Module 4: AWS Cloud Security

Tuesday, September 9, 2025 8:51 PM

A. Learning Outcomes (LOs) and Topics:

- Topics:
  + AWS shared responsibility model
  + AWS Identity and Access Management (IAM)
  + Securing a new AWS account
  + Securing accounts
  + Securing data on AWS
  + Working to ensure compliance
- Learning Outcomes (LOs):
  + Recognize the shared repsonsibility model
  + Identify the responsibility of the customer and AWS
  + Recognize the IAM users, groups, and roles.
  + Describe different types of security credentials in IAM
  + Identify the steps to securing a new AWS account
  + Explore IAM groups and users
  + Recognize how to secure AWS data
  + Recognize AWS compliance programs.

B. AWS Shared Responsibility Models

- AWS Responsibilities:
  + Physical security of data centers: Controlled, need-based acess
  + Hardware and software infrastructure: Storage decommissioning, host opearting system (OS) access logging, and auditing.
  + Network infrastructure: Intrusion detection
  + Virtualization infrastructure: Instance isolation
- Customer responsibilities:
  + Amazon Elastic Compute Cloud (Amazon EC2) instance **operating system**: Including patching, maintenanace
  + **Applications**: Passwords, role-based access, etc.
  + **Security group** configuration
  + OS or host-based **firewalls**: Including intrusion detecttion or prevention systems.
  + **Network** configurations
  + Account management: Login and permission settings for each user
- Infrastructure as a Service (IaaS):
  + Customer has more flexibility over configuring networking and storage settings.
  + Customer is responsible for managing more aspects of the security
  + Customer configures the access controls.
- Platform as a Service (PaaS):
  + Customer does not need to manage the underlying infrastructure.
  + AWS handles the operating system, database patching, firewall configuration, and disaster recovery
  + Customer can focus on managing code or data
- Software as a Service (SaaS):
  + Software is centrally hosted
  + Licensed on a subscription model or pay-as-you-go basis
  + Services are typically accessed via web browser, mobile app, or application programming interface (API)
  + Customers do not need to manage the infrastucture that supports the service

C. AWS Identity and Access Management (IAM)

- Use IAM to manage access to AWS resources -
  + A resource is an entity in an AWS account that you work with
  + Example resources; An Amazon EC2 instance or an Amazon S3 bucket
- Example - Control who can terminate Amazon EC2 instances
- Define fine-grained access rights-
  + Who can access the resource
  + Which resources can be accessed and what can the user do to the resource
  + How resources can be acccessed
- IAM is a no-cost AWS account feature

- IAM: Essential components
  + IAM **User**: A **person** or **application** that can authenticate with an AWS account
  + IAM **Group**: A **collection of IAM** users that are granted identical authorization
  + IAM **Policy**: The document that defines **which resources can be accessed** and the **level of access** to each resource.
  + IAM **Role:** Useful mechanism to grant a set of permissions for making AWS service requests.
- Authenticate as an **IAM user** to gain access: When you define an IAM user, you select what **types of access** the user is permitted to use
  + ***Programmatic Access***:
    * Authenticate using: **Access key ID and Secret access key**
    * Provides AWS CLI and AWS SDK access
  + AWS Management Console access:
    * Authenticate using: 12-digit Account ID/alias, IAM user name, and IAM password
    * If enabled, multi-factor authentication (MFA) prompts for an authentication code.
- IAM MFA:
  + MFA provides increased security
  + In addition to user name and password, MFA requires a unique authentication code to access AWS services.
- IAM Authorization:
  + Assign permissions by creating an IAM policy
  + Permissions determine which **resources and operations** are allowed:
    * All permissions are implicity denied by default
    * If something is explicity denied, it is never allowed

Best practice: Follow the **principle of least privilege**

Note: The scope of IAM service confgurations is global. Settings apply across all AWS Regions

- IAM Policies:
  + An IAM policy is a document that define permissions: Enables fire-grained access control
  + Two types of policies: identity-based and resource-based
  + Identity-based policies:
    * Attach a policy to any IAM entity: An IAM user, an IAM group, or an IAM role
    * Pocicies specify: Actions that may/may not be oerformed by the entity
    * A single policy can be attached to multiple entities.
    * A single entity can have multiple plicies attached to it.
  + Resource-based policies: Attached to a resource (such as an S3 bucket).
- Resource-based policies:
  + Identity-based policies are attached to a user, group, or role.
  + Resource-based policies are attached to a resource (not to a user, group or role)
  + Characteristics of resource-based:
    * Specifies who has access to the resource and what actions they can perform on it
    * The policies are inline only, not managed
  + Resource-based policies are supported only by some AWS services
- IAM groups:
  + An IAM group is a collection of IAM users
  + A group is used to grant same permissions to multiple users: Permissions granted by attaching IAM policy or policies to the group.
  + A user can belong to multiple groups
  + There is no default group
  + Groups cannot be nested
- IAM roles:
  + An IAM role is an IAM identity with specific permissions
  + Similar to an IAM user: Attach permissions policies to it
  + Different from an IAM user
    * Not uniquely associated with one person
    * Intended to be assumable by a person, application, or service
  + Role provides temporary security credentials
  + Examples of how IAM roles are used to delegate access:
    * Used by an IAM user in the same AWS account as the role
    * Used by an AWS service - such as Amazon EC2 - in the same account as the role
    * Used by an IAM user in a different AWS aacount than the role

D. Securing a new AWS account

- Best practice: Do not use the AWS account root user except when necessary
  + Access to the account root user requires logging in with the email address (and password) that you used to create the account.
- Example actions that can only be done wirh the account root user:
  + Update the account root user password

+ Change the AWS Support plan
+ Restore an IAM user's permissions
+ Change account settings (for example, contact information, allowed Regions)

Step 1: Stop using the account root user as soon as possible (The account root user has unrestricted access to al your resources)

1. While you are logged in as the account root user, create an IAM user, for yourself. Save the access keys if needed
2. Create an IAM group, give it full administrator permissions, and add the IAM user to the group
3. Disable and remove your account root user access keys, if they exist
4. Enable a password policy for users.
5. Sign in with your new IAM user credentials
6. Store your account root user credentials in a secure place.

Step 2: Enable multi-factor authentication (MFA):

- Require MFA for your account root user and for all IAM users
- You can also use MFA to control access to AWS service APIs
- Options for retrieving the MFA token -
  + Virtual MFA-compliant applications:
    * Google Authentiicator
    * Authy Authenticator (Windows phone app)
  + U2F security key devices: For instance, YubiKey
  + Hardware MFA options: Key fob or display card offered by Gemalto

Step 3: Use AWS CloudTrail

- CloudTrail tracks user activity on the account: Logs all API request to resources in all supported services by the account
- Basic AWS CloudTrail event history is enabled by default and is free: It contains all management event data on latest 90 days of account activity
- To access CloudTrail:
  1. Log in to the AWS Management Console and choose the CloudTrail service
  2. Click Event History to view, filter, and search the last 90 days of events.
- To enable logs beyond 90 days and enable specified event alerting, create a trail.
  1. From the CloudTrail Console trails page, click Create trail
  2. Give it a name, apply to Regions, and generate new Amazon S3 bucket for log storage
  3. Configure access restrictions on the S3 buckte.

Step 4: Enable a biling report, such as the AWS Cost and Usage Report

- Billing report provides information about you use of AWS resources and estimated costs for that use.
- AWS delivers the reports to an Amazon S3 bucket that you specify: Report is updated at least once per day;
- The AWS Cost and Usage Report tracks yout AWS usage and provides estimated charges associated with your AWS account, either by the hour or by the day.

E. Securing Accounts

- AWS Organizations enables you to consolidate multiple AWS accounts so that you centrally manage them
- Security features of AWS Organizations:
  + Group AWS accounts into organizationa units (OUs) and attach different access policies to each OU
  + Integration and support for IAM: Permissions to a user are the intersection of what is allowed by AWS Organizations and what is granted by IAM in that account
  + Use service control policies to establish control over the AWS servies and API actions that each AWS account can access
- AWS Organizations: Service control policies
  + Service control policies (SCPs) offer centralized control over accounts. Limit permissions that are available in an account that is part of an organization.
  + Ensures that accounts comply with access control guidelines.
  + SCPs are similar to IAM permissions policies:
    * They use similar syntax
    * However, an SCO never grants permissions
    * Instead, SCPs specify the maximum permissions for an organization.
- AWS Key Management Service (AWS KMS) features:
  + Enables you to create and manage encryption keys
  + Enables you to control the use of encryption across AWS services and in you applications.
  + Integrates with AWS CloudTrail to log all key usage.
  + Uses hardware security modules (HSMs) that are validated by Federal Information Processing Standards (FIPS) 140-2 to protect keys.
- Amazon Cognito features:

+ Adds user sign-up, sign-in, and access control to your web and mobile applications.
+ Scales to millions of users.
+ Supports sign-in with social identity providers such as Facebook, Google, and enterprise identity providers ncluding Microsoft Active Directory via Security Assertion Markup Language (SAML 2.0)

- AWS Shield features:
+ is a managed distributed denial of service (DDoS) protection service
+ Safeguards application running on AWS
+ Provides always-on detection and automatic inline mitigations
+ AWS Shield Standard enabled for at no additional cost. AWS Shield Advanced is an optional paid service
+ Use to minimize application downtime and latency.

F. Securing Data

- Encryption encodes data with a secret key, which makes it unreadable
+ Only those who have the secret key can decode the data.
+ AWS KMS can manage your secret keys

- AWS supports encryption of data at rest.
+ Data at rest = Data stored physically (on disk or on tape)
+ Enrypting data stored in any service that is supported by AWS KMS including Amazon S3, Amazon EBS, Amazon Elastic File System (Amazon EFS), and Amazon RDS managed databases

- Encryption of data in transit
+ Encryption of data in transit (data moving across a network)
* Transport Layer Security (TLS) - formly SSL - is an open standard protocol
* AWS Certificate Manager provides a way to manag, deploy, and renew TLS or SSL certificates.
+ Secure HTTP (HTTPS) creates a secure tunnel: Uses TLS or SSL for the bidirectional exchange of data
+ AWS services support data in transit encryption

- Securing Amazon S3 buckets and objects:
+ Newly created S3 buckets and objects are private and protected by default.
+ When use cases require sharing data objects on Amazon S3:
* It is essential to manage and control the data access
* Follow the permissions that follow the principle of least priviledge and consider using Amazon S3 encryption
+ Tools and options for controlling access to S3 data include:
* Amazon S3 Block Public Access: Simple to use
* IAM policies: A good option when the user can authenticate using IAM
* Bucket policies
* Access control lists (ACLs): A legacy access control mechanism
* AWS Trusted Advisor: bucket permission check - A free feature

G. Workign to Ensure Compliance

- Customers are subject to many different security and cmpliance regulations and requirements
- AWS engages with certifying bodies and independent auditors to provide customers with detailed information about the processes, policies, and controls that are established and operated by AWS.
- Compliance programs can be broadly categorized:
+ Certifications and attestations: Assessed by a third-party, independent auditors
+ Laws, regulations, and privacy: AWS provides security features and legal agreements to support compliance
+ Alignments and frameworks: Industry or function specific security or compliance requirements.

- AWS Config:
+ Assess, audit, and evaluate the configurations of AWS resources.
+ Use for continuous monitoring of configurations
+ Automatically evaluate recorded configurations versus desired configurations.
+ Review configuration changes.
+ View detailed configuration histories
+ Simplify compliance auditing and security analysis

- AWS Artifact: is a resource for compliance-related information
+ Provide access to security and compliance reports, and select online agreements.
+ Can access example downloads:
* AWS ISO certificates
* Payment Card Industry (PCI) and Service Organization Control (SOC) reports.
+ Access AWS Artifact directly from the AWS Managemenr Console: Under Security, Identify & Compliance, click Artifact