Data Security and Encryption

By: Ahsan Farooqui

Roadmap of Data Security & Encryption

- Data security Control
- Cloud data Storage
- Data Migration into Cloud
- Securing Cloud Data Transfer
- Securing Data in the Cloud
- Storage-at rest
- Tokenization and Key Management
- Data Security Architecture
- Monitoring,Auditing and Alerting
- ERM
- Data Masking and Test Data Generation
- Enforcing Security Life Security
- AWS Data Protection

Data security Control

Data security controls tend to fall into three buckets. We cover all of these in this section:

- Controlling what data goes into the cloud (and where).
- Protecting and managing the data in the cloud. The key controls and processes are:
  - Access controls
  - Encryption
  - Architecture Monitoring/alerting (of usage, configuration, lifecycle state, etc.)
  - Additional controls, including those related to the specific product/service/platform of your cloud provider,
    - data loss prevention, and enterprise rights management.
- Enforcing information lifecycle management security.
  - Managing data location/residency.
  - Ensuring compliance, including audit artifacts (logs, configurations).
  - Backups and business continuity,

Cloud Data Storage

Since cloud storage is virtualized it tends to support different data storage types including

- Object storage: Object storage is similar to a file system. “Objects” are typically files, which are then stored using a cloud-platform specific mechanism. Most access is through APIs, not standard file sharing protocols, although cloud providers may also offer front-end interfaces to support those protocols.
- Volume storage: This is essentially a virtual hard drive for instances/virtual machines.
- Database: Cloud platforms and providers may support a variety of different kinds of databases, including existing commercial and open source options, as well as their own proprietary systems.Proprietary databases typically use their own APIs. Commercial or open source databases by the provider and typically use existing standards for connections. These can be relational or non-relational—the latter includes NoSQL and other key/value storage systems, or file system-based databases (e.g. HDFS).
- Application/platform: Examples of these would be a content delivery network (CDN), files stored in SaaS, caching, and other novel options.

Managing Data Migration into Cloud

- Before securing the data in the cloud, most organizations want some means of managing what data is stored in private and public cloud providers. This is often essential for compliance as much or more than for security.
- To start, define your policies for which data types are allowed and where they are allowed, then tie these to your baseline security requirements. For example, “Personally Identifiable Information (PII) is allowed on x services assuming it meets y encryption and access control requirements.”
- Then identify your key data repositories. Monitor them for large migrations/activity using tools such as Database Activity Monitoring and File Activity Monitoring. This is essentially building an “early warning system” for large data transfers, but it’s also an important data security control to detect all sorts of major breaches and misuse scenarios.

Tools For Managing Data Migration into Cloud

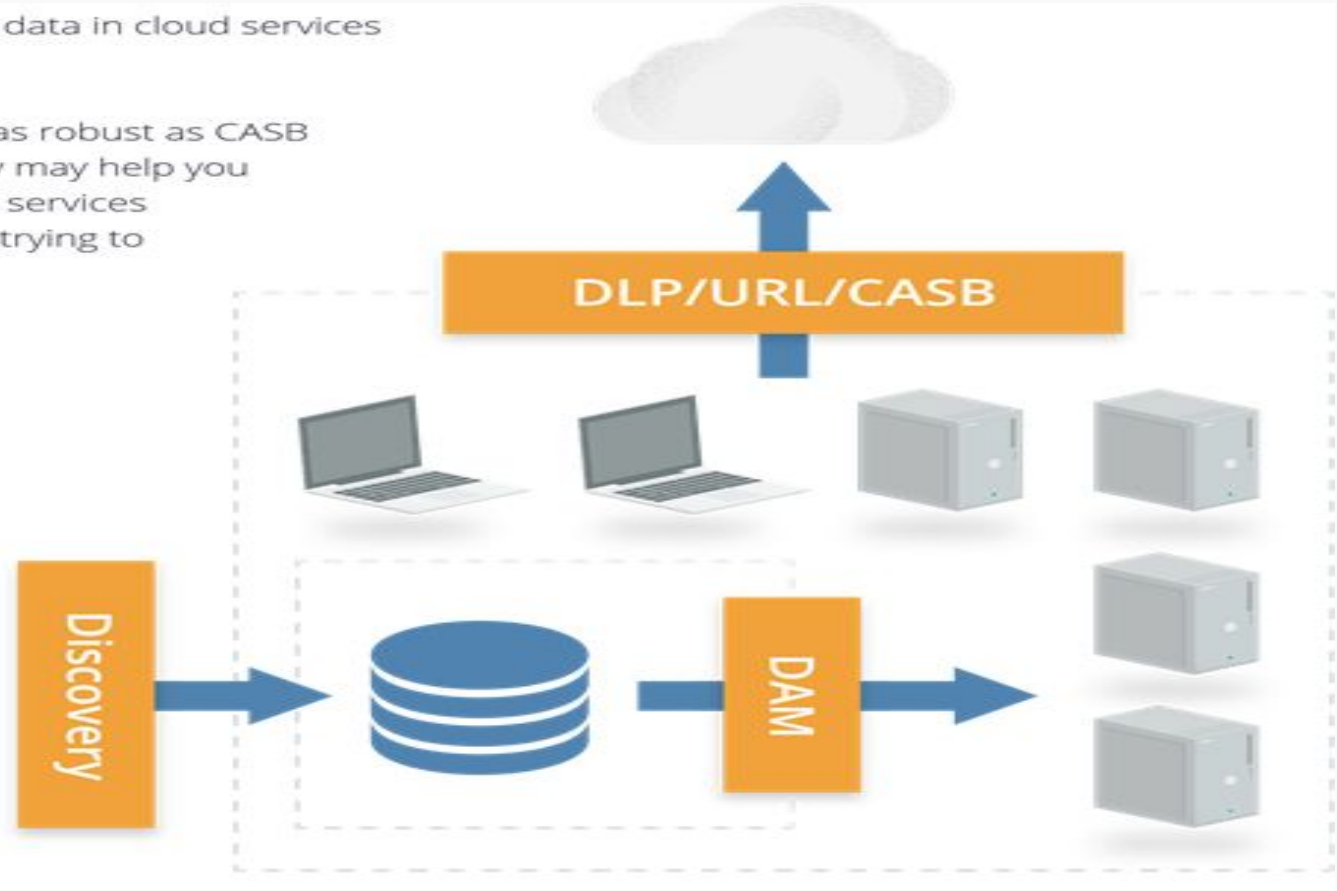
- CASB: Cloud Access and Security Brokers (also known as Cloud Security Gateways) discover internal use of cloud services using various mechanisms such as network monitoring, integrating with an existing network gateway or monitoring tool, or even by monitoring DNS queries. After discovering which services your users are connecting to, most of these products then offer monitoring of activity on approved services through API connections (when available) or inline interception (man in the middle monitoring). Many support DLP and other security alerting and even offer controls to better manage use of sensitive data in cloud services (SaaS/PaaS/and IaaS).
- URL filtering: While not as robust as CASB a URL filter/web gateway may help you understand which cloud services your users are using (or trying to use).
- DLP: If you monitor web traffic (and look inside SSL connections) a Data Loss Prevention (DLP) tool may also help detect data migrations to cloud services. However, some cloud SDKs and APIs may encrypt portions of data and traffic that DLP tools can't unravel, and thus they won't be able to understand the payload.

Tools For Managing Data Migration into Cloud

manage use of sensitive data in cloud services (SaaS/PaaS/and IaaS).

URL filtering: While not as robust as CASB a URL filter/web gateway may help you understand which cloud services your users are using (or trying to use).

DLP: If you monitor web traffic (and look inside SSL connections) a Data Loss Prevention (DLP) tool may also help detect data migrations to cloud services. However, some cloud SDKs and APIs may encrypt portions of data and traffic that DLP tools can't unravel, and thus they won't be able to understand the payload.



Securing Cloud Data transfer

- Ensure that you are protecting your data as it moves to the cloud. This necessitates understanding your provider's data migration mechanisms,
  - as leveraging provider mechanisms is often more secure and cost effective than "manual" data transfer methods such as Secure File Transfer Protocol (SFTP).
  - For example, sending data to a provider's object storage over an API is likely much more reliable and secure than setting up your own SFTP server on a virtual machine in the same provider.
- There are a few options for in-transit encryption depending on what the cloud platform supports.
  - Client-side encryption-
    - One way is to encrypt before sending to the cloud
  - Network encryption (TLS/ SFTP/etc.)
    - Most cloud provider APIs use Transport Layer Security (TLS) by default;
    - if not, pick a different provider, since this is an essential security capability.
  - Proxy-based encryption
    - where you place an encryption proxy in a trusted area between the cloud user and the cloud provider and the proxy manages the encryption before transferring the data to the provider.
- In some instances you may have to accept public or untrusted data.
  - If you allow partners or the public to send you data, ensure you have security mechanisms in place to sanitize it before processing or mixing it with your existing data. Always isolate and scan this data before integrating it.

Securing Data in the Cloud

- Securing data in the cloud by
  - Cloud Data Access Control
    - Management Plane
    - Public and Internal Sharing Control
    - Application Level Control
- Storage(At-Rest)
  - Encryption
  - Tokenization
- Key Management (Including Customer-Managed Keys)
  - HSM/appliance
  - Virtual appliance/software
  - Cloud provider service
  - Hybrid
- Customer-Managed Keys
- Data Security Architectures
- Monitoring, Auditing, and Alerting
- Additional Data Security Controls
  - Cloud Platform/Provider-Specific Controls
  - Data Loss Prevention
- Enterprise Rights Management
- Data Masking and Test Data Generation

Cloud Data Access Control

- Cloud Data Access Control
  - Access controls should be implemented with a minimum of three layers:
    - Management Plane
      - These are your controls for managing access of users that directly access the cloud platform's management plane. For example, logging in to the web console of an IaaS service will allow that user to access data in object storage. Fortunately, most cloud platforms and providers start with default deny access control policies.
    - Public and Internal Sharing Control
      - If data is shared externally to the public or partners that don't have direct access to the cloud platform, there will be a second layer of controls for this access.
    - Application Level Control
      - As you build your own applications on the cloud platform you will design and implement your own controls to manage access.
  - Options for access controls will vary based on cloud service model and provider-specific features.Create an entitlement matrix based on platform-specific capabilities.

Cloud Data Access Control

- **Entitlement Matrix Document**
  - **An entitlement matrix documents which users, groups, and roles should access which resources and functions.**

Entitlement	Super-Admin	Service-Admin	Storage-Admin	Dev	Security-Audit	Security-Admin
Volume Describe	X	X		X	X	X
Object Describe	X		X	X	X	X
Volume Modify	X	X		X		X
Read Logs	X				X	X

Frequently validate that your controls meet your requirements, paying particular attention to any public shares. Consider setting up alerts for all new public shares or for changes in permissions that allow public access.

Cloud Data Access Control

- Fine-Grained Access Controls and Entitlement Mappings
  - The depth of potential entitlements will vary greatly from technology to technology. Some databases may support row-level security, others little more than broad access. Some will allow you to tie entitlements to identity and enforcement mechanisms built into the cloud platform, while others rely completely on the storage platform itself merely running in virtual machines. It's important to understand your options, map them out, and build your matrix. This applies to more than just file access, of course; it also applies to databases and all your cloud data stores.

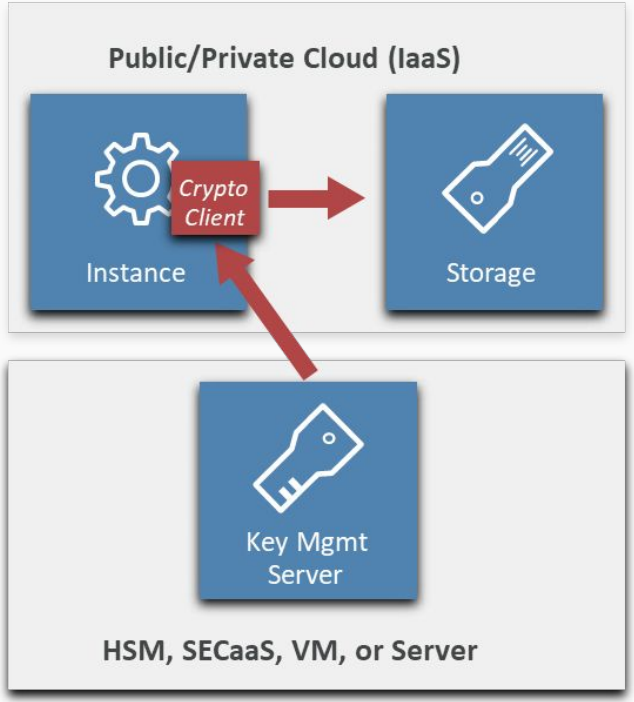
Storage (At-Rest)

- Encryption
  - Encryption protects data by applying a mathematical algorithm that “scrambles” the data, which then can only be recovered by running it through an unscrambling (decryption) process with a corresponding key. The result is a blob of “ciphertext”.
  - Encryption Components
    - There are three components of an encryption system:
      - Data
        - The data is, of course, the information that you’re encrypting.
      - The encryption engine
        - The engine is what performs the mathematical process of encryption?
      - Key management
        - Finally, the key manager handles the keys for the encryption.
    - The overall design of the system focuses on where to put each of these components.
    - When designing an encryption system, you should start with a threat model. For example, do you trust a cloud provider to manage your keys? How could the keys be exposed? Where should you locate the encryption engine to manage the threats you are concerned with?

Storage (At-Rest)

- IAAS Encryption
  - IaaS Encryption IaaS volumes can be encrypted using different methods, depending on your data.
- Volume storage encryption
  - Instance-managed encryption: The encryption engine runs within the instance, and the key is stored in the volume but protected by a passphrase or keypair.
  - Externally managed encryption:The encryption engine runs in the instance, but the keys are managed externally and issued to the instance on request.
- Object and file storage
  - Client-side encryption: When object storage is used as the back-end for an application (including mobile applications), encrypt the data using an encryption engine embedded in the application or client.
  - Server-side encryption: Data is encrypted on the server (cloud) side after being transferred in. The cloud provider has access to the key and runs the encryption engine.
  - Proxy encryption: In this model, you connect the volume to a special instance or appliance/ software, and then connect your instance to the encryption instance. The proxy handles all crypto operations and may keep keys either onboard or externally.

Externally managed volume encryption



Storage (At-Rest)

- **PaaS Encryption**
  - PaaS encryption varies tremendously due to all the different PaaS platforms.
    - Application layer encryption: Data is encrypted in the PaaS application or the client accessing the platform.
    - Database encryption: Data is encrypted in the database using encryption that's built in and is supported by a database platform like Transparent Database Encryption (TDE) or at the field level.
    - Other: These are provider-managed layers in the application, such as the messaging queue. There are also IaaS options when that is used for underlying storage.
- **SaaS Encryption**
  - SaaS providers may use any of the options previously discussed. It is recommended to use per-customer keys when possible, in order to better enforce multitenancy isolation. The following options are for SaaS consumers:
    - Provider-managed encryption: Data is encrypted in the SaaS application and generally managed by the provider.
    - Proxy encryption: Data passes through an encryption proxy before being sent to the SaaS application.

Tokenization

- Tokenization, on the other hand, takes the data and replaces it with a random value. It then stores the original and the randomized version in a secure database for later recovery.

- Tokenization is often used when the format of the data is important (e.g. replacing credit card numbers in an existing system that requires the same format text string).
- Format Preserving
  - Encryption encrypts data with a key but also keeps the same structural format as tokenization.
  - but it may not be as cryptographically secure due to the compromises.

Key Management

- The main considerations for key management are
  - performance, accessibility, latency, and security
  - Can you get the right key to the right place at the right time while also meeting your security and compliance requirements?
- There are four potential options for handling key management:
  - HSM/appliance: Use a traditional hardware security module (HSM) or appliance-based key manager, which will typically need to be on-premises, and deliver the keys to the cloud over a dedicated connection.
  - Virtual appliance/software: Deploy a virtual appliance or software-based key manager in the cloud.
  - Cloud provider service: This is a key management service offered by the cloud provider. Before selecting this option, make sure you understand the security model and SLAs to understand if your key could be exposed.
  - Hybrid: You can also use a combination, such as using a HSM as the root of trust for keys but then delivering application-specific keys to a virtual appliance that's located in the cloud and only manages keys for its particular context.
-

Key Management-Customer Managed Key

- A customer-managed key allows
  - a cloud customer to manage their own encryption key
  - while the provider manages the encryption engine.
- For example, using your own key to encrypt SaaS data within the SaaS platform. Many providers encrypt data by default, using keys completely in their control. Some may allow you to substitute your own key, which integrates with their encryption system. Make sure your vendor's practices align with your requirements.
- Some providers may require you to use a
  - service within the provider to manage the key.
  - Thus, although the key is customer-managed, it is still potentially available to provider. This doesn't necessarily mean it is insecure: Since the key management and data storage systems can be separated, it would require collusion on the part of multiple employees at the provider to potentially compromise data.
  - However, keys and data could still be exposed by a government request, depending on local laws. You may be able to store the keys externally from the provider and only pass them over on a per-request basis.

Data Security Architecture

- Application architecture impacts data security. The features your cloud provider offers can reduce the attack surface, but make sure to demand strong metastructure security.
- For example, gap networks by using cloud storage or a queue service that communicates on the provider's network, not within your virtual network. That forces attackers to either fundamentally compromise the cloud provider or limit themselves to application-level attacks, since network attack paths are closed.
- An example would be using object storage for data transfers and batch processing, rather than SFTP-ing, to static instances. Another is message queue gapping—run application components on different virtual networks that are only bridged by passing data through the cloud provider's message queue service. This eliminates network attacks from one portion of the application to the other.

Monitoring ,Auditing and Alerting

- These should tie into overall cloud monitoring.
  - Identify (and alert about) any public access or entitlement changes on sensitive data.
  - Use tagging to support alerting, when it's available.
  - You'll need to monitor both API and storage access, since data may be exposed through either—in other words, accessing data in object storage via an API call or via a public sharing URL.
  - Activity monitoring, including Database Activity Monitoring, may be an option.
  - Make sure to store your logs in a secure location, like a dedicated logging account.

Additional Data Security Controls

- Cloud Platform/Provider-Specific Controls
  - A cloud platform or provider may have data security controls that are not covered elsewhere in this domain. Although typically they will be some form of access control and encryption, this Guidance can't cover all possible options.
- Data Loss Prevention
  - Data Loss Prevention (DLP) is typically a way to monitor and protect data that your employees access via monitoring local systems, web, email, and other traffic. It is not typically used within data centers, and thus is more applicable to SaaS than PaaS or IaaS, where it is typically not deployed.
    - CASB: Some CASBs include basic DLP features for the sanctioned services they protect. Forexample, you could set a policy that a credit card number is never stored in a particular cloud service. The effectiveness depends greatly on the particular tool, the cloud service, and how the CASB is integrated for monitoring. Some CASB tools can also route traffic to dedicated DLP platforms for more robust analysis than is typically available when the CASB offers DLP as a feature.
    - Cloud provider feature: The cloud provider themselves may offer DLP capabilities, such as a cloud file storage and collaboration platform that scans uploaded files for content and applies corresponding security policies.

Enterprise Rights Management

- As with DLP, this is typically an employee security control that isn't always as applicable in cloud. Since all Digital Rights Management (DRM)/Enterprise Rights Management (ERM) is based on encryption, existing tools may break cloud capabilities, especially in SaaS.
  - Full DRM: This is traditional, full digital rights management using an existing tool. For example,applying rights to a file before storing it in the cloud service. As mentioned, it may break cloud provider features, such as browser preview or collaboration, unless there is some sort of integration (which is rare at the time of this writing).
  - Provider-based control: The cloud platform may be able to enforce controls very similar to full DRM by using native capabilities. For example, user/device/view versus edit: a policy that only allows certain users to view a file in a web browser, while other users can download and/or edit the content. Some platforms can even tie these policies to specific devices, not just on a user level.

Data Masking and Test Data Generation

- These are techniques to protect data used in development and test environments, or to limit real-time access to data in applications.
  - Test data generation: This is the creation of a database with non-sensitive test data based on a “real” database. It can use scrambling and other randomization techniques to create a data set that resembles the source in size and structure but lacks sensitive data.
  - Dynamic masking: Dynamic masking rewrites data on the fly, typically using a proxy mechanism,to mask all or part of data delivered to a user. It is usually used to protect some sensitive data in applications, for example masking out all but the last digits of a credit card number when presenting it to a user.

Enforcing Life Cycle Security

- Managing data location/residency: At certain times, you'll need to disable unneeded locations.Use encryption to enforce access at the container or object level. Then, even if the data moves to an unapproved location, the data is still protected unless the key moves with it.
- Ensuring compliance: You don't merely need to implement controls to maintain compliance, you need to document and test those controls. These are "artifacts of compliance;" this includes any audit artifacts you will have.
- Backups and business continuity:

AWS Data Protections

How do you classify your data?
Classification provides a way to categorize data, based on

- ❑ **criticality and**
- ❑ **sensitivity**

in order to help you determine appropriate protection and retention controls.

Best practices

- ❑ **Identify the data within your workload ?**
- ❑ **Define data protection controls**
- ❑ **Automate identification and classification**
- ❑ **Define data lifecycle management**

AWS Data Protections

Data protection Questions

- ❑ How do you classify your data?
- ❑ How do you protect your data at rest?
- ❑ How do you protect your data in transit?

How do you classify your data?

It's critical to understand the type and classification of data your workload is processing, the associated business processes:

- where the data is stored, and
- who is the data owner.
- You should also have an understanding of the applicable legal and compliance requirements of your workload, and what data controls need to be enforced. Identifying data is the first step in the data classification journey.
- Data classification allows workload owners
  - to identify locations that store sensitive data and
  - Determine how that data should be accessed and shared.
- Data classification aims to answer the following questions:
  - What type of data do you have?
  - Who can perform create, read, update, and delete (CRUD) operations?
  - What business impact might occur if the data is disclosed unintentionally, altered, or deleted?

What,Who and What type of data queries?

What type of data do you have?
This could be data such as:

- **Intellectual property (IP) such as trade secrets, patents, or contract agreements.**
- **Protected health information (PHI) such as medical records that contain medical history information connected to an individual.**
- **Personally identifiable information (PII), such as name, address, date of birth, and national ID or registration number.**
- **Credit card data, such as the Primary Account Number (PAN), cardholder name, expiration date, and service code number.**

Who can perform create, read, update, and delete (CRUD) operations?

- **Account for potential escalation of privileges by understanding who can manage permissions to the data.**

What business impact might occur if the data is disclosed unintentionally, altered, or deleted?

- **Understand the risk consequence if data is modified, deleted, or inadvertently disclosed.**

Queries Answer?

By knowing the answers to these questions, you can take the following actions:

- ❑ Decrease sensitive data scope (such as the number of sensitive data locations) and limit access tosen sitive data to only approved users.
- ❑ Gain an understanding of different data types so that you can implement appropriate data protection mechanisms and techniques, such as encryption, data loss prevention, and identity and access management.
- ❑ Optimize costs by delivering the right control objectives for the data.
- ❑ Confidently answer questions from regulators and auditors regarding the types and amount of data, and how data of different sensitivities are isolated from each other.

AWS Products -Data Discovery?

- Use services such as Amazon Macie to automate at scale both the discovery and classification of sensitive data.
  - Within Amazon S3 Buckets
  - Within RDS Database
  - Within Dynamo Database
- Other services, such as Amazon EventBridge and AWS Config, can be used to automate remediation for data security issues such as unencrypted Amazon Simple Storage Service (Amazon S3) buckets and Amazon EC2 EBS volumes or untagged data resources.
- Detecting PII in unstructured data such as customer emails, support tickets, product reviews, and social media, is possible by using Amazon Comprehend, which is a natural language processing (NLP) service that uses machine learning (ML) to find insights and relationships like people, places, sentiments,
- and topics in unstructured text.
- AWS resource tagging. Tagging allows you to assign metadata to your AWS resources that you can use to manage, identify, organize, search for, and filter resources. AWS Organization

Define data protection controls?

- Protect data according to its classification level. For example,
  - data classified as public by using relevant recommendations
  - while protecting sensitive data with additional controls.
  - By using resource tags, separate AWS accounts per sensitivity (and potentially also for each caveat,enclave, or community of interest), IAM policies, AWS Organizations SCPs, AWS Key Management Service (AWS KMS), and AWS CloudHSM, you can define and implement your policies for data classification and protection with encryption.

For example, if you have a project with S3 buckets that contain highly critical data or Amazon Elastic Compute Cloud (Amazon EC2) instances that process confidential data, they can be tagged with a Project=ABC tag. Only your immediate team knows what the project code means, and it provides a way to use attribute-based access control. You can define levels of access to the AWS KMS encryption keys through key policies and grants to ensure that only appropriate services have access to the sensitive content through a secure mechanism. If you are making authorization decisions based on tags you should make sure that the permissions on the tags are defined appropriately using tag policies in AWS Organizations.

Automate identification and classification

- Automating the identification and classification of data can help you implement the correct controls.
- Using automation for this instead of direct access from a person reduces the risk of human error and exposure.
- You should evaluate using a tool, such as Amazon Macie, that uses machine learning to automatically discover, classify, and protect sensitive data in AWS. Amazon Macie recognizes sensitive data, such as personally identifiable information (PII) or intellectual property, and provides you with dashboards and alerts that give visibility into how this data is being accessed or moved.
- Amazon S3 Inventory
  - Consider Amazon Macie: Amazon Macie uses machine learning to automatically discover and classify data stored in Amazon S3.

Define Data Lifecycle Management

- Your defined lifecycle strategy should be based on sensitivity level as well as legal and organization requirements.
- Aspects including the duration for which you retain data, data destruction processes, data access management, data transformation, and data sharing should be considered.
- When choosing a data classification methodology, balance usability versus access.
- You should also accommodate the multiple levels of access and nuances for implementing a secure, but still usable, approach for each level.
- Always use a defense in depth approach and reduce human access to data and mechanisms for transforming,deleting, or copying data. For example, require users to strongly authenticate to an application, and give the application, rather than the users, the requisite access permission to perform action at a distance.
- In addition, ensure that users come from a trusted network path and require access to the decryption keys.
- Use tools, such as dashboards and automated reporting, to give users information from the data rather than giving them direct access to the data.

How do you protect your data at rest?

Protect your data at rest by implementing multiple controls, to reduce the risk of unauthorized access or mishandling.

Best practices

- ❑ **Implement secure key management**
- ❑ **Enforce encryption at rest**
- ❑ **Automate data at rest protection**
- ❑ **Enforce access control**
- ❑ **Use mechanisms to keep people away from data**

Implement secure key management

By defining an encryption approach that includes

- the storage, rotation, and access control of keys,you can help provide protection for your content against unauthorized users and against unnecessary exposure to authorized users.
- AWS Key Management Service (AWS KMS) helps you manage encryption keys and integrates with many AWS services. This service provides durable, secure, and redundant storage for your AWS KMS keys.
- You can define your key aliases as well as key-level policies. The policies help you define key administrators as well as key users.
- Additionally, AWS CloudHSM is a cloud-based

hardware security module (HSM) that allows you to easily generate and use your own encryption keys in the AWS Cloud. It helps you meet corporate, contractual, and regulatory compliance requirements for data security by using FIPS 140-2 Level 3 validated HSMs.

Enforce Encryption at rest

- You should enforce the use of encryption for data at rest. Encryption maintains the confidentiality of sensitive data in the event of unauthorized access or accidental disclosure.
- AWS Key Management Service (AWS KMS) integrates with many AWS services to make it easier to encrypt your data at rest. For example, in Amazon Simple Storage Service (Amazon S3), you can set default encryption on a bucket so that new objects are automatically encrypted.
- When using AWS KMS, consider how tightly the data needs to be restricted. Default and service-controlled AWS KMS keys are managed and used on your behalf by AWS.
- For sensitive data that requires fine-grained access to the underlying encryption key, consider customer managed keys (CMKs). You have full control over CMKs, including rotation and access management through the use of key policies.
- Additionally, Amazon Elastic Compute Cloud (Amazon EC2) and Amazon S3 support the enforcement of encryption by setting default encryption. You can use AWS Config Rules to check automatically that you are using encryption, for example, for Amazon Elastic Block Store (Amazon EBS) volumes, Amazon Relational Database Service (Amazon RDS) instances, and Amazon S3 buckets.

Automate data at rest protection

- Data at rest represents any data that you persist in non-volatile storage for any duration in your workload. This includes
- block storage, object storage, databases, archives, IoT devices, and any other storage medium on which data is persisted.
- Protecting your data at rest reduces the risk of unauthorized access, when encryption and appropriate access controls are implemented.
- Enforce encryption at rest: You should ensure that the only way to store data is by using encryption.
- AWS KMS integrates seamlessly with many AWS services to make it easier for you to encrypt all your data at rest. For example, in Amazon Simple Storage Service (Amazon S3) you can set default encryption on a bucket so that all new objects are automatically encrypted. Additionally, Amazon EC2 and Amazon S3 support the enforcement of encryption by setting default encryption.
- You can use AWS Managed Config Rules to check automatically that you are using encryption, for example, for EBS volumes, Amazon Relational Database Service (Amazon RDS) instances, and Amazon S3 buckets.

Enforce Access Control

- To help protect your data at rest, enforce access control using mechanisms, such as isolation and versioning, and apply the principle of least privilege. Prevent the granting of public access to your data.
- Verify that only authorized users can access data on a need-to-know basis. Protect your data with regular backups and versioning to prevent against intentional or inadvertent modification or deletion of data. Isolate critical data from other data to protect its confidentiality and data integrity.

Multiple controls can help protect your data at rest, including access (using least privilege), isolation, and versioning. Access to your data should be audited using detective mechanisms, such as AWS CloudTrail, and service level logs, such as Amazon Simple Storage Service (Amazon S3) access logs.
Amazon S3 Glacier Vault Lock and Amazon S3 Object Lock provide mandatory access control for objects in Amazon S3—once a vault policy is locked with the compliance option, not even the root user can change it until the lock expires.

Enforce Access Control

Enforce access control: Enforce access control with least privileges, including access to encryption keys.

- Separate data based on different classification levels: Use different AWS accounts for data classification levels, and manage those accounts using AWS Organizations.
- Review AWS Key Management Service (AWS KMS) policies: Review the level of access granted in AWS KMS policies.
- Review Amazon S3 bucket and object permissions: Regularly review the level of access granted in S3 bucket policies. Best practice is to avoid using publicly readable or writeable buckets.
- Consider using AWS Config to detect buckets that are publicly available, and Amazon CloudFront to serve content from Amazon S3. Verify that buckets that should not allow public access are properly configured to prevent public access. By default, all S3 buckets are private, and can only be accessed by users that have been explicitly granted access.
- Use AWS IAM Access Analyzer: IAM Access Analyzer analyzes Amazon S3 buckets and generates a finding when an S3 policy grants access to an external entity.
- Use Amazon S3 versioning and object lock when appropriate.
- Use Amazon S3 Inventory: Amazon S3 Inventory can be used to audit and report on the replication and encryption status of your S3 objects.
- Review Amazon EBS and AMI sharing permissions: Sharing permissions can allow images and volumes to be shared with AWS accounts that are external to your workload.
- Review AWS Resource Access Manager Shares periodically to determine whether resources should continue to be shared. Resource Access Manager allows you to share resources, such as AWS Network Firewall policies, Amazon Route 53 resolver rules, and subnets, within your Amazon VPCs. Audit shared resources regularly and stop sharing resources which no longer need to be shared.

Use Mechanism to keep peoples away

- Keep all users away from directly accessing sensitive data and systems under normal operational Circumstances.
- For example, use a change management workflow to manage Amazon Elastic Compute Cloud (Amazon EC2) instances using tools instead of allowing direct access or a bastion host.
- This can be achieved using AWS Systems Manager Automation, which uses automation documents that contain steps you use to perform tasks.
-

How do you protect your data in transit?

- Protect your data in transit by implementing multiple controls to reduce the risk of unauthorized access

or loss.

Best practices

- Implement secure key and certificate management
- Enforce encryption in transit
- Automate detection of unintended data access
- Authenticate network communications

How do you protect your data in transit?

Implement secure key and certificate management

- Store encryption keys and certificates securely and rotate them at appropriate time intervals with strict access control. The best way to accomplish this is to use a managed service, such as
- AWS Certificate Manager (ACM). It lets you easily provision, manage, and deploy public and private Transport Layer
- Security (TLS) certificates for use with AWS services and your internal connected resources.
- ACM integrates with AWS resources, such as Elastic Load Balancers (ELBs), AWS distributions, and APIs on API Gateway, also handling automatic certificate
- renewals. If you use ACM to deploy a private root CA, both certificates and private keys can be provided by it for use in Amazon Elastic Compute Cloud (Amazon EC2) instances, containers, and so on.
-

Enforce Encryption in transit

- Enforce your defined encryption requirements based on your organization's policies, regulatory obligations and standards to help meet organizational, legal, and compliance requirements.
- Only use protocols with encryption when transmitting sensitive data outside of your virtual private cloud (VPC).
- Encryption helps maintain data confidentiality even when the data transits untrusted networks.
- AWS services provide HTTPS endpoints using TLS for communication, providing encryption in transit
- when communicating with the AWS APIs.
- .you can use VPN connectivity into your VPC from an external network or AWS Direct Connect to facilitate encryption of traffic.
- Insecure protocols like HTTP can be audited and blocked in aVPC through the use of security groups.
- HTTP requests can also be automatically redirected to HTTPS in Amazon CloudFront or on an Application Load Balancer

How do you protect your data in transit?

Automate Detection of Unintended data access

- Use tools such as Amazon GuardDuty to automatically detect suspicious activity or attempts to move data outside of defined boundaries.
- For example, GuardDuty can detect Amazon Simple Storage Service (Amazon S3) read activity that is unusual with the Exfiltration:S3/AnomalousBehavior finding.
- In addition to GuardDuty, Amazon VPC Flow Logs, which capture network traffic information, can be used with Amazon EventBridge to detect connections, both successful and denied.
- Amazon S3 Access Analyzer can help assess what data is accessible to who in your Amazon S3 buckets.

Authenticate Network communication

- Verify the identity of communications by using protocols that support authentication, such as Transport Layer Security (TLS) or IPsec.
- Using network protocols that support authentication, allows for trust to be established between the parties. This adds to the encryption used in the protocol to reduce the risk of communications being altered or intercepted. Common protocols that implement authentication include Transport Layer
- Security (TLS), which is used in many AWS services, and IPsec, which is used in AWS Virtual Private
- Network (AWS VPN).
-