network monitoring services, and enabling it will allow you to detect security and access issues such as overly permissive security groups, network access control lists (ACLs), and alert on anomalous activities such as rejected connection requests or unusual levels of data transfer.

Secure authentication

1. **Require multifactor authentication (MFA) to delete CloudTrail buckets.**

   Once an AWS account has been compromised, one of the first steps the hacker will likely take is to delete CloudTrail logs to cover his tracks and delay detection. By requiring multifactor authentication in order for a user to delete an S3 bucket containing CloudTrail logs, the hacker will find it more difficult to remove the logs and stay hidden.

2. **Turn on multifactor authentication for the "root" account.**

   When signing up for AWS for the first time, the user account that is generated is known as the root account. It's the most privileged user type in an AWS environment, with access to every AWS resource. For this reason, MFA should be turned on as early as possible. As an additional measure, root account MFA shouldn't be tied to a personal device. There should exist a dedicated mobile device that is secured independent of a user's personal device. This adds an additional layer of security and ensures the root account is accessible even if a personal device is lost or the individual owning the device is no longer at the company.

3. **Turn on multifactor authentication for IAM users.**

   MFA is often the last line of defense against a compromised account. All IAM users with a console password should have it enabled.

4. **IAM users must be enabled for multi-mode access.**

   IAM users must be enabled for both API access and for console access to reduce the risk of unauthorized access in case IAM user credentials (access keys or passwords) are compromised. Application users should use only access keys to programmatically access data in AWS, and administrators who need console access should only use passwords to manage AWS resources.

5. **Ensure IAM policies are attached to groups or roles.**

   Instead of assigning policies and permissions to users directly, provision permissions to users at the group and role level. Doing so makes managing permissions more efficient, makes it simpler to remove or reassign permissions based on a change in responsibilities, and minimizes the risk of an individual user getting excessive and unnecessary permissions or privileges by accident.

6. **Rotate IAM access keys regularly, and standardize on the selected number of days.**

   Sending requests from the AWS Command Line Interface (CLI) to AWS APIs requires an access key, which is comprised of an access key ID and secret access key. Rotating access keys regularly ensures that data cannot be accessed with a potential lost or stolen key.

7. **Set up a strict password policy.**

   When left to their own volition, users often create passwords that, while easy to remember, are also easy to guess. Configuring a strict password policy not only ensures that users can't get around it, but also protects an account from brute force login attempts. The actual policy may vary, but at a minimum, require passwords to have at least one upper-case letter, one lower-case letter, one number, one symbol, and a minimum length of 14 characters.

8. **Set the password expiration period to 90 days, and ensure the IAM password policy prevents reuse.**

   Employees tend to use the same password across multiple services despite the inherent security risks of doing so. As a best practice, configure the IAM password policy to record at least the past 24 passwords for each user, preventing password reuse. Prompt users to change their passwords no sooner than every 90 days, as setting this too high exposes the organization to account compromise from credentials that are phished or stolen, and too low encourages users to employ easy-to-remember (and therefore easy-to-guess) passwords.

9. **Don't use expired SSL/TLS certificates.**

   Using expired SSL/TLS certificates with AWS services may lead to errors on services such as ELB or custom applications hosted on AWS, which would impact business productivity.

Secure configuration

1. **When using CloudFront, ensure CloudFront distributions use HTTPS.**

   Enabling SSL/TLS ensures all traffic to and from CloudFront is encrypted and minimizes the risk of a man-in-the-middle attack.

2. **Restrict access to CloudTrail bucket.**

   Unrestricted access to CloudTrail logs should never be enabled for any user or administrator account. While most AWS users and administrators will not have any malicious intent to cause harm, they are still susceptible to phishing attacks that could expose their account credentials and lead to an account compromise. Restricting access to CloudTrail logs will decrease the risk of unauthorized access to the logs.

3. **Encrypt CloudTrail log files at rest.**

   In order to decrypt encrypted CloudTrail log files, a user must have decryption permission by the customer-created key management, or customer master key (CMK), policy along with permission for accessing the S3 buckets containing the logs. This means that only users whose job duties require it should have both decryption permission and access permission to S3 buckets containing CloudTrail logs.

4. **Encrypt Elastic Block Store (EBS) database.**

   As an added layer of data security, ensure that the EBS database is being encrypted. Keep in mind that this can only be done at the time when you create the EBS volume. In order to encrypt EBS volumes that weren't encrypted at creation, you must create a new encrypted EBS volume and transfer the data from the unencrypted volume to the encrypted one.

5. **Provision access to resources using IAM roles.**

   Provisioning access to resources using IAM roles is recommended versus providing an individual set of credentials for access. This ensures that credentials are not lost or misplaced accidentally, leading to account compromise.

6. **Ensure EC2 security groups don't have large ranges of ports open.**

   With large port ranges open, vulnerabilities could be exposed. An attacker can scan the ports and identify vulnerabilities of hosted applications without easy traceability due to large port ranges being open.

7. **Configure EC2 security groups to restrict inbound access to EC2.**

   Excessive permission to access EC2 instances should not be allowed. Instead of whitelisting large IP ranges to access EC2 instances, be specific and only whitelist individual private IP addresses for EC2 instance access.

8. **Avoid using root user accounts.**

   The root user is created automatically when creating an AWS account for the first time. This user has access to all services and resource in the AWS account, making it the most privileged user account. As such, the root user account should only be used in the instance of creating the first IAM user. Beyond that, root user credentials should be securely locked away and access to them forbidden.

9. **Use secure SSL ciphers when connecting between the client and ELB.**

   Using stale or insecure ciphers, or those with known vulnerabilities, could lead to an insecure connection between the client and the ELB load balancer.

10. **Use secure SSL versions when connecting between the client and ELB.**

    Using stale or deprecated SSL versions, or those with known vulnerabilities, could lead to an insecure connection or man-in-the-middle attack between the client and the ELB load balancer.

11. **Use standard naming (tagging) convention for EC2.**

    EC2 instances must use a standard/custom convention to reduce the risk of misconfiguration. For example, you can use ec2-RegionCode-AvailabilityZoneCode-EnvironmentCode-ApplicationCode, which could turn into ec2-us-west-1-2b-p-nodejs.

12. **Encrypt Amazon’s Relational Database Service (RDS).**

    As an added layer of security and to ensure you’re compliant with possible data security policies, ensure that your database is being encrypted.

13. **Ensure access keys are not being used with root accounts.**

    Using access keys with the root account is a direct vector for account compromise. Anyone who gets access to the key has access to all the services in the AWS account. Creating role-based accounts with appropriate privileges and access keys is the recommended best practice.

14. **Use secure CloudFront SSL versions.**

    Using stale, deprecated, or SSL versions with known vulnerabilities could lead to an unsecure connection or man-in-the-middle attack in your CloudFront traffic.

15. **Enable the require_ssl parameter in all Redshift clusters.**

    To encrypt all Redshift traffic on the wire and minimize the risk of a man-in-the-middle attack, all Redshift clusters must have the require_ssl parameter enabled.

16. **Rotate SSH keys periodically.**

    Rotating SSH keys periodically will significantly reduce the risk of account compromise due to developers accidently sharing SSH keys inappropriately.

17. **Minimize the number of discrete security groups.**

    Enterprises must consciously minimize the number of discrete security groups to decrease the risk of misconfiguration leading to account compromise.

Inactive entities

1. **Reduce the number of IAM groups.**

   Reducing unused or stale IAM groups also reduces the risk of accidentally provisioning entities with older security configurations.

2. **Terminate unused access keys.**

   Unused access keys increase the threat surface of an enterprise to a compromised account or insider threat. It is highly recommended that any access keys unused for over 30 days be terminated.

3. **Disable access for inactive or unused IAM users.**

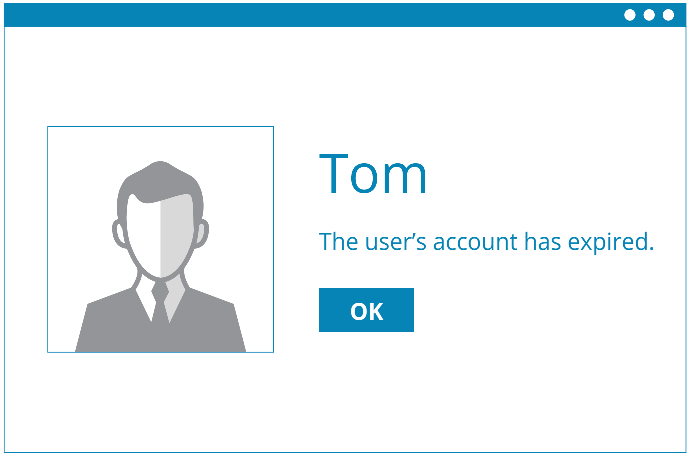
   As a best practice, unused IAM user accounts, or users who haven't logged into their AWS accounts in over 90 days, should have their accounts disabled. This reduces the likelihood of an abandoned account being compromised and leading to a data breach.

4. **Remove unused IAM access keys.**

   Removing unused AWS IAM credentials can significantly reduce the risk of unauthorized access to your AWS resources. Access must not be enabled to resources for IAM users who have left the organization, or applications or tools that are no longer using these resources.

5. **Delete unused SSH Public Keys.**

   Deleting unused SSH Public Keys lowers the risk of unauthorized access to data using SSH from unrestricted locations.



Tom

The user's account has expired.

Access restrictions

1. **Restrict access to Amazon Machine Images (AMIs).**

   Unrestricted access to AMIs makes these AMIs available in the Community AMIs, where everyone with an AWS account can use them to launch EC2 instances. Most of the time, AMIs will contain snapshots of enterprise-specific applications (including configuration and application data). Hence, unrestricted access to AMIs is not recommended.

2. **Disallow unrestricted ingress access on uncommon ports.**

   Allowing unrestricted inbound access to uncommon ports can increase opportunities for malicious activity such as hacking, data loss, brute-force attacks, DoS attacks, and others.

3. **Restrict access to EC2 security groups.**

   Unrestricted access to EC2 security groups opens an enterprise to malicious attacks such as brute-force attacks, DoS attacks, or man-in-the-middle attacks.

4. **Restrict access to RDS instances.**

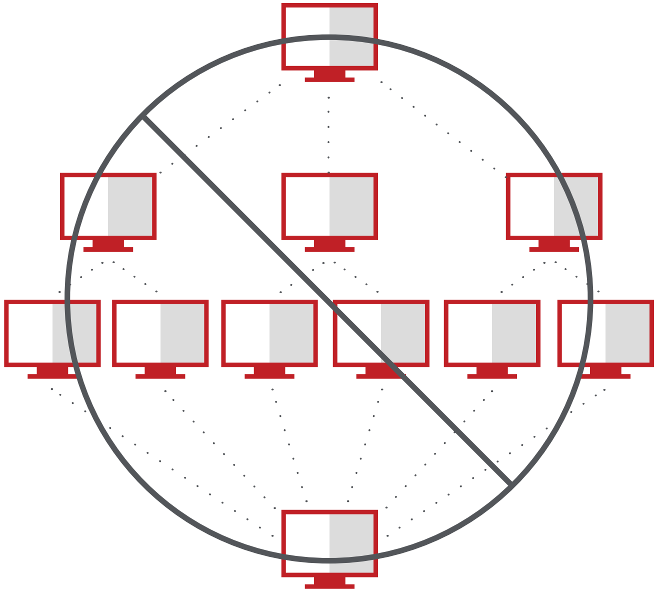
   When the VPC security group associated with an RDS instance allows unrestricted access (that is, the source is set to 0.0.0.0/0), entities on the Internet can establish a connection to your database. This increases the risk of malicious activities such as brute force attacks, SQL injections, or DoS attacks.

5. **Restrict access to Redshift clusters.**

   When the Redshift clusters are publicly accessible, entities on the Internet can establish a connection to your databases. This increases the risk for malicious activities such as brute-force attacks, SQL injections, or DoS attacks.

6. **Restrict outbound access.**

   Outbound access from ports must be restricted to required entities only, such as specific ports or specific destinations.



7. **Restrict access to well-known ports such as:**
   a. **CIFS**—Ensure that access through port 445 is restricted to required entities only. CIFS is a commonly used protocol for communication and sharing data. Unrestricted access could potentially lead to unauthorized access to data.
   b. **FTP**—Ensure that access through port 20/21 is restricted to required entities only. FTP is a commonly used protocol for sharing data, and if left unrestricted, could lead to unauthorized access to data or an accidental breach.
   c. **ICMP**—Ensure that access for ICMP is restricted to required entities only. Unrestricted access could lead to unauthorized access to data, as attackers could use ICMP to test for network vulnerabilities or employ DoS against the infrastructure.
   d. **MongoDB**—Ensure that access through port 27017 is restricted to required entities only.
   e. **MSSQL**—Ensure that access through port 1433 is restricted to required entities only.
   f. **MySQL**—Ensure that access through port 3306 is restricted to required entities only.
   g. **Oracle DB**—Ensure that access through port 1521 is restricted to required entities only.
   h. **PostgreSQL**—Ensure that access through port 5432 is restricted to required entities only.
   i. **Remote desktop**—Ensure that access through port 3389 is restricted to required entities only.
   j. **RPC**—Ensure that access through port 135 is restricted to required entities only
   k. **SMTP**—Ensure that access through port 25 is restricted to required entities only. Unrestricted SMTP access can be misused to spam your enterprise, and launch DoS and other attacks.
   l. **SSH**—Ensure that access through port 22 is restricted to required entities only.
   m. **Telnet**—Ensure that access through port 23 is restricted to required entities only.
   n. **DNS**—Ensure that access through port 53 is restricted to required entities only.

Custom Applications Security Best Practices

Information security is a shared responsibility, and not just between Amazon and its customers. While developers may prioritize speed above all, IT security needs to be part of the software development process. Whether you follow a waterfall or agile methodology, there is a role for IT security in the architecture planning, auditing, and testing of applications. Experience has demonstrated that application security is improved when the IT security team is involved from the beginning, instead of bringing in the security team after an application has been developed.

What follows are recommendations for creating a successful DevOps workflow that integrates security.

1. Inventory and categorize all existing custom applications by the types of data stored, their compliance requirements, and the possible threats they face.

The first step in securing custom application development and usage is to inventory all existing applications and the data uploaded to them. IT security and audit teams should have visibility not only into the number of these applications running on AWS but also on whether sensitive data is being uploaded. Visibility into sensitive data enables the security team to identify which internal and external regulations apply to an app and its data, and what kind of security controls must be in place to protect it.

2. IT security should be involved in testing throughout the development process.

DevOps should invite the IT security team to bring their own application testing tools and methodologies when pushing production code, without slowing down the process. Security should team up with the QA team to define test cases and qualifying parameters that should be met before code can be promoted.

Developing a secure application isn't enough. IT security should also ensure that end users are using the application in a secure manner. With that in mind, there are a few steps the security team can take to ensure appropriate use of these applications.

3. Grant the fewest privileges possible for application users.

Unrestricted or overly permissive user accounts increase the risk and damage of an external or internal threat. Internally, a user with too many permissions might inadvertently cause data loss. Externally, a hacker who compromises an account with too many permissions can easily wreak havoc. For this reason, application administrators should limit a user's permissions to a level where they can only do what's necessary to accomplish their job duties.

4. Enforce a single set of data loss prevention policies across custom applications and all other cloud services.

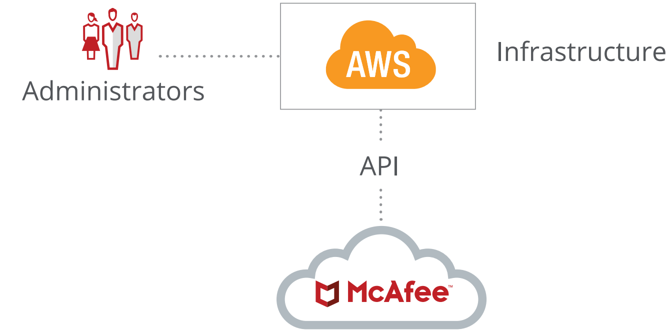
The first step in enforcing DLP policies is inventorying existing DLP policies for all cloud services, on-premises applications, and endpoints, and identifying the policies that would apply to custom applications. Enterprises also need to understand how a custom application is being used, including the number of files containing sensitive data, the number of files being shared, and anomalous usage events indicative of threats. Some of the types of sensitive data that should be protected are:

- Credit card numbers
- Social Security numbers
- Account numbers
- Salaries
- IP addresses
- Account credentials
- Intellectual property

5. **Encrypt highly sensitive data such as protected health information (PHI) or personally identifiable information (PII).**

   It is important to encrypt files containing sensitive information such as Social Security numbers or protected health information. In some cases, policies may be put in place that would block users from uploading files containing sensitive data that may not be encrypted.

How a Cloud Access Security Broker Helps Secure AWS



Amazon offers many built-in security capabilities, giving enterprises the ability to enforce a wide range of security, compliance, and governance policies. However, AWS settings can be very deep. In sprawling AWS environments, it can be prohibitive from a resource standpoint to manually check security configurations and user permissions for potential risks, and next to impossible to sift through the events provided by AWS CloudTrail to uncover potential threats. A cloud access security broker (CASB) helps automate the process of securing AWS—both the AWS platform and services, as well as the custom applications you deploy in AWS.

On the infrastructure side, a mature CASB can provide comprehensive threat protection, monitoring, auditing, and remediation to secure all your AWS accounts. What follows are some of the things a CASB enables you to do.

1. **Detect compromised accounts, insider threats, and privileged access misuse across AWS**

   CASBs combine machine learning and user and entity behavior analytics (UEBA) to analyze cloud activity across multiple heuristics. This allows a CASB to develop a model for typical user behavior and detect anomalous activity patterns across AWS accounts and other cloud services that may be indicative of a threat. For example, if a user generally logs into AWS from Austin, Texas, and one day they log in from Beijing, China in a time frame so short it would be impossible to travel, a CASB can highlight this event and flag it for further investigation.

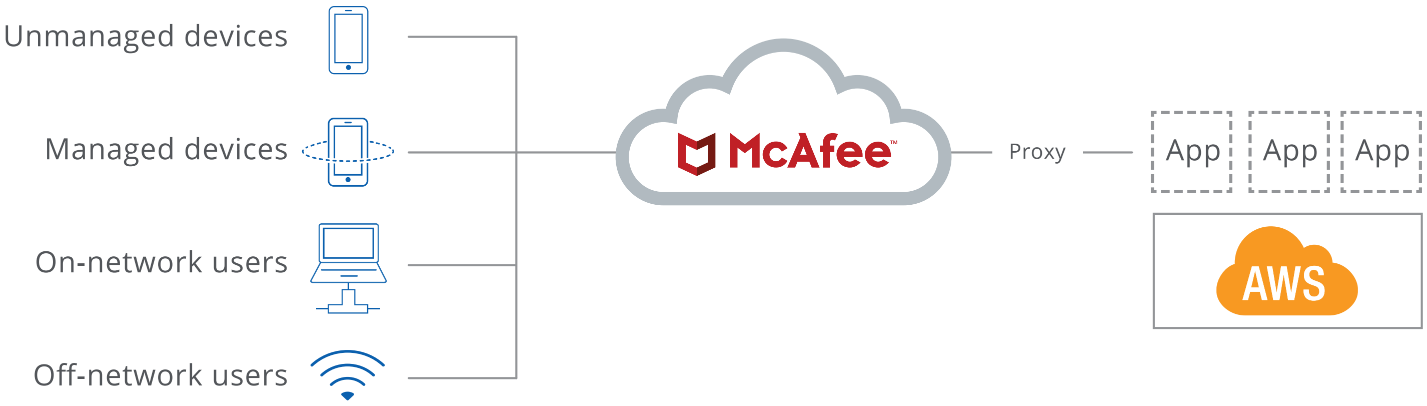
   CASBs detect compromised accounts based on impossible travel as well as excessive failed login attempts, brute-force attacks, login attempts from untrusted or disparate locations, and other scenarios. CASBs can also detect potential insider or privileged user threats by monitoring inappropriate escalation of privileges or repeated authorization failures. Machine learning makes it possible to detect these threats without configuring any rules or policies. CASBs also support the ability to tune the sensitivity of detection to narrow in on certain threats or cast a wider net.

2. **Perform forensic investigations with a complete audit trail of user activity**
   CASBs provide complete and granular visibility into how users are using AWS, including root, IAM, and federated users. Using a CASB, an enterprise can readily detect (in real time) creation, modification, or removal of AWS resources by any user. Security analysts can view the entire audit trail and filter by activity type, user, geography, and other dimensions. In doing so, a CASB can dramatically accelerate post-incident investigations and decrease incident response time.
3. **Identify excessive IAM permissions and dormant accounts**
   A CASB can highlight dormant user accounts that have a heightened risk of being compromised. For example, if an IAM user has been inactive for 90 days, a security alert can be triggered by the CASB for further investigation. Once the IAM user has been disabled, the alert will be resolved automatically without requiring any further action by the AWS administrator. CASBs can also identify excessive IAM permissions such as when an account has access to the AWS console, has unrestricted access to a CloudTrail S3 bucket, or when excessive IAM permissions and policies are associated to a user account instead of IAM groups or roles.
4. **Analyze and audit AWS security configurations to ensure compliance and lower risk**
   While Amazon has provided a set of configurable controls to help protect an AWS account, it is entirely up to the customer to ensure that these settings are configured appropriately, and that any misconfiguration or change in configuration is identified and remediated in real time. To that end, CASBs provide enterprises with the necessary security auditing that can automatically flag any security misconfiguration across all AWS accounts. Misconfigurations flagged include:
   - CloudTrail not enabled
   - Unrestricted access to a CloudTrail S3 bucket
   - Multifactor authentication not required to delete a CloudTrail S3 bucket
   - Access logging not enabled for a CloudTrail S3 bucket
   - MFA not enabled for root accounts
   - MFA not enabled for IAM users with console password

How a Cloud Access Security Broker Helps Secure Custom Applications Deployed in AWS

Traditionally, CASBs have focused on securing SaaS applications such as Salesforce, Box, and Office 365 by providing DLP, activity monitoring, threat protection, access control, and encryption. Today, a mature CASB can extend these same controls to all custom applications an enterprise deploys in AWS. One of the challenges today is that IT security is often not involved in the development process for new custom applications. A CASB makes it possible to extend these security controls to custom applications without any code changes to the application, enabling organizations to secure their existing applications without any development resources. What follows are some of the things a CASB enables you to do to secure custom applications.

> "By 2020, 85% of large enterprises will use a cloud access security broker product for their cloud services, which is up from fewer than 5% today."
>
> —Gartner, "Market Guide For Cloud Access Security Brokers"



1. **Capture a complete audit trail of user activity within the application**

   CASBs provide complete and granular visibility into user and administrator activity within custom applications and across all other cloud services. CASBs reveal who is accessing which applications, what types of data are being uploaded or downloaded, with what kind of device, and by whom. This level of visibility into activity supports compliance efforts and helps accelerate post-incident forensic investigation while decreasing incident response time.

2. **Limit access to data on BYOD devices, and enforce other access control policies as needed**

   While deploying custom applications on AWS provides the fundamental benefit of letting employees access critical resources from anywhere, at any time, using any device, it also introduces security risks. This is because sensitive data could be exposed after downloading to an unmanaged or unsecure bring-your-own-device (BYOD) endpoint.

   CASBs provide contextual access controls that enforce distinct access policies for custom applications based on whether the device is managed or unmanaged, if the IP is blacklisted or safe, or whether the traffic originates from a trusted or untrusted location. CASBs can also force additional authentication steps if predefined risk conditions are met.

3. **Detect insider/external threats and compromised accounts**

   Given the ubiquity of compromised accounts, insider threats, and privileged user threats, CASBs provide threat protection to custom applications hosted in the cloud. CASBs not only analyze anomalous activities within a custom application, but also correlate activities across all custom and SaaS applications to sift through the noise and identify true threats. CASBs can create an alert when, for example, an internal employee downloads a large amount of data onto a personal device right before taking a position at a competitor company, or when a privileged user performs an unwarranted permissions escalation. CASBs can also detect external threats such as when a third party attempts to log into an account using compromised credentials.

4. Enforce data loss prevention

Controlling the upload and sharing of sensitive data is one of the common use cases for a CASB. Organizations in highly regulated industries such as financial services, healthcare, and government, who want to take advantage of the benefits of cloud computing while staying compliant with internal and external policies, turn to CASBs for their cloud data loss prevention requirements.

The recommended platform approach to cloud DLP ensures that the same policies that protect data in SaaS and on-premises applications can be used to protect data in custom applications hosted on AWS. CASBs provide multiple remediation options when a policy violation occurs, including blocking the upload, coaching the user, or notifying an administrator.

5. Encrypt data uploaded to custom applications

Enterprises looking for an additional layer of security can use a CASB to encrypt data in custom applications using their own encryption keys. Using enterprise-owned encryption keys ensures that the IaaS provider cannot decrypt and view the data.

Aside from strengthening the security of the custom application, storing data encrypted has another benefit. Numerous regional and industry-specific laws, including HIPAA-HITECH, require organizations to notify customers whose data has been compromised in a breach. However, if that data has been made indecipherable with encryption, organizations are exempt from the breach notification requirement.

Learn More about McAfee® Skyhigh Security Cloud for AWS

McAfee Skyhigh Security Cloud for Amazon Web Services (AWS) is a comprehensive monitoring, auditing, and remediation solution for your AWS environment and custom applications.

www.skyhighnetworks.com

Download a data sheet for a complete list of product capabilities.

- Skyhigh for AWS
- Skyhigh for Custom Apps

1. Cloud Security Alliance, “Custom Applications and IaaS Trends 2017, CSA Report.”
2. Gartner, “Gartner Says Worldwide Cloud Infrastructure-as-a-Service Spending to Grow 32.8 Percent in 2015.”
3. Structure Research, “Market Share Report: Massive Scale Cloud.”
4. Gartner, “Magic Quadrant for Cloud Infrastructure as a Service, Worldwide 2016,” 2016.
5. Verizon, “2016 Data Breach Investigations Report.”

About McAfee

McAfee is the device-to-cloud cybersecurity company. Inspired by the power of working together, McAfee creates business and consumer solutions that make our world a safer place. By building solutions that work with other companies' products, McAfee helps businesses orchestrate cyber environments that are truly integrated, where protection, detection, and correction of threats happen simultaneously and collaboratively. By protecting consumers across all their devices, McAfee secures their digital lifestyle at home and away. By working with other security players, McAfee is leading the effort to unite against cybercriminals for the benefit of all.

www.mcafee.com

McAfee™
Together is power.

2821 Mission College Blvd.
Santa Clara, CA 95054
888.847.8766
www.mcafee.com