Data Security in the Cloud

A Comprehensive Guide

Introduction

Data security is one of the biggest concerns of cloud-based organizations. Security incidents happen every day and can cost companies their reputation, customers, as well as fines.

In this eBook, we will understand how to encrypt data, what is data classification and why is it important, as well as how to apply in your AWS environment all the best practices recommended to ensure a robust security posture.

1. How to secure data

To protect users' data, the first step is ensuring confidentiality. This can be achieved through encryption.

1.1 Types of Encryption for in Motion, in Use, at Rest Data

Encryption is the process of altering data in order to hide its content and ensure confidentiality. Entities that do not have the decryption key in their possession cannot decrypt the data and, therefore, read its content.

How does encryption work?

Plaintext data is transformed, using an encryption algorithm and a secret key, to ciphertext, which is unreadable text.

There are two types of encryption algorithms:

- Symmetric,
- Asymmetric.

In symmetric algorithms, the key used to perform the encryption is the same as the one used to decrypt it and is, therefore, secret.

Examples of symmetric algorithms are:

- DES (Data Encryption Standard),
- 3DES (Triple DES),
- AES (Advanced Encryption Standard).

The latter one is, in 2022, the industry standard and is recommended to be used with 128 bits keys.

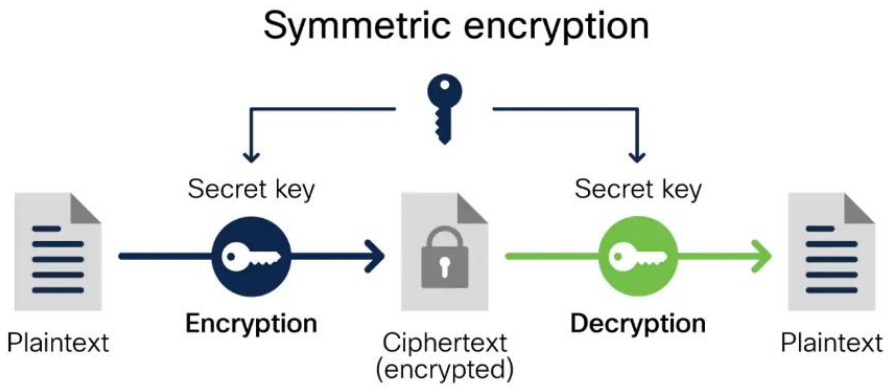


Image source – cisco.com

Asymmetric algorithms use two different keys: a public key for encryption and a private key for decryption.

Asymmetric algorithm examples are:

- RSA (Rivest-Shamir-Adleman),
- ECC (Elliptic Curve Cryptography).

Asymmetric algorithms are not commonly used for encryption because they are slower. For example, the RSA algorithm requires keys between 1024 and 4096 bits, which slows down the encryption and decryption process.

These algorithms can be used, however, to encrypt symmetric algorithm keys when they are distributed.

A more common usage of asymmetric algorithms is digital signatures. They are mathematical algorithms that are used to cryptographically validate the authenticity and integrity of a message or media on the internet.

What is encryption used for?

Encryption ensures confidentiality of data. The unreadable ciphertext keeps the data private from all parties that do not possess the decryption key.

Data has three states:

- In motion,
- In use,
- At rest.

It is essential to understand these states and ensure that the data is always encrypted. It is not enough to encrypt data only when it is stored if, when in transit, a malicious party can still read it.

Therefore, we will look at encryption mechanisms for all three data states.

In Motion Encryption

Data in motion, or in transit, is data that is moved from one location to another, for example, between:

- computers,
- services,
- virtual machines,
- applications,
- networks.

Examples of data in motion are:

- emails,
- files,
- messages.

Data in motion can be encrypted using SSL/TLS. TLS (Transport Layer Security) and SSL (Secure Sockets Layer) are transport layer protocols that protect the data in transit. TLS is a newer and improved version of SSL.

SSL/TLS ensure confidentiality through encryption. Firstly, a session is created between the two parties exchanging a message using asymmetric encryption. Then, after the secure session is established, symmetric algorithms are used to encrypt the data in motion.

Using one of the mentioned protocols prevents attackers from reading the data in motion.

Websites should use HTTPS (Hypertext Transfer Protocol Secure) instead of HTTP to ensure encryption between websites and browsers. HTTPS uses SSL/TLS.

What is in motion data vulnerable to?

Eavesdropping attacks. In this situation, malicious entities can analyze traffic sent over the internet and read unencrypted data.

In Use Encryption

Data currently accessed and used is considered in use.

Examples of in use data are:

- files that are currently open,
- databases,
- RAM data.

Because data needs to be decrypted to become in use, it is essential that data security is taken care of before the actual use of data begins.

To do this, you need to ensure a good authentication mechanism. Technologies like Single Sign-On (SSO) and Multi-Factor Authentication (MFA) can be implemented to increase security.

Moreover, after a user authenticates, access management is necessary. Users should not be allowed to access any available resources, only the ones they need to, in order to perform their job.

A method of encryption for data in use is Secure Encrypted Virtualization (SEV). It requires specialized hardware, and it encrypts RAM memory using an AES-128 encryption engine and an AMD EPYC processor. Other hardware vendors are also offering memory encryption for data in use, but this area is still relatively new.

What is in use data vulnerable to?

In use data is vulnerable to **authentication attacks**. These types of attacks are used to gain access to the data by bypassing authentication, brute-forcing or obtaining credentials, and others.

At Rest Encryption

Data at rest is data that is not currently used or transmitted between computer systems.

This state of data is usually the most sought-for by attackers. Data at rest can be stored in:

- Storage cloud assets such as buckets,
- Databases,
- Files, and others.

The most common method of protecting data at rest is through encryption. In this chapter, we will look at ways to perform encryption and understand its importance.

Why is it important to encrypt data at rest?

There are three main risks regarding data at rest:

- loss,
- leakage,
- theft.

In all three cases, your data at rest will probably end up in somebody's hands. You can forget your USB drive in a coffee shop, you can accidentally disclose your data to someone else or the public or it can be stolen by a malicious attacker.

For these reasons, a safeguarding mechanism is encrypting your data.

Although data at rest is the easiest to secure out of all three states, it is usually the point of focus for attackers.

Data at rest is vulnerable to **exfiltration attacks.** The most common way at rest data is compromised is through exfiltration attacks, which means that hackers try to steal that data. For this reason, implementing a very robust encryption scheme is important.

Another essential thing to note is that, when data is exfiltrated, even if it is encrypted, attackers can try to brute-force cryptographic keys offline for a long period of time. Therefore, a long, random encryption key should be used (and rotated regularly).

How do you encrypt data at rest?

There are a few best practices that need to be considered when undergoing the encryption process:

1. Use an industry-recommended standard with an appropriate key length.

For data at rest, symmetric encryption algorithms are usually used. An industry-recommended standard is AES-256 (Advanced Encryption Standard with a key of 256 bits).

2. Classify data and decide what to encrypt.

Make sure you don't leave any sensitive data unencrypted. Use data classification to decide what data should be encrypted.

Alternatively, perform full disk encryption to protect all data, especially in case you lose the hardware.

Key management

Now that we've established how to encrypt data at rest, let's talk keys.

If your key management is poor, no matter how strong and well-done encryption is, it can become totally useless.

Follow the best practices we're recommending to ensure textbook key management.

1. Use a random key generation algorithm for your keys.

Most random number generator algorithms are not truly random; they are called Pseudo-Random Number Generators (PRNGs).

If you're using programmatic functions such as random() or rand() from C++, Java, and other languages, you're not generating random keys; they use a seed (which always gives the same result), can be predicted, and are not for cryptographic usage.

For this reason, you need to use tools that utilize Cryptographically Secure Random Number Generators (CSRNGs).

To generate a random, secure key, you can use:

- GenerateRandom, a tool provided by AWS,
- the GenerateRandomBytes API from Google Cloud,
- SecureRandom, a class from Java, and others.

2. Store your keys separately from your ciphertext.

Do not store your keys in the same place as encrypted data, and do not hardcode them in the source code. You can use:

- dedicated hardware devices such as Hardware security modules (HSM),
- key management systems such as Azure Key Vault and AWS KMS,
- open source KMS such as HashiCorp Vault.

3. Rotate the keys.

Change the keys regularly (every 90 days or less). This process involves retiring an encryption key and generating a new one.

Moreover, if a key is compromised, immediately replace it and assess which data is at risk.

4. Implement access control for keys

Ensure that access to keys is heavily restricted in the following ways:

Implement the Least Privilege Principle. Only the individuals that need the keys should be able to access them. You can also implement time windows when keys can be accessed.

Only authorized personnel should be able to access keys. Ensure that, after you've granted access rights to people, only they can see and use the keys.

5. Manage key deletion

If a key is permanently deleted, all data encrypted with that key is lost. The key should be appropriately destroyed after all the encrypted data is decrypted and re-encrypted with a new key.

Solutions from cloud vendors for safe key deletion are:

- Soft delete in Azure Key Vault,
- Key deletion scheduling in AWS.

Server-side encryption - cloud solutions

Server-side encryption is implemented in the cloud, and the cloud vendor usually takes care of the key. This method of encryption is easier to use, since the cloud provider takes care of the algorithm, the key management system, and other troubles you may have.

AWS, Azure, and GCP provide data at rest encryption and key management solutions. Let's look at the available options and how to make sure you're using them correctly.

Encryption in AWS

The following services in AWS support data at rest encryption capabilities:

- Amazon EBS,
- Amazon S3,
- Amazon RDS,
- Amazon Redshift,
- AWS Lambda, and many others.

Key management is done using the AWS Key Management Service, which allows users to utilize their own keys or let AWS deal with them.

Encryption in Azure

In Microsoft Azure, users have the following options:

- Azure Disk Encryption, for Virtual Machines,
- Azure Storage and Azure SQL Database, which encrypt all data at rest.

For key management, Azure provides the following services:

- Azure Key Vault,
- Vault Managed Hardware Security Model (HSM).

Encryption in GCP

For key management, GCP provides the Google Key Management Service. As an additional layer of security, the encryption key, named DEK (Data Encryption Key), is also encrypted using a KEK (Key-encryption key).

1.2 What is Data Classification and Why is it Important?

Data classification is a way of grouping data to ensure easy sorting, retrieval, and prioritization.

The data is divided into categories, and a label, or a tag, is applied to make it easily searchable.

The three commonly used types of data classification are:

- **Content-based**, which is done solely based on the information involved,
- **Context-based**, which takes into account the location of the data, the owner, the application it is used in, and others,
- **User-based**, which requires users to label data based on internal rules.

In this chapter, we will understand how valuable data classification is for a company using cloud services, as well as how AWS, Azure, and GCP handle the process of labeling/tagging assets.

Benefits of data classification

1. Risk management

According to AWS, data classification is a foundational step in cybersecurity risk management. The reason behind this is that applying labels to data and establishing security requirements such as:

- the level of confidentiality,
- the need for integrity checking,
- the sensitivity of data,

can help your company manage risks efficiently.

2. Compliance

When implementing compliance with international standards, you must know what type of data your company is managing and storing.

Data classification should be done correctly to understand which of the data you're storing is confidential/sensitive. You cannot comply with recognized frameworks unless you correctly handle confidential data (and you cannot do this unless you know which data is confidential).

Let's look at a **scenario** – if you're storing customer PII, but you are not aware of the criticality of that data, you may not even think of protecting it, for example, by encrypting it.

Therefore, your company may not be compliant with standards like:

- SOC 2,
- PCI-DSS,
- GDPR, and others.

3. Security

Organizing data into categories and using labels can help you maintain:

- **confidentiality**, because you will turn your focus to the most sensitive data,
- **integrity**, because you can mark the need for integrity as high for some data using labels,
- **availability**, which can be explicitly ensured for data that needs to be highly available and is labeled as such.

Much of the data that used to be saved on-premises is now saved and processed in the cloud, in databases, assets of type storage, and others.

Data classification – a cloud overview

We will look at the top 3 cloud vendors – AWS, Microsoft Azure, and GCP – to see how data classification can be implemented and the different types of tags that can be applied depending on the cloud service selected.

1. Amazon Web Services

In the AWS documentation, a three-tiered classification is recommended, with the following tag names:

- Unclassified,
- Official,
- Secret and above.

Moreover, AWS presents the following three labels used by NIST (National Institute of Standards and Technology), a United States government agency, and recommends them:

- Low,
- Moderate,
- High,

which classify the impact a potential data breach would cause on that data.

However, these tags are recommendations and users can implement their own tags. Later in this chapter, you will find best practices on how to implement labeling for your cloud environment.

When you create a resource in AWS, you can add tags (key-value pairs) to the resource to associate it to labels used in data classification.

2. Google Cloud Platform

For data classification in GCP, we can find both labels and tags, which are two different things.

A label is described as a key-value pair that you can create using the Resource Manager API and the Google Cloud console. These can be used to separate resources in terms of billing, to add information about resource state, and so on.

Tags, however, are the tools that allow GCP customers to classify data and establish rules based on their classification.

The difference between labels and tags in Google Cloud is that labels are simply metadata added to resources, while tags categorize assets and can be used when defining policies and rules (for example, who is allowed to access a certain asset) in your GCP environment.

3. Microsoft Azure

In Microsoft Azure, we can use the Microsoft Purview service to ensure data labeling of cloud assets.

Microsoft Purview is a solution offered by Microsoft that brings together your cloud, on-premises, and SaaS data and helps you manage it through different solutions:

- Data Map,
- Data Catalog,
- Data Sharing,
- Data Estate Insights, and others.

An important aspect is that Data Map powers most of the solutions offered by Microsoft Purview and is a paid service.

In terms of data classification, there are a few services that can help you manage your cloud resources:

- the Microsoft Purview Data Catalog uses sensitivity labels that can be added to cloud assets.
- the Microsoft Purview Information Protection service, which has the following features: data classification, trainable classifiers, sensitive information types,
- the Azure Information Protection unified labeling client, a downloadable client that also provides sensitivity labels.

Microsoft Azure suggests that you apply tags that contain additional information about resources (do not include any PII or sensitive data in the tags) to:

- add context to your resources and understand them better,
- be able to use complex filters.

Azure also suggests a "Data classification" tag to describe the sensitivity of data stored or processed by a resource. If an organization does not have their own labels defined, they may use the following values supplied by Microsoft:

- Non-business,
- Public,
- General,
- Confidential,
- Highly confidential.

Moreover, Azure recommends that you also use a formal data classification process. In the next section, we will explain best practices to keep in mind when classifying your resources.

How do you implement data classification?

An important rule to follow when implementing data classification is that the entire organization should use the same classification tags/labels. Using a policy or a procedure for this process that regulates:

- the classification process as a whole,
- the tags' names, and others

is essential to ensuring consistent data classification.

2. Data Security in AWS: an In-depth Analysis

Understanding and managing all your assets and services in the cloud are demanding tasks. It is easy to overlook even the smallest configuration and introduce a vulnerability in your cloud infrastructure.

In this chapter, you will find a comprehensive guide that will help you secure your AWS cloud resources and fix misconfigurations.

2.1 Encryption

For data in transit, AWS provides the following solutions:

1. **Encrypt the data using SSL/TLS.**

2. **Perform client-side encryption.** This solution requires the user to encrypt the data before uploading it to the cloud, but it is more difficult since the client has to deal with the encryption process, key management, and other services.

For data at rest, AWS provides encryption for the following services:

- Amazon EBS,
- Amazon S3,
- Amazon RDS,
- Amazon Redshift,
- AWS Lambda, and many others.

AWS uses the 256-bit Advanced Encryption Standard (AES-256) encryption algorithm.

For key management, AWS provides AWS KMS (AWS Key Management System). It is a comprehensive solution that helps users deal with all the trouble that comes with cryptographic keys.

AWS KMS helps you:

- Create cryptographic keys,
- Define policies and control how the keys are used,
- Audit the keys usage to ensure they are used legitimately.

According to AWS, this service can be used:

1. Through the AWS Management Console,
2. Using the AWS KMS APIs.

2.2 Data classification

While the AWS documentation recommends using a three-tiered approach with the tags mentioned in the Data classification chapter, users can tailor the classification to their needs and use their own tags. In addition, **tag policies** can be used to standardize their creation and ensure consistency across all assets.

To accomplish classification using tags in your AWS environment, you have the following options:

- Using the Amazon console, at resource level, where tags can be added either at creation or after,
- Programmatically, using the Amazon API, AWS CLI, or AWS SDK.

According to the AWS documentation, restrictions regarding tags include:

- There cannot be more than 50 tags per resource,
- Each tag key must be unique for each resource,
- The maximum key length is 128 Unicode characters in UTF-8,
- The maximum value length is 256 Unicode characters in UTF-8.

2.3 Access control

Regulating access control is an essential step to your cloud data security program.

To manage access control in AWS, you can use policies, which can be assigned at the following levels:

- Users,
- Groups of users,
- Roles,
- Resources.

Policies define permissions. To correctly implement them, use the Least Privilege Principle to only allow access rights to the necessary users for the minimum amount of time possible.

Let's look at an example where a policy is applied to an S3 bucket.

A bucket policy contains rules based on which access is allowed or denied and is written in JSON.

```
{
  "Version": "2012-10-17",
  "Id": "S3PolicyId1",
  "Statement": [
    {
      "Sid": "IPAllow",
      "Effect": "Deny",
      "Principal": "*",
      "Action": "s3:*",
      "Resource": [
        "arn:aws:s3:::DOC-EXAMPLE-BUCKET",
        "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*"
      ],
      "Condition": {
        "NotIpAddress": {
          "aws:SourceIp": "54.240.143.0/24"
        }
      }
    }
  ]
}
```

Policy source – docs.aws.amazon.com

Analyzing the image above, we understand that the policy is applied to a bucket resource, the rule for the permission is “Deny”, and the result of the bucket policy is denying access to the objects stored in the specified bucket unless the requests are made with source IPs in the subnet 54.240.143.0/24.

2.4 Data loss prevention

Data loss prevention (DLP) is a protection mechanism for sensitive data that ensures that no unintentional or malicious disclosures occur. DLP prevents data breaches by ensuring no confidential data is accidentally leaked, lost, or stolen.

Amazon Macie is a data security and privacy service that protects users’ sensitive data using machine learning technologies and pattern matching.

This tool identifies sensitive data using **sensitive data discovery jobs**, which analyze S3 buckets. Sensitive data discovery jobs use pre-defined or user-defined lists (or a combination of both) to single out confidential data by matching patterns to the lists.

A passport number is an example of sensitive data that Amazon Macie could match. This is because it has a set number of digits, some corresponding to the owner's region or country.

After identifying sensitive data, Amazon Macie can:

- use IAM policies to filter traffic to it,
- encrypt and decrypt data,
- perform logging and monitoring through AWS CloudTrail integration, and others.

2.5 Availability

Availability means that users should be able to access their data without disruptions at any point.

A solution for availability in the cloud is **availability zones**.

An availability zone is a geographical area where groups of data centers are located. These data centers contain replicated data and provide redundancy regarding electrical power, networking, and connectivity.

An AWS region contains multiple AWS availability zones, all within 100km of each other, which are independent and provide redundancy.

Some AWS regions around the globe are:

- North America,
- South America,
- Europe,
- China,
- South Africa, and others.

These regions provide high availability.

Another solution for availability is DDoS Protection. DDoS (Distributed Denial of Service) attacks are attempts to bring down a service or a resource by sending a large amount of traffic to them using controlled machines.

AWS Shield is the AWS DDoS Protection service that protects applications hosted in the cloud.

AWS Shield has two tiers:

- Standard,
- Advanced.

Besides the features that come with the Standard plan, which are at network and transport layer, the Advanced tier of AWS Shield provides:

- integration with AWS WAF (Web Application Firewall),
- real-time visibility into attacks,
- additional detection and mitigation of sophisticated DDOS attacks, and others.

2.6 S3 Bucket Security

In 2017, 4 million records with customer information, login credentials, and source code were made publicly available due to 2 unsecured AWS S3 storage buckets owned by Time Warner Cable.

The consequences of this attack were disastrous, and this event showed the entire cloud industry how important security is.

What is an Amazon S3 bucket?

An Amazon S3 bucket is a storage cloud asset that acts as a container for data stored in the public cloud. Buckets are object storage services and are similar to folders; this type of storage is flexible and scalable and is ideal for large files and unstructured data.

Common S3 Bucket Misconfigurations

1. Public access to a bucket is allowed.

Sometimes, Amazon S3 buckets are required to be publicly accessible. For example, this use case occurs when the owner intends to make data accessible to the internet.

However, breaches occur when a bucket that has sensitive information such as PII (Personal Identifiable Information) allows:

- Public “READ” access,
- Public “WRITE” access.

You can grant and deny access to a bucket using **access lists** and **bucket policies**.

An access control list (ACL) is a set of rules that limit access to buckets through permissions. It defines an account's access level over a bucket (for example, READ or WRITE).

A bucket policy also contains rules based on which access is allowed or denied, but it is a more modern solution because it can enable more complex filtering. It is a JSON-based access policy language.

Amazon recommends that you no longer use ACLs beside special cases, in which you need to filter access to objects individually.

2. No at rest encryption is performed for old data.

   As of January 2023, AWS encrypts all new objects in S3 buckets by default. However, the user needs to manage the encryption of data added before that. AWS provides multiple server-side encryption options to protect data at rest, such as with Amazon S3 Managed Keys (SSE-S3) and AWS Key Management Service (SSE-KMS).

3. Logging is disabled.

   Logging an S3 bucket is an essential step in securing your data. With logging, you can record actions taken by users, keep log files for compliance purposes and understand what roles have permission to access data inside a bucket.

   There are two solutions for AWS bucket logging:

   - Server access logging, and
   - AWS CloudTrail.

   With server access logging, you obtain detailed records regarding requests that are made to a bucket.

   AWS CloudTrail is a comprehensive service that tracks user activity and API calls. It can be used to keep a record of who sends requests to a bucket.

   It is important to keep in mind that AWS CloudTrail does not log failed authentication attempts through incorrect credentials. However, it does track requests made by anonymous or unauthorized users.

4. No regular backups are performed.

   Attackers may not only try to steal your sensitive data, but they can also delete it. Therefore, ensuring regular and consistent backups is essential to configuring your buckets and providing availability.

   Using AWS Backup, you can perform S3 bucket backups. Amazon supports the following types of backups:

   - **Continuous backups**, which allow data restoration from any moment in the last 35 days,
   - **Periodic backups**, which can be performed every 1 hour, 12 hours, or less often.

An important feature of AWS Backup is that tags, access control lists, and other metadata are also saved along with your data.

An additional layer of security can be added by using the MFA delete feature in AWS. This option requires a successful MFA before allowing a user to delete an object or bucket.

Moreover, you can keep multiple versions of an object inside a bucket. This process is called versioning and can be used to recover objects from accidental deletion.

3. How to Identify Misconfigurations

Finding so many misconfigurations and correctly remediating them can be a very tedious and time-consuming job. We can help you establish a plan to secure the entire cloud environment and keep it that way.

3.1 Keep Track of Findings with Controls

Cyscale has over 400 controls that automatically check for any wrong configurations or vulnerabilities in your organization's cloud infrastructure. These controls work across multiple cloud service providers such as AWS, Azure and GCP.

Here are some examples of controls for AWS that check if you're implementing the best practices described in the previous chapter:

- *Ensure S3 bucket ACL grants permissions only to specific AWS accounts*
- *Ensure all S3 buckets employ encryption-at-rest*
- *Ensure no SQL Databases allow ingress 0.0.0.0/0 (ANY IP)*
- *Ensure CloudTrail logs are encrypted at rest*

3.2 Obtain Visibility Through the Data Security Dashboard

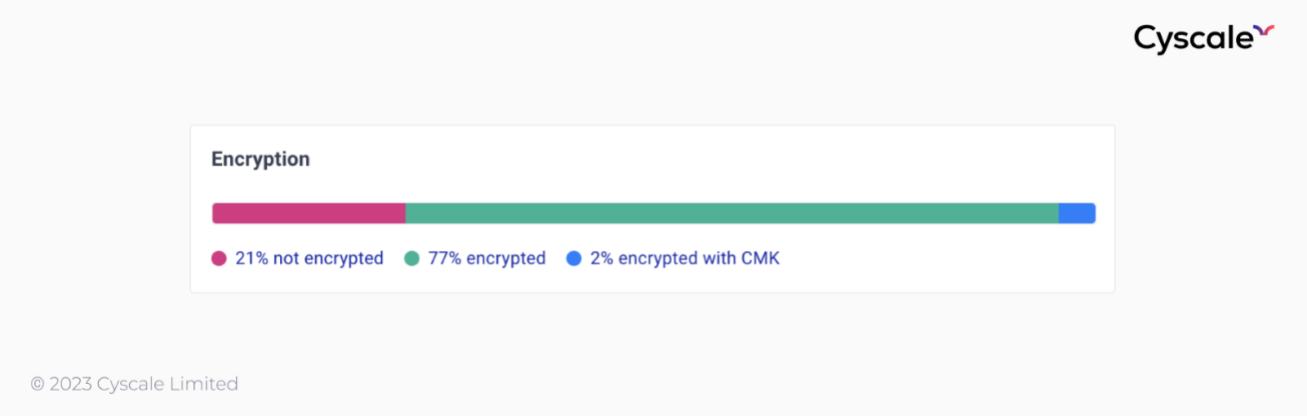
Cyscale's Data Security Dashboard provides the visibility you need for your cloud. The Dashboard displays information about:

- Encryption,
- The management of cryptographic keys,
- Publicly accessible storage assets such as VMs, databases, buckets, and others,
- Databases and misconfigurations related to them,
- Object containers such as blobs and buckets that may vulnerable, and others.

These DSPM (Data Security Posture Management) capabilities enable users to detect attack paths for data storage assets and quickly mitigate them.

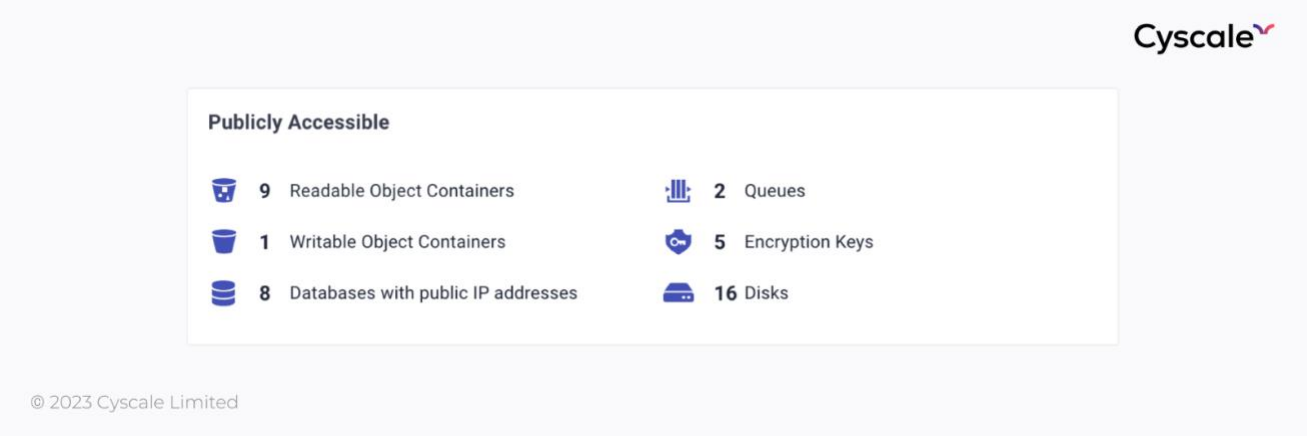
The first section of the dashboard shows the percentage of storage assets that are:

- unencrypted,
- encrypted with provider-managed-key, and
- encrypted with CMK (Customer Managed Key).



This card is a good indicator of progress, and, by clicking on each section of the status bar, we see which assets fit in each of those states. Using this feature, you are at a click away from finding out which of your storage cloud assets are unencrypted.

The next section contains the Publicly Accessible card, which provides visibility over a multitude of assets, as you can see in the image below. When you click on each element, you see a list of affected assets, along with the associated risk.



Let's look at this feature in more detail to understand how this helps secure your cloud infrastructure.

The "**Readable Object Containers**" and "**Writable Object Containers**" refer to storage assets such as buckets and blobs. Object containers should not be publicly accessible unless it is necessary, since individuals could then read or overwrite possibly sensitive data without having to perform any kind of authentication or authorization.

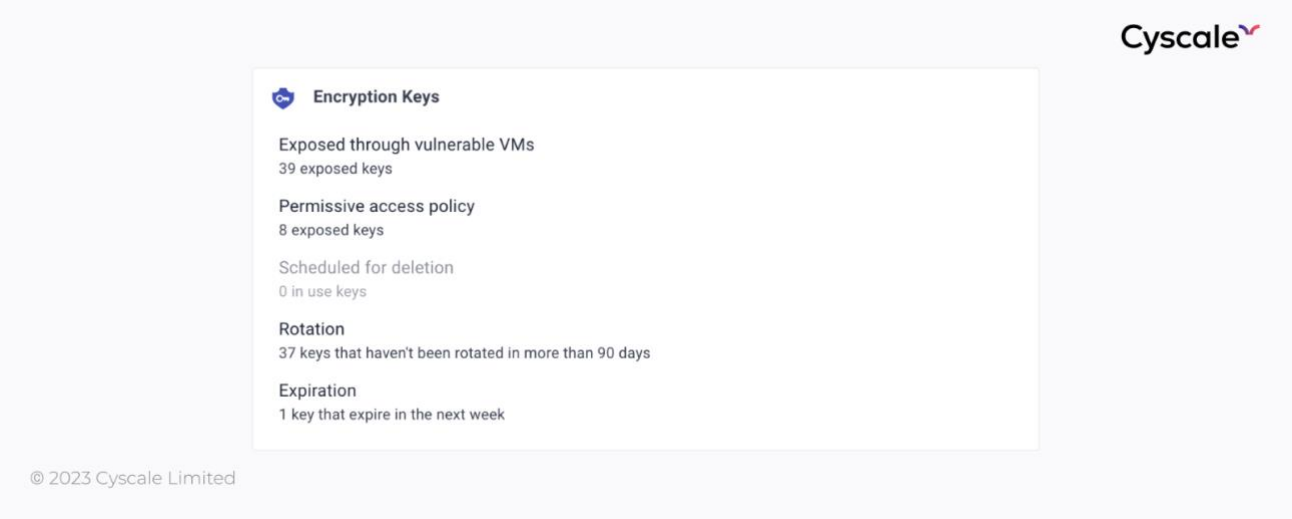
The next item in this section highlights **database instances that have public IP addresses**. Databases should be configured with private IP addresses to reduce attack surface and increase security.

The last items in this list are publicly accessible:

- queues,
- encryption keys, and
- disks.

The control regarding encryption keys checks for attached policies that may allow public access to the key. The other control ensures there are no disks attached to VMs reachable from the internet.

The next card on the Data Security Dashboard provides an overview of the encryption keys used in your cloud infrastructure.

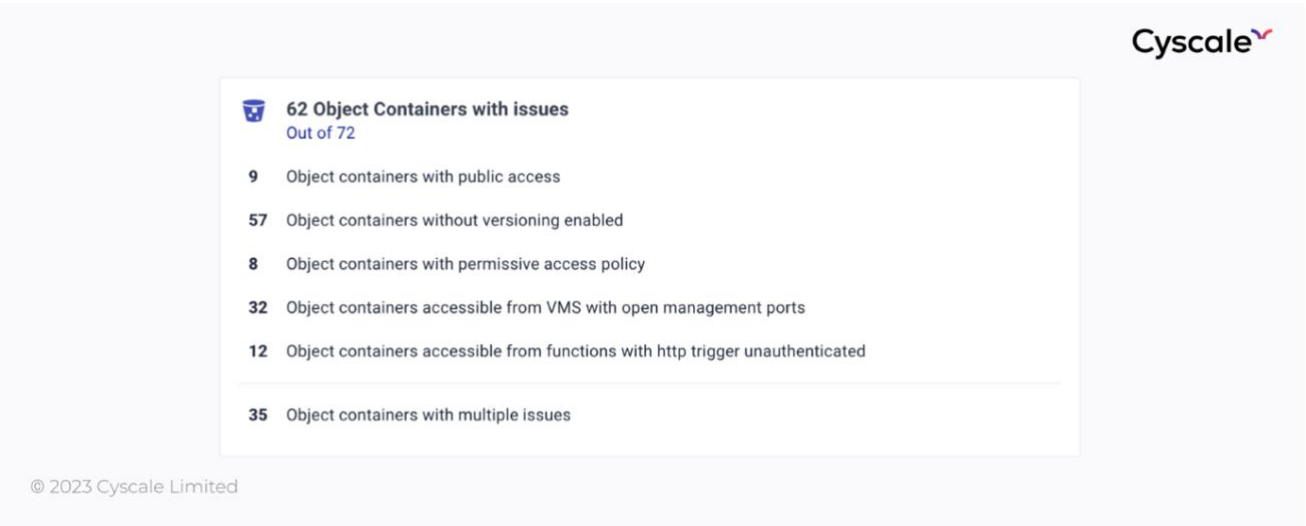


Cyscale checks if the encryption keys are stored on a vulnerable VM or if they have a permissive access policy to identify possible attack paths. Moreover, important information is highlighted, such as:

- keys that are in use and are scheduled for deletion,
- keys that haven't been rotated in a long time, and
- keys that will expire soon.

The next two sections in this dashboard focus on object containers, such as buckets or blobs, and on databases. Here, you can see some of the categories of vulnerabilities Cyscale has identified and checked your cloud environment against.

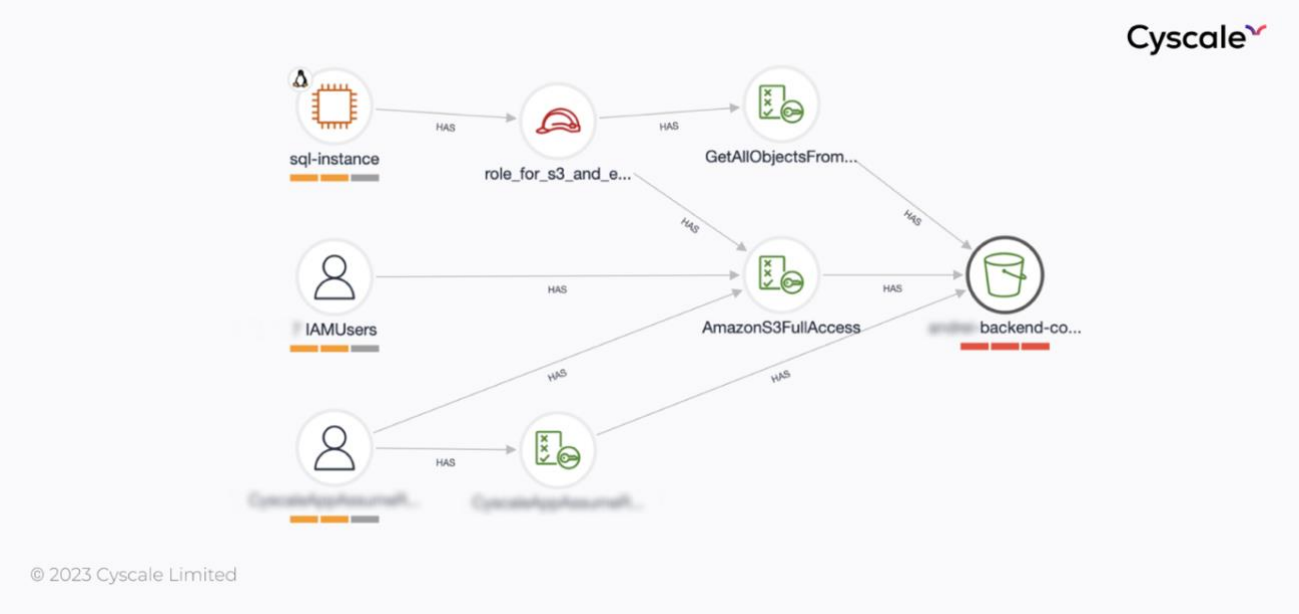
The first card presents the state of your object containers.



Using this feature, we identify attack paths that may compromise your cloud assets and help you solve them.

For example, a common attack is exploiting VMs that have open management ports. If you have a VM that has permissions on a bucket, and that VM is compromised, your bucket may be compromised as well.

Using the Cyscale Knowledge Graph, you can see that the VM named "sql-instance" has an instance profile that gives it full access to the bucket on the far right, and the VM also has port 22 (SSH) open. The VM is thus connected to the internet and therefore puts the data stored in the bucket at risk.

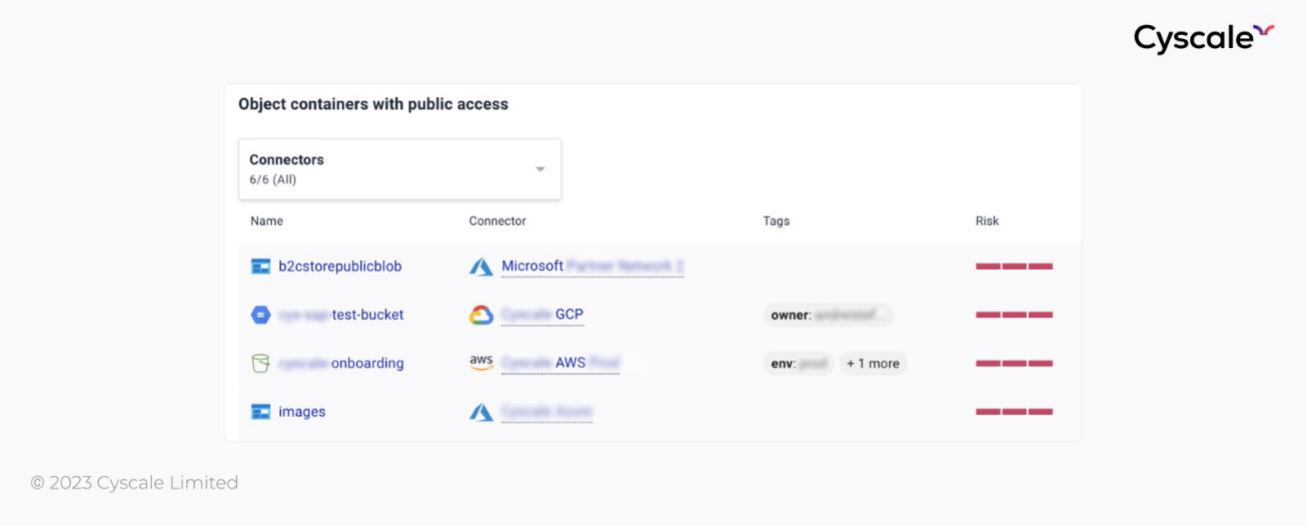


Other critical issues highlighted for object containers include:

- Enabling public access to storage assets,
- Having an overly-permissive access policy,
- Not enabling versioning, and others.

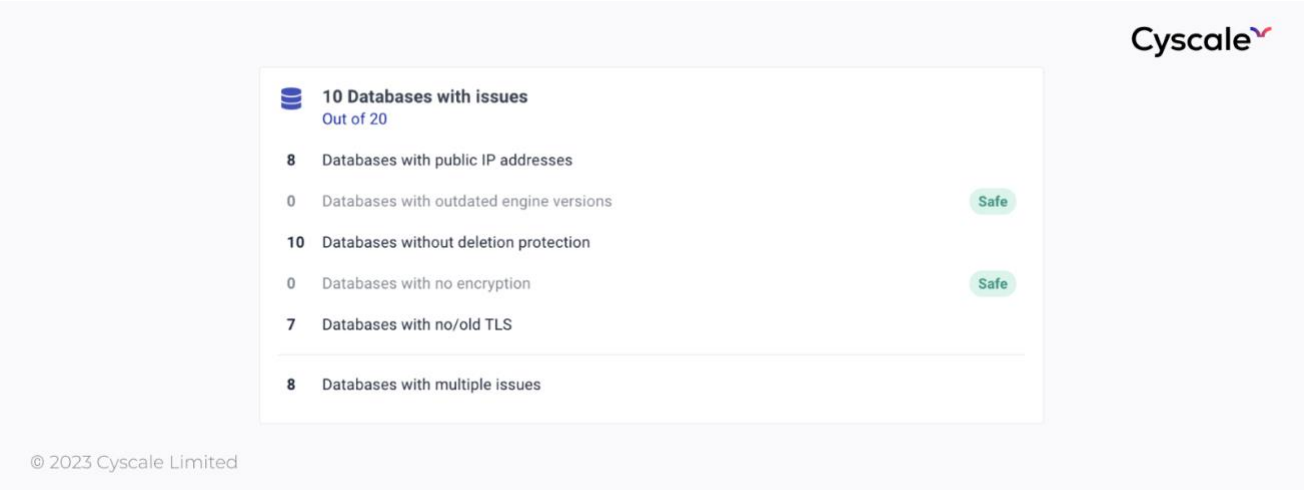
Clicking on “Object containers with public access”, we get a list of misconfigured assets as well as details like:

- Connectors, which represents the cloud or identity provider account,
- Tags,
- Risks.



Moving on to the databases card, using controls, Cyscale checks for the following misconfigurations:

- **Databases with public IP addresses,**
- **Databases with outdated engine versions,**
- **Databases without deletion protection,**
- **Databases with no encryption,**
- **Databases with no/old TLS.**



Use the multitude of features present in the Cyscale Data Security Dashboard to eliminate data exposure through [data storage misconfigurations](#).

3.3 Conclusion

Whether you're working trying to classify data, enable encryption, secure S3 buckets, or you're trying to assess the overall cloud security posture of your organization, Cyscale ensures that your cloud environment is secure.

Check out our product in the playground or schedule a demo with us to start your cloud security journey!