AWS S3 Security Considerations

Omid Vahdaty, Big Data ninja

Concepts

- protecting data while
  - **in-transit** (as it travels to and from Amazon S3) , 2 ways:
    - by using SSL
    - client-side encryption.
  - **at rest** (while it is stored on disks in Amazon S3 data centers) 2 ways:
    - Server Side encryption. (SSE)
    - client-side encryption.

Encryption Types

- Server Side
  - encrypt your object before saving it on S3 disks
  - decrypt it when you download the objects from S3.
- Client Side
  - Client-side encryption refers to encrypting data **before sending** it to Amazon S3
    - Use an AWS KMS-managed customer master key
    - Use a client-side master key
    - Disadvantage: Less matching the AWS ecosystem. You need to manage keys.

Client side master key

- Your client-side master keys and your unencrypted data are never sent to AWS
- manage your own encryption keys
- If you lose them, you won't be able to decrypt your data.
- **When uploading an object**
  - You provide a client-side master key to the Amazon S3 encryption client
  - for each object,encryption client locally generates a one-time-use symmetric key
  - The client uploads the encrypted data key and its material description as part of the object metadata
  - The material description helps the client later determine which client-side master key to use for decryption
  - The client then uploads the encrypted data to Amazon S3 and also saves the encrypted data key as object metadata
- **When downloading an object**
  - The client first downloads the encrypted object from Amazon S3 along with the metadata
  - Using the material description in the metadata, the client first determines which master key to use to decrypt
  - the encrypted data key.

Client Side KMS–Managed Customer Master Key (CMK)

- you provide only an AWS KMS customer master key ID (CMK ID)
- you don't have to worry about providing any encryption keys to the Amazon S3 encryption client (for example, the `AmazonS3EncryptionClient` in the AWS SDK for Java). 2options
    - A plain text version
    - A cipher blob
- unique data encryption key for each object it uploads.

Server Side Encryption (SSE)

- Server-side encryption is about data encryption at rest
  - Amazon S3 encrypts your data at the object level as it writes it to disks
  - decrypts it for you when you access it.
  - As long as you authenticate your request and you have access permissions
  - You can't apply different types of server-side encryption to the same object simultaneously.
- 3 methods
  - **Server-Side Encryption with Customer-Provided Keys (SSE-C)**
    - You manage the encryption keys and Amazon S3 manages the encryption, as it writes to disks, and decryption, when you access your objects
  - **S3-Managed Keys (SSE-S3)**
  - **AWS KMS-Managed Keys (SSE-KMS)**

S3-Managed Keys (SSE-S3)

- Each object is encrypted with a unique key employing strong multi-factor encryption
- it encrypts the key itself with a master key that it regularly rotates
- 256-bit Advanced Encryption Standard (AES-256), to encrypt

AWS KMS-Managed Keys (SSE-KMS)

- Similar to SSE-S3
- There are separate permissions for the use of an envelope key (that is, a key that protects your data's encryption key)
- provides you with an audit trail of when your key was used and by whom
- you have the option to create and manage encryption keys yourself, or use a default key that is unique to you, the service you're using, and the region you're working in.

Additional Safeguards

1. VPN (site to site)
2. Identity
3. IP ACL
4. Write Only permissions.

Security Diagram