# S3 Bucket Security

# S3 encryption

There are four flavors.

You can also set up a bucket policy to only allow files that are encrypted by one of the following server sided encryption. You would provide that in the S3 bucket JSON policy list. For example: You can specify to only accept files that are encrypted by KMS, and deny all other files encrypted by other schemes.

## Server-side encryption S3

Enabled by default for new buckets and new objects. The encryption use keys that is handled managed and owned by AWS S3.

The object is uploaded then when it reaches S3 it will be encrypted with the key owned by the S3 bucket, and stored into the bucket.

Set header for **x-amz-server-side-encryption: AES256**

## Server-side encryption KMS

The encryption uses key that's handled and managed by AWS KMS. It is also encrypted on the server side

This is good because it has user control and logs every key usage.

Limitation: You will be using KMS APIs which will have their limitation. Limited 5500, 10000, 30000 request per region.

Set header for **x-amz-server-side-encryption: aws:kms**

## Server-side encryption-C (custom key)

The custom key are sent to AWS to be used. **The encryption key is never stored they are discarded right after.**

HTTPS must be used because you will be sending the encryption key over the wire.

## Client-side encryption

Client encrypt the file themselves before sending to S3, and the client must decrypt the file themselves when retrieving it from S3.

Therefore, the client manages the key and encryption cycle themselves.

The files is encrypted during upload, and S3 won't do any encryption. When the file is downloaded the client will be responsible for decrypting it.

### Encryption in transit

How is data encrypted when they are being downloaded.

HTTPS for S3 bucket endpoint have encryption in flight.

HTTP for S3 endpoint have no encryption in flight.

You don't have to worry since most client uses HTTPS by default.

# S3 CORS

origin = protocol + host + port

CORS tells the browser to whether allow the current website to access the resource they fetched from other places. Same origin is allowed by default, if they have the same origin it is allowed by default (same origin policy). If they aren't then CORS needs to explicitly say that this website are allowed to access the resources they requested, otherwise, the website cannot use it.

If client make cross-origin request on your S3 bucket, then it needs to have the correct CORS headers. You can just configure it so that the header allow for specific origin or for * regions.

# S3 MFA delete

Force user to generate a code from their device before doing important operations on S3

Deleting object, suspend versioning on the bucket. They are dangerous operations. Versioning must be enabled to use MFA delete.

Only root account can enable/disable MFA delete.

# S3 Access logs

You might want to log all access to your S3 bucket. The data can then be analyzed.

The log will be deposited to another S3 bucket and the logging bucket must be in the same AWS region.

Do not monitor logging bucket, recursion you will pay lots of money in the end!

# S3 Pre-signed URL

You can generate these URL using console, CLI, or SDK.

There will be an expiration date for the URL. The pre-signed URL inherit the permissions of the user that generated the URL.

This is used for giving out the files stored in a private S3 bucket to other users. So that they can access the file as well.

They will be able to upload to the precise file location as well!

# S3 Glacier vault lock

Adopt the write once read many model. Once you lock the vault it cannot be changed or deleted anymore after you put the object into it.

This is helpful or compliance and data retention.

## S3 object lock

Enable versioning first, also let you do WORM model. It is a lock for each object, it prevents deletion for a specified amount of time.

1. Compliance mode: Object version can't be overwritten or deleted even root user. Retention mode can't be changed or even the period won't be shorten
2. Governance: Most users can't overwrite or delete an object version or alter lock setting, but special users can

Retention period: How long you are going to protect the object, can be extended not shorten.

Legal hold: Protect object indefinitely, independent from retention object. They can be removed.

# S3 access points

You can define access point which is associated with a policy, that grant R/W permission to a specific prefix for S3 bucket, for a particular IAM user. Each access point will have their own DNS name.

This allows you to control who is able to read what "folders" from S3 bucket.

You can also define VPC origin, to only allow access point to be accessed within VPC. To make this work you need to create VPC endpoint policy to allow traffics out from the VPC.

# S3 object lambda

Use case for access point. Change the object before is retrieved from the access point.

Uses a lambda function, the lambda can sit in front of the access point to modify the object before it is returned to the user at the access point.

You need **Access Point + Object Lambda access point** to make this work.

Example use cases: Redacting lambda function for the object before it is returned to the user, or enrich the data before is returned to the application.

Revision #2
Created 19 February 2023 20:38:35 by Tamarine
Updated 20 February 2023 17:48:32 by Tamarine