Qualys CloudView

Policy Document

AWS Best Practices Policy

Qualys, Inc.
919 E Hillsdale Blvd
4th Floor
Foster City, CA 94404
1 (650) 801 6100

Table of Contents

Control ID - 3: Ensure access keys unused for 90 days or greater are disabled

Criticality: HIGH

Specification

AWS IAM users can access AWS resources using different types of credentials, such as passwords or access keys. It is recommended that all users access keys that have been unused in 90 or greater days be removed or deactivated.

Rationale

Disabling or removing unnecessary credentials will reduce the window of opportunity for credentials associated with a compromised or abandoned account to be used.

Evaluation

Check IAM Users having active access keys and have not used for 90 days or more.

Changes in account credentials may take up to 4 hours to get reflected in the AWS IAM evaluations. The time taken depends on when the last credential report was fetched by the Cloud View service and the time when changes were made in AWS IAM

Remediation

Perform the following to deactivate or delete credentials (AccessKeys):

1. Sign in to the AWS Management Console and open the IAM console at https://console.aws.amazon.com/iam/.
2. In the navigation pane, choose **Users**.
3. Choose the name of the user whose access key(s) have not been used in 90 Days and then choose the **Security Credentials** tab.
4. If needed, expand the **Access Keys** section and do any of the following:
    - To disable an active access key, choose **Make Inactive**.
    - To delete an access key, click **X** and then choose **Delete** to confirm.

Changes in account credentials may take up to 4 hours to get reflected in the AWS IAM evaluations. The time taken depends on when the last credential report was fetched by the Cloud View service and the time when changes were made in AWS IAM

Reference:

https://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_access-keys.html#Using_CreateAccessKey

(search for header- "To create, modify, or delete a user's access keys")

Using AWS CLI:

aws iam update-access-key --access-key-id <Key-ID> --status <Active/Inactive>
For command usage refer: https://docs.aws.amazon.com/cli/latest/reference/iam/update-access-key.html

Reference

- https://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_access-keys.html
- https://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_finding-unused.html

Control ID - 6: Ensure IAM Password Policy is Enabled

Criticality: HIGH

Specification

Password policies are set to enforce password complexity requirements, password reset and expiry methods to all the IAM users.

Rationale

Setting a password complexity, expiry time, password reset policy reduces chances of credentials getting compromised and misused.

Evaluation

Check IAM Password policy is applied.

Remediation

To create or change a password policy using AWS Management Console:

1. Sign in to the AWS Management Console and open the IAM console at https://console.aws.amazon.com/iam/.
2. In the navigation pane, click **Account Settings**.
3. Click on Set Password Policy
4. In the **Password Policy** section, select the options you want to apply to your password policy.
5. Click **Apply Password Policy**.

To create or change a password policy Using

- **AWS CLI:** `aws iam update-account-password-policy`
- **Tools for Windows PowerShell:** `Update-IAMAccountPasswordPolicy`
- **AWS API:** `UpdateAccountPasswordPolicy`

Reference:

https://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_passwords_account-policy.html#IAMPasswordPolicy

Reference

- https://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_passwords_account-policy.html
- https://docs.amazonaws.cn/en_us/config/latest/developerguide/iam-password-policy.html

Control ID - 7: Ensure IAM password policy requires at least one uppercase letter

Criticality: HIGH

Specification

Password policies are, in part, used to enforce password complexity requirements. IAM password policies can be used to ensure password are comprised of different character sets. It is recommended that the password policy require at least one uppercase letter.

Rationale

Setting a password complexity policy increases account resiliency against brute force login attempts.

Evaluation

Check "Requires at least one uppercase letter" is checked under "Password Policy"

Remediation

Perform the following steps to apply the password policies on AWS :

Method 1 : Via AWS Console

1. Login to AWS Console (with appropriate permissions to View Identity Access Management Account Settings)
2. Go to IAM Service on the AWS Console
3. Click on Account Settings on the Left Pane
4. Click on Update Password Policy
5. Check "Requires at least one uppercase letter"
6. Click "Apply password policy"

Method 2 : Via AWS Command Line Interface
Run a command : # aws iam update-account-password-policy --require-uppercase-characters
Note: All commands starting with "aws iam update-account-password-policy" can be combined into a single command.

Reference:

https://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_passwords_account-policy.html

Using AWS CLI:

aws iam update-account-password-policy --require-uppercase-characters
For command usage refer: https://docs.aws.amazon.com/cli/latest/reference/iam/update-account-password-policy.html

Reference

- https://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_passwords_account-policy.html
- https://docs.amazonaws.cn/en_us/config/latest/developerguide/iam-password-policy.html

Control ID - 8: Ensure IAM password policy require at least one lowercase letter

Criticality: HIGH

Specification

Password policies are, in part, used to enforce password complexity requirements. IAM password policies can be used to ensure password are comprised of different character sets. It is recommended that the password policy require at least one lowercase letter.

Rationale

Setting a password complexity policy increases account resiliency against brute force login attempts.

Evaluation

Check "Requires at least one lowercase letter" is checked under "Password Policy"

Remediation

Perform the following steps to apply the password policies on AWS :

Method 1 : Via AWS Console

1. Login to AWS Console (with appropriate permissions to View Identity Access Management Account Settings)
2. Go to IAM Service on the AWS Console
3. Click on Account Settings on the Left Pane
4. Click on Update Password Policy
5. Check "Requires at least one lowercase letter"
6. Click "Apply password policy"

Method 2 : Via AWS Command Line Interface
Run a command : # aws iam update-account-password-policy --require-lowercase-characters
Note: All commands starting with "aws iam update-account-password-policy" can be combined into a single command.

For command usage refer: https://docs.aws.amazon.com/cli/latest/reference/iam/update-account-password-policy.html

Reference:

https://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_passwords_account-policy.html

Reference

- https://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_passwords_account-policy.html

- https://docs.amazonaws.cn/en_us/config/latest/developerguide/iam-password-policy.html

Control ID - 9: Ensure IAM password policy require at least one symbol

Criticality: HIGH

Specification

Password policies are, in part, used to enforce password complexity requirements. IAM password policies can be used to ensure password are comprised of different character sets. It is recommended that the password policy require at least one symbol.

Rationale

Setting a password complexity policy increases account resiliency against brute force login attempts.

Evaluation

Check "Require at least one non-alphanumeric character" is checked under "Password Policy"

Remediation

Perform the following steps to apply the password policies on AWS :

Method 1 : Via AWS Console

1. Login to AWS Console (with appropriate permissions to View Identity Access Management Account Settings)
2. Go to IAM Service on the AWS Console
3. Click on Account Settings on the Left Pane
4. Click on Update Password Policy
5. Check "Requires at least one non-alphanumeric character"
6. Click "Apply password policy"

Method 2 : Via AWS Command Line Interface
Run a command : # aws iam update-account-password-policy --require-symbols
Note: All commands starting with "aws iam update-account-password-policy" can be combined into a single command.

For command usage refer: https://docs.aws.amazon.com/cli/latest/reference/iam/update-account-password-policy.html

Reference:

https://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_passwords_account-policy.html

Reference

- https://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_passwords_account-policy.html
- https://docs.amazonaws.cn/en_us/config/latest/developerguide/iam-password-policy.html

Control ID - 10: Ensure IAM password policy require at least one number

Criticality: HIGH

Specification

Password policies are, in part, used to enforce password complexity requirements. IAM password policies can be used to ensure password are comprised of different character sets. It is recommended that the password policy require at least one number.

Rationale

Setting a password complexity policy increases account resiliency against brute force login attempts.

Evaluation

Check "Require at least one number " is checked under "Password Policy"

Remediation

Perform the following steps to apply the password policies on AWS :

Method 1 : Via AWS Console

1. Login to AWS Console (with appropriate permissions to View Identity Access Management Account Settings)
2. Go to IAM Service on the AWS Console
3. Click on Account Settings on the Left Pane
4. Click on Update Password Policy
5. Check "Requires at least one number"
6. Click "Apply password policy"

Method 2 : Via AWS Command Line Interface
Run a command : # aws iam update-account-password-policy --require-numbers
Note: All commands starting with "aws iam update-account-password-policy" can be combined into a single command.

For command usage refer: https://docs.aws.amazon.com/cli/latest/reference/iam/update-account-password-policy.html

Reference:

https://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_passwords_account-policy.html

Reference

- https://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_passwords_account-policy.html
- https://docs.amazonaws.cn/en_us/config/latest/developerguide/iam-password-policy.html

Control ID - 13: Ensure IAM password policy expires passwords within 90 days or less

Criticality: HIGH

Specification

IAM password policies can require passwords to be rotated or expired after a given number of days. It is recommended that the password policy expire passwords after 90 days or less.

Rationale

Reducing the password lifetime increases account resiliency against brute force login attempts. Additionally, requiring regular password changes help in the following scenarios:

1. Passwords can be stolen or compromised sometimes without your knowledge.
2. This can happen via a system compromise, software vulnerability, or internal threat.
3. Certain corporate and government web filters or proxy servers have the ability to intercept and record traffic even if it's encrypted.
4. Many people use the same password for many systems such as work, email, and personal.
5. Compromised end user workstations might have a keystroke logger.

Evaluation

Check **Password expiration period (in days)** is set to 90 or less.

Remediation

Perform the following steps to apply the password policies on AWS :

Method 1 : Via AWS Console

1. Login to AWS Console (with appropriate permissions to View Identity Access Management Account Settings)
2. Go to IAM Service on the AWS Console
3. Click on Account Settings
4. Click on Update Password Policy
5. Check "Enable password expiration"
6. Set "Password expiration period (in days):" to 90 or less
7. Click "Apply password policy"

Method 2 : Via AWS Command Line Interface
Run a command : # aws iam update-account-password-policy --max-password-age <value 90 or less>
Note: All commands starting with "aws iam update-account-password-policy" can be combined into a single command.

For command usage refer: https://docs.aws.amazon.com/cli/latest/reference/iam/update-account-password-policy.html

Reference:

https://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_passwords_account-policy.html

Reference

- https://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_passwords_account-policy.html

Control ID - 17: Ensure IAM policies are attached only to groups or roles

Criticality: MEDIUM

Specification

By default, IAM users, groups, and roles have no access to AWS resources. IAM policies are the means by which privileges are granted to users, groups, or roles. It is recommended that IAM policies be applied directly to groups and roles but not users.

Rationale

Assigning privileges at the group or role level reduces the complexity of access management as the number of users grow. Reducing access management complexity may in-turn reduce opportunity for a principal to inadvertently receive or retain excessive privileges.

Evaluation

Check IAM policies are not attached directly to users.

Remediation

Perform the following to create an IAM group and assign a policy to it:

1. Sign in to the AWS Management Console and open the IAM console at https://console.aws.amazon.com/iam/.
2. In the navigation pane, click Groups and then click Create New Group.
3. In the Group Name box, type the name of the group and then click Next Step.
4. In the list of policies, select the check box for each policy that you want to apply to all members of the group. Then click Next Step.
5. Click Create Group

Perform the following to add a user to a given group:

1. Sign in to the AWS Management Console and open the IAM console at https://console.aws.amazon.com/iam/.
2. In the navigation pane, click Groups
3. Select the group to add a user to
4. Under Users tab, click Add Users To Group
5. Select the users to be added to the group
6. Click Add Users

Perform the following to remove a direct association between a user and policy:

1. Sign in to the AWS Management Console and open the IAM console at https://console.aws.amazon.com/iam/.
2. In the left navigation pane, click on Users
3. Select the user
4. Click on the Permissions tab
5. Expand Permissions Policies
6. Click Detach/Delete icon (cross) for all managed and inline policies

Using AWS CLI:

aws iam create-group --group-name <GrpName>
For command usage refer: https://docs.aws.amazon.com/cli/latest/reference/iam/create-group.html

aws iam attach-group-policy --policy-arn <PolicyArn> --group-name <GrpName>
For command usage refer: https://docs.aws.amazon.com/cli/latest/reference/iam/attach-group-policy.html

aws iam add-user-to-group --user-name <UserName> --group-name <GrpName>
For command usage refer: https://docs.aws.amazon.com/cli/latest/reference/iam/add-user-to-group.html

aws iam detach-user-policy --user-name <UserName> --policy-arn <PolicyArn>
For command usage refer: https://docs.aws.amazon.com/cli/latest/reference/iam/detach-user-policy.html

aws iam delete-user-policy --user-name <UserName> --policy-name <PolicyName>
For command usage refer: https://docs.aws.amazon.com/cli/latest/reference/iam/delete-user-policy.html

Reference

- https://docs.aws.amazon.com/IAM/latest/UserGuide/access_policies_managed-vs-inline.html#inline-policies
- https://docs.aws.amazon.com/IAM/latest/UserGuide/access_policies_manage-attach-detach.html

Control ID - 45: S3 Bucket Access Control List Grant Access to Everyone or Authenticated Users

Criticality: HIGH

Specification

Ensure that your S3 buckets access control list does not allow unrestricted public to read or write access. Exposing your S3 buckets to everyone or any authenticated AWS users can lead to data leaks, data loss and unexpected charges for the S3 service.

Rationale

Allowing unrestricted access increases opportunities for loss of data. It is recommended to promptly review your S3 buckets and their contents to ensure that you are not accidentally making objects available to users that you don't intend.

Evaluation

Control checks whether bucket access control list allows read or write access to Everyone or AWS Authenticated Users

Remediation

Perform the following:

1. Sign in to the AWS management console and open the amazon s3 console at https://console.aws.amazon.com/s3/.
2. Select the bucket and click **Permissions**.
3. In the permissions pane, navigate to **Public Access** section.
4. The section shows a list of permissions assigned to everyone. Uncheck all the permissions granted to everyone.

Using AWS CLI:

aws s3api put-bucket-acl --bucket <BucketName>
For command usage refer: https://docs.aws.amazon.com/cli/latest/reference/s3api/put-bucket-acl.html

Reference

https://docs.aws.amazon.com/AmazonS3/latest/dev/acl-overview.html

Control ID - 46: S3 Bucket Policy Grant Access to Everyone

Criticality: HIGH

Specification

Allowing public access to S3 bucket using bucket policy can allow any user to read, upload, modify or delete contents of the bucket resulting in data loss and unexpected charges for the S3 service.

Rationale

Allowing unrestricted access increases opportunities for loss of data. It is recommended to promptly review your S3 buckets and their contents to ensure that you are not accidentally making objects available to u

Evaluation

Control checks whether bucket policy allows read or write access to Everyone.

Remediation

Perform the following:

1. Sign in to the AWS management console and open the amazon s3 console at https://console.aws.amazon.com/s3/.
2. Select the bucket and click **Permissions**.
3. In the permissions pane, navigate to **Bucket Policy** section.
4. In the bucket policy editor, update value for **Principal** by removing wildcard * which represents open access and configuring appropriate account-arn(s) or canonical user ID(s).
   For advanced configuration using bucket policy, refer to the bucket policy samples mentioned at https://docs.aws.amazon.com/AmazonS3/latest/dev/example-bucket-policies.html
5. Save bucket policy.

Using AWS CLI:

aws s3api put-bucket-policy --bucket <BucketName> --policy <value>
For command usage refer: https://docs.aws.amazon.com/cli/latest/reference/s3api/put-bucket-policy.html

Reference

https://docs.aws.amazon.com/AmazonS3/latest/user-guide/set-permissions.html

Control ID - 47: Ensure access logging is enabled for S3 buckets

Criticality: HIGH

Specification

S3 Bucket Access Logging generates a log that contains access records for each request made to your S3 bucket. An access log record contains details about the request, such as the request type, the resources specified in the request worked, and the time and date the request was processed. It is recommended that bucket access logging be enabled on the CloudTrail S3 bucket.

Rationale

By enabling S3 bucket logging on target S3 buckets, it is possible to capture all events which may affect objects within a target bucket. Configuring logs to be placed in a separate bucket allows access to log information which can be useful in security and incident response workflows.

Evaluation

Control checks whether the logging is enabled on S3 buckets.

Remediation

Perform the following to enable server access logging:

1. Sign in to the AWS management console and open the amazon s3 console at https://console.aws.amazon.com/s3/.
2. Select the bucket and click **Properties**.
3. On Properties page, Choose **Server Access Logging** and click **Enable Logging**.
4. Set **Target Bucket** to receive the log record objects. Set **Target Prefix** (Optional).
5. Choose save.

Using AWS CLI:

aws s3api put-bucket-logging --bucket <BucketName> --bucket-logging-status <value>
For command usage refer: https://docs.aws.amazon.com/cli/latest/reference/s3api/put-bucket-logging.html

Reference

https://docs.aws.amazon.com/AmazonS3/latest/user-guide/server-access-logging.html

Control ID - 48: Ensure versioning is enabled for S3 buckets

Criticality: HIGH

Specification

Versioning is a means of keeping multiple variants of an object in the same bucket. You can use versioning to preserve, retrieve, and restore every version of every object stored in your Amazon S3 bucket. With versioning, you can easily recover from both unintended user actions and application failures.

Rationale

Versioning can protect data from being deleted and from being overwritten accidentally.

Evaluation

Control checks whether the versioning is enabled on S3 buckets.

Remediation

Perform the following to enable versioning of S3 Buckets:

1. Sign in to the AWS management console and open the amazon s3 console at https://console.aws.amazon.com/s3/.
2. Select the bucket and click **Properties** and then select **Versioning**.
3. Choose **Enable Versioning** and select Save.

Using AWS CLI:

aws s3api put-bucket-versioning --bucket <BucketName> --versioning-configuration <value>
For command usage refer: https://docs.aws.amazon.com/cli/latest/reference/s3api/put-bucket-versioning.html

Reference

https://docs.aws.amazon.com/AmazonS3/latest/user-guide/enable-versioning.html

Control ID - 57: Ensure that bucket policy enforces encryption in transit

Criticality: HIGH

Specification

This control ensures that encryption in transit is enforced on a bucket using bucket policy.

Rationale

Configuring bucket policy to enforce encryption in transit mitigates the risk of data leakage and disclosure of sensitive data while data in transit. This provides protection from sniffing attacks especially when buckets and objects are being accessed outside of the trusted network.

By default, bucket has no bucket policy enforced.

Note:

In Bucket Policy for "SecureTransport", setting resource to "Resource": "arn:aws:s3:::<Bucket-Name>/*" will enforce "SecureTransport" configuration to all the objects inside the bucket but not on bucket url itself. However, Setting resource to "Resource": "arn:aws:s3:::<Bucket-Name>" will enforce "SecureTransport" configuration to the bucket itself but not on objects inside the bucket.

Evaluation

This control ensures that bucket policy enforcing "SecureTransport" exists for all the objects inside of a bucket

Remediation

To update/apply bucket policy to enforce encryption in transit while accessing any object inside the bucket:

Using AWS Console:

1. Sign in to the AWS Management Console and open the Amazon S3 console at https://console.aws.amazon.com/s3/
2. In the Bucket name list, choose the name of the bucket that you want
3. Choose Permissions
4. Choose Bucket Policy
5. If there is no existing bucket policy for a bucket define one with json:

```
{
   "Version": "2012-10-17",
   "Statement": [{
          "Effect": "Deny",
          "Principal":
          {
                 "AWS": "*"
          },
          "Action": "s3:*",
          "Resource": [
                 "arn:aws:s3:::<Bucket-Name>",
                 "arn:aws:s3:::<Bucket-Name>/*"
          ],
          "Condition": {
                 "Bool":
                 {
                        "aws:SecureTransport": "false"
                 }
          }
    }]
  }
```

28. If there is already a bucket policy, in Statement section append json mentioned below:

```
 {
   "Effect": "Deny",
   "Principal":
   {
```

```
34.                         "AWS": "*"
35.                 },
36.                 "Action": "s3:*",
37.                 "Resource": [
38.                         "arn:aws:s3:::<Bucket-Name>",
39.                         "arn:aws:s3:::<Bucket-Name>/*"
40.                 ],
41.                 "Condition": {
42.                         "Bool":
43.                         {
44.                                 "aws:SecureTransport": "false"
45.                         }
46.                 }
47.         }
```

48. Choose Save

Using AWS CLI:

Use command put-bucket-policy. For command usage refer:
https://docs.aws.amazon.com/cli/latest/reference/s3api/put-bucket-policy.html

Impact:

Enforcing "SecureTransport" on a bucket may break existing communication channels with no support for SSL/TLS

Note:

In Bucket Policy for "SecureTransport", Setting resource to "Resource": "arn:aws:s3:::<Bucket-Name>/*" will enforce "SecureTransport" configuration to all the objects inside the bucket but not on bucket url itself. However, Setting resource to "Resource": "arn:aws:s3:::<Bucket-Name>" will enforce "SecureTransport" configuration to the bucket itself but not on objects inside the bucket.

Reference

https://docs.aws.amazon.com/AmazonS3/latest/user-guide/default-bucket-encryption.html

Control ID - 58: Ensure that the key expiry is set for CMK with external key material

Criticality: MEDIUM

Specification

Ensure that all CMKs with external key material have an expiration time set.

Rationale

In case of CMK with external key material, key expiration is managed by the customer as AWS cannot rotate the key. Expiration period can be customized at the time of importing the key material. AWS KMS will automatically delete the key material after the expiration period. You can delete the imported key material yourself. In both cases, the key material is deleted but the CMK reference and the metadata remain so that you can import key material afterward.

This ensures that the keys cannot be used beyond their assigned lifetimes.

By default, key expiration is not enabled.

Evaluation

This control ensures that expiry is set for CMK with external key material.

Remediation

Using AWS Console:

1. Sign in to the AWS Management Console and open the AWS Key Management Service (KMS) console at https://console.aws.amazon.com/kms
2. Choose "Customer managed keys" option
3. Choose the alias or key ID of a CMK
4. Under "Key material", click on the "Delete key material" button
5. Select "Confirm that you want to delete this key material" and click "Delete Key material" option from the pop-up
6. Under "Key material", click on the "Download wrapping key and import token" button (If you have already downloaded this in past 24hr you can use that)
7. Choose encryption algorithm to use and select Download and close
8. Now follow the steps to Encrypt Key Material with Openssl : https://docs.aws.amazon.com/kms/latest/developerguide/importing-keys-encrypt-key-material.html
9. Under "Key material", click on the "Upload key material" button
10. For "Wrapped key material", click on the "Choose file" option
11. Upload the file that contains wrapped (encrypted) key material
12. Upload the Import token by clicking the "Choose file" option for the "Import token" option
13. Under Expiration option, enter expiration date and time for the "Key material expires at" option
14. Click on "Upload key material"

Note: AWS has provided a new console called the "AWS KMS console", which will help users organize and maintain keys in a better way.
Link to AWS KMS console: https://us-east-2.console.aws.amazon.com/kms

Using AWS CLI:

aws kms delete-imported-key-material For command usage refer:
https://docs.aws.amazon.com/cli/latest/reference/kms/delete-imported-key-material.html

aws kms import-key-material For command usage refer:
https://docs.aws.amazon.com/cli/latest/reference/kms/import-key-material.html

Reference

- https://docs.aws.amazon.com/kms/latest/developerguide/overview.html
- https://docs.aws.amazon.com/kms/latest/developerguide/importing-keys.html

Control ID - 63: Ensure Block new public bucket policies for an account is set to true

Criticality: HIGH

Specification

The account level public access setting 'Block new public bucket policies' ensures that a bucket policy cannot be updated to grant public access.

Note:
AWS now refers this option as "Block public access to buckets and objects granted through new public bucket or access point policies"

Rationale

The account level public access setting 'Block new public bucket policies' prevents adding a new public policy on any bucket that belongs to that account. When bucket owner tries to set public policy, it displays an access denied error message. This setting does not change the behavior of an existing bucket policies with public access.

Evaluation

This control ensures that the account level public access setting 'Block new public bucket policies' is set to true.

Remediation

To enable Block new public bucket policies in public access settings for account:

Using AWS Console:

1. Sign in to the AWS Management Console and open the Amazon S3 console at https://console.aws.amazon.com/s3/
2. Click on the "Public access settings for this account" option
3. Click edit
4. In "Manage public bucket policies for this account" section, check the box for "Block public access to buckets and objects granted through new public bucket or access point policies"
5. Choose Save
6. When you're asked for confirmation, enter confirm. Then choose Confirm to save your changes

Using AWS CLI:

aws s3control put-public-access-block --account-id <AccountID> --public-access-block-configuration BlockPublicPolicy=true For command usage refer:
https://docs.aws.amazon.com/cli/latest/reference/s3control/put-public-access-block.html

Reference

https://docs.aws.amazon.com/AmazonS3/latest/user-guide/block-public-access-account.html

Control ID - 64: Ensure that Block public and cross-account access if bucket has public policies for the account is set to true

Criticality: HIGH

Specification

The account level public access setting 'Block public and cross-account access if bucket has public policies' blocks public and cross-account access to all the buckets from the account by overriding existing bucket policies.

Note:
AWS now refers this option as "Block public and cross-account access to buckets and objects through any public bucket or access point policies"

Rationale

The account level public access setting 'Block new public bucket policies' prevents public or cross-account access granted to all the buckets from the account by bucket policies. If this option is set, access to buckets that are publicly accessible will be limited to the bucket owner and to AWS services. This option can be used to protect buckets that have public policies while you work to remove the policies.

Evaluation

This control ensures that account level public access setting 'Block public and cross-account access if bucket has public policies' is set to true.

Remediation

To enable Block public and cross-account access in public access settings for account:

Using AWS Console:

1. Sign in to the AWS Management Console and open the Amazon S3 console at https://console.aws.amazon.com/s3/
2. Click on the "Public access settings for this account" option
3. Click edit
4. In "Manage public bucket policies for this account" section, check the box for "Block public and cross-account access to buckets and objects through any public bucket or access point policies"
5. Choose Save
6. When you're asked for confirmation, enter confirm. Then choose Confirm to save your changes

Using AWS CLI:

aws s3control put-public-access-block --account-id <AccountID> --public-access-block-configuration RestrictPublicBuckets=true For command usage refer:
https://docs.aws.amazon.com/cli/latest/reference/s3control/put-public-access-block.html

Reference

https://docs.aws.amazon.com/AmazonS3/latest/user-guide/block-public-access-account.html

Control ID - 65: Ensure that Block new public ACLs and uploading public objects for the account is set to true

Criticality: HIGH

Specification

The account level public access setting 'Block new public ACLs and uploading public objects' ensures that any bucket or object ACL from the account cannot be updated to grant public access.

Note:
AWS now refers this option as "Block public access to buckets and objects granted through new access control lists (ACLs)"

Rationale

The account level public access setting 'Block new public ACLs and uploading public objects' prevents adding public access through ACL to buckets or Objects from the account. It does not affect existing buckets or objects. Use this setting to protect against future attempts to use ACLs to make buckets or objects public. If an application tries to upload an object with a public ACL or if an administrator tries to apply a public access setting to the bucket, this setting will block the public access setting for the bucket or the object.

Evaluation

This control ensures that account level public access setting 'Block new public ACLs and uploading public objects' is set to true.

Remediation

To enable Block new public ACLs and uploading public objects in public access settings for account:

Using AWS Console:

1. Sign in to the AWS Management Console and open the Amazon S3 console at https://console.aws.amazon.com/s3/
2. Click on the "Public access settings for this account" option
3. Click edit
4. In "Manage public access control lists (ACLs) for this account" section, check the box for "Block public access to buckets and objects granted through new access control lists (ACLs)"

5. Choose Save
6. When you're asked for confirmation, enter confirm. Then choose Confirm to save your changes

Using AWS CLI:

aws s3control put-public-access-block --account-id <AccountID> --public-access-block-configuration BlockPublicAcls=true For command usage refer:
https://docs.aws.amazon.com/cli/latest/reference/s3control/put-public-access-block.html

Reference

https://docs.aws.amazon.com/AmazonS3/latest/user-guide/block-public-access-account.html

Control ID - 66: Ensure that Remove public access granted through public ACLs for the account is enabled

Criticality: HIGH

Specification

The account level public access setting 'Remove public access granted through public ACLs' ensures that any of the existing S3 buckets from the account does not to evaluate any public ACL when authorizing a request.

Note:
AWS now refers this option as "Block public access to buckets and objects granted through any access control lists (ACLs)"

Rationale

This account level option enforces all the S3 buckets from the account not to evaluate any public ACL when authorizing a request, ensuring that no bucket or object can be made public by using ACLs. This setting overrides any current or future public access settings for current and future objects in the bucket. If an existing application is currently uploading objects with public ACLs to the bucket, this setting will override the setting on the object.

Evaluation

This control ensures that account level public access setting 'Remove public access granted through public ACLs' is set to true.

Remediation

To enable Remove public access granted through public ACLs in public access settings for account:

Using AWS Console:

1. Sign in to the AWS Management Console and open the Amazon S3 console at https://console.aws.amazon.com/s3/
2. Click on the "Public access settings for this account" option
3. Click edit
4. In "Manage public access control lists (ACLs) for this account" section, check the box for "Block public access to buckets and objects granted through any access control lists (ACLs)"
5. Choose Save
6. When you're asked for confirmation, enter confirm. Then choose Confirm to save your changes

Using AWS CLI:

aws s3control put-public-access-block --account-id <AccountID> --public-access-block-configuration IgnorePublicAcls=true For command usage refer: https://docs.aws.amazon.com/cli/latest/reference/s3control/put-public-access-block.html

Reference

https://docs.aws.amazon.com/AmazonS3/latest/user-guide/block-public-access-account.html

Control ID - 67: Ensure Server Side Encryption (SSE) is enabled for S3 bucket

Criticality: HIGH

Specification

Ensure Server Side Encryption (SSE) is enabled for the S3 bucket.

Rationale

Configuring SSE for a bucket ensures that data stored in S3 bucket is encrypted at rest.

By default, SSE is not configured for a bucket.

Note: Setting Default Encryption (SSE) for an existing bucket does not encrypt existing objects in the bucket.

Evaluation

This control ensures that "ServerSideEncryptionConfiguration" exists for a bucket.

Remediation

To enable default encryption on an S3 bucket:

Using AWS Console:

1. Sign in to the AWS Management Console and open the Amazon S3 console at https://console.aws.amazon.com/s3/

2. In the Bucket name list, choose the name of the bucket that you want
3. Choose Properties
4. Choose Default encryption
5. Choose AES-256 or AWS-KMS
6. Choose Save

Using AWS CLI:

aws s3api put-bucket-encryption For command usage refer:
https://docs.aws.amazon.com/cli/latest/reference/s3api/put-bucket-encryption.html

Impact:

Enabling default encryption may require an update in bucket policy. If AWS KMS option is used for default encryption configuration, it is subjected to the RPS (requests per second) limits of AWS KMS.

Note:

Setting Default Encryption (SSE) for an existing bucket does not encrypt existing objects in the bucket.

Reference

https://docs.aws.amazon.com/AmazonS3/latest/user-guide/default-bucket-encryption.html

Control ID - 114: Ensure Images (AMIs) owned by an AWS account are not public

Criticality: HIGH

Specification

- By default, Account owned (i.e. account user-created) Image is shared privately only to account providing LaunchPermissions as [].
- By default, the snapshot associated with the image (AMI) is shared privately only to account for providing createVolumePermissions as [].
- On UI, While Sharing Image Privately, there is an optional provision to share associated snapshot as well

Rationale

Sharing Image (AMI) publicly allows any AWS user from any AWS account to list and launch EC2 instances. Sharing of AMI is not recommended because:

1. Discloses Sensitive information stored on root device i.e Audit logs, Stored credentials, source codes, software licenses, Application Data etc.
2. May create licensing issues with the applications installed on Root Device
3. To avoid information disclosure, Images should be shared privately so that only users from the same account owning Image and trusted AWS accounts can list and launch the same.

By default, the parameter "LaunchPermissions" is set to [], allowing private sharing of Image to the users from the same (owner) account.

When there is a business need to share an image (AMI) with users from multiple accounts, Image sharing permissions provide the configuration that can allow sharing of the image privately only with the configured trusted AWS accounts.

Evaluation

The control checks for restricted image (AMI) sharing permissions

Remediation

After you make an AMI public, it is available in **Community AMIs** when you launch an instance in the same region using the console. Note that it can take a short while for an AMI to appear in **Community AMIs** after you make it public. It can also take a short while for an AMI to be removed from **Community AMIs** after you make it private again.

To private an AMI using the console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **AMIs**.
3. Select your AMI from the list, and then choose **Actions**, **Modify Image Permissions**.
4. Choose **Private** and choose **Save**.

To make private an AMI image using AWS CLI

```
aws ec2 modify-image-attribute --region <your-region> 1 --image-id <ImageId> --launch-permission "Remove=[{Group=all}]"
```

Command reference :

https://docs.aws.amazon.com/cli/latest/reference/ec2/modify-image-attribute.html

Reference

https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/sharingamis-intro.html

Control ID - 115: Ensure that EBS Volumes attached to EC2 instances are encrypted

Criticality: MEDIUM

Specification

The Control Ensure that all EBS volumes that are attached to instances (either as root Device or Block device) are encrypted.

Rationale

AWS features encrypt data stored on an EBS volume at rest and in motion by setting a single option. When an encrypted EBS volume is created and attached to a supported instance type, data on the volume, disk I/O, and snapshots created from the volume are all encrypted. The encryption occurs on the servers that host the EC2 instances, providing encryption of data as it moves between EC2 instances and EBS storage.

Root device of an instance may store sensitive information like boot information, operating system files, applications installed, application licenses, application data, source codes, audit logs etc.
Encrypting root device volumes ensures compliance with requirement - encryption-at-rest and encryption-in-transit.

To check EBS encryption supported instance types refer:
https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EBSEncryption.html#EBSEncryption_supported_instances

Evaluation

The control check for Encryption property for all attached EBS Volumes

Remediation

To encrypt EBS volume Attached to an EC2 Instance

1. On AWS Console, Navigate to the Volumes under "Elastic Block Store" Under the Attachment information note down the Path example: /dev/xvda and navigate to the corresponding instance to which EBS Volume is attached. Stop Instance Associated with the unencrypted EBS volume.
   Reference: https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/Stop_Start.html#starting-stopping-instances
2. Create a snapshot of an unencrypted volume. This snapshot will be unencrypted.
   Reference:https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ebs-creating-snapshot.html
3. Select the Snapshot, from Actions, choose, Create Volume fill in appropriate settings and tick the Encryption checkbox, Set Encryption key to AWS-Managed or Customer-Managed CMK (Recommended), as per compliance requirement.
   Reference: https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ebs-restoring-volume.html
4. Detach the unencrypted volume from Instance.
   Reference: https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ebs-detaching-volume.html
5. To attach encrypted volume as the root device, Navigate to the Volumes under "Elastic Block Store"
   - Select Volume
   - From Actions, Click attach
   - It will open “Attach Volume” dialogue box
   - Choose Instance
   - Choose Device (Mount Path) as per noted from Step1
   - Click Attach
   Reference: https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ebs-attaching-volume.html
6. Start the instance and Ensure that Instance functions as expected.
7. Delete Older unencrypted Volume as we already have encrypted
   Reference: https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ebs-deleting-volume.html
8. Delete corresponding unencrypted Snapshot as we already have encrypted snapshot for the same.
   Reference: https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ebs-deleting-snapshot.html

Note:
- Steps 3,4 can be done while the instance is running.Instance can be Stopped at Step 5, just before detaching the volume from the instance.
- However, to preserve the consistency of root device as data is continuously being updated, It is recommended to stop the instance at initial steps.

Reference

https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EBSEncryption.html#EBSEncryption_considerations
https://docs.aws.amazon.com/kms/latest/developerguide/services-ebs.html#ebs-encrypt

https://docs.aws.amazon.com/kms/latest/developerguide/services-ebs.html#ebs-encryption-context
https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EBSEncryption.html#EBSEncryption_supported_instances

Control ID - 116: Ensure that Unattached EBS Volumes are encrypted

Criticality: MEDIUM

Specification

The Control Ensure that all unattached EBS volumes are encrypted.

Rationale

AWS features encrypt data stored on an EBS volume at rest and in motion by setting a single option. When an encrypted EBS volume is created and attached to a supported instance type, data on the volume, disk I/O, and snapshots created from the volume are all encrypted. The encryption occurs on the servers that host the EC2 instances, providing encryption of data as it moves between EC2 instances and EBS storage.
Unattached Volumes can be root as well as block device and it may store sensitive and confidential data. Encrypting volumes ensures compliance with requirement - encryption-at-rest and encryption-in-transit.

Evaluation

The control check for Encryption property for all unattached EBS Volumes

Remediation

To encrypt EBS volume that are not attached.

1. On AWS Console, Navigate to the Volumes under "Elastic Block Store" choose the unencrypted volume. Create a snapshot of an unencrypted volume. This snapshot will be unencrypted.
   Reference:https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ebs-creating-snapshot.html
2. Select the Snapshot, from Actions, choose, Create Volume fill in appropriate settings and tick the Encryption checkbox, Set Encryption key to AWS-Managed or Customer-Managed CMK (Recommended), as per compliance requirement.
   Reference: https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ebs-restoring-volume.html
3. Delete Older unencrypted Volume as we already have encrypted
   Reference: https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ebs-deleting-volume.html
4. Delete corresponding unencrypted Snapshot as we already have encrypted snapshot for the same.
   Reference: https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ebs-deleting-snapshot.html

Reference

https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EBSEncryption.html#EBSEncryption_considerations
https://docs.aws.amazon.com/kms/latest/developerguide/services-ebs.html#ebs-encrypt
https://docs.aws.amazon.com/kms/latest/developerguide/services-ebs.html#ebs-encryption-context

Control ID - 119: Ensure no AWS default KMS Key is used to protect Secrets

Criticality: HIGH

Specification

AWS Secrets Manager service provides secure management of information such as database credentials, passwords, third-party API keys, and arbitrary text. This information is termed as secrets and can be retrieved from centralized storage whenever needed. Use of KMS CMK is recommended to encrypt the secrets when stored at rest.

Rationale

All Secrets in AWS Secret Manager service are stored with encryption at rest. A default AWS managed KMS key is used by default. The user has no control over the access, key material, and life cycle of the key. Use of a KMS customer Managed Key (CMK) for secret encryption provides access and life cycle management over the encrypting key. Custom key material can also be used for additional confidentiality.

Evaluation

This control ensures that all Secrets in AWS secret manager are encrypted using KMS CMK.

Remediation

Using AWS Console:

1. Sign in to the AWS Management Console and open the AWS Secrets Manager console at https://console.aws.amazon.com/secretsmanager/home.
2. In the Navigation pane, choose `Secrets`.
3. Click on the secret to be modified.
4. Click on `Actions` and select `Edir encryption key`.
5. Select an appropriate KMS Customer Managed Key (CMK) from the list.
6. Check `Create new version of secret with new encryption key` option.
7. Click `Save`.

Using AWS CLI:

Use the followinf command to update the KMS key for the secret:

```
aws secretsmanager update-secret --secret-id <secret_id> --kms-key-id
<kms_key_id>
```

Note:

1. The `secret_id` above can be ARN or the friendly name of the secret.
2. The `kms_key_id` in above command can be either key id or alias of the KMS key

Reference

- https://docs.aws.amazon.com/secretsmanager/latest/userguide/data-protection.html

- https://docs.aws.amazon.com/cli/latest/reference/secretsmanager/update-secret.html

Control ID - 120: Ensure No CMK is marked for deletion

Criticality: MEDIUM

Specification

The deletion of AWS KMS CMK is destructive in nature. Deleting CMK results in the deletion of the associated key material and the associated metadata. If the effect radius is not monitored correctly, deleting a CMK could lead to the unavailability of data as any data encrypted with this key cannot be decrypted. To prevent such cases, a waiting period is enforced by AWS during which the key remains in Pending for Deletion state and can be recovered. During this, most of the functional use of the key such as encryption and decryption is unavailable.

Rationale

When the CMK is deleted, the associated key material and the metadata is deleted irreversibly. Thus no CMKs should be transitioned as Pending for Deletion state unless it is verified the key is no longer needed.

Evaluation

This control ensures no required CMK is marked for deletion.

Remediation

Using AWS console:

1. Sign in to the **AWS Management Console** and open the **Amazon KMS console** at https://console.aws.amazon.com/kms/.
2. Select the appropriate region from the top right corner.
3. In the navigation pane, choose **Customer managed keys**, and then choose the CMK that you want to cancel deletion for.
4. Click "**Key actions**" button and click "**Cancel Key Deletion**".

Using AWS CLI:

```
# aws kms cancel-key-deletion --key-id <KeyID/KeyARN>
```

For command usage refer: https://docs.aws.amazon.com/cli/latest/reference/kms/cancel-key-deletion.html

Reference

https://docs.aws.amazon.com/kms/latest/developerguide/deleting-keys.html

Control ID - 121: Ensure only Root user of the AWS Account should be allowed full access on the CMK

Criticality: HIGH

Specification

Access control for AWS KMS Customer Managed Keys (CMKs) is primarily accomplished using Key policies and can be used in conjunction with IAM policies and Grants. Unlike many AWS services, the AWS account's root user does not have access to CMK implicitly. Thus, it is recommended to allow access to AWS account's root user to reduce the risk of CMK becoming unmanageable in case the account with such privileges is either not accessible or deleted. Further, to enable IAM policies for the CMK, the Root user of the AWS account needs to have full access on the key.

Rationale

AWS KMS key access is managed by its resource based policy known as Key Policy. If the key policy doesn't explicitly allow, even the AWS account's root user will not be able to access the CMK. In cases such as the user with the privileges to access and manage the CMK is either inaccessible or deleted, the CMK will become unmanageable. Furthermore, to enable IAM policies in conjunction with Key Policies, full access to the AWS account's Root user is required in the Key policy for the CMK. Thus it is recommended to allow full access to AWS account's root over the CMK as this account cannot be deleted.

Note:

1. The control works with the following KMS Policy JSON Elements:
    - Principal
    - Action
    - Resource
    - SID
2. Use of other elements may result policy read error.
3. The NotAction and NotResource works with "allow all deny by exception" model which is not recommended and hence are not covered.
4. For Principal, following elements are supported:
    - AWS
    - Service
    - Federated
5. Sid element for statement is optional
6. Full access is detected as granting "*****" or "**kms:***" action to a prinicipal on "*****" resource. Granting all indivisual actions with or without using wildcards will not be considered as granting full permissions.

Evaluation

This control evaluates whether the AWS account's Root user is assigned full access to the CMK using Key policy.

Remediation

To assign the Root user of the account as the Owner of the key:

Using AWS console:

1. Sign in to the **AWS Management Console** and open the **Amazon KMS console** at https://console.aws.amazon.com/kms/.
2. Select the appropriate region from the top right corner.
3. In the navigation pane, choose **Customer managed keys**, and then choose the CMK that you want to modify.
4. Navigate to "**Key policy**" and click **Switch to Policy View** button. Click **Edit**.

5. Add/modify the policy statement such that only AWS account's Root user has privileges to perform any action (**kms:***) on any resource (*).
6. Click **Save changes**.

Using AWS CLI:

```
# aws kms put-key-policy --key-id <KeyId/KeyARN> --policy-name default --
policy <PolicyJsonString>
```

For command usage refer: https://docs.aws.amazon.com/cli/latest/reference/kms/put-key-policy.html

Reference

https://docs.aws.amazon.com/kms/latest/developerguide/key-policies.html#key-policy-default-allow-root-enable-iam

Control ID - 122: Permissions to delete key is not granted to any Principal other than the Root user of AWS Account

Criticality: HIGH

Specification

The deletion of AWS KMS CMK is destructive in nature. Deleting CMK results in the deletion of the associated key material and metadata. If the effect radius is not monitored correctly, deleting a CMK could lead to the unavailability of data as any data encrypted with this key cannot be decrypted. To prevent such cases, a waiting period is enforced by AWS during which the key remains in Pending for Deletion state and can be recovered. During this, most of the functional use of the key such as encryption and decryption is unavailable.

Rationale

When the CMK is deleted all the key material and the metadata is deleted irreversibly. Unavailability of the CMK can cause a malfunction in the system as the deleted CMK dependent encryption/decryption cannot be performed. Following the Principle of Least Privileges, restricting the delete permissions over the CMK ("**kms:ScheduleKeyDeletion**") to just the root user of the AWS account helps to ensure the CMK is not deleted accidentally or maliciously.

Note:

1. The control works with the following KMS Policy JSON Elements:
    - Principal
    - Action
    - Resource
    - SID
2. Use of other elements may result policy read error.
3. The NotAction and NotResource works with "allow all deny by exception" model which is not recommended and hence are not covered.
4. For Principal, following elements are supported:
    - AWS
    - Service
    - Federated
5. Sid element for statement is optional

6. A resource will be flagged if either "`kms:ScheduleKeyDeletion`" or "`*`" or "`kms:*`" is explicitly granted to a principal with exception of root user of AWS account. Granting the action using wildcards will not be considered.

Evaluation

This control ensures no Principal other than the AWS account's root user has permissions to delete the CMK.

Remediation

Using AWS console:

1. Sign in to the **AWS Management Console** and open the **Amazon KMS console** at https://console.aws.amazon.com/kms/.
2. Select the appropriate region from the top right corner.
3. In the navigation pane, choose **Customer managed keys**, and then choose the CMK that you want to modify.
4. Navigate to **Key policy** and click **Switch to policy view**. Click **Edit**.
5. Remove "**kms:ScheduleKeyDeletion**" privilege from any Principal other than the AWS account's root user.

Using AWS CLI:

Use the following command to update the key policy of a CMK which has no principal other than the AWS account's root user with "kms:ScheduleKeyDeletion" privilege.

```
# aws kms put-key-policy --key-id <KeyId/KeyARN> --policy-name default --
policy <PolicyJsonString>
```

For command usage refer: https://docs.aws.amazon.com/cli/latest/reference/kms/put-key-policy.html

Reference

https://docs.aws.amazon.com/kms/latest/developerguide/deleting-keys.html

Control ID - 123: Ensure CMK administrators are not the user of the key

Criticality: HIGH

Specification

The CMK administrators have privileges to manage the CMK including modifications to Key Policy, delete key, update aliases and manage key material. An administrator with key use permissions such as encryption and decryption using the key can be used maliciously. It is recommended to follow the Principle of Separation of Duties and restrict administrators from having user privileges for the CMKs.

Rationale

Following the Principle of Separation of Duties while granting privileges ensures that no single entity has enough permissions to cause severe issues and help in minimizing the issues and frauds due to malicious intent. Restricting administrators from using CMKs for cryptographic functions (encryption and decryption) will reduce the chances of misuse of CMK for unwanted/unauthorized actions.

Any principal is considered as a CMK administrator (without delete permissions) if any single statement in the key policy contains the following details:

- Effect: Allow
- Resource: "*****"
- Statement ID: "Allow access for Key Administrators"
- Actions: "kms:Create*", "kms:Describe*", "kms:Enable*", "kms:List*", "kms:Put*", "kms:Update*", "kms:Revoke*", "kms:Disable*", "kms:Get*", "kms:Delete*", "kms:TagResource", "kms:UntagResource"
- In case of key ith external key material, administrators are granted additional action of "kms:ImportKeyMaterial"

Any principal is considered as a CMK administrator (with delete permissions) if any single statement in the key policy contains the following details:

- Effect: Allow
- Resource: "*****"
- Statement ID: "Allow access for Key Administrators"
- Actions: "kms:Create*", "kms:Describe*", "kms:Enable*", "kms:List*", "kms:Put*", "kms:Update*", "kms:Revoke*", "kms:Disable*", "kms:Get*", "kms:Delete*", "kms:TagResource", "kms:UntagResource", "kms:ScheduleKeyDeletion", "kms:CancelKeyDeletion"
- In case of key ith external key material, administrators are granted additional action of "kms:ImportKeyMaterial"

Any principal is considered as a CMK user if any single statement in the key policy contains the following details:

- Effect: Allow
- Resource: "*****"
- Statement ID: "Allow use of the key"
- Actions: "kms:Encrypt", "kms:Decrypt", "kms:ReEncrypt*", "kms:GenerateDataKey*", "kms:DescribeKey"

Note:

1. The control works with the following KMS Policy JSON Elements:
    - Principal
    - Action
    - Resource
    - SID
2. Use of other elements may result policy read error.
3. The NotAction and NotResource works with "allow all deny by exception" model which is not recommended and hence are not covered.
4. For Principal, following elements are supported:
    - AWS
    - Service
    - Federated
5. Sid element for statement is optional
6. The control's behavior is similar to the AWS console behavior to detect key admins and users.

Evaluation

This control ensures that the CMK administrators are not the user of the key.

Remediation

Using AWS console:

1. Sign in to the **AWS Management Console** and open the **Amazon KMS console** at https://console.aws.amazon.com/kms/.
2. Select the appropriate region from the top right corner.
3. In the navigation pane, choose **Customer managed keys**, and then choose the CMK that you want to modify.
4. Navigate to "**Key policy**" and click **Switch to Policy View** button. Click **Edit**.
5. Add/modify the policy such that no principal with administrative privileges on the CMK is allowed user permissions on the CMK.
6. Click **Save changes**.

Using AWS CLI:

```
# aws kms put-key-policy --key-id <KeyId/KeyARN> --policy-name default --
policy <PolicyJsonString>
```

For command usage refer: https://docs.aws.amazon.com/cli/latest/reference/kms/put-key-policy.html

Reference

https://docs.aws.amazon.com/kms/latest/developerguide/key-policies.html

Control ID - 124: Ensure all Custom key stores are connected to their CloudHSM clusters

Criticality: HIGH

Specification

Custom Key Stores can be used if more control is required over the storage of the key material of a AWS KMS CMK. A custom key store is backed by Hardware Security Modules (HSMs) that are used for creating and storing cryptographic key material. These HSMs are a part of AWS CloudHSM cluster and the custom key store needs to be connected to the CloudHSM cluster to function as intended.

Rationale

The Custom key stores provide managed hardware for the creation and storage of the AWS KMS CMKs. A CloudHSM cluster is used to provide the necessary infrastructure and thus both needs to be connected for the proper function of the system. If the Custom Key Store is not connected to the associated CloudHSM cluster, even though the custom Key store and its CMKs can be viewed and managed, the no new CMKs cannot be created. Furthermore, when disconnected, the existing CMKs in the custom key store cannot be used for any cryptographic operations such as encryption and decryption.

Evaluation

This control ensures all the Custom Key stores are connected to the associated CloudHSM cluster.

Remediation

Custom Key Store is discnnected from associated CloudHSM cluster.

Reference

https://docs.aws.amazon.com/kms/latest/developerguide/custom-key-store-overview.html

Control ID - 126: Ensure AMIs owned by an AWS account are encrypted

Criticality: MEDIUM

Specification

An Amazon Machine Image (AMI) provides the information required to launch an instance. AMI supports Instance Store and EBS for the data storage. Encryption is supported only for EBS backed volumes snapshot attached to AMI.

Note: Encrypting AMI with CMK might incur additional cost after an encrypted instance is launched from AMI.

Rationale

AMI provides all the information required to launch an instance. AMI can consist of sensitive and business critical data and to protect it's confidentiality and integrity, it is highly recommended to enable encryption.

Encrypting AMI ensures compliance with the requirement - encryption-at-rest.

Evaluation

This control ensures that EBS snapshots attached to Account Owned AMI are encrypted.

Remediation

Using AWS console

1. Sign in to AWS console and navigate to https://console.aws.amazon.com/ec2/v2/home
2. Navigate to AMIs Section in the left navigation menu.
3. Select the actionable AMI, Choose **Actions**.
4. From the list of available options choose **Copy AMI**.
5. Select the destination region and tick the **Encryption** Option.
6. Under the **master key** option select appropriate KMS Key (It is recommended to choose Customer managed KMS key for AMI consisting of highly sensitive data).
7. Now **Copy AMI**, Newly created AMI will be Encrypted.
8. You can safely delete the older AMI.

Using AWS CLI

```
$> aws ec2 copy-image --encrypted <kms-key-id> --description <description-for-
ami>
    --kms-key-id <kms-key-id-for-destinated-region> --name <name-of-ami> --
source-image-id <actionable-image-id>
    --source-region <actionable-ami-region> --region <destination-region>
```

Command Reference: https://docs.aws.amazon.com/cli/latest/reference/ec2/copy-image.html

```
$> aws ec2 delete-image --name <old-ami> --region <region>
```

Command Reference: https://docs.aws.amazon.com/cli/latest/reference/appstream/delete-image.html

Note: Using the KMS CMK key might incur additional cost.

Reference

https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/AMIEncryption.html

Control ID - 127: Ensure AWS EBS Volume snapshots are encrypted

Criticality: MEDIUM

Specification

EBS snapshots are incremental backups for EBS volumes. AWS support encryption of snapshot using AWS managed KMS key or using CMK.

Note: Using Customer manged CMK for encryption might incur additional charges.

Rationale

Data is an important asset to an organization. EBS snapshot data may consist of confidential data, proprietary source code. Encryption at rest ensures that your data is protected from unauthorized access and malicious threat. AWS supports AES-256 encryption of EBS volumes using KMS keys for providing encryption at rest.This ensures that confidentiality and integrity of data is maintained. Encryption and decryption are transparently handled by AWS for the underlying storage.

Encrypting the EBS snapshots ensure compliance with the requirement - encryption-at-rest.

Evaluation

This control ensures that the EBS snapshots are encrypted.

Remediation

Using the AWS console:

1. Sign in to AWS console and navigate to https://console.aws.amazon.com/ec2/v2/home
2. From the left navigation choose **snapshots** option.
3. Select the actionable snapshot, choose **actions**.
4. Choose the **Copy Snapshot** option.
5. Check the **encryption** option.
6. Under the **Master key** choose the kms key (It is recommended to choose KMS CMK when EBS snapshot consist of highly sensitive data).
7. Click on **copy**.
8. Once the newly encrypted snapshot is ready old snapshot should be deleted.

Using AWS CLI

```
$> aws ec2 copy-snapshot --description <description-for-new-snapshot> --
destination-region <region-name>
```

```
    --encrypted --kms-key-id <kms-key-id> ---source-region <region-name> --
source-snapshot-id <actionable-snapshot-id>
```

Command reference: https://docs.aws.amazon.com/cli/latest/reference/ec2/copy-snapshot.html

```
$> aws ec2 delete-snapshot --snapshot-id <actionable-snapshot-id> --region
    <region-name>
```

Command reference: https://docs.aws.amazon.com/cli/latest/reference/ec2/delete-snapshot.html

Note:

1. Applications using old snapshot id should point to new snapshot id.
2. Using KMS CMK to encrypt snapshots might incur additional cost.

Reference

https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ebs-creating-snapshot.html

Control ID - 128: Ensure access log is enabled for Application load balancer

Criticality: MEDIUM

Specification

Access logs for a ELBv2 Load Balancer contains detailed information about requests sent to the load balancer. These include time the request was received, the client's IP address, latencies, request paths, and server responses. These logs can be monitored and reviewed to analyze traffic patterns and to troubleshoot issues and security incidents.

Rationale

Access logs for Application Load balancer provide visibility on the access requests to the load balancer and can be used anomalous and malicious traffic and calls. These logs can also play a vital role in troubleshooting and forensic analysis.

Access logging is disabled by default.

Evaluation

This control verifies that Access Logs are enabled for all Application Load Balancers.

Remediation

Note:

1. To enable access logs for ELBv2 Load balancer, an S3 bucket is required in the same region as the Load Balancer with Amazon S3-Managed Encryption Keys (SSE-S3) encryption.
2. The bucket must have a bucket policy that grants Elastic Load Balancing permission to write the access logs to your bucket. For more details on required permissions and policy, please refer to

https://docs.aws.amazon.com/elasticloadbalancing/latest/application/load-balancer-access-logs.html#enable-access-logging

Using AWS Console:

- To create a new S3 bucket

1. Go to Amazon S3 console at https://console.aws.amazon.com/s3/
2. Select `Create Bucket.`
3. Enter a unique bucket name and select the region where you created your load balancer.
4. Click `Create`.

- To allow write access to ELBv2 Load Balancer on S3 bucket, edit the bucket policy

1. Go to Amazon S3 console at https://console.aws.amazon.com/s3/
2. Select the S3 bucket, choose `Permissions`, choose `Bucket Policy`.
3. Edit the policy to allow the required permissions for Load Balancer to write the logs to the bucket. Please refer to https://docs.aws.amazon.com/elasticloadbalancing/latest/application/load-balancer-access-logs.html#enable-access-logging for more details on required permissions and bucket policy.

- To Enable Access Logs for ELBv2 Load Balancer

1. Go to Amazon EC2 console at https://console.aws.amazon.com/ec2/v2/home
2. In the navigation pane, under `Load Balancing`, choose `Load Balancers`.
3. Select the load balancer.
4. Under `Description` tab, choose `Edit attributes`.
5. Select `Enable` for `Access logs`.
6. For `S3 location`, select an existing S3 bucket or create a new bucket.
7. Click `Save`.

Using AWS CLI:

- To create a new S3 bucket

```
aws s3api create-bucket --bucket my-bucket --region us-east-1
```

- To allow write access to ELBv2 Load Balancer on S3 bucket, edit the bucket policy

1. Create a policy JSON file to allow the required permissions for Load Balancer to write the logs to the bucket. Please refer to https://docs.aws.amazon.com/elasticloadbalancing/latest/application/load-balancer-access-logs.html#enable-access-logging for more details on required permissions and bucket policy.
2. Use the command below to attach the policy to the bucket
   `aws s3api put-bucket-policy --bucket <bucket_name> --policy <path_to_json>`

Note: Using the above command will overwrite the current policy. Please keep the include all statements in the JSON that are required for the policy.

- To Enable Access Logs for existing ELBv2 Load Balancer

Use the following command to enable the access logs for existing Load Balancer

```
aws elbv2 modify-load-balancer-attributes --load-balancer-arn
<load_balancer_arn> --attributes Key=access_logs.s3.enabled,Value=true
Key=access_logs.s3.bucket,Value=<bucket_name>
Key=access_logs.s3.prefix,Value=<log_prefix>
```

Reference

- https://docs.aws.amazon.com/elasticloadbalancing/latest/application/load-balancer-access-logs.html
- https://docs.aws.amazon.com/cli/latest/reference/s3api/put-bucket-policy.html
- https://docs.aws.amazon.com/cli/latest/reference/s3api/create-bucket.html
- https://docs.aws.amazon.com/cli/latest/reference/elbv2/modify-load-balancer-attributes.html

Control ID - 129: Ensure access log is enabled for Classic Elastic load balancer

Criticality: MEDIUM

Specification

Access logs for a Classic Load Balancer contains detailed information about requests sent to the load balancer. These include time the request was received, the client's IP address, latencies, request paths, and server responses. These logs can be monitored and reviewed to analyze traffic patterns and to troubleshoot issues and security incidents.

Rationale

Access logs for Classic Load balancer provide visibility on the access requests to the load balancer and can be used anomalous and malicious traffic and calls. These logs can also play a vital role in troubleshooting and forensic analysis.

Evaluation

This control verifies that Access Logs are enabled for all Classic Load Balancers.

Remediation

Note: To enable access logs for Classic Load balancer, an S3 bucket is required in the same region as the Load Balancer with Amazon S3-Managed Encryption Keys (SSE-S3) encryption.

Using AWS Console:

- To create a new S3 bucket

1. Go to Amazon S3 console at https://console.aws.amazon.com/s3/
2. Select `Create Bucket.`
3. Enter a unique bucket name and select the region where you created your load balancer.
4. Click `Create`.

- To allow write access to Load Balancer on S3 bucket, edit the bucket policy

1. Go to Amazon S3 console at https://console.aws.amazon.com/s3/
2. Select the S3 bucket, choose `Permissions`, choose `Bucket Policy`.

3. Edit the policy to allow `s3:PutObject` action to the root user of the AWS account that corresponds to the Region for your load balancer. For the list of AWS accounts corresponding to the region, please refer to https://docs.aws.amazon.com/elasticloadbalancing/latest/classic/enable-access-logs.html

- To Enable Access Logs for Classic Load Balancer

1. Go to Amazon EC2 console at https://console.aws.amazon.com/ec2/v2/home
2. In the navigation pane, under `Load Balancing`, choose `Load Balancers`.
3. Select the load balancer.
4. Under `Description` tab, choose `Configure Access Logs`.
5. Select `Enable access logs`.
6. Enter the `Interval` as required.
7. For `S3 location`, select an existing S3 bucket or create a new bucket.
8. Click `Save`.

Using AWS CLI:

- To create a new S3 bucket

```
aws s3api create-bucket --bucket my-bucket --region us-east-1
```

- To allow write access to Load Balancer on S3 bucket, edit the bucket policy

1. Create a policy JSON file to allow `s3:PutObject` action to the root user of the AWS account that corresponds to the Region for your load balancer. For the list of AWS accounts corresponding to the region, please refer to https://docs.aws.amazon.com/elasticloadbalancing/latest/classic/enable-access-logs.html
2. Use the command below to attach the policy to the bucket
   `aws s3api put-bucket-policy --bucket <bucket_name> --policy <path_to_json>`

Note: Using the above command will overwrite the current policy. Please keep the include all statements in the JSON that are required for the policy.

- To Enable Access Logs for existing Classic Load Balancer

1. Create a JSON file containing the load balancer configuration
   `{ "AccessLog": { "Enabled": true, "S3BucketName": "<bucket_name>", "EmitInterval": <interval>, "S3BucketPrefix": "<optional_prefix>" } }`
2. Use the following command to enable the access logs for existing Classic Load Balancer

```
aws elb modify-load-balancer-attributes --load-balancer-name my-loadbalancer --load-balancer-attributes <path_to_json_file>
```

Reference

- https://docs.aws.amazon.com/elasticloadbalancing/latest/classic/access-log-collection.html
- https://docs.aws.amazon.com/cli/latest/reference/s3api/put-bucket-policy.html
- https://docs.aws.amazon.com/cli/latest/reference/s3api/create-bucket.html

Control ID - 130: Ensure Classic Elastic load balancer is not using unencrypted protocol

Criticality: HIGH

Specification

A load balancer serves as the single point of contact, distributing the incoming traffic across multiple targets for high availability. Classic Load Balancers can listen to multiple protocols such as TCP, SSL, HTTP, and HTTPS. The use of unencrypted protocols is not recommended.

Rationale

Encrypted communication helps in maintaining the confidentiality and integrity of the data. The use of unencrypted protocols poses major security risks, are easy to compromise and data is available in plain text for anyone to see. Thus use of unencrypted protocols should be avoided.

Encrypted protocols such as HTTPS and SSL should be used.

Evaluation

This control verifies that Classic Load Balancer listens for only encrypted protocols.

Remediation

Note: Removing listeners from Classic load balancers will stop balancing the protocol listener was listening on and may require further changes on the application that was using the load balancer

- To delete a listener for Classic Load Balancer

Using AWS Console:

1. Go to Amazon EC2 console at https://console.aws.amazon.com/ec2/v2/home
2. In the navigation pane, under `Load Balancing`, choose `Load Balancers`.
3. Select the load balancer.
4. Got to `Listeners` tab, click `Edit`.
5. Click `Remove` for all entries working on unencrypted protocols such as HTTP.
6. Click `Save`.

Using AWS CLI:

Use the following command to delete a listener from existing Classic Load Balancer

```
aws elb delete-load-balancer-listeners --load-balancer-name
<load_balancer_name> --load-balancer-ports <port>
```

Note: The port in the above command refers to listener entry for the Classic ELB that is using an unencrypted protocol such as TCP or HTTP.

Reference

- Listeners for Your Classic Load Balancer - Elastic Load Balancing

- https://docs.aws.amazon.com/cli/latest/reference/elb/delete-load-balancer-listeners.html

Control ID - 131: Ensure Elastic load balancer listener is not using unencrypted protocol

Criticality: HIGH

Specification

A load balancer serves as the single point of contact, distributing the incoming traffic across multiple targets for high availability. ELBv2 Load Balancers can listen to multiple protocols such as TCP, UDP, and HTTPS. The use of unencrypted protocols is not recommended.

Rationale

Encrypted communication helps in maintaining the confidentiality and integrity of the data. Use of unencrypted protocols poses major security risks, are easy to compromise and data is available in plain text for anyone to see. Thus use of unencrypted protocols should be avoided.

Encrypted protocols such as HTTPS for ELBv2 application load balancers and TLS protocol for network load balancers should be used.

Evaluation

This control verifies that ELBv2 Load Balancer listens for only encrypted protocols.

Remediation

Note: Removing listeners from ELBv2 load balancers will stop balancing the protocol listener was listening on and may require further changes on the application that was using the load balancer

- To delete a listener for ELBv2 Load Balancer

Using AWS Console:

1. Go to Amazon EC2 console at https://console.aws.amazon.com/ec2/v2/home
2. In the navigation pane, under `Load Balancing`, choose `Load Balancers`.
3. Select the load balancer.
4. Got to `Listeners` tab.
5. Select all listeners working on unencrypted protocols such as HTTP.
6. Select `Delete`, When prompted for confirmation, choose `Yes, Delete`.

Using AWS CLI:

Use the following command to delete a listener from existing ELBv2 Load Balancer

```
aws elbv2 delete-listener --listener-arn <listner_arn>
```

Reference

- https://docs.aws.amazon.com/elasticloadbalancing/latest/application/delete-listener.html

- https://docs.aws.amazon.com/cli/latest/reference/elbv2/delete-listener.html

Control ID - 144: Ensure EFS Encryption is enabled for data at rest

Criticality: MEDIUM

Specification

Amazon Elastic File System (Amazon EFS) provides simple, scalable, fully manged NFS file System to store data. EFS support Encryption at rest, this property can only be enabled while creating EFS. AWS supports encryption using AWS KMS keys or customer managed keys.

Note: Using Customer managed keys might incur additional charges.

Rationale

AWS EFS supports encryption for data at rest. Enabling encryption at rest helps to protect data against the threat from malicious activity ensuring confidentiality and integrity of your data.

Amazon uses the industry-standard AES-256 encryption algorithm to encrypt your data on the server that hosts your EFS. Once your data is encrypted, Amazon handles authentication of access and decryption of your data transparently with a minimal impact on performance. You don't need to modify your database client applications to use encryption.

Encrypting EFS helps to achieve compliance requirements needed within your organization.

Evaluation

This control ensures that EFS data is encrypted at rest.

Remediation

Note: Once an EFS is created AWS doesn't allow to modify the encryption setting. This guide provides general guidelines to create a new Encrypted EFS and copy data from unencrypted EFS.

Step 1: Create Encrypted EFS

Using AWS Console:

1. Login into AWS console and navigate to https://console.aws.amazon.com/efs/home
2. From left navigation, choose **File Systems**.
3. Select the **actionable** File System and expand for details and make note of the existing configuration.
4. To create a new encrypted EFS, choose **Create File System**.
5. Configure **network access** configuration as per actionable EFS.
6. Choose next steps, Under Enable encryption select **Enable encryption at rest**.
7. Choose the KMS key (It is recommended to choose AWS Customer managed key if your EFS consist of sensitive data).
8. Choose **next steps**, review configurations and choose **create File System**.

Using AWS CLI

Use This command to create a file system with an existing configuration.

```
aws efs create-file-system --creation-token <unique-token> --performance-mode
<as-per-existing-config>
    --throughput-mode <as-per-existing-config> --region <as-per-existing-
config> --tags Key=key,Value=value
  --profile <as-per-existing-config> --encrypted --kms-key-id <new-kms-key-id>
```

Note: Unique Token Ensure that duplicate file systems are not created in case of network failures.

Command reference: https://docs.aws.amazon.com/cli/latest/reference/efs/create-file-system.html

Documentation reference: https://docs.aws.amazon.com/efs/latest/ug/creating-using-create-fs.html

Step 2: Copy data to encrypted EFS

Once the Encrypted EFS is created. Mount the old and newly created EFS on EC2 Instance or On-premise Server.
Follow the AWS guide for mount instructions.

Documentation reference: https://docs.aws.amazon.com/efs/latest/ug/mounting-fs.html

1. Copy the Data from source EFS to new one.

2. Once the Copy is completed. Verify the integrity of data in new EFS and once verification of data is completed, it is safe to delete the old unencrypted EFS.

Note: Using KMS CMK might incur additional cost.

Reference

https://docs.aws.amazon.com/efs/latest/ug/mounting-fs.html

https://docs.aws.amazon.com/cli/latest/reference/efs/create-file-system.html

https://docs.aws.amazon.com/cli/latest/reference/efs/describe-file-systems.html

Control ID - 145: Ensure EFS File system resource is encrypted by KMS using a customer managed Key (CMK)

Criticality: MEDIUM

Specification

Amazon Elastic File System (Amazon EFS) provides simple, scalable, fully manged NFS file System to store data. EFS support Encryption at rest, this property can only be enabled while creating EFS. AWS supports encryption using AWS KMS keys or customer managed keys.

Note: Using Customer managed keys might incur additional charges.

Rationale

AWS EFS supports encryption for data at rest. Enabling encryption at rest helps to protect data against the threat from malicious activity ensuring confidentiality and integrity of your data.

Amazon uses the industry-standard AES-256 encryption algorithm to encrypt your data on the server that hosts your EFS. Once your data is encrypted, Amazon handles authentication of access and decryption of your data transparently with a minimal impact on performance. You don't need to modify your database client applications to use encryption.

Encrypting EFS using CMK helps to achieve compliance requirements needed within your organization.

Evaluation

This control ensures that EFS data is encrypted at rest using customer master key.

Remediation

Note: Once an EFS is created AWS doesn't allow to modify the encryption setting. This guide provides general guidelines to create a new Encrypted EFS and copy data from unencrypted EFS.

Step 1: Create Encrypted EFS

Using AWS Console:

1. Login into AWS console and navigate to https://console.aws.amazon.com/efs/home
2. From left navigation, choose **File Systems**.
3. Select the **actionable** File System and expand for details and make note of the existing configuration.
4. To create a new encrypted EFS, choose **Create File System**.
5. Configure **network access** configuration as per actionable EFS.
6. Choose next steps, Under Enable encryption select **Enable encryption at rest**.
7. Choose the AWS Customer managed key.
8. Choose **next steps**, review configurations and choose **create File System**.

Using AWS CLI

Use This command to create a file system with an existing configuration.

```
aws efs create-file-system --creation-token <unique-token> --performance-mode
<as-per-existing-config>
    --throughput-mode <as-per-existing-config> --region <as-per-existing-
config> --tags Key=key,Value=value
  --profile <as-per-existing-config> --encrypted --kms-key-id <new-kms-key-id>
```

Note: Unique Token Ensure that duplicate file systems are not created in case of network failures.

Command reference: https://docs.aws.amazon.com/cli/latest/reference/efs/create-file-system.html

Documentation reference: https://docs.aws.amazon.com/efs/latest/ug/creating-using-create-fs.html

Step 2: Copy data to encrypted EFS

Once the Encrypted EFS is created. Mount the old and newly created EFS on EC2 Instance or On-premise Server.
Follow the AWS guide for mount instructions.

Documentation reference: https://docs.aws.amazon.com/efs/latest/ug/mounting-fs.html

1. Copy the Data from source EFS to new one.

2. Once the Copy is completed. Verify the integrity of data in new EFS and once verification of data is completed, it is safe to delete the old unencrypted EFS.

Note: Using KMS CMK might incur additional cost.

Reference

https://docs.aws.amazon.com/efs/latest/ug/mounting-fs.html

https://docs.aws.amazon.com/cli/latest/reference/efs/create-file-system.html

https://docs.aws.amazon.com/cli/latest/reference/efs/describe-file-systems.html

Control ID - 146: Ensure that AWS Elastic Block Store (EBS) volume snapshots are not public

Criticality: HIGH

Specification

Amazon Elastic Block Store (Amazon EBS) provides block-level storage volumes for use with EC2 instances. EBS volumes behave like raw, unformatted block devices. You can mount these volumes as devices on your instances. EBS volume snapshots can be made private through snapshot permissions.

Rationale

EBS volume snapshots contain a mirror copy of your application which might contain sensitive data also. So if this EBS volume snapshots are shared publicly which will give them permission to both copy the snapshot and create a volume from it.

Note:

Public snapshots of encrypted volumes are not supported, but you can share an encrypted snapshot with specific accounts.

Evaluation

This control ensures that AWS Elastic Block Store (EBS) volume snapshots are not public.

Remediation

Using AWS Console:

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. Choose Snapshots in the navigation pane.
3. Select the snapshot and then choose Actions, Modify Permissions.
4. Choose the Private radio button.
5. Choose Save.

Using AWS CLI:

Execute the following command:

```
aws ec2 modify-snapshot-attribute --snapshot-id <snapshot_id> --attribute
createVolumePermission --operation-type remove --group-names all
```

Command Reference:
https://docs.aws.amazon.com/cli/latest/reference/ec2/modify-snapshot-attribute.html

Reference

https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ebs-modifying-snapshot-permissions.html

Control ID - 147: Ensure that AWS ElastiCache Memcached clusters are not associated with default VPC

Criticality: LOW

Specification

Amazon ElastiCache is a web service that makes it easy to set up, manage, and scale a distributed in-memory data store or cache environment in the cloud. It provides a high-performance, scalable, and cost-effective caching solution. At the same time, it helps remove the complexity associated with deploying and managing a distributed cache environment. You can choose in which subnet group(VPC) to create the ElastiCache cluster at the time of creation.

Rationale

Resources which may not actually need to be in network also use default VPC. And this resource will have access throughout the default network. It is a best practice to create elasticache resource in a custom VPC so that if there is any breach in default VPC your resource would be isolated from it.

Evaluation

This control ensures that AWS ElastiCache redis clusters are not associated with default VPC.

Remediation

Changing subnet group is only available while creating the ElastiCache Memcached Cluster. So to modify the subnet group we have to first backup the ElastiCache Memcached cluster and delete the misconfigured ElastiCache Memcached Cluster. After this operation we can restore the backup with modified port. After the restoring the cluster, application changes might be required for application to point to the new cluster endpoint.

Using AWS Console

To Backup the ElastiCache redis cluster

1. Open AWS ElastiCache console at https://console.aws.amazon.com/elasticache/home.
2. In the navigation pane, choose Memcached.
3. Choose the box to the left of the name of the Redis cluster you want to back up.
4. Now in actions dropdown choose backup.

To Delete the ElastiCache Memcached cluster

1. Open AWS ElastiCache console at https://console.aws.amazon.com/elasticache/home.
2. In the navigation pane, choose `Memcached`.
3. Choose the box to the left of the name of the Memcached cluster you want to delete.
4. Now in actions dropdown choose `Delete`.

To Restore the Backup

1. Open AWS ElastiCache console at https://console.aws.amazon.com/elasticache/home.
2. In the navigation pane, choose Backups.
3. Select the Backup you want to restore.
4. Click on the restore button.
5. In the Restore Cluster window ensure that `Choose a Subnet group` is set with a subnet group not associated with default vpc.

Using AWS CLI

To Backup the ElastiCache Memcached cluster reffer https://docs.aws.amazon.com/AmazonElastiCache/latest/red-ug/backups-manual.html.

To restore the backup reffer https://docs.aws.amazon.com/AmazonElastiCache/latest/red-ug/backups-restoring.html#backups-restoring-CLI.

Reference

- https://docs.aws.amazon.com/AmazonElastiCache/latest/mem-ug/Clusters.Delete.html
- https://docs.aws.amazon.com/AmazonElastiCache/latest/red-ug/backups-restoring.html
- https://docs.aws.amazon.com/AmazonElastiCache/latest/mem-ug/Clusters.Create.CLI.html
- https://docs.aws.amazon.com/AmazonElastiCache/latest/red-ug/SubnetGroups.Creating.html#SubnetGroups.Creating.CLI

Control ID - 148: Ensure that AWS ElastiCache Redis clusters are not associated with default VPC

Criticality: LOW

Specification

Amazon ElastiCache is a web service that makes it easy to set up, manage, and scale a distributed in-memory data store or cache environment in the cloud. It provides a high-performance, scalable, and cost-effective caching solution. At the same time, it helps remove the complexity associated with deploying and managing a distributed cache environment. You can choose in which subnet group(VPC) to create the elasticache cluster at the time of creation.

Rationale

Resources which may not actually need to be in network also use default VPC. And this resource will have access throughout the default network. It is a best practice to create elasticache resource in a custom VPC so that if there is any breach in default VPC your resource would be isolated from it.

Evaluation

This control ensures that AWS ElastiCache redis clusters are not associated with default VPC.

Remediation

Changing subnet group is only available while creating the ElastiCache Redis Cluster. So to modify the subnet group we have to first backup the ElastiCache redis cluster and delete the misconfigured ElastiCache Redis Cluster. After this operation we can restore the backup with modified port. After the restoring the cluster, application changes might be required for application to point to the new cluster endpoint.

Using AWS Console

To Backup the ElastiCache redis cluster

1. Open AWS ElastiCache console at https://console.aws.amazon.com/elasticache/home.
2. In the navigation pane, choose Redis.
3. Choose the box to the left of the name of the Redis cluster you want to back up.
4. Now in actions dropdown choose backup.

To Delete the ElastiCache Redis cluster

1. Open AWS ElastiCache console at https://console.aws.amazon.com/elasticache/home.
2. In the navigation pane, choose `Redis`.
3. Choose the box to the left of the name of the Redis cluster you want to delete.
4. Now in actions dropdown choose `Delete`.

To Restore the Backup

1. Open AWS ElastiCache console at https://console.aws.amazon.com/elasticache/home.
2. In the navigation pane, choose Backups.
3. Select the Backup you want to restore.
4. Click on the restore button.
5. In the Restore Cluster window ensure that `Choose a Subnet group` is set with a subnet group not associated with default vpc.

Using AWS CLI

To Backup the ElastiCache redis cluster reffer https://docs.aws.amazon.com/AmazonElastiCache/latest/red-ug/backups-manual.html.

To restore the backup reffer https://docs.aws.amazon.com/AmazonElastiCache/latest/red-ug/backups-restoring.html#backups-restoring-CLI.

Reference

- https://docs.aws.amazon.com/AmazonElastiCache/latest/mem-ug/Clusters.Delete.html
- https://docs.aws.amazon.com/AmazonElastiCache/latest/red-ug/backups-restoring.html
- https://docs.aws.amazon.com/AmazonElastiCache/latest/mem-ug/Clusters.Create.CLI.html
- https://docs.aws.amazon.com/AmazonElastiCache/latest/red-ug/SubnetGroups.Creating.html#SubnetGroups.Creating.CLI

Control ID - 149: Ensure that AWS ElastiCache redis clusters are not using their default endpoint ports

Criticality: LOW

Specification

Applications to access AWS ElastiCache Redis cluster, user has to set up the endpoint port. AWS ElastiCache Redis cluster by default sets this endpoint port to **6379**.

Rationale

Configuring AWS ElastiCache Redis cluster to non-default port helps in preventing or delaying the attacks by malicious users as attackers/malicious-users will require to initiate network scans which may increase the probability of anomaly detection which will give an opportunity to administrators to take compensatory actions like blocking malicious-users/IPs.

The default port for AWS ElastiCache Redis cluster is **6379**.

Evaluation

This control ensures that AWS ElastiCache Redis clusters are not using their default endpoint ports.

Remediation

Changing default endpoint ports is only available while creating the ElastiCache Redis Cluster. So to modify the port we have to first backup the ElastiCache Redis cluster and delete the misconfigured ElastiCache Redis Cluster. After this operation we can restore the backup with modified port. After the restoring the cluster, application changes might be required for application to point to the new cluster endpoint.

Using AWS Console

To Backup the ElastiCache Redis cluster

1. Open AWS ElastiCache console at https://console.aws.amazon.com/elasticache/home.
2. In the navigation pane, choose Redis.
3. Choose the box to the left of the name of the Redis cluster you want to back up.

4. Now in actions dropdown choose backup.

To Delete the ElastiCache Redis cluster

1. Open AWS ElastiCache console at https://console.aws.amazon.com/elasticache/home.
2. In the navigation pane, choose `Redis`.
3. Choose the box to the left of the name of the Redis cluster you want to delete.
4. Now in actions dropdown choose `Delete`.

To Restore the Backup

1. Open AWS ElastiCache console at https://console.aws.amazon.com/elasticache/home.
2. In the navigation pane, choose Backups.
3. Select the Backup you want to restore.
4. Click on the restore button.
5. In the Restore Cluster window ensure that port is not set to 6379.

Using AWS CLI

To Backup the ElastiCache redis cluster reffer https://docs.aws.amazon.com/AmazonElastiCache/latest/red-ug/backups-manual.html.

To restore the backup reffer https://docs.aws.amazon.com/AmazonElastiCache/latest/red-ug/backups-restoring.html.

Reference

- https://docs.aws.amazon.com/AmazonElastiCache/latest/mem-ug/Clusters.Delete.html
- https://docs.aws.amazon.com/AmazonElastiCache/latest/red-ug/backups-restoring.html
- https://docs.aws.amazon.com/AmazonElastiCache/latest/mem-ug/Clusters.Create.CLI.html

Control ID - 150: Ensure that AWS ElastiCache memcached clusters are not using their default endpoint ports

Criticality: LOW

Specification

Applications to access AWS ElastiCache Memcached cluster user has to set up the endpoint port.AWS ElastiCache Memcached cluster by default sets this endpoint port to 11211.

Rationale

Configuring AWS ElastiCache Memcached cluster to non-default port helps in preventing or delaying the attacks by malicious users as attackers/malicious-users will require to initiate network scans which may increase the probability of anomaly detection which will give an opportunity to administrators to take compensatory actions like blocking malicious-users/IPs.

The default port for AWS ElastiCache Memcached cluster is **11211**.

Evaluation

This control ensures that AWS ElastiCache Memcached clusters are not using their default endpoint ports.

Remediation

Changing default endpoint ports is only available while creating the ElastiCache Memcached Cluster. So to modify the port we have to first backup the ElastiCache Memcached cluster and delete the misconfigured ElastiCache Memcached Cluster. After this operation we can restore the backup with modified port. After the restoring the cluster, application changes might be required for application to point to the new cluster endpoint.

Using AWS Console

To Backup the ElastiCache Memcached cluster

1. Open AWS ElastiCache console at https://console.aws.amazon.com/elasticache/home.
2. In the navigation pane, choose Memcached.
3. Choose the box to the left of the name of the Redis cluster you want to back up.
4. Now in actions dropdown choose backup.

To Delete the ElastiCache Memcached cluster

1. Open AWS ElastiCache console at https://console.aws.amazon.com/elasticache/home.
2. In the navigation pane, choose `Memcached`.
3. Choose the box to the left of the name of the Memcached cluster you want to delete.
4. Now in actions dropdown choose `Delete`.

To Restore the Backup

1. Open AWS ElastiCache console at https://console.aws.amazon.com/elasticache/home.
2. In the navigation pane, choose Backups.
3. Select the Backup you want to restore.
4. Click on the restore button.
5. In the Restore Cluster window ensure that port is not set to 11211.

Using AWS CLI

To Backup the ElastiCache redis cluster reffer https://docs.aws.amazon.com/AmazonElastiCache/latest/red-ug/backups-manual.html.

To restore the backup reffer https://docs.aws.amazon.com/AmazonElastiCache/latest/red-ug/backups-restoring.html.

Reference

- https://docs.aws.amazon.com/AmazonElastiCache/latest/mem-ug/Clusters.Delete.html
- https://docs.aws.amazon.com/AmazonElastiCache/latest/red-ug/backups-restoring.html
- https://docs.aws.amazon.com/AmazonElastiCache/latest/mem-ug/Clusters.Create.CLI.html

Control ID - 151: Ensure AWS ElastiCache Redis cluster with Multi-AZ Automatic Failover feature is set to enabled

Criticality: LOW

Specification

Multi-AZ is necessary if the ElastiCache Redis cluster needs to achieve high availability. Redis asynchronously replicates the data form the primary to the read replicas. ElastiCache also propagates the DNS name of the promoted replica do that there is no endpoint change required on the application side. Auto-failover and Multi-AZ both should be enabled for Mutli-AZ Automatic Failover to work.

Rationale

Running ElastiCache Redis cluster in Multi-AZ will improve higher availability and a smaller need for administration. If an ElastiCache Redis primary node failure occurs, the impact on the ability to read/write to primary is limited to the time it takes for automatic failover to complete.

Evaluation

Control Ensures that AWS ElastiCache Redis cluster with Multi-AZ Automatic Failover is enabled.

Remediation

using AWS Console

To enable Multi-AZ on Redis Cluster

1. Open AWS ElastiCache console at https://console.aws.amazon.com/elasticache/home.
2. In the navigation pane, choose Redis.
3. Choose the box to the left of the name of the Redis cluster for which you want to enable Multi-AZ.
4. In Actions dropdown choose Modify.
5. In the Modify cluster window ensure that Multi-AZ option is checked.
6. Click Modify.

using AWS CLI

```
aws elasticache modify-replication-group --replication-group-id myReplGroup --
automatic-failover-enabled --multi-az-enabled
```

Reference

- https://docs.aws.amazon.com/AmazonElastiCache/latest/red-ug/AutoFailover.html
- https://docs.aws.amazon.com/AmazonElastiCache/latest/red-ug/Replication.Modify.html

Control ID - 152: Ensure AWS ElastiCache Redis cluster with Redis AUTH feature is enabled

Criticality: MEDIUM

Specification

Amazon ElastiCache in-transit encryption is an optional feature that allows you to increase the security of your data at its most vulnerable points—when it is in transit from one location to another. Redis authentication tokens enable Redis to require a token (password) before allowing clients to execute commands, thereby improving data security.

Rationale

Enabling Authentication adds an additional layer of security that restricts unauthorized access.

Impact

Application using this Redis cluster might have downtime during the backup and restore time.

Evaluation

Ensures AWS ElastiCache Redis Clusters are having Redis AUTH feature enabled.

Remediation

Redis AUTH feature is only available while creating the ElasticCache Redis Cluster. So to modify the option we have to first backup the ElasticCache Redis Cluster and delete the Cluster. After this operation, we can restore the backup with the Redis AUTH feature option set to enabled.
using AWS Console:
To Backup the ElastiCache redis cluster

1. Open AWS ElastiCache console at https://console.aws.amazon.com/elasticache/home.
2. In the navigation pane, choose `Redis`.
3. Choose the box to the left of the name of the Redis cluster you want to back up.
4. Now in actions dropdown choose `Backup`.

To Delete the ElastiCache redis cluster

1. Open AWS ElastiCache console at https://console.aws.amazon.com/elasticache/home.
2. In the navigation pane, choose `Redis`.
3. Choose the box to the left of the name of the Redis cluster you want to delete.
4. Now in actions dropdown choose `Delete`.

To Restore the Backup

1. Open AWS ElastiCache console at https://console.aws.amazon.com/elasticache/home.
2. In the navigation pane, choose `Backups`.
3. Choose the box to the left of the name of the backup you want to restore.
4. Click `Restore`
5. In the Restore Cluster window ensure `Encryption in-transit` option is checked for Redis AUTH option to be visible.
6. Ensure to check `Redis AUTH` option.

using AWS CLI:
To Backup the ElastiCache Redis cluster refer https://docs.aws.amazon.com/AmazonElastiCache/latest/red-ug/backups-manual.html
To restore the backup refer https://docs.aws.amazon.com/AmazonElastiCache/latest/red-ug/backups-restoring.html
To enable In-Transit Encryption on Redis refer
https://docs.aws.amazon.com/AmazonElastiCache/latest/red-ug/in-transit-encryption.html
To enable Redis AUTH refer https://docs.aws.amazon.com/AmazonElastiCache/latest/red-ug/auth.html.

Reference

- https://docs.aws.amazon.com/AmazonElastiCache/latest/red-ug/backups-manual.html
- https://docs.aws.amazon.com/AmazonElastiCache/latest/red-ug/backups-restoring.html

- https://docs.aws.amazon.com/AmazonElastiCache/latest/red-ug/in-transit-encryption.html

Control ID - 153: Ensure that AWS ElastiCache Redis clusters are In-Transit encrypted

Criticality: HIGH

Specification

Amazon ElastiCache is a web service that makes it easy to set up, manage, and scale a distributed in-memory data store or cache environment in the cloud. It provides a high-performance, scalable, and cost-effective caching solution.Amazon ElastiCache in-transit encryption is an optional feature that allows you to increase the security of your data at its most vulnerable points. Providing in-transit encryption capability, ElastiCache gives you a tool you can use to help protect your data when it is moving from one location to another.

Rationale

We might move data from a primary node to a read replica node within a replication group, or between your replication group and your application.

The in-transit encryption option in Amazon ElastiCache helps protect your data when it is moving from one location to another.

Evaluation

This control ensures that AWS ElastiCache Redis clusters are In-Transit encrypted.

Remediation

Encryption in transit option is only available while creating the ElastiCache Redis Cluster. So to modify the option we have to first backup the ElastiCache Redis Cluster and delete the Cluster. After this operation, we can restore the backup with encryption in transit option enabled.

Using AWS Console
To Backup the ElastiCache redis cluster

1. Open AWS ElastiCache console at https://console.aws.amazon.com/elasticache/home.
2. In the navigation pane, choose Redis.
3. Choose the box to the left of the name of the Redis cluster you want to back up.
4. Now in actions dropdown choose backup.

To Delete the ElastiCache Redis cluster

1. Open AWS ElastiCache console at https://console.aws.amazon.com/elasticache/home.
2. In the navigation pane, choose Redis.
3. Choose the box to the left of the name of the Redis cluster you want to delete.
4. Now in actions dropdown choose delete.

To Restore the Backup

1. Open AWS ElastiCache console at https://console.aws.amazon.com/elasticache/home.

2. In the navigation pane, choose Backups.
3. Select the Backup you want to restore.
4. Click on the restore button.
5. In the Restore Cluster window ensure Encryption in-transit option is checked.

Using AWS CLI

To Backup the ElastiCache redis cluster reffer https://docs.aws.amazon.com/AmazonElastiCache/latest/red-ug/backups-manual.html.

To restore the backup reffer https://docs.aws.amazon.com/AmazonElastiCache/latest/red-ug/backups-restoring.html.

To enable In-Transit Encryption on Redis reffer https://docs.aws.amazon.com/AmazonElastiCache/latest/red-ug/in-transit-encryption.html.

Reference

- https://docs.aws.amazon.com/AmazonElastiCache/latest/red-ug/backups-manual.html
- https://docs.aws.amazon.com/AmazonElastiCache/latest/red-ug/backups-restoring.html
- https://docs.aws.amazon.com/AmazonElastiCache/latest/red-ug/in-transit-encryption.html
- https://docs.aws.amazon.com/AmazonElastiCache/latest/red-ug/Clusters.Delete.html
- https://docs.aws.amazon.com/AmazonElastiCache/latest/red-ug/Replication.DeletingRepGroup.html
- https://docs.aws.amazon.com/AmazonElastiCache/latest/red-ug/Clusters.Create.CLI.html
- https://docs.aws.amazon.com/AmazonElastiCache/latest/red-ug/encryption.html

Control ID - 154: Ensure that AWS ElastiCache Redis clusters are Data At-Rest encrypted

Criticality: HIGH

Specification

Amazon ElastiCache is a web service that makes it easy to set up, manage, and scale a distributed in-memory data store or cache environment in the cloud. It provides a high-performance, scalable, and cost-effective caching solution.Amazon ElastiCache at-rest encryption is a feature that allows you to meet compliance requirement - encryption of data at-rest.

Rationale

Encrypting on-disk data can help prevent the exposure of data that falls into the wrong hands in case of any data breach.

Note: Enabling At-Rest encryption can have a performance impact during encryption operations.

Evaluation

This control ensures that AWS ElastiCache Redis clusters are At-Rest encrypted.

Remediation

Encryption at-rest option is only available while creating the ElastiCache Redis Cluster. So to modify the option we have to first backup the ElastiCache Redis Cluster and delete the Cluster. After this operation, we can restore the backup with Encryption at-rest option enabled.

Using AWS Console
To Backup the ElastiCache Redis cluster

1. Open AWS ElastiCache console at https://console.aws.amazon.com/elasticache/home.
2. In the navigation pane, choose Redis.
3. Choose the box to the left of the name of the Redis cluster you want to back up.
4. Now in actions dropdown choose backup.

To Delete the ElastiCache Redis cluster

1. Open AWS ElastiCache console at https://console.aws.amazon.com/elasticache/home.
2. In the navigation pane, choose Redis.
3. Choose the box to the left of the name of the Redis cluster you want to delete.
4. Now in actions dropdown choose delete.

To Restore the Backup

1. Open AWS ElastiCache console at https://console.aws.amazon.com/elasticache/home.
2. In the navigation pane, choose Backups.
3. Select the Backup you want to restore.
4. Click on the restore button.
5. In the Restore Cluster window ensure Encryption at-rest option is checked.

Using AWS CLI

To Backup the ElastiCache Redis cluster reffer https://docs.aws.amazon.com/AmazonElastiCache/latest/red-ug/backups-manual.html.

To restore the backup reffer https://docs.aws.amazon.com/AmazonElastiCache/latest/red-ug/backups-restoring.html.

To enable Encryption At-Rest on Redis reffer https://docs.aws.amazon.com/AmazonElastiCache/latest/red-ug/at-rest-encryption.html.

Reference

- https://docs.aws.amazon.com/AmazonElastiCache/latest/red-ug/backups-manual.html
- https://docs.aws.amazon.com/AmazonElastiCache/latest/red-ug/backups-restoring.html
- https://docs.aws.amazon.com/AmazonElastiCache/latest/red-ug/in-transit-encryption.html
- https://docs.aws.amazon.com/AmazonElastiCache/latest/red-ug/Clusters.Delete.html
- https://docs.aws.amazon.com/AmazonElastiCache/latest/red-ug/Replication.DeletingRepGroup.html
- https://docs.aws.amazon.com/AmazonElastiCache/latest/red-ug/Clusters.Create.CLI.html
- https://docs.aws.amazon.com/AmazonElastiCache/latest/red-ug/encryption.html

Control ID - 155: Ensure that AWS ElastiCache Redis clusters are Data At-Rest encrypted with CMK

Criticality: HIGH

Specification

Amazon ElastiCache is a web service that makes it easy to set up, manage, and scale a distributed in-memory data store or cache environment in the cloud. It provides a high-performance, scalable, and cost-effective caching solution.Amazon ElastiCache At-Rest encryption is an optional feature that allows you to increase the security of your data by encrypting on-disk data.

Rationale

Encrypting on-disk data prevent unauthorized access of data that falls into the wrong hands in case of any data breach.By using AWS KMS CMK for encryption gives more control over access to data you encrypt.AWS KMS CMK will helps to maintain control over who can use your customer master keys(CMK) and gain access to encrypted data.

Note: Enabling At-Rest encryption can have a performance impact during encryption operations.

Evaluation

This control ensures that AWS ElastiCache Redis clusters are At-Rest encrypted.

Remediation

Encryption at-rest option is only available while creating the ElasticCache Redis Cluster. So to modify the option we have to first backup the ElastiCache Redis Cluster and delete the Cluster. After this operation, we can restore the backup with Encryption At-Rest option enabled.

Using AWS Console
To Backup the ElastiCache Redis cluster

1. Open AWS ElastiCache console at https://console.aws.amazon.com/elasticache/home.
2. In the navigation pane, choose Redis.
3. Choose the box to the left of the name of the Redis cluster you want to back up.
4. Now in actions dropdown choose backup.

To Delete the ElastiCache redis cluster

1. Open AWS ElastiCache console at https://console.aws.amazon.com/elasticache/home.
2. In the navigation pane, choose Redis.
3. Choose the box to the left of the name of the Redis cluster you want to delete.
4. Now in actions dropdown choose delete.

To Restore the Backup

1. Open AWS ElastiCache console at https://console.aws.amazon.com/elasticache/home.
2. In the navigation pane, choose Backups.
3. Select the Backup you want to restore.
4. Click on the restore button.

5. In the Restore Cluster window ensure Encryption At-Rest option is checked and Customer Managed Customer Master Key option is used instead of Default.

Using AWS CLI

To Backup the ElastiCache Redis cluster reffer https://docs.aws.amazon.com/AmazonElastiCache/latest/red-ug/backups-manual.html.

To restore the backup reffer https://docs.aws.amazon.com/AmazonElastiCache/latest/red-ug/backups-restoring.html.

To enable Encryption At-Rest on Redis reffer https://docs.aws.amazon.com/AmazonElastiCache/latest/red-ug/at-rest-encryption.html.

Reference

- https://docs.aws.amazon.com/AmazonElastiCache/latest/red-ug/backups-manual.html
- https://docs.aws.amazon.com/AmazonElastiCache/latest/red-ug/backups-restoring.html
- https://docs.aws.amazon.com/AmazonElastiCache/latest/red-ug/in-transit-encryption.html
- https://docs.aws.amazon.com/AmazonElastiCache/latest/red-ug/Clusters.Delete.html
- https://docs.aws.amazon.com/AmazonElastiCache/latest/red-ug/Replication.DeletingRepGroup.html
- https://docs.aws.amazon.com/AmazonElastiCache/latest/red-ug/Clusters.Create.CLI.html
- https://docs.aws.amazon.com/AmazonElastiCache/latest/red-ug/encryption.html

Control ID - 156: Ensure node-to-node encryption feature is enabled for AWS Elasticsearch Service domains

Criticality: HIGH

Specification

AWS Elasticsearch Service (Amazon ES) is easy to deploy, operate, and scale Elasticsearch clusters in the AWS Cloud. Node-to-node encryption feature provided by Amazon ES provides an additional layer of security. Node-to-node encryption is not enabled by default on Amazon ES.

Rationale

Node-to-node encryption enables TLS 1.2 encryption for all communications between the nodes of the Elasticsearch Service clusters. This helps in preventing intercepting traffic between Elasticsearch Service cluster nodes.

NOTE: Node-to-node encryption is only supported by domains having Elasticsearch versions above 6.0

Evaluation

This control ensures node-to-node encryption is enabled on AWS Elasticsearch Service domains.

Remediation

NOTE: You cannot enable node-to-node configuration for the existing Elasticsearch Service domain. you need to create a new Elasticsearch Service domain with node-to-node encryption enabled and migrate data from the old domain.

To Migrate data from one ES to another ES in AWS:

1. Register the same manual snapshot repository on both source and destination by referring https://docs.aws.amazon.com/elasticsearch-service/latest/developerguide/es-managedomains-snapshots.html#es-managedomains-snapshot-registerdirectory.
2. Take a manual snapshot of the source Elasticsearch domain by referring https://docs.aws.amazon.com/elasticsearch-service/latest/developerguide/es-managedomains-snapshots.html#es-managedomains-snapshot-create.
3. Restore the snapshot to the destination domain by referring https://docs.aws.amazon.com/elasticsearch-service/latest/developerguide/es-managedomains-snapshots.html#es-managedomains-snapshot-restore.

Creating new Elasticsearch Service domain using AWS Console:

1. Go to the Elasticsearch Service dashboard by visiting https://console.aws.amazon.com/es/home.
2. Click **Create a new domain** button.
3. Fill in appropriate settings for the new Elasticsearch Service domain.
4. In the Encryption, section ensures **node-to-node encryption** option is checked

Creating new Elasticsearch Service domain using AWS CLI:

You can create Elasticsearch Service domain with appropriate parameter using below command:

```
aws es create-elasticsearch-domain --domain-name <value> --node-to-node-encryption-options Enabled=true
```

Reference

- https://docs.aws.amazon.com/elasticsearch-service/latest/developerguide/ntn.html

Control ID - 157: Ensure AWS Elasticsearch Service domains have enabled the support for publishing slow logs to AWS CloudWatch Logs

Criticality: LOW

Specification

AWS Elasticsearch Service (Amazon ES) is easy to deploy, operate, and scale Elasticsearch clusters in the AWS Cloud. Elasticsearch Service provides support for publishing slow logs to AWS Cloudwatch Logs.

Rationale

In case of any incident, slow logs can help in identifying patterns and anomalies of the Elasticsearch Service cluster.

Evaluation

This control ensures that Elasticsearch Service domains have enabled support for publishing slow logs to AWS Cloudwatch Logs.

Remediation

NOTE: Search slow logs and Index slow logs must be enabled for supporting publishing of slow logs.

Using AWS Console:

1. Go to the Elasticsearch Service dashboard by visitinghttps://console.aws.amazon.com/es/home.
2. In the navigation pane, under My domains, choose the domain that you want to update.
3. On the Logs tab under the Set up **Search slow logs** click **Setup**.
4. Create a **CloudWatch log group**, or choose an existing one.
5. Choose an appropriate access policy or create a new one using the JSON editor the console provides.
6. Click **Enable**.
7. On the Logs tab under Set up **Index slow logs** section click **Setup**.
8. Create a **CloudWatch log group**, or choose an existing one.
9. Choose an appropriate access policy or create a new one using the JSON editor the console provides.
10. Click **Enable**.

Using AWS CLI:

1. To create a log group execute the below command:

   ```
   aws logs create-log-group --log-group-name <log_group_name>
   ```

2. Find the created log group's ARN using below command:

   ```
   aws logs describe-log-groups --log-group-name <log_group_name>
   ```

3. Enable slow logs for Elasticsearch Service domain using below command:

   ```
   aws es update-elasticsearch-domain-config --domain-name <es_domain_name>
   --log-publishing-options
   SEARCH_SLOW_LOGS=CloudWatchLogsLogGroupArn=<arn_from_above_command>,Enab
   led=true,INDEX_SLOW_LOGS=CloudWatchLogsLogGroupArn=<arn_from_above_comma
   nd>,Enabled=true
   ```

Reference

- https://docs.aws.amazon.com/elasticsearch-service/latest/developerguide/es-createdomain-configure-slow-logs.html

Control ID - 158: Ensure AWS Elasticsearch Service domains are not publicly accessible

Criticality: HIGH

Specification

AWS Elasticsearch Service (Amazon ES) is easy to deploy, operate, and scale Elasticsearch clusters in the AWS Cloud. AWS Elasticsearch Service domain provides an option to create a domain in VPC(Virtual Private Cloud).

Rationale

AWS Elasticsearch Service domains that are within VPC have an additional layer of security as all traffic remains secure within the AWS Cloud.

Evaluation

This control ensures that AWS Elasticsearch Service domains are not publicly accessible.

Remediation

NOTE: Network configuration for the Elasticsearch Service domain can't be changed after the domain is created. To change the Network configuration of the domain create a new domain within VPC and migrate old domain data to a new domain.

To Migrate data from one ES to another ES in AWS:

1. Register the same manual snapshot repository on both source and destination by referringhttps://docs.aws.amazon.com/elasticsearch-service/latest/developerguide/es-managedomains-snapshots.html#es-managedomains-snapshot-registerdirectory.
2. Take a manual snapshot of the source Elasticsearch domain by referring https://docs.aws.amazon.com/elasticsearch-service/latest/developerguide/es-managedomains-snapshots.html#es-managedomains-snapshot-create.
3. Restore the snapshot to the destination domain by referring https://docs.aws.amazon.com/elasticsearch-service/latest/developerguide/es-managedomains-snapshots.html#es-managedomains-snapshot-restore.

Creating new Elasticsearch Service domain within VPC using Console:

1. Go to the Elasticsearch Service dashboard by visiting https://console.aws.amazon.com/es/home.
2. Click Create a new domain button.
3. Fill in appropriate settings for the new Elasticsearch Service domain.
4. In the Network configuration section ensure to select VPC access (Recommended) option.

Creating new Elasticsearch Service domain within VPC using CLI:

You can create Elasticsearch Service domain within VPC appropriate parameters using below command:

```
aws es create-elasticsearch-domain --domain-name <value> --vpc-options
SubnetIds=<subnet_id>,SecurityGroupIds=<security_group_id>
```

Reference

- https://docs.aws.amazon.com/cli/latest/reference/es/create-elasticsearch-domain.html

Control ID - 159: Ensure AWS Elasticsearch Service domains are using the latest version of Elasticsearch engine

Criticality: MEDIUM

Specification

AWS Elasticsearch Service (Amazon ES) is easy to deploy, operate, and scale Elasticsearch clusters in the AWS Cloud. AWS Elasticsearch Service domain gives the flexibility to choose the Elasticsearch engine version at the time of creation and upgrade option after creation.

Rationale

Using the latest version for the Elasticsearch engine ensures regular security patches.

Evaluation

This Control ensures that Elasticsearch Service domains are using the latest version of the Elasticsearch engine.

Remediation

NOTE: AWS Elasticsearch Service does not support in-place Elasticsearch upgrades For Elasticsearch engine versions 1.5 and 2.3. To upgrade these versions refer https://docs.aws.amazon.com/elasticsearch-service/latest/developerguide/es-version-migration.html.

In-place Domain upgrade for Elasticsearch Service domain using AWS Console:

1. Go to the Elasticsearch Service dashboard by visiting https://console.aws.amazon.com/es/home.
2. In the navigation pane, under My domains, choose the domain that you want to upgrade the engine version.
3. In the Actions, dropdown chooses the Upgrade domain option.
4. In the Version to upgrade to field choose the latest version from the dropdown.
5. Click Submit.

In-place Domain upgrade for Elasticsearch Service domain using AWS CLI:

Use below command to upgrade the Elasticsearch Service domain engine to the desired version:

```
aws es upgrade-elasticsearch-domain --domain-name <domain_name> --target-version <target_version>
```

Reference

- https://docs.aws.amazon.com/elasticsearch-service/latest/developerguide/es-version-migration.htm

Control ID - 162: Ensure AWS Route 53 Registered domain has Transfer lock enabled

Criticality: MEDIUM

Specification

Amazon Route 53 is a highly available and scalable cloud Domain Name System (DNS) web service. AWS Route 53 Registered domain have Transfer lock feature. Once the Transfer lock is enabled domain can not be transferred until the lock is disabled.

Rationale

Enabling Transfer lock for the route 53 registered domain may provide additional layer of protection against unauthorized domain transfers.

Evaluation

The control ensures that Route 53 Registered domains have transfer lock enabled.

Remediation

Using AWS Console:

1. Go to Route 53 console at https://console.aws.amazon.com/route53/.
2. In the navigation pane, choose Registered Domains.
3. Choose the name of the domain that you want to update.
4. For Transfer lock property click enable.

Using AWS CLI:

Execute the below command

```
aws route53domains enable-domain-transfer-lock --domain-name <domain_name>
```

Reference

- https://docs.aws.amazon.com/cli/latest/reference/route53domains/enable-domain-transfer-lock.html

Control ID - 163: Ensure AWS Route 53 Registered domain has Auto renew Enabled

Criticality: MEDIUM

Specification

Amazon Route 53 is a highly available and scalable cloud Domain Name System (DNS) web service. AWS Route 53 Registered domains have Auto renew feature which automatically renews domains 30 days before expiration.

Rationale

Configuring Auto renew eliminates the possibility of losing the domain.

Evaluation

This control ensures that AWS Route 53 Registered domains have Auto renew enabled.

Remediation

Using AWS Console:

1. Go to Route 53 console at https://console.aws.amazon.com/route53/.
2. In the navigation pane, choose Registered Domains.
3. Choose the name of the domain that you want to update.
4. For Auto renew property click enable.

Using AWS CLI:

Execute the below command:

```
aws route53domains enable-domain-auto-renew --domain-name <domain_name>
```

Reference

- https://docs.aws.amazon.com/cli/latest/reference/route53domains/enable-domain-auto-renew.html

Control ID - 164: Ensure AWS Route 53 Registered domain is not expired

Criticality: HIGH

Specification

Amazon Route 53 is a highly available and scalable cloud Domain Name System (DNS) web service. Expired Route 53 Registered domains can be renewed before late renewal time.

Rationale

Renewal of an expired domain adds an additional layer of protection against malicious entities to re-register this domain to gathering the insider data.

Evaluation

This control ensures that AWS Route 53 Registered domains are not expired.

Remediation

Using AWS Console:

If the domain is still listed in Registered domains then it is still in the late renewal period. Follow the below steps for extending the registration period:

1. Go to Route 53 console at https://console.aws.amazon.com/route53/.
2. In the navigation pane, choose Registered Domains.
3. Choose the name of the domain for which you want to extend the registration period.
4. In the Expires on field click extend.
5. In the Extend registration for list, choose the number of years that you want to extend the registration for.
6. Choose Extend domain registration.

For domains that are not in the late renewal period refer to https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/domain-restore-expired.html.

Reference

- https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/domain-restore-expired.html
- https://docs.aws.amazon.com/Route53/latest/DeveloperGuide/domain-renew.html

Control ID - 165: Ensure AWS Kinesis Data Firehose delivery stream with Direct PUT and other sources as source has Server-side encryption configured

Criticality: HIGH

Specification

It is recommended to have service-side encryption enabled for Amazon Kinesis Delivery Streams

Rationale

If you send data to your delivery stream using PutRecord or PutRecordBatch, or if you send the data using AWS IoT, Amazon CloudWatch Logs, or CloudWatch Events, you can turn on server-side encryption by using the StartDeliveryStreamEncryption operation.

Evaluation

This control ensure that AWS Kinesis Data Firehose delivery stream with Direct PUT and other sources as source has Server-side encryption configured

Remediation

Via AWS Console

1. Login to AWS Console
2. Go to Amazon Kinesis on the AWS Console
3. Goto each kinesis Data firehose delivery stream
4. Click on Encryption

5. Click Edit
6. Mark the box to Enable server-side encryption for source records in delivery stream
7. Click Save

Reference

- https://docs.aws.amazon.com/firehose/latest/dev/encryption.html

Control ID - 166: Ensure AWS Kinesis Data Firehose delivery stream with Kinesis Data stream as source has Server-side encryption configured

Criticality: HIGH

Specification

It is recommended to have service-side encryption enabled for Amazon Kinesis Delivery Streams.

Rationale

When you configure a Kinesis data stream as the data source of a Kinesis Data Firehose delivery stream, Kinesis Data Firehose no longer stores the data at rest. Instead, the data is stored in the data stream. Enabling the encryption on data stream will enable encryption on delivery stream too.

Evaluation

Ensure AWS Kinesis Data Firehose delivery stream with Kinesis Data stream as source has Server-side encryption configured.

Remediation

Via AWS Console

1. Login to AWS Console
2. Go to Amazon Kinesis on the AWS Console
3. For each kinesis Data firehose delivery stream click on source kinesis data stream
4. Click on Configuration
5. Navigate to Encryption
6. Click Edit
7. Mark the box to Enable server-side encryption for source records in delivery stream
8. Select Use Customer-managed CMK
9. Select the required key in the dropdown
10. Click Save

Reference

- https://docs.aws.amazon.com/firehose/latest/dev/encryption.html

Control ID - 166: Ensure AWS Kinesis Data Firehose delivery stream with Direct PUT and other sources as source has Server-side encryption configured with KMS Customer Managed Keys

Criticality: HIGH

Specification

It is recommended to have service-side encryption enabled for Amazon Kinesis Delivery Streams with customer-managed key.

Rationale

If you send data to your delivery stream using PutRecord or PutRecordBatch, or if you send the data using AWS IoT, Amazon CloudWatch Logs, or CloudWatch Events, you can turn on server-side encryption by using the StartDeliveryStreamEncryption operation.

Evaluation

Ensure AWS Kinesis Data Firehose delivery stream with Kinesis Data stream as source has Server-side encryption configured with customer-managed key

Remediation

Via AWS Console

1. Login to AWS Console
2. Go to Amazon Kinesis on the AWS Console
3. Goto each kinesis Data firehose delivery stream
4. Click on Encryption
5. Click Edit
6. Mark the box to Enable server-side encryption for source records in delivery stream
7. Select Use Customer-managed CMK
8. Select the required key in the dropdown
9. Click Save

Reference

- https://docs.aws.amazon.com/firehose/latest/dev/encryption.html

Control ID - 168: Ensure AWS Kinesis Data Firehose delivery stream with Kinesis Data stream as source has Server-side encryption configured with KMS Customer Managed Keys

Criticality: HIGH

Specification

It is recommended to have service-side encryption enabled for Amazon Kinesis Delivery Streams.

Rationale

When you configure a Kinesis data stream as the data source of a Kinesis Data Firehose delivery stream, Kinesis Data Firehose no longer stores the data at rest. Instead, the data is stored in the data stream. Enabling the encryption on data stream will enable encryption on delivery stream too.

Note: Server-side encryption should be enabled to get this control evaluated.

Evaluation

This controls ensures that AWS Kinesis Data Firehose delivery stream with Kinesis Data stream as source has Server-side encryption configured

Remediation

Via AWS Console

1. Login to AWS Console
2. Go to Amazon Kinesis on the AWS Console
3. For each kinesis Data firehose delivery stream click on source kinesis data stream
4. Click on Configuration
5. Navigate to Encryption
6. Click Edit
7. Mark the box to Enable server-side encryption for source records in delivery stream
8. Select Use Customer-managed CMK
9. Select the required key in the dropdown
10. Click Save

Reference

- https://docs.aws.amazon.com/firehose/latest/dev/encryption.html

Control ID - 174: Ensure that Customer managed KMS keys use external key material

Criticality: HIGH

Specification

A customer managed AWS KMS Symmetric key can either use AWS managed key material or external key material to encrypt the data. When using external key material, customer has to provide and manage the key material and has full control over the key.

Rationale

When using external key material, customer has full control over the life-cycle and the key material of the symmetric KMS key.

Evaluation

The control ensures that the symmetric customer managed KMS key uses external key material.

Remediation

The key material origin cannot be changed after creating a AWS KMS symmetric key. The key with AWS managed key material must be deleted and either a new symmetric key with custom key material must be created or an existing key with custom key material should be used.

Note: Deleting a KMS key will render the data encrypted using it unusable. Proper measures should be taken to verify that no required data is still encrypted with the key before key is deleted.

To create a new symmetric AWS KMS key with custom key material:

Note: Refer to Importing key material step 3: Encrypt the key material for steps to wrap the key material.

Using AWS Console:

1. Go to AWS KMS console at https://ap-south-1.console.aws.amazon.com/kms.
2. In the navigation pane on the left side of the console, choose `Customer managed keys`.
3. Click **`Create Key`** button.
4. Select **`Symmetric`** as key type.
5. Expand **`Advanced options`**, select **`External`**.
6. Check the box with note **`I understand the security, availability, and durability implications of using an imported key`**.
7. Click **`Next`**.
8. Add alias and tags as needed. Click **`Next`**.
9. Add administrators and key deletion permissions as needed. Click **`Next`**.
10. Add key usage permissions as needed. Click **`Next`**.
11. Review policy, click **`Finish`**.
12. Select wrapping algorithm and Download wrapping key and import token. Cick **Next**.
13. Upload the Wrapped key material and Import token.
14. Set the expiration options.
15. Click **`Upload key material`**.

Using AWS CLI:

1. To create a new CMK with no key material, use the below command:

```
aws kms create-key --origin EXTERNAL
```

Note: For more details on command, please refer to https://docs.aws.amazon.com/cli/latest/reference/kms/create-key.html

2. To create a new CMK with no key material, use the below command:

```
aws kms get-parameters-for-import --key-id <key_id> --wrapping-algorithm
<algo> --wrapping-key-spec <key_spec>
```

Note: For more details on command, please refer to https://docs.aws.amazon.com/cli/latest/reference/kms/get-parameters-for-import.html

3. To import wrapped Import key material, use the below command:

```
aws kms import-key-material --key-id <key_id> --encrypted-key-material
<key_file> --import-token <token_file> --expiration-model
<KEY_MATERIAL_EXPIRES|KEY_MATERIAL_DOES_NOT_EXPIRE>
```

Note: For more details on command, please refer to https://docs.aws.amazon.com/cli/latest/reference/kms/import-key-material.html

Reference

- https://docs.aws.amazon.com/kms/latest/developerguide/importing-keys.html
- https://docs.aws.amazon.com/kms/latest/developerguide/importing-keys-encrypt-key-material.html
- https://docs.aws.amazon.com/cli/latest/reference/kms/create-key.html
- https://docs.aws.amazon.com/cli/latest/reference/kms/get-parameters-for-import.html
- https://docs.aws.amazon.com/cli/latest/reference/kms/import-key-material.html

Control ID - 179: Ensure MFA is enabled in AWS Directory

Criticality: HIGH

Specification

Enabling multi-factor authentication (MFA) for your AWS Managed Microsoft AD directory allows you to increase security when your users specify their AD credentials to access Supported Amazon Enterprise applications.

When MFA is enabled, your users enter their username and password (first factor) as usual, and they must also enter an authentication code (the second factor) they obtain from your virtual or hardware MFA solution. These factors together provide additional security by preventing access to your Amazon Enterprise applications, unless users supply valid user credentials and a valid MFA code.

Rationale

Enabling multi-factor authentication (MFA) for your AWS Managed Microsoft AD directory allows you to increase security when your users specify their AD credentials to access Supported Amazon Enterprise applications.

When MFA is enabled, your users enter their username and password (first factor) as usual, and they must also enter an authentication code (the second factor) they obtain from your virtual or hardware MFA solution. These factors together provide additional security by preventing access to your Amazon Enterprise applications, unless users supply valid user credentials and a valid MFA code.

This control ensures MFA is enabled on AWS Directories.

Evaluation

This control ensures MFA is enabled on AWS Directories.

Remediation

Using AWS Console:

1. Go to AWS console at https://console.aws.amazon.com/
2. Select the region as the Directory region.

3. Go to **Directory Service** dashboard.
4. Within the Directory Service Console, in the left pane, click on Directories
5. Select the required Directory ID, and do one of the following:
    - If you have multiple Regions showing under **Multi-Region replication**, select the Region where you want to enable MFA, and then choose the **Networking & security** tab
    - If you do not have any Regions showing under Multi-Region replication, choose the **Networking & security** tab.
6. In the **Multi-factor authentication section**, choose **Actions**, and then choose **Enable**
7. On the **Enable multi-factor authentication (MFA) page**, provide the following values:
    - Display label
    - RADIUS server DNS name or IP addresses
    - Port
    - Shared secret code
    - Confirm shared secret code
    - Protocol
    - Server timeout (in seconds)
    - Max RADIUS request retries
8. Choose **Enable**

Using AWS CLI:

Save the configuration details required my AWS in radius-mfa-config.json, the file should look something like this:

```
{
   "RadiusServers": ["[Server IP Address OR DNS NAME]"],
   "RadiusPort": [PORT],
   "RadiusTimeout": [SERVER TIMEOUT IN SECONDS],
   "RadiusRetries": [MAX RADIUS REQUEST RETRIES],
   "SharedSecret": "[SHARED SECRET CODE]",
   "AuthenticationProtocol": "[PROTOCOL]",
   "DisplayLabel": "[DISPLAY LABEL]",
   "UseSameUsername": [true|false]
 }
```

To WorkDocs on Workspace Directory, use the below command:

```
aws ds enable-radius --region [REGION] --directory-id [DIRECTORY_ID] --radius-settings file://radius-mfa-config.json
```

Note: For more details on command, please refer to https://docs.aws.amazon.com/cli/latest/reference/ds/enable-radius.html

Reference

- https://docs.aws.amazon.com/directoryservice/latest/admin-guide/ms_ad_mfa.html

Control ID - 181: Ensure proper protocol is configured for Radius server in AWS Directory

Criticality: MEDIUM

Specification

Enabling multi-factor authentication (MFA) for your AWS Managed Microsoft AD directory allows you to increase security when your users specify their AD credentials to access Supported Amazon Enterprise applications.

When MFA is enabled, your users enter their username and password (first factor) as usual, and they must also enter an authentication code (the second factor) they obtain from your virtual or hardware MFA solution. These factors together provide additional security by preventing access to your Amazon Enterprise applications, unless users supply valid user credentials and a valid MFA code.

Rationale

Enabling multi-factor authentication (MFA) for your AWS Managed Microsoft AD directory allows you to increase security when your users specify their AD credentials to access Supported Amazon Enterprise applications.

When MFA is enabled, your users enter their username and password (first factor) as usual, and they must also enter an authentication code (the second factor) they obtain from your virtual or hardware MFA solution. These factors together provide additional security by preventing access to your Amazon Enterprise applications, unless users supply valid user credentials and a valid MFA code.

This control ensures MFA is enabled on AWS Directories.

Evaluation

This control ensure proper protocol is configured for Radius server in AWS Directory

Remediation

Using AWS Console:

1. Go to AWS console at https://console.aws.amazon.com/
2. Select the region as the Directory region.
3. Go to **Directory Service** dashboard.
4. Within the Directory Service Console, in the left pane, click on Directories
5. Select the required Directory ID, and do one of the following:
    - If you have multiple Regions showing under **Multi-Region replication**, select the Region where you want to enable MFA, and then choose the **Networking & security** tab
    - If you do not have any Regions showing under Multi-Region replication, choose the **Networking & security** tab.
6. In the **Multi-factor authentication section**, choose **Actions**, and then choose **Enable**
7. On the **Enable multi-factor authentication (MFA) page**, provide the following values:
    - Display label
    - RADIUS server DNS name or IP addresses
    - Port
    - Shared secret code
    - Confirm shared secret code
    - Protocol, Protocol should have the **MS-CHAPv2**
    - Server timeout (in seconds)
    - Max RADIUS request retries
8. Choose **Enable**

Using AWS CLI:

Save the configuration details required my AWS in radius-mfa-config.json, the file should look something like this:

```
{
   "RadiusServers": ["[Server IP Address OR DNS NAME]"],
   "RadiusPort": [PORT],
   "RadiusTimeout": [SERVER TIMEOUT IN SECONDS],
   "RadiusRetries": [MAX RADIUS REQUEST RETRIES],
   "SharedSecret": "[SHARED SECRET CODE]",
   "AuthenticationProtocol": "MS-CHAPv2",
   "DisplayLabel": "[DISPLAY LABEL]",
   "UseSameUsername": [true|false]
}
```

To WorkDocs on Workspace Directory, use the below command:

```
aws ds enable-radius --region [REGION] --directory-id [DIRECTORY_ID] --radius-settings file://radius-mfa-config.json
```

Note: For more details on command, please refer to https://docs.aws.amazon.com/cli/latest/reference/ds/enable-radius.html

Reference

- https://docs.aws.amazon.com/directoryservice/latest/admin-guide/ms_ad_mfa.html

Control ID - 182: Ensure SNS Topics do not Allow Everyone to Publish

Criticality: LOW

Specification

This control ensures that SNS topics do not allow "Everyone" to publish. SNS topic policy should not contain allow Everyone with Action:"SNS:Publish".

Rationale

AWS services, users that can publish to your SNS topics can be unrestricted. We have to make sure SNS topics are not available for everyone to protect it from unauthorized access.

Evaluation

This control ensures that SNS topics are not accessible to publish in public.

Remediation

Using AWS Console:

1. Go to AWS Console SNS dashboard at https://console.aws.amazon.com/sns/
2. In the left navigation panel select **Topics**, select effected **topic**.

3. Under **Access policy** section, set policy as below,

```
{
    "Version": "2008-10-17",
    "Id": "__default_policy_ID",
    "Statement": [
      {
        "Sid": "__default_statement_ID",
        "Effect": "Allow",
        "Principal": {
          "AWS": "*"
        },
        "Action": [
          "SNS:GetTopicAttributes",
          "SNS:SetTopicAttributes",
          "SNS:AddPermission",
          "SNS:RemovePermission",
          "SNS:DeleteTopic",
          "SNS:Subscribe",
          "SNS:ListSubscriptionsByTopic",
          "SNS:Publish",
          "SNS:Receive"
        ],
        "Resource": "[SNS_TOPIC_ARN]",
        "Condition": {
          "StringEquals": {
            "AWS:SourceOwner": "[Account_ID]"
          }
        }
      }
    ]
}
```

35. Click on **Save changes**.

Using AWS CLI:

```
aws sns set-topic-attributes --topic-arn [topic_arn] --attribute-name Policy --attribute-value file://[Policy.json]
```

Note: For more details on command, please refer to https://docs.aws.amazon.com/cli/latest/reference/sns/set-topic-attributes.html

Reference

- https://docs.aws.amazon.com/cli/latest/reference/sns/set-topic-attributes.html

Control ID - 183: Ensure SNS Topics do not Allow Everyone to Subscribe

Criticality: LOW

Specification

This control ensures that SNS topics do not allow "Everyone" to subscribe. SNS topic policy should not contain allow Everyone with Action:"SNS:Subscribe" and "SNS:Receive".

Rationale

AWS services, users that can subscribe to your SNS topics can be unrestricted. We have to make sure SNS topics are not available for everyone to protect it from unauthorized access.

Evaluation

This control ensures that SNS topics do not allow "Everyone" to subscribe.

Remediation

Using AWS Console:

1. Go to AWS Console SNS dashboard at https://console.aws.amazon.com/sns/
2. In the left navigation panel select **Topics**, select effected **topic**.
3. Under **Access policy** section, set policy as below,

```
{
    "Version": "2008-10-17",
    "Id": "__default_policy_ID",
    "Statement": [
      {
        "Sid": "__default_statement_ID",
        "Effect": "Allow",
        "Principal": {
          "AWS": "*"
        },
        "Action": [
          "SNS:GetTopicAttributes",
          "SNS:SetTopicAttributes",
          "SNS:AddPermission",
          "SNS:RemovePermission",
          "SNS:DeleteTopic",
          "SNS:Subscribe",
          "SNS:ListSubscriptionsByTopic",
          "SNS:Publish",
          "SNS:Receive"
        ],
        "Resource": "[SNS_TOPIC_ARN]",
        "Condition": {
          "StringEquals": {
            "AWS:SourceOwner": "[Account_ID]"
          }
        }
      }
    ]
}
```

35. Click on **Save changes**.

Using AWS CLI:

```
aws sns set-topic-attributes --topic-arn [topic_arn] --attribute-name Policy --attribute-value file://[Policy.json]
```

Note: For more details on command, please refer to https://docs.aws.amazon.com/cli/latest/reference/sns/set-topic-attributes.html

Reference

- https://docs.aws.amazon.com/cli/latest/reference/sns/set-topic-attributes.html

Control ID - 184: Ensure there are no Internet facing Application load balancers

Criticality: HIGH

Specification

Application load balancers need to be configured in right scheme for maintainig secure architecture.

Rationale

To maintain a secure load balancing architecture it is essential to Application Load balancers to be configured with right scheme. An internet-facing load balancer has a publicly resolvable DNS name, so it can route requests from clients over the internet to the EC2 instances that are registered with the load balancer. An internal load balancer routes requests to targets using private IP addresses.

Evaluation

This control ensures no Application Load Balancers is configured for internet facing scheme.

Remediation

1. Go to AWS Management EC2 dashboard at https://console.aws.amazon.com/ec2/
2. In the left navigation panel under **LOAD BALANCING**, select **Load Balancers**.
3. Click Create load balancer from the dashboard top menu, select Application Load Balancer then click Continue.
4. On the Step 1: Configure Load Balancer page, provide a unique name for your new AWS ALB then set the load balancer Scheme to internal. Configure the necessary listeners and availability zones then once all these are configured, use the Add tag button, available in the Tags section, to attach tags to your new ALB. Click Next: Configure Security Settings to continue the setup process.
5. On the Step 2: Configure Security Settings page, create the necessary HTTPS listener for your new ELB. If your Application Load Balancer is not using an HTTPS listener just skip this page and click Next: Configure Security Groups to continue the process.
6. On the Step 3: Configure Security Groups page, select Create a new security group and provide a name and a short description for the new security group. This security group should contain a rule that allows traffic to the port that you configured your ALB to use. Click Next: Configure Routing.
7. On the Step 4: Configure Routing page, choose an existing Target Group or set a new one based on your requirements. In the Health checks section, click Advanced health check settings and configure the new load balancer health checks. Click Next: Register Targets to continue the ALB setup process.
8. On the Step 5: Register Targets page, use the Add to registered button to attach the necessary backend instances to the internal ALB. The new ALB will start routing requests to the registered EC2 instances as soon as the setup process is complete and the instances pass the initial health checks. Click Next: Review.

9. On the Step 6: Review page, examine all the required configuration details then click Create to build your new internal Application Load Balancer (ALB).

Using AWS CLI:

Run create-load-balancer command to create a internal AWS Application Load balancer

```
aws elbv2 create-load-balancer --region <region> --name <load balancer name> --scheme internal --subnets <subnet id>  --security-groups <security group id>
```

Note: For more details on command, please refer to
https://docs.aws.amazon.com/cli/latest/reference/elbv2/create-load-balancer.html

Reference

- https://docs.aws.amazon.com/elasticloadbalancing/latest/classic/elb-internet-facing-load-balancers.html
- https://docs.aws.amazon.com/elasticloadbalancing/latest/application/create-application-load-balancer.html
- https://docs.aws.amazon.com/cli/latest/reference/elbv2/create-load-balancer.html

Control ID - 185: Ensure ALB using listener type HTTPS must have SSL Security Policy

Criticality: MEDIUM

Specification

Application load balancers are using latest security policy for SSL configuration.

Rationale

Using insecure and deprecated security policies for SSL negotiation configuration within Load Balancers will expose the connection between the client and the load balancer to various SSL/TLS vulnerabilities. Latest security policy secures SSL negotiation configuration in order to follow security best practices and protect their front-end connections.

Evaluation

This control ensures that HTTPS listeners have security policies configured for Application Load Balancer.

Remediation

1. Go to AWS Management EC2 dashboard at https://console.aws.amazon.com/ec2/
2. In the left navigation panel under **LOAD BALANCING**, select **Load Balancers**.
3. Select the Load Balancer that you want to reconfigure.
4. Choose the **Listeners** tab from the bottom panel.
5. Select the **Non HTTPS listener**, click the Actions dropdown button from the panel top menu and select Edit.
6. Under Protocol and Port select HTTPS, under Default actions choose desired action and target.
7. Under **Security Policy** section select desired **ELBSecurityPolicy** policy.

Using AWS CLI:

Run modify-listener command using the ARN of the HTTPS listener that you want to reconfigure as identifier to update its predefined security policy

```
aws elbv2 modify-listener --region <region> --listener-arn <listener arn> --
protocol HTTPS --port 443 --certificates CertificateArn=<Certificate arn> --
ssl-policy  <ELBSecurityPolicy name> --default-actions Type=<Action
Type>,TargetGroupArn=<Target Group Arn>
```

Note: For more details on command, please refer to
https://docs.aws.amazon.com/cli/latest/reference/elbv2/modify-listener.html

Reference

- https://docs.aws.amazon.com/elasticloadbalancing/latest/application/listener-update-certificates.html
- https://docs.aws.amazon.com/elasticloadbalancing/latest/application/create-https-listener.html

Control ID - 186: Ensure that ALB using listener type HTTP must be redirected to HTTPS

Criticality: HIGH

Specification

Application Load balancer listener needs to be configured to redirect HTTP traffic to HTTPS.

Rationale

Redirecting HTTP traffic to HTTPS within Load Balancer listener's configuration simplifies deployments while benefiting from the scale, the availability, and the reliability of Amazon Elastic Load Balancing. The ALB's capability to redirect HTTP requests to HTTPS allows to meet the goal of secure browsing and achieve better search ranking and high SSL/TLS score.

Evaluation

This control ensures that the Application load balancer listener has HTTP traffic redirected to HTTPS.

Remediation

1. Go to AWS Management EC2 dashboard at https://console.aws.amazon.com/ec2/
2. In the left navigation panel under **LOAD BALANCING**, select **Load Balancers**.
3. Select the Listeners tab to access the list of listeners configured for the load balancer.
4. Click on the View/edit rules link available in the Rules column for the HTTP listener.
5. On the Rules page, perform the following actions:
    1. Choose Edit rules tab, and click on the edit rule icon to modify the existing default rule in order to redirect all HTTP requests to HTTPS.
    2. Within the Edit Rule mode, under THEN, delete the existing condition.
    3. Choose Add action to add the new condition with the Redirect to action.
    4. In the Redirect to action configuration box, enter 443 for the HTTPS port and keep the defaults for the remaining options.

5. Click on the checkmark icon to save the configuration changes.
6. Choose Update to apply the changes to the selected load balancer listener rule.

Using AWS CLI:

Define the HTTP listener rule configuration that redirects HTTP traffic to HTTPS, name the configuration document to a JSON file redirect-config.json:

```
[
    {
        "Type": "redirect",
        "Order": 1,
        "RedirectConfig": {
            "Protocol": "HTTPS",
            "Host": "#{host}",
            "Query": "#{query}",
            "Path": "/#{path}",
            "Port": "443",
            "StatusCode": "HTTP_301"
        }
    }
]
```

Run modify-listener command to modify the rule configuration for the specified HTTP listener, using the configuration document defined at the previous step:

```
aws elbv2 modify-listener --region <region> --listener-arn <listener arn> --
default-actions file://redirect-config.json
```

Note: For more details on command, please refer to

- https://docs.aws.amazon.com/elasticloadbalancing/latest/application/listener-update-rules.html
- https://docs.aws.amazon.com/cli/latest/reference/elbv2/create-rule.html

Reference

- https://docs.aws.amazon.com/elasticloadbalancing/latest/application/listener-update-rules.html
- https://docs.aws.amazon.com/elasticloadbalancing/latest/application/create-listener.html
- https://docs.aws.amazon.com/elasticloadbalancing/latest/application/create-https-listener.html

Control ID - 187: Ensure that ALB listeners have HTTPS enabled Target Groups

Criticality: MEDIUM

Specification

Application Load balancer target group needs to be configured with HTTPS on port 443.

Rationale

Each target group is used to route requests to one or more registered targets. Each created listener rule is specified with a target group and conditions. When a rule condition is met, traffic is forwarded to the

corresponding target group. Multiple target groups can be created per different requirements. These target groups can be configured to perform health check for load balancer instances.

Evaluation

This control ensures that Application Load balancer target group is configured using HTTPS on port 443.

Remediation

1. Go to AWS Management EC2 dashboard at https://console.aws.amazon.com/ec2/
2. In the left navigation panel under **LOAD BALANCING**, select **Target Groups**.
3. Click on Create Target Group button and follow below steps
    1. Choose a target type from Instance, IP Address or Lambda Function as per requirement.
    2. Give the target group a name.
    3. Choose protocol as HTTPS and Port as 443.
    4. Select Protocol version and Health check protocol as desired.
    5. Click next button and add Instances as needed and click Create target group to submit.

Using AWS CLI:

Run create-target-group command to create target group with protocol HTTPS and port 443

```
aws elbv2 create-target-group --name <region> --protocol HTTPS --port 443 --
target-type instance --vpc-id <VPC ID>
```

Note: For more details on command, please refer to
https://docs.aws.amazon.com/cli/latest/reference/elbv2/create-target-group.html

Reference

- https://docs.aws.amazon.com/elasticloadbalancing/latest/application/load-balancer-target-groups.html
- https://docs.aws.amazon.com/elasticloadbalancing/latest/application/create-target-group.html

Control ID - 188: Ensure IncreaseVolumeSize is Disabled for Workspace directories in all regions

Criticality: LOW

Specification

You can increase the size of the root and user volumes for a WorkSpace, up to 2000 GB each. WorkSpace root and user volumes come in set groups that can't be changed.

You can expand the root and user volumes whether they are encrypted or unencrypted, and you can expand both volumes once in a 6-hour period. However, you can't increase the size of the root and user volumes at the same time.

Rationale

You can increase the size of the root and user volumes for a WorkSpace, up to 2000 GB each. WorkSpace root and user volumes come in set groups that can't be changed.

This control ensures Increase volume size is enabled for Workspaces in directory

Evaluation

This control ensures Increase volume size is enabled for Workspaces in directory

Remediation

Using AWS Console:

1. Go to AWS console at https://console.aws.amazon.com/
2. Select the region as the Directory region.
3. Go to **Workspaces** dashboard.
4. Within the Workspace Console, in the left pane, click on Directories
5. Select the required Directory, and click on **Actions** and select **Update Details**
6. Open the **User Self Service Permissions** Accordion
7. Enable **Increase volume size**
8. Click on Update and Exit

Using AWS CLI:

To Enable Restart WorkSpace from client on Workspace Directory, use the below command:

```
aws workspaces modify-selfservice-permissions --resource-id <directory-id> --
selfservice-permissions IncreaseVolumeSize="ENABLED"
```

Note: For more details on command, please refer to
https://docs.aws.amazon.com/cli/latest/reference/workspaces/modify-selfservice-permissions.html

Reference

- https://docs.aws.amazon.com/directoryservice/latest/admin-guide/ms_ad_best_practices.html

Control ID - 193: Ensure that NLB balancer listener is not using unencrypted protocol

Criticality: MEDIUM

Specification

Network load balancer listeners are not using unencrypted protocols.

Rationale

Using unencrypted protocols and no security policies for SSL negotiation configuration within Load Balancers will expose the connection between the client and the load balancer to various SSL/TLS vulnerabilities. Encrypted protocols paired with secure port and security policy will secure SSL negotiation configuration in order to follow security best practices and protect their front-end connections.

Evaluation

This control ensures that Network Load Balancer listeners are using encrypted protocols.

Remediation

1. Go to AWS Management EC2 dashboard at https://console.aws.amazon.com/ec2/
2. In the left navigation panel under **LOAD BALANCING**, select **Load Balancers**.
3. Select the Load Balancer that you want to reconfigure.
4. Choose the **Listeners** tab from the bottom panel.
5. Select the **Non TLS listener**, click the Actions dropdown button from the panel top menu and select Edit.
6. Under Protocol and Port select TLS, under Default actions choose desired action and target.
7. Under **Security Policy** section select desired **ELBSecurityPolicy** policy.

Using AWS CLI:

Run modify-listener command using the ARN of the TLS listener that you want to reconfigure as identifier to update its predefined security policy

```
aws elbv2 modify-listener --region <region> --listener-arn <listener arn> --
protocol TLS --port 443 --certificates CertificateArn=<Certificate arn> --ssl-
policy  <ELBSecurityPolicy name> --default-actions Type=<Action
Type>,TargetGroupArn=<Target Group Arn>
```

Note: For more details on command, please refer to
https://docs.aws.amazon.com/cli/latest/reference/elbv2/modify-listener.html

Reference

- https://docs.aws.amazon.com/elasticloadbalancing/latest/network/listener-update-certificates.html
- https://docs.aws.amazon.com/elasticloadbalancing/latest/network/create-tls-listener.html

Control ID - 194: Ensure that Classic Elastic load balancer is not internet facing

Criticality: HIGH

Specification

Elastic load balancers need to be configured in the right scheme for maintainig secure architecture.

Rationale

To maintain a secure load balancing architecture it is essential for Elastic Load balancers to be configured with the right scheme. An internet-facing load balancer has a publicly resolvable DNS name, so it can route requests from clients over the internet to the EC2 instances that are registered with the load balancer. An internal load balancer routes requests to targets using private IP addresses.

Evaluation

This control ensures no Elastic Load Balancers is configured for internet facing scheme.

Remediation

1. Go to AWS Management EC2 dashboard at https://console.aws.amazon.com/ec2/
2. In the left navigation panel under **LOAD BALANCING**, select **Load Balancers**.
3. Click Create load balancer from the dashboard top menu, select Classic Load Balancer then click create.
4. On Step 1: Define Load Balancer page, provide a unique name for your new AWS ELB and mark the option 'Create an internal load balancer'. Configure the necessary security groups,security settings, health check and add relevent tags.

Using AWS CLI:

Run create-load-balancer command to create a internal AWS ELB

```
aws elb create-load-balancer --region <region> --load-balancer-name <load
balancer name> --scheme internal --listeners
"Protocol=<protocol>,LoadBalancerPort=<port>,InstanceProtocol=<port>,InstanceP
ort=<port>" --subnets <subnet id> --security-groups <security group>
```

Note: For more details on command, please refer to
https://docs.aws.amazon.com/cli/latest/reference/elb/create-load-balancer.html

Reference

- https://docs.aws.amazon.com/elasticloadbalancing/latest/classic/elb-internet-facing-load-balancers.html
- https://docs.aws.amazon.com/cli/latest/reference/elb/create-load-balancer.html

Control ID - 195: Ensure Classic Elastic Load balancer must have SSL Security Policy

Criticality: HIGH

Specification

Elastic load balancers are using latest security policy for SSL configuration.

Rationale

Using insecure and deprecated security policies for SSL negotiation configuration within Load Balancers will expose the connection between the client and the load balancer to various SSL/TLS vulnerabilities. Latest security policy secures SSL negotiation configuration in order to follow security best practices and protect their front-end connections.

Evaluation

This control ensures that HTTPs/SSL listeners have security policies configured for Elastic Load Balancer.

Remediation

1. Go to AWS Management EC2 dashboard at https://console.aws.amazon.com/ec2/
2. In the left navigation panel under **LOAD BALANCING**, select **Load Balancers**.
3. Select the Load Balancer that you want to reconfigure.

4. Choose the **Listeners** tab from the bottom panel.
5. Click Edit, from this list of listeners choose **Non HTTPS/SSL listener**, from protocol dropdown panel select HTTPS or SSL protocol.
6. Click Change for Cipher and selcet appropriate Security Policy and click save.

Using AWS CLI:

AWS CLI can be used to create a new HTTPS/SSL Listener

```
aws elb create-load-balancer-listeners --load-balancer-name <Load balancer
name> --listeners Protocol=<HTTPS OR
SSL>,LoadBalancerPort=443,InstanceProtocol=<protocol>,InstancePort=<portn>,SSL
CertificateId=<Certificate arn>
```

Note: For more details on command, please refer to
https://docs.aws.amazon.com/elasticloadbalancing/latest/classic/elb-add-or-delete-listeners.html

Reference

- https://docs.aws.amazon.com/elasticloadbalancing/latest/classic/elb-listener-config.html
- https://docs.aws.amazon.com/elasticloadbalancing/latest/APIReference/API_ModifyListener.html

Control ID - 196: Ensure AWS VPC subnets have automatic public IP assignment disabled

Criticality: HIGH

Specification

AWS Subnets have an attribute that determines whether a network interface created in the subnet automatically receives a public IPv4 address. Therefore, when launching an instance into a subnet that has this attribute enabled, a public IP address is assigned to the primary network interface (eth0) that's created for the instance. A public IP address is mapped to the primary private IP address through network address translation (NAT).

Rationale

AWS Subnets have an attribute that determines whether a network interface created in the subnet automatically receives a public IPv4 address. Therefore, when launching an instance into a subnet that has this attribute enabled, a public IP address is assigned to the primary network interface (eth0) that's created for the instance. A public IP address is mapped to the primary private IP address through network address translation (NAT).

This control ensures AWS VPC subnets have automatic public IP assignment disabled

Evaluation

This control ensures AWS VPC subnets have automatic public IP assignment disabled

Remediation

Using AWS Console:

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/
2. In the navigation pane, choose **Subnets**
3. Select the identified Subnet and choose the option **Modify auto-assign IP settings** under the **Actions** dropdown
4. Disable the **Auto-Assign IPv4** option and save it

Using AWS CLI:

```
aws ec2 modify-subnet-attribute --subnet-id [RESOURCE_ID] --region [REGION]  --no-map-public-ip-on-launch
```

Note: For more details on command, please refer to https://docs.aws.amazon.com/cli/latest/reference/ec2/modify-subnet-attribute.html

Reference

- https://docs.aws.amazon.com/vpc/latest/userguide/vpc-ip-addressing.html

Control ID - 197: Ensure to encrypt the User Volumes and Root Volumes with the customer managed master keys for AWS WorkSpace

Criticality: HIGH

Specification

Workspaces is integrated with the AWS Key Management Service (AWS KMS). This allows you to encrypt storage volumes of WorkSpaces using customer master keys (CMKs). When you launch a WorkSpace, you can encrypt the root volume (for Microsoft Windows, the C drive; for Linux, /) and the user volume (for Windows, the D drive; for Linux, /home). Doing so ensures that the data stored at rest, disk I/O to the volume, and snapshots created from the volumes are all encrypted.

Rationale

Workspaces is integrated with the AWS Key Management Service (AWS KMS). This allows you to encrypt storage volumes of WorkSpaces using customer master keys (CMKs)

- You can't encrypt an existing WorkSpace. You must encrypt a WorkSpace when you launch it
- Disabling encryption for an encrypted WorkSpace is not currently supported
- WorkSpaces launched with root volume encryption enabled might take up to an hour to provision

This control ensures Root and User volumes are encrypted using a Customer Managed Key

Evaluation

This control ensures Root and User volumes are encrypted using a Customer Managed Key

Remediation

Note: Root/User Volume encryption cannot be changed once workspace has been created. You'll need to terminate the resource and create a new one.

Note: The Control would evaluate only for the Workspaces which have there Root and User Volumes Encrypted

Using AWS Console:

1. Go to AWS console at https://console.aws.amazon.com/
2. Select the region as the Directory region.
3. Go to **Workspaces** dashboard.
4. Within the Workspace Console, in the left pane, click on **Workspace**
5. Click on **Launch Workspaces**
6. For the **WorkSpaces Configuration** step, do the following:
    - Select the volumes to encrypt: **Root Volume** and **User Volume**
    - For **Encryption Key**, select a CMK that you created.
    - Click **Next Step**
7. Choose **Launch WorkSpaces**

Using AWS CLI:

To Create workspace with encrypted volumes, use the below command:

```
aws workspaces create-workspaces --workspaces DirectoryId=[DIRECTORY-ID],UserName=[USER-NAME],BundleId=[BUNDLE-ID],VolumeEncryptionKey=[KMS-KEY],UserVolumeEncryptionEnabled=true,RootVolumeEncryptionEnabled=true,WorkspaceProperties={RunningMode=[AUTO_STOP|ALWAYS_ON],RunningModeAutoStopTimeoutInMinutes=[TIME-OUT-VALUE],RootVolumeSizeGib=[SIZE-VALUE],UserVolumeSizeGib=[SIZE-VALUE],ComputeTypeName=[COMPUTE-TYPE-VALUE]}
```

Reference

- https://docs.aws.amazon.com/workspaces/latest/adminguide/encrypt-workspaces.html

Control ID - 198: Ensure Workspace directory must have a vpc endpoint so that the API traffic associated with the management of workspaces stays within the vpc

Criticality: HIGH

Specification

VPC endpoints enables private connections between your VPC and supported AWS services and VPC endpoint services powered by AWS PrivateLink. AWS PrivateLink is a technology that enables you to privately access services by using private IP addresses. Traffic between your VPC and the other service does not leave the Amazon network.
VPC endpoints are virtual devices. They are horizontally scaled, redundant, and highly available VPC components. They allow communication between instances in your VPC and services without imposing availability risks.

Rationale

VPC endpoints enables private connections between your VPC and supported AWS services and VPC endpoint services powered by AWS PrivateLink. AWS PrivateLink is a technology that enables you to privately access services by using private IP addresses. Traffic between your VPC and the other service does not leave the Amazon network.
VPC endpoints are virtual devices. They are horizontally scaled, redundant, and highly available VPC components. They allow communication between instances in your VPC and services without imposing availability risks.

This control ensures Workspace directory have a vpc endpoint so that the API traffic associated with the management of workspaces stays within the vpc

Evaluation

This control ensures Workspace directory have a vpc endpoint so that the API traffic associated with the management of workspaces stays within the vpc

Remediation

Using AWS Console:

1. Open the Amazon VPC console at https://console.aws.amazon.com/vpc/
2. In the navigation pane, choose **Endpoints** and then **Create Endpoint**
3. Under **Service category**, select **AWS services**
4. Under **Service Name**, search for **Workspaces**
5. The Service name required would look something like -

   ```
   com.amazonaws.[REGION].workspaces
   ```

6. Under **VPC**, select the ID for the VPC used by our Workspace's Directory
7. Choose appropriate settings for Subnet, DNS, Security Groups, Policy and Tags
8. Click on **Create Endpoint**

Using AWS CLI:

```
aws ec2 create-vpc-endpoint --vpc-id [VPC_ID] --vpc-endpoint-type Interface --
service-name [WORKSPACE_SERVICE_NAME]
```

Note: Before creating endpoint, please ensure DNS Hostnames and DNS Support is enabled for the given VPC

Note: For more details on command, please refer to
https://docs.aws.amazon.com/cli/latest/reference/ec2/create-vpc-endpoint.html

Reference

- https://docs.aws.amazon.com/vpc/latest/privatelink/vpc-endpoints.html

Control ID - 200: Ensure to log state machine execution history to CloudWatch Logs

Criticality: LOW

Specification

Standard Workflows record execution history in AWS Step Functions, although you can optionally configure logging to Amazon CloudWatch Logs.

Unlike Standard Workflows, Express Workflows don't record execution history in AWS Step Functions. To see execution history and results for an Express Workflow, you must configure logging to Amazon CloudWatch Logs. Publishing logs doesn't block or slow down executions

Rationale

Standard Workflows record execution history in AWS Step Functions, although you can optionally configure logging to Amazon CloudWatch Logs.

Unlike Standard Workflows, Express Workflows don't record execution history in AWS Step Functions. To see execution history and results for an Express Workflow, you must configure logging to Amazon CloudWatch Logs. Publishing logs doesn't block or slow down executions

Evaluation

This control ensures that Cloudwatch logging is enabled for step functions

Remediation

Using AWS Console:

1. Go to AWS Console step functions dashboard at https://console.aws.amazon.com/states/
2. In the left navigation panel, select **State machines**.
3. Select the state machine that you want to reconfigure.
4. Click on **Edit** buton.
5. Under **Logging** section, select one of the log levels from the dropdown.
6. under **CloudWatch log group**, select the log group to use for logging.
7. Click on **Save**.

Using AWS CLI:

```
aws stepfunctions update-state-machine --state-machine-arn [stateMachineArn] --logging-configuration file://[jsonFilePath]
```

Add below json to logging configuration file used :

```
{
   "level": "ALL",
   "includeExecutionData": true,
   "destinations": [
     {
       "cloudWatchLogsLogGroup": {
         "logGroupArn": "[CloudWatchLogGroupArn]"
       }
     }
   ]
 }
```

Note: For more details on command, please refer to https://docs.aws.amazon.com/cli/latest/reference/stepfunctions/update-state-machine.html

Reference

- https://docs.aws.amazon.com/step-functions/latest/dg/cw-logs.html

Control ID - 202: Ensure to update the Security Policy of the Network Load Balancer

Criticality: MEDIUM

Specification

Elastic Load Balancing uses a TLS negotiation configuration, known as a security policy, to negotiate TLS connections between a client and the load balancer. A security policy is a combination of protocols and ciphers, these help establish a secure connection between a client and a server and ensures that all data passed between the client and your load balancer is private.

Rationale

Elastic Load Balancing uses a TLS negotiation configuration, known as a security policy, to negotiate TLS connections between a client and the load balancer. A security policy is a combination of protocols and ciphers, these help establish a secure connection between a client and a server and ensures that all data passed between the client and your load balancer is private.

This control ensure the Security Policy of the Network Load Balancer is updated

Evaluation

This control ensure the Security Policy of the Network Load Balancer is updated

Remediation

Using AWS Console:

1. Sign in to the AWS Management Console and open the EC2 dashboard console at https://console.aws.amazon.com/ec2/
2. In the left navigation panel, under **Load Balancing** section, choose **Load Balancers**
3. Select the Network Load balancer you want to edit
4. Choose the **Listeners** tab from the dashboard bottom panel
5. Select the listener with ID **TLS : 443**, then click the **Edit** button
6. On **Listeners** page, select one of the following policies from the **Security policy** dropdown list
    - ELBSecurityPolicy-2016-08
    - ELBSecurityPolicy-TLS-1-1-2017-01
    - ELBSecurityPolicy-FS-2018-06
    - ELBSecurityPolicy-TLS-1-2-Ext-2018-06

Using AWS CLI:

To modify a Load Balancer's Listener Protocol, use the following AWS CLI command:

```
aws elbv2 modify-listener --region [REGION] --listener-arn [LISTENER_ARN] --
ssl-policy [POLICY_NAME]
```

Note: For more details on command, please refer to the following:
https://docs.aws.amazon.com/cli/latest/reference/elbv2/modify-listener.html

Reference

- https://docs.aws.amazon.com/elasticloadbalancing/latest/network/create-tls-listener.html

Control ID - 203: Ensure EBS Volume is encrypted by KMS using a customer managed Key (CMK)

Criticality: HIGH

Specification

An Amazon EBS volume is a durable, block-level storage device that you can attach to your instances. After you attach a volume to an instance, you can use it as you would use a physical hard drive. EBS volumes are flexible.

Rationale

AWS allows Encryption via Key generated by AWS or a Key generated by the Customer, the reason Customer Managed Key(CMK) are preferred because they allow the user to manage the Key Material.

Evaluation

This control ensures to encrypt the EBS volumes with customer managed master keys in all regions

Remediation

Note: The Volume has to be recreated to remediate this control

You can create new Volume using AWS Console:

1. Go to AWS console at https://console.aws.amazon.com/
2. Select the region as the Volume region.
3. Go to **EC2** dashboard.
4. Within the Ec2 Dashboard, Click on Volumes from the Resources Section
5. Click on **Create Volume** button.
6. Check the Encryption check box
7. In the Master Key selection, select the Customer Created KMS key
8. Proceed with remaining process of Volume Creation as usual

Using AWS CLI:

To create a EC2 Volume, use the below command:

```
aws ec2 create-volume --availability-zone <zone> --encrypted --kms-key-id
<key-id> --size <volume-size>
```

Note: For more details on command, please refer to
https://docs.aws.amazon.com/cli/latest/reference/ec2/create-volume.html

Reference

- https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ebs-volumes.html

Control ID - 204: Ensure AWS EBS Volume snapshots are encrypted with KMS using a customer managed Key (CMK)

Criticality: HIGH

Specification

You can back up the data on your Amazon EBS volumes to Amazon S3 by taking point-in-time snapshots.

Snapshots are incremental backups, which means that only the blocks on the device that have changed after your most recent snapshot are saved. This minimizes the time required to create the snapshot and saves on storage costs by not duplicating data.

Each snapshot contains all of the information that is needed to restore your data (from the moment when the snapshot was taken) to a new EBS volume.

Rationale

AWS allows Encryption via Key generated by AWS or a Key generated by the Customer, the reason Customer Managed Key(CMK) are preferred because they allow the user to manage the Key Material.

Evaluation

This control ensures to encrypt the EBS Snapshot with customer managed master keys in all regions

Remediation

Note: Encryption of EBS Snapshot depends of the EBS Volume, if the volume is not encrypted then the snapshot wouldn't be encrypted

You can create an encrypted EBS Volume using AWS Console:

1. Go to AWS console at https://console.aws.amazon.com/
2. Select the region as the Volume region.
3. Go to **EC2** dashboard.
4. Within the Ec2 Dashboard, Click on Volumes from the Resources Section
5. Click on **Create Volume** button.
6. Check the Encryption check box
7. In the Master Key selection, select the Customer Created KMS key
8. Proceed with remaining process of Volume Creation as usual

Using AWS CLI:

To create a EC2 Volume, use the below command:

```
aws ec2 create-volume --availability-zone <zone> --encrypted --kms-key-id
<key-id> --size <volume-size>
```

Note: For more details on command, please refer to
https://docs.aws.amazon.com/cli/latest/reference/ec2/create-volume.html

Reference

- https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EBSSnapshots.html

Control ID - 205: Ensure RestartWorkspace is Enabled for Directories in all regions

Criticality: LOW

Specification

If you are experiencing issues with your WorkSpace, you can restart (reboot) it. Restarting a WorkSpace disconnects you from your WorkSpace, so that it can be shut down and restarted.

Your user data, operating system, and system settings are not affected. This process takes several minutes to finish.

Rationale

If you are experiencing issues with your WorkSpace, you can restart (reboot) it. Restarting a WorkSpace disconnects you from your WorkSpace, so that it can be shut down and restarted.

To avoid losing changes, save any open documents and other application files before you restart your WorkSpace.

This control ensures Restart WorkSpace from client is enabled in directory.

Evaluation

This control ensures Restart WorkSpace from client is enabled

Remediation

Using AWS Console:

1. Go to AWS console at https://console.aws.amazon.com/
2. Select the region as the Directory region.
3. Go to **Workspaces** dashboard.
4. Within the Workspace Console, in the left pane, click on Directories
5. Select the required Directory, and click on **Actions** and select **Update Details**
6. Open the **User Self Service Permissions** Accordion
7. Enable **Restart WorkSpace from client**
8. Click on Update and Exit

Using AWS CLI:

To Enable Restart WorkSpace from client on Workspace Directory, use the below command:

```
aws workspaces modify-selfservice-permissions --resource-id <directory-id> --
selfservice-permissions RestartWorkspace="ENABLED"
```

Note: For more details on command, please refer to
https://docs.aws.amazon.com/cli/latest/reference/workspaces/modify-selfservice-permissions.html

Reference

- https://docs.aws.amazon.com/directoryservice/latest/admin-guide/ms_ad_best_practices.html

Control ID - 208: Ensure WorkDocs is not enabled in Workspace Directories

Criticality: MEDIUM

Specification

Amazon WorkDocs is a fully managed, secure content creation, storage, and collaboration service. With Amazon WorkDocs, you can easily create, edit, and share content, and because it's stored centrally on AWS, access it from anywhere on any device. Amazon WorkDocs makes it easy to collaborate with others, and lets you easily share content, provide rich feedback, and collaboratively edit documents.

Rationale

You can use Amazon WorkDocs to store, sync, and share your files. WorkDocs can automatically back up documents on your WorkSpace and sync documents to and from other devices such as a PC or Mac, so that you can access your data regardless of which desktop you are using.

Evaluation

This control ensures AWS WorkDocs is not enabled.

Remediation

Using AWS Console:

1. Go to AWS console at https://console.aws.amazon.com/
2. Select the region as the Directory region.
3. Go to **Directory Service** dashboard.
4. Within the Directory Service Console, in the left pane, click on Directories
5. Select the required Directory ID, and click the Application Management Tab
6. Under **AWS apps & services**, check **Amazon WorkDocs**.
7. If its enabled, you'll see a URL to the Application. Note that URL
8. Click on **Amazon WorkDocs** or You can search for it in the console's main search bar
9. This will open the WorkDocs console, The WorkDoc (with the URL we noted before) will listed there
10. Select the WorkDoc, Click on **Actions** and Click on **Delete WorkDoc Site**
11. (Optional) If you want to delete the Directory as well, the new window opened would have the option for that
12. The new window opened would ask you to confirm deletion by Typing DELETE in a given text box

Using AWS CLI:

To WorkDocs on Workspace Directory, use the below command:

```
aws workspaces modify-workspace-creation-properties --resource-id <directory-
id> --workspace-creation-properties EnableWorkDocs=false
```

Note: For more details on command, please refer to https://docs.aws.amazon.com/cli/latest/reference/workspaces/modify-workspace-creation-properties.html

Reference

- https://docs.aws.amazon.com/workdocs/latest/userguide/what_is.html

Control ID - 209: Ensure Access to Internet is not enabled in Workspace Directories

Criticality: MEDIUM

Specification

Enable this setting to assign a public IP to WorkSpaces in this directory by default.

This will allow outbound Internet access from your WorkSpaces when using an Internet Gateway in the Amazon VPC in which your WorkSpaces are located. You should leave this setting disabled if you are using a Network Address Translation (NAT) configuration for outbound Internet access from your VPC.

Any changes to this setting will apply to new WorkSpaces you launch or existing WorkSpaces that you rebuild.

Rationale

Internet Access to Directory could allow OS updates and Application Deployment regardless with the user is logged in.

Evaluation

This control ensures Internet Access is not enabled

Remediation

Using AWS Console:

1. Go to AWS console at https://console.aws.amazon.com/
2. Select the region as the Directory region.
3. Go to **Workspaces** dashboard.
4. Within the Workspace Console, in the left pane, click on Directories
5. Select the required Directory, and click on **Actions** and select **Update Details**
6. Open the **Access to Internet** Accordian, Click on **Disable**
7. Click on Update and Exit

Using AWS CLI:

To Disable Internet Access on Workspace Directory, use the below command:

```
aws workspaces modify-workspace-creation-properties --resource-id <directory-
id> --workspace-creation-properties EnableInternetAccess=false
```

Note: For more details on command, please refer to https://docs.aws.amazon.com/cli/latest/reference/workspaces/modify-workspace-creation-properties.html

Reference

- https://docs.aws.amazon.com/directoryservice/latest/admin-guide/ms_ad_best_practices.html

Control ID - 210: Ensure Local Administrator setting is not enabled in Workspace Directories

Criticality: MEDIUM

Specification

When a Directory is created, AWS creates a Admin Account which has permissions to perform actions like Add/delete users, create policies.

It is recommended to follow least privilege and assign admin roles any when required

Rationale

When a Directory is created, AWS creates a Admin Account which has permissions to perform actions like Add/delete users, create policies.

Unless a Administrator is assigned to the Directory, it is recommended to disable this account to prevent unauthorized access

This control ensures Local Administrator is not enabled in directory.

Evaluation

This control ensures Local Administrator is not enabled

Remediation

Using AWS Console:

1. Go to AWS console at https://console.aws.amazon.com/
2. Select the region as the Directory region.
3. Go to **Workspaces** dashboard.
4. Within the Workspace Console, in the left pane, click on Directories
5. Select the required Directory, and click on **Actions** and select **Update Details**
6. Open the **Local Administrator Setting** Accordion, Click on **Disable**
7. Click on Update and Exit

Using AWS CLI:

To Disable Local Administrator on Workspace Directory, use the below command:

```
aws workspaces modify-workspace-creation-properties --resource-id <directory-
id> --workspace-creation-properties UserEnabledAsLocalAdministrator=false
```

Note: For more details on command, please refer to https://docs.aws.amazon.com/cli/latest/reference/workspaces/modify-workspace-creation-properties.html

Reference

- https://docs.aws.amazon.com/directoryservice/latest/admin-guide/ms_ad_best_practices.html

Control ID - 211: Ensure Maintenance Mode is not enabled in Workspace Directories

Criticality: MEDIUM

Specification

AWS uses maintenance mode to schedule OS and Application updates to Directories

Enables maintenance on your AutoStop WorkSpaces.

If this setting is enabled, AutoStop WorkSpaces will be automatically started once every month to keep them current with the latest Windows updates.

Rationale

AWS runs automated maintenance on Directories between 00h00 to 04h00. During this time the directory is unavailable.

It is advised to switch to Manual Maintenance as it gives you control over the maintenance period according to you organization's working hours

This control ensures Maintenance Mode is not enabled in directory.

Evaluation

This control ensures Maintenance Mode is not enabled

Remediation

Using AWS Console:

1. Go to AWS console at https://console.aws.amazon.com/
2. Select the region as the Directory region.
3. Go to **Workspaces** dashboard.
4. Within the Workspace Console, in the left pane, click on Directories
5. Select the required Directory, and click on **Actions** and select **Update Details**
6. Open the **Maintenance Mode** Accordion, Click on **Disable**
7. Click on Update and Exit

Using AWS CLI:

To Disable Maintenance Mode on Workspace Directory, use the below command:

```
aws workspaces modify-workspace-creation-properties --resource-id <directory-
id> --workspace-creation-properties EnableMaintenanceMode=false
```

Note: For more details on command, please refer to https://docs.aws.amazon.com/cli/latest/reference/workspaces/modify-workspace-creation-properties.html

Reference

- https://docs.aws.amazon.com/directoryservice/latest/admin-guide/ms_ad_best_practices.html

Control ID - 212: Ensure Device Type Windows Access Control is not enabled in Workspace Directories

Criticality: MEDIUM

Specification

Before restricting WorkSpaces access to trusted devices, confirm that client certificates are deployed to all of your trusted devices.

Rationale

AWS provides a number of ways to access Directories, to ensure a user is able to access Directories via there Windows Device, it is advised to allow DeviceType Windows.

Evaluation

This control ensures Windows Devices are Allowed access to Workspace directory

Remediation

Using AWS Console:

1. Go to AWS console at https://console.aws.amazon.com/
2. Select the region as the Directory region.
3. Go to **Workspaces** dashboard.
4. Within the Workspace Console, in the left pane, click on Directories
5. Select the required Directory, and click on **Actions** and select **Update Details**
6. Open the **Access Control Options** Accordion
7. Enable the **Windows and MacOS** option
8. Click on Update and Exit

Using AWS CLI:

To Allow Windows devices on Workspace Directory, use the below command:

```
aws workspaces modify-workspace-access-properties --resource-id <directory-id>
--workspace-access-properties DeviceTypeWindows="ALLOW"
```

Note: For more details on command, please refer to https://docs.aws.amazon.com/cli/latest/reference/workspaces/modify-workspace-access-properties.html

Reference

- https://docs.aws.amazon.com/directoryservice/latest/admin-guide/ms_ad_best_practices.html

Control ID - 213: Ensure Device Type MacOS Access Control is not enabled in Workspace Directories

Criticality: MEDIUM

Specification

Before restricting WorkSpaces access to trusted devices, confirm that client certificates are deployed to all of your trusted devices.

Rationale

AWS provides a number of ways to access Directories, to ensure a user is able to access Directories via there MacOS Device, it is advised to allow DeviceType MacOS.

Evaluation

This control ensures MacOS Devices are Allowed access to directory

Remediation

Using AWS Console:

1. Go to AWS console at https://console.aws.amazon.com/
2. Select the region as the Directory region.
3. Go to **Workspaces** dashboard.
4. Within the Workspace Console, in the left pane, click on Directories
5. Select the required Directory, and click on **Actions** and select **Update Details**
6. Open the **Access Control Options** Accordion
7. Enable the **Windows and MacOS** option
8. Click on Update and Exit

Using AWS CLI:

To Allow MacOS devices on Workspace Directory, use the below command:

```
aws workspaces modify-workspace-access-properties --resource-id <directory-id>
--workspace-access-properties DeviceTypeOsx="ALLOW"
```

Note: For more details on command, please refer to https://docs.aws.amazon.com/cli/latest/reference/workspaces/modify-workspace-access-properties.html

Reference

- https://docs.aws.amazon.com/directoryservice/latest/admin-guide/ms_ad_best_practices.html

Control ID - 214: Ensure Device Type Web Access Control is not enabled in Workspace Directories

Criticality: MEDIUM

Specification

Web Access allows to user to connect to the directory via a Web Browser. This provides an ease of access and avoiding setting up any pre-requisite applications

Rationale

AWS provides a number of ways to access Directories, to ensure a user is able to access Directories via there Web browser, it is advised to allow DeviceType Web.

Evaluation

`This control ensures Web Access is Allowed access in directory

Remediation

Using AWS Console:

1. Go to AWS console at https://console.aws.amazon.com/
2. Select the region as the Directory region.
3. Go to **Workspaces** dashboard.
4. Within the Workspace Console, in the left pane, click on Directories
5. Select the required Directory, and click on **Actions** and select **Update Details**
6. Open the **Access Control Options** Accordion
7. Enable the **Other Platforms** option
8. Also, ensure the **Web Access** option is checked
9. Click on Update and Exit

Using AWS CLI:

To Allow Windows devices on Workspace Directory, use the below command:

```
aws workspaces modify-workspace-access-properties --resource-id <directory-id>
--workspace-access-properties DeviceTypeWeb="ALLOW"
```

Note: For more details on command, please refer to https://docs.aws.amazon.com/cli/latest/reference/workspaces/modify-workspace-access-properties.html#modify-workspace-access-properties

Reference

- https://docs.aws.amazon.com/directoryservice/latest/admin-guide/ms_ad_best_practices.html

Control ID - 215: Ensure Device Type iOS Access Control is not enabled in Workspace Directories

Criticality: MEDIUM

Specification

This allows to users to connect to the directory via a iOS Device. This provides an ease of access and allows users to access directories on the go

Rationale

AWS provides a number of ways to access Directories, to ensure a user is able to access Directories via there iOS Device, it is advised to allow DeviceType iOS.

Evaluation

This control ensures Access from iOS devices is Allowed access to directory

Remediation

Using AWS Console:

1. Go to AWS console at https://console.aws.amazon.com/
2. Select the region as the Directory region.
3. Go to **Workspaces** dashboard.
4. Within the Workspace Console, in the left pane, click on Directories
5. Select the required Directory, and click on **Actions** and select **Update Details**
6. Open the **Access Control Options** Accordion
7. Enable the **Other Platforms** option
8. Also, ensure the **iOS** option is checked
9. Click on Update and Exit

Using AWS CLI:

To Allow Windows devices on Workspace Directory, use the below command:

```
aws workspaces modify-workspace-access-properties --resource-id <directory-id>
--workspace-access-properties DeviceTypeIos="ALLOW"
```

Note: For more details on command, please refer to https://docs.aws.amazon.com/cli/latest/reference/workspaces/modify-workspace-access-properties.html#modify-workspace-access-properties

Reference

- https://docs.aws.amazon.com/directoryservice/latest/admin-guide/ms_ad_best_practices.html

Control ID - 216: Ensure Device Type Android Access Control is not enabled in Workspace Directories

Criticality: MEDIUM

Specification

This allows to users to connect to the directory via a Android Device. This provides an ease of access and allows users to access directories on the go

Rationale

AWS provides a number of ways to access Directories, to ensure a user is able to access Directories via there Android Device, it is advised to allow Device Type Android.

This control ensures Access from Android devices is Allowed in directory.

Evaluation

TThis control ensures Access from Android devices is Allowed

Remediation

Using AWS Console:

1. Go to AWS console at https://console.aws.amazon.com/
2. Select the region as the Directory region.
3. Go to **Workspaces** dashboard.
4. Within the Workspace Console, in the left pane, click on Directories
5. Select the required Directory, and click on **Actions** and select **Update Details**
6. Open the **Access Control Options** Accordion
7. Enable the **Other Platforms** option
8. Also, ensure the **Android** option is checked
9. Click on Update and Exit

Using AWS CLI:

To Allow Windows devices on Workspace Directory, use the below command:

```
aws workspaces modify-workspace-access-properties --resource-id <directory-id>
--workspace-access-properties DeviceTypeAndroid="ALLOW"
```

Note: For more details on command, please refer to https://docs.aws.amazon.com/cli/latest/reference/workspaces/modify-workspace-access-properties.html#modify-workspace-access-properties

Reference

- https://docs.aws.amazon.com/directoryservice/latest/admin-guide/ms_ad_best_practices.html

Control ID - 217: Ensure Device Type ChromeOS Access Control is not enabled in Workspace Directories

Criticality: MEDIUM

Specification

This allows to users to connect to the directory via a ChromeOS Device. This provides an ease of access and allows users to access directories on the go

Rationale

AWS provides a number of ways to access Directories, to ensure a user is able to access Directories via there ChromeOS Device, it is advised to allow DeviceType ChromeOS.

Evaluation

This control ensures Access from ChromeOS devices is Allowed to access directory

Remediation

Using AWS Console:

1. Go to AWS console at https://console.aws.amazon.com/
2. Select the region as the Directory region.
3. Go to **Workspaces** dashboard.
4. Within the Workspace Console, in the left pane, click on Directories
5. Select the required Directory, and click on **Actions** and select **Update Details**
6. Open the **Access Control Options** Accordion
7. Enable the **Other Platforms** option
8. Also, ensure the **ChromeOS** option is checked
9. Click on Update and Exit

Using AWS CLI:

To Allow Windows devices on Workspace Directory, use the below command:

```
aws workspaces modify-workspace-access-properties --resource-id <directory-id>
--workspace-access-properties DeviceTypeChromeOs="ALLOW"
```

Note: For more details on command, please refer to https://docs.aws.amazon.com/cli/latest/reference/workspaces/modify-workspace-access-properties.html#modify-workspace-access-properties

Reference

- https://docs.aws.amazon.com/directoryservice/latest/admin-guide/ms_ad_best_practices.html

Control ID - 218: Ensure Device Type ZeroClient Access Control is not enabled in Workspace Directories

Criticality: MEDIUM

Specification

This allows to users to connect to the directory via a Zero Client Device. This provides an ease of access and allows users to access directories on devices like kiosk and dumb terminals

Rationale

AWS provides a number of ways to access Directories, to ensure a user is able to access Directories via there Zero Client, it is advised to allow DeviceType Zero Client.

This control ensures Access from Android devices is Allowed in directory.

Evaluation

This control ensures Access from Zero Client devices is Allowed

Remediation

Using AWS Console:

1. Go to AWS console at https://console.aws.amazon.com/
2. Select the region as the Directory region.
3. Go to **Workspaces** dashboard.
4. Within the Workspace Console, in the left pane, click on Directories
5. Select the required Directory, and click on **Actions** and select **Update Details**
6. Open the **Access Control Options** Accordion
7. Enable the **Other Platforms** option
8. Also, ensure the **Zero Clients** option is checked
9. Click on Update and Exit

Using AWS CLI:

To Allow Windows devices on Workspace Directory, use the below command:

```
aws workspaces modify-workspace-access-properties --resource-id <directory-id>
--workspace-access-properties DeviceTypeZeroClient="ALLOW"
```

Note: For more details on command, please refer to https://docs.aws.amazon.com/cli/latest/reference/workspaces/modify-workspace-access-properties.html#modify-workspace-access-properties

Reference

- https://docs.aws.amazon.com/directoryservice/latest/admin-guide/ms_ad_best_practices.html

Control ID - 221: Ensure ChangeComputeType is Disabled in all regions for Workspace Directories

Criticality: LOW

Specification

You can switch a WorkSpace between the Value, Standard, Performance, Power, and PowerPro bundles.

When you request a bundle change, Workspace is rebooted using the new bundle. Workspace would preserve the operating system, applications, data, and storage settings for the WorkSpace.

Rationale

You can request a larger bundle once in a 1-hour period or a smaller bundle once every 30 days. For a newly launched WorkSpace, you must wait 1 hour before requesting a larger bundle.

This control ensures Change Compute Type is disabled for Workspaces in directory

Evaluation

This control ensures Change Compute Type is disabled for directory

Remediation

Using AWS Console:

1. Go to AWS console at https://console.aws.amazon.com/
2. Select the region as the Directory region.
3. Go to **Workspaces** dashboard.
4. Within the Workspace Console, in the left pane, click on **Directories**
5. Select the required Directory, and click on **Actions** and select **Update Details**
6. Open the **User Self Service Permissions** Accordion
7. Disable **Change Compute Type**
8. Click on Update and Exit

Using AWS CLI:

To Disable Change Compute Type on Workspace Directory, use the below command:

```
aws workspaces modify-selfservice-permissions --resource-id <directory-id> --
selfservice-permissions ChangeComputeType="DISABLED"
```

Note: For more details on command, please refer to
https://docs.aws.amazon.com/cli/latest/reference/workspaces/modify-selfservice-permissions.html

Reference

- https://docs.aws.amazon.com/directoryservice/latest/admin-guide/ms_ad_best_practices.html

Control ID - 222: Ensure SwitchRunningMode is Disabled in all regions for Workspace Directories

Criticality: LOW

Specification

The running mode of a WorkSpace directory determines its immediate availability and how you pay for it (monthly or hourly)

AlwaysOn: Use when paying a fixed monthly fee for unlimited usage of your WorkSpaces. This mode is best for users who use their WorkSpace full time as their primary desktop.

AutoStop: Use when paying for your WorkSpaces by the hour. With this mode, your WorkSpaces stop after a specified period of disconnection, and the state of apps and data is saved.

Rationale

Keeping the ability to changing Running mode disabled, allows the Admins to have complete control over workspace running modes. This works as a check for any workspace left On/Off causing blockers in critical business application

This control ensures Switch Running Mode is disabled for Workspaces in directory

Evaluation

This control ensures Switch Running Mode is disabled for directory

Remediation

Using AWS Console:

1. Go to AWS console at https://console.aws.amazon.com/
2. Select the region as the Directory region.
3. Go to **Workspaces** dashboard.
4. Within the Workspace Console, in the left pane, click on **Directories**
5. Select the required Directory, and click on **Actions** and select **Update Details**
6. Open the **User Self Service Permissions** Accordion
7. Disable **Switch Running Mode**
8. Click on Update and Exit

Using AWS CLI:

To Disable Switch Running Mode on Workspace Directory, use the below command:

```
aws workspaces modify-selfservice-permissions --resource-id <directory-id> --
selfservice-permissions SwitchRunningMode="DISABLED"
```

Note: For more details on command, please refer to
https://docs.aws.amazon.com/cli/latest/reference/workspaces/modify-selfservice-permissions.html

Reference

- https://docs.aws.amazon.com/directoryservice/latest/admin-guide/ms_ad_best_practices.html

Control ID - 223: Ensure RebuildWorkspace is Disabled in all regions for Workspace Directories

Criticality: MEDIUM

Specification

This recreates the root volume, the user volume, and the primary elastic network interface.

The root volume is refreshed with the most recent image of the bundle that the WorkSpace was created from. The user volume is recreated from the most recent snapshot. The current contents of the user volume are overwritten. The primary elastic network interface is recreated. The WorkSpace receives a new private IP address

Rationale

Keeping the ability to rebuilding workspace disabled, help avoid accidental rebuilds and allows proper pre-requisite troubleshooting to be performed

You can rebuild a WorkSpace only if the following conditions are met:
The WorkSpace must have a state of AVAILABLE, ERROR, UNHEALTHY, STOPPED, or REBOOTING. To rebuild a WorkSpace in the REBOOTING state, you must use the RebuildWorkspaces API operation or the rebuild-workspaces AWS CLI command.
A snapshot of the user volume must exist.

This control ensures Rebuild WorkSpace Mode is disabled for directory

Evaluation

This control ensures Rebuild WorkSpace Mode is disabled for directory

Remediation

Using AWS Console:

1. Go to AWS console at https://console.aws.amazon.com/
2. Select the region as the Directory region.
3. Go to **Workspaces** dashboard.
4. Within the Workspace Console, in the left pane, click on **Directories**
5. Select the required Directory, and click on **Actions** and select **Update Details**
6. Open the **User Self Service Permissions** Accordion
7. Disable **Rebuild Workspace**
8. Click on Update and Exit

Using AWS CLI:

To Disable Rebuild Workspace on Workspace Directory, use the below command:

```
aws workspaces modify-selfservice-permissions --resource-id <directory-id> --
selfservice-permissions SwitchRunningMode="DISABLED"
```

Note: For more details on command, please refer to

- https://docs.aws.amazon.com/cli/latest/reference/workspaces/modify-selfservice-permissions.html

Reference

- https://docs.aws.amazon.com/directoryservice/latest/admin-guide/ms_ad_best_practices.html

Control ID - 224: Ensure only AD Connector directory type is allowed for AWS Directories

Criticality: MEDIUM

Specification

AD Connector is a directory gateway with which you can redirect directory requests to your on-premises Microsoft Active Directory without caching any information in the cloud. AD Connector comes in two sizes, small and large. You can spread application loads across multiple AD Connectors to scale to your performance needs. There are no enforced user or connection limits.

Rationale

AD Connector is a directory gateway with which you can redirect directory requests to your on-premises Microsoft Active Directory without caching any information in the cloud.

AD Connector cannot be shared with other AWS accounts. If this is a requirement, consider using AWS Managed Microsoft AD to Share your directory. AD Connector is also not multi-VPC aware, which means that AWS applications like Workspaces are required to be provisioned into the same VPC as your AD Connector.

This control ensures only AD Connector are allowed for directory

Evaluation

This control ensures only AD Connector are allowed for directory

Remediation

Note:AD Connector Directories cannot be create via CLI and once a directory is created its type cannot be changed

Using AWS Console:

1. Go to AWS console at https://console.aws.amazon.com/
2. Select the region as the Directory region.
3. Go to **Workspaces** dashboard.
4. Within the Workspace Console, in the left pane, click on **Directories**
5. Click on **Set up Directory**
6. From Select directory type page, select **AD Connector**
7. Now you proceed with the directory creation by entering the VPC and AD details

Reference

- https://docs.aws.amazon.com/directoryservice/latest/admin-guide/directory_ad_connector.html

Control ID - 225: Ensure to enable the encryption of the Root volumes for Workspaces in all regions

Criticality: HIGH

Specification

Workspaces is integrated with the AWS Key Management Service (AWS KMS). This allows you to encrypt storage volumes of WorkSpaces using customer master keys (CMKs). When you launch a WorkSpace, you can encrypt the root volume (for Microsoft Windows, the C drive; for Linux, /) and the user volume (for Windows, the D drive; for Linux, /home). Doing so ensures that the data stored at rest, disk I/O to the volume, and snapshots created from the volumes are all encrypted.

Rationale

Workspaces is integrated with the AWS Key Management Service (AWS KMS). This allows you to encrypt storage volumes of WorkSpaces using customer master keys (CMKs)

- You can't encrypt an existing WorkSpace. You must encrypt a WorkSpace when you launch it
- Disabling encryption for an encrypted WorkSpace is not currently supported
- WorkSpaces launched with root volume encryption enabled might take up to an hour to provision

This control ensures encryption for Root volumes is enabled

Evaluation

This control ensures encryption for Root volumes is enabled

Remediation

Note: Root Volume encryption cannot be changed once workspace has been created. You'll need to terminate the resource and create a new one.

Using AWS Console:

1. Go to AWS console at https://console.aws.amazon.com/
2. Select the region as the Directory region.
3. Go to **Workspaces** dashboard.
4. Within the Workspace Console, in the left pane, click on **Workspace**
5. Click on **Launch Workspaces**
6. For the **WorkSpaces Configuration** step, do the following:
    - Select the volumes to encrypt: **Root Volume**
    - For **Encryption Key**, select an AWS KMS CMK, either the AWS managed CMK created by Amazon Workspaces or a CMK that you created. The CMK that you select must be symmetric. Amazon Workspaces does not support asymmetric CMKs
    - Click **Next Step**
7. Choose **Launch WorkSpaces**

Reference

- https://docs.aws.amazon.com/workspaces/latest/adminguide/encrypt-workspaces.html

Control ID - 226: Ensure to enable the encryption of the User volumes for Workspaces in all regions

Criticality: HIGH

Specification

Workspaces is integrated with the AWS Key Management Service (AWS KMS). This allows you to encrypt storage volumes of WorkSpaces using customer master keys (CMKs). When you launch a WorkSpace, you can encrypt the root volume (for Microsoft Windows, the C drive; for Linux, /) and the user volume (for Windows, the D drive; for Linux, /home). Doing so ensures that the data stored at rest, disk I/O to the volume, and snapshots created from the volumes are all encrypted.

Rationale

Workspaces is integrated with the AWS Key Management Service (AWS KMS). This allows you to encrypt storage volumes of WorkSpaces using customer master keys (CMKs)

- You can't encrypt an existing WorkSpace. You must encrypt a WorkSpace when you launch it
- Disabling encryption for an encrypted WorkSpace is not currently supported
- WorkSpaces launched with root volume encryption enabled might take up to an hour to provision

This control ensures encryption for User volumes is enabled

Evaluation

This control ensures encryption for User volumes is enabled

Remediation

Note: User Volume encryption cannot be changed once workspace has been created. You'll need to terminate the resource and create a new one.

Using AWS Console:

1. Go to AWS console at https://console.aws.amazon.com/
2. Select the region as the Directory region.
3. Go to **Workspaces** dashboard.
4. Within the Workspace Console, in the left pane, click on **Workspace**
5. Click on **Launch Workspaces**
6. For the **WorkSpaces Configuration** step, do the following:
    - Select the volumes to encrypt: **User Volume**
    - For **Encryption Key**, select an AWS KMS CMK, either the AWS managed CMK created by Amazon Workspaces or a CMK that you created. The CMK that you select must be symmetric. Amazon Workspaces does not support asymmetric CMKs
    - Click **Next Step**
7. Choose **Launch WorkSpaces**

Reference

- https://docs.aws.amazon.com/workspaces/latest/adminguide/encrypt-workspaces.html

Control ID - 227: Ensure Amazon API Gateway APIs are only accessible through private API endpoints in all regions

Criticality: HIGH

Specification

Using Amazon API Gateway, you can create private REST APIs that can only be accessed from your virtual private cloud in Amazon VPC by using an interface VPC endpoint. This is an endpoint network interface that you create in your VPC.

Rationale

Using Amazon API Gateway, you can create private REST APIs that can only be accessed from your virtual private cloud in Amazon VPC by using an interface VPC endpoint. This is an endpoint network interface that you create in your VPC.

To restrict access to your private API to specific VPCs and VPC endpoints, you must add aws:SourceVpc and aws:SourceVpce conditions to your API's resource policy.

API gateway apis are only accessible through Private Api endpoints

Evaluation

API gateway apis are only accessible through private api endpoints

Remediation

Using AWS Console:

To create an interface VPC endpoint for API Gateway execute-api

1. Sign in to the AWS Management Console and open the Amazon VPC console at https://console.aws.amazon.com/vpc/
2. In the navigation pane, choose **Endpoints** ,**Create Endpoint**
3. For **Service category**, ensure that **AWS services** is selected
4. For **Service Name**, choose the API Gateway service endpoint, including the AWS Region that you want to connect to. This is in the form **com.amazonaws.region.execute-api**
5. Complete the following information:
    - For **VPC**, choose the VPC that you want to create the endpoint in
    - For **Subnets**, choose the subnets (Availability Zones) in which to create the endpoint network interfaces
    - For **Enable Private DNS Name**, leave the check box selected. Private DNS is enabled by default
    - For **Security group**, select the security group to associate with the VPC endpoint network interfaces
6. Choose **Create endpoint**

Create a private API using the API Gateway console

1. Sign in to the API Gateway console and choose **+ Create API**
2. Under **Create new API**, choose the **New API** option
3. Type a name for **API name**
4. For **Endpoint Type**, choose **Private**
5. Choose **Create API**

Using AWS CLI:

To create a Private API for API gateway, use the below command:

```
aws apigateway create-rest-api --name '<api-name>' --description '<api-
descriptoin>' --region <api-region> --endpoint-configuration '{ "types":
["PRIVATE"] }'
```

Note: For more details on command, please refer to

- https://docs.aws.amazon.com/apigateway/latest/developerguide/apigateway-private-apis.html#apigateway-private-api-create-interface-vpc-endpoint

Reference

- https://aws.amazon.com/blogs/compute/introducing-amazon-api-gateway-private-endpoints/

Control ID - 228: Ensure to disable default route table association for Transit Gateways in all regions

Criticality: HIGH

Specification

A transit gateway acts as a Regional virtual router for traffic flowing between your virtual private clouds (VPCs) and on-premises networks. A transit gateway scales elastically based on the volume of network traffic.

Default route table association: Automatically associate attachments with the default route table.

Rationale

A transit gateway acts as a Regional virtual router for traffic flowing between your virtual private clouds (VPCs) and on-premises networks. A transit gateway scales elastically based on the volume of network traffic.

Default route table association: Automatically associate attachments with the default route table.

This control ensures default route table association is disabled for Transit Gateways

Evaluation

This control ensures default route table association is disabled for Transit Gateways

Remediation

Using AWS Console:

1. Go to AWS console at https://console.aws.amazon.com/
2. Select the region as the Gateway region.
3. Go to **VPC** dashboard.
4. Within the VPC Console, in the left pane, click on **Transit Gateway**
5. Select the required Gateway, and click on **Actions** and select **Modify**
6. Uncheck **Default route table association**
7. Click on **Modify Transit Gateway**

Using AWS CLI:

To Disable Default route table association on Transit gateway, use the below command:

```
aws ec2 modify-transit-gateway --transit-gateway-id <gateway-id> --options
DefaultRouteTableAssociation=disable
```

Note: For more details on command, please refer to

- https://docs.aws.amazon.com/cli/latest/reference/ec2/modify-transit-gateway.html

Reference

- https://docs.aws.amazon.com/vpc/latest/tgw/what-is-transit-gateway.html

Control ID - 229: Ensure to disable default route table propagation for Transit Gateways in all regions

Criticality: HIGH

Specification

A transit gateway acts as a Regional virtual router for traffic flowing between your virtual private clouds (VPCs) and on-premises networks. A transit gateway scales elastically based on the volume of network traffic

Default route table propagation - Automatically propagate routes from attachments into the default route table. This will allow anything connected to the default route table to route to each other (assuming the VPC route table is also configured)

Rationale

A transit gateway acts as a Regional virtual router for traffic flowing between your virtual private clouds (VPCs) and on-premises networks. A transit gateway scales elastically based on the volume of network traffic.

Default route table propagation - Automatically propagate routes from attachments into the default route table. This will allow anything connected to the default route table to route to each other (assuming the VPC route table is also configured)

This control ensures default route table propagation is disabled for Transit Gateways

Evaluation

This control ensures default route table propagation is disabled for Transit Gateways

Remediation

Using AWS Console:

1. Go to AWS console at https://console.aws.amazon.com/
2. Select the region as the Gateway region.
3. Go to **VPC** dashboard.
4. Within the VPC Console, in the left pane, click on **Transit Gateway**
5. Select the required Gateway, and click on **Actions** and select **Modify**
6. Uncheck **Default route table propagation**
7. Click on **Modify Transit Gateway**

Using AWS CLI:

To Disable Default route table propagation on Transit gateway, use the below command:

```
aws ec2 modify-transit-gateway --transit-gateway-id <gateway-id> --options
DefaultRouteTablePropagation=disable
```

Note: For more details on command, please refer to

- https://docs.aws.amazon.com/cli/latest/reference/ec2/modify-transit-gateway.html

Reference

- https://docs.aws.amazon.com/vpc/latest/tgw/what-is-transit-gateway.html

Control ID - 230: Ensure to enable config for the all resources for Config Service

Criticality: HIGH

Specification

AWS Config enables you to assess, audit, and evaluate the configurations of your AWS resources. Config continuously monitors and records your AWS resource configurations and allows you to automate the evaluation of recorded configurations against desired configurations. With Config, you can review changes in configurations and relationships between AWS resources, dive into detailed resource configuration histories, and determine your overall compliance against the configurations specified in your internal guidelines. This enables you to simplify compliance auditing, security analysis, change management, and operational troubleshooting.

Rationale

With AWS Config, you're able to continuously monitor and record configuration changes of your AWS resources. Config also enables you to inventory your AWS resources, the configurations of your AWS resources, as well as software configurations within EC2 instances at any point in time.

AWS Config allows you to continuously audit and assess the overall compliance of your AWS resource configurations with your organization's policies and guidelines

This control ensures config for the all resources is enabled for Config Service

Evaluation

This control ensures config for the all resources is enabled for Config Service

Remediation

Note: This Control evaluates Config service for all regions collectively, i.e the control will fail if any region fails. If any region is disabled, this would be counted as a region failure.

Note: This control evaluates the Pass/Fail scenario by checking whether required settings is configured or not. Whether the Config Recorder is turned on is not considered. Please check CID-23 to ensure Config Service is enabled in all regions.

Using AWS Console:

1. Go to AWS Config console at https://console.aws.amazon.com/config/
2. Select the region as the Config region.
3. Within the Config Console, in the left pane, click on **Settings**
4. On the Settings Page ,Click on **Edit**
5. Under **General settings**, select **Record all resources supported in this region**
6. Click on **Save**

Using AWS CLI:

To configure aggregator for Config Service, use the below command:

```
aws configservice put-configuration-recorder --configuration-recorder
name=[config-name],roleARN=[iam-role-arn]  --recording-group allSupported=true
```

Note: For more details on command, please refer to

- https://docs.aws.amazon.com/cli/latest/reference/configservice/put-configuration-recorder.html

Reference

- https://docs.aws.amazon.com/config/latest/developerguide/gs-console.html
- https://docs.aws.amazon.com/config/latest/developerguide/select-resources.html

Control ID - 231: Ensure to enable config for the global resources like IAM for Config Service

Criticality: HIGH

Specification

AWS Config enables you to assess, audit, and evaluate the configurations of your AWS resources. Config continuously monitors and records your AWS resource configurations and allows you to automate the evaluation of recorded configurations against desired configurations. With Config, you can review changes in configurations and relationships between AWS resources, dive into detailed resource configuration histories, and determine your overall compliance against the configurations specified in your internal guidelines. This enables you to simplify compliance auditing, security analysis, change management, and operational troubleshooting.

Rationale

With AWS Config, you're able to continuously monitor and record configuration changes of your AWS resources. Config also enables you to inventory your AWS resources, the configurations of your AWS resources, as well as software configurations within EC2 instances at any point in time.

AWS Config allows you to continuously audit and assess the overall compliance of your AWS resource configurations with your organization’s policies and guidelines

This control ensures config for the global resources like IAM is enabled for Config Service

Evaluation

This control ensures config for the global resources like IAM is enabled for Config Service

Remediation

Note: This Control evaluates Config service for all regions collectively, i.e the control will fail if any region fails. If any region is disabled, this would be counted as a region failure.

Note: This control evaluates the Pass/Fail scenario by checking whether required settings is configured or not. Whether the Config Recorder is turned on is not considered. Please check CID-23 to ensure Config Service is enabled in all regions.

Using AWS Console:

1. Go to AWS Config console at https://console.aws.amazon.com/config/
2. Select the region as the Config region.
3. Within the Config Console, in the left pane, click on **Settings**
4. On the Settings Page ,Click on **Edit**
5. Under **General settings**, select **Record all resources supported in this region**
6. Check **Include global resources (e.g., AWS IAM resources)**
7. Click on **Save**

Using AWS CLI:

To configure aggregator for Config Service, use the below command:

```
aws configservice put-configuration-recorder --configuration-recorder
name=[config-name],roleARN=[iam-role-arn]  --recording-group
allSupported=true,includeGlobalResourceTypes=true
```

Note: For more details on command, please refer to

- https://docs.aws.amazon.com/cli/latest/reference/configservice/put-configuration-recorder.html

Reference

- https://docs.aws.amazon.com/config/latest/developerguide/gs-console.html
- https://docs.aws.amazon.com/config/latest/developerguide/select-resources.html

Control ID - 232: Ensure to configure data retention period for the configuration items for Config Service

Criticality: HIGH

Specification

AWS Config is a service that enables you to assess, audit, and evaluate the configurations of your AWS resources. Config continuously monitors and records your AWS resource configurations and allows you to automate the evaluation of recorded configurations against desired configurations. With Config, you can review changes in configurations and relationships between AWS resources, dive into detailed resource configuration histories, and determine your overall compliance against the configurations specified in your internal guidelines. This enables you to simplify compliance auditing, security analysis, change management, and operational troubleshooting.

Rationale

Amazon Config allows you to delete your data by specifying a retention period for your Configuration Items. When you specify a retention period, Amazon Config retains your Configuration Items for that specified period. You can choose a period between a minimum of 30 days and a maximum of 7 years (2557 days). Amazon Config deletes data older than your specified retention period. If you do not specify a retention period, Amazon Config continues to store ConfigurationItems for the default period of 7 years (2557 days)

When recording is switched on, the current state of the resource is when a Configuration Item is recorded and until the next change (a new ConfigurationItem) is recorded.

This control ensures to configure data retention period of 7 years for the configuration items for Config Service

Evaluation

This control ensures to configure data retention period of 7 years for the configuration items for Config Service

Remediation

Note: This Control evaluates Config service for all regions collectively, i.e the control will fail if any region fails. If any region is disabled, this would be counted as a region failure.

Note: This control evaluates the Pass/Fail scenario by checking whether required settings is configured or not. Whether the Config Recorder is turned on is not considered. Please check CID-23 to ensure Config Service is enabled in all regions.

Using AWS Console:

1. Go to AWS Config console at https://console.aws.amazon.com/config/
2. Select the region as the Config region.
3. Within the Config Console, in the left pane, click on **Settings**
4. On the Settings Page ,Click on **Edit**
5. Under **General settings**, select **Retain AWS Config data for 7 years (2557 days)**
6. Click on **Save**

Using AWS CLI:

To configure data retention period on Config Service, use the below command:

```
aws configservice put-retention-configuration --retention-period-in-days 2557
```

Note: For more details on command, please refer to

- https://docs.aws.amazon.com/cli/latest/reference/configservice/put-retention-configuration.html

Reference

- https://docs.aws.amazon.com/config/latest/developerguide/gs-console.html
- https://docs.amazonaws.cn/en_us/config/latest/developerguide/delete-config-data-with-retention-period.html

Control ID - 233: Ensure to configure s3 buckets which contains details for the resources that Config records

Criticality: HIGH

Specification

AWS Config enables you to assess, audit, and evaluate the configurations of your AWS resources. Config continuously monitors and records your AWS resource configurations and allows you to automate the evaluation of recorded configurations against desired configurations. With Config, you can review changes in configurations and relationships between AWS resources, dive into detailed resource configuration histories, and determine your overall compliance against the configurations specified in your internal guidelines. This enables you to simplify compliance auditing, security analysis, change management, and operational troubleshooting.

Rationale

AWS Config delivers configuration items of the AWS resources that it is recording to the Amazon S3 bucket that you specified when you configured your delivery channel.

This control ensures to configure s3 buckets which contains details for the resources that Config records for Config Service

Evaluation

This control ensures to configure s3 buckets which contains details for the resources that Config records for Config Service

Remediation

Note: This Control evaluates Config service for all regions collectively, i.e the control will fail if any region fails. If any region is disabled, this would be counted as a region failure.

Note: This control evaluates the Pass/Fail scenario by checking whether required settings is configured or not. Whether the Config Recorder is turned on is not considered. Please check CID-23 to ensure Config Service is enabled in all regions.

Using AWS Console:

1. Go to AWS Config console at https://console.aws.amazon.com/config/
2. Select the region as the Config region.
3. Within the Config Console, in the left pane, click on **Settings**
4. On the Settings Page ,Click on **Edit**
5. Under **Delivery method**, select appropriate option from **Amazon S3 bucket**
6. Enter bucket details in **S3 bucket name**
7. Click on **Save**

Using AWS CLI:

To configure S3 bucket for delivery on Config Service, use the below command:

```
aws configservice put-delivery-channel --delivery-channel name=[channel-
name],s3BucketName=[bucket-name]
```

Note:CLI Command requires s3BucketName of an existing S3 bucket

Note: For more details on command, please refer to

- https://docs.aws.amazon.com/cli/latest/reference/configservice/put-delivery-channel.html

Reference

- https://docs.aws.amazon.com/config/latest/developerguide/gs-console.html
- https://docs.aws.amazon.com/config/latest/developerguide/s3-bucket-policy.html

Control ID - 234: Ensure to configure certificate provider type to custom in EMR security configuration

Criticality: MEDIUM

Specification

AWS EMR cluster, data will be shared between different applications in the cluster which if not encrypted can be vulnerable.

Rationale

AWS EMR cluster, data will be shared between different applications in the cluster which if not encrypted can be vulnerable. Data InTransit should be encrypted with a certicate from custom certificate provider.

Certificates from other providers cannot be used for encryption for safety concerns.

Evaluation

This control ensures that in transit data in an EMR Cluster will always be encrypted with custom(our own) certificate provider.

Remediation

Using AWS Console:

Note : We cannot modify an existing EMR cluster, new EMR cluster should be created with required configuration.

To configure a security configuration with custom certificate provider for In Transit data encryption, follow the below procedure:

1. Sign in to AWS Console and navigate to https://console.aws.amazon.com/elasticmapreduce/home.
2. In the Navigation pane, choose **Security configurations**.
3. Click on **Create** button.
4. Check **Data in transit encryption** checkbox, select **Custom** option in **Certificate provider type** dropdown
5. In **Custom key provider location** text box, provide location of jar file of the **custom provider certificate**.
6. Enter **Certificate provider class** name in the given location.
7. Click **Create** button.

To Create and Configure an EMR cluster with Secuirty Configuration having InTransit Data Encryption Enabled.

1. Sign in to AWS Console and navigate to https://console.aws.amazon.com/elasticmapreduce/home.
2. In the Navigation pane, choose **Clusters**.
3. Click on **Create cluster** button.
4. Select **Go to advanced options**.
5. In steps 1 - 3, configure options according to the requirements.
6. In **step 4**, select **Security Configuration**.
7. Select above created security configuration with InTransit data encryption enabled.
8. Click **Create cluster** button.

Using AWS CLI:

```
       aws emr create-security-configuration --name MySecurityConfig --
security-configuration '{
        "EncryptionConfiguration": {
                "EnableInTransitEncryption" : true,
                "EnableAtRestEncryption" : true,
                "InTransitEncryptionConfiguration" : {
                        "TLSCertificateConfiguration" : {
                                "CertificateProviderType" : "Custom",
                                "S3Object" : <S3BucketCertificateLocation>
                        }
                }
            }
        }'
```

For command usage refer: https://docs.aws.amazon.com/cli/latest/reference/emr/create-security-configuration.html

Reference

- https://docs.aws.amazon.com/emr/latest/ManagementGuide/emr-data-encryption-options.html

Control ID - 235: Ensure to enable data in transit encryption for EMR security configuration

Criticality: HIGH

Specification

AWS EMR cluster, data will be shared between different applications in the cluster which if not encrypted can be vulnerable.

Rationale

AWS EMR cluster, data will be shared between different applications in the cluster which if not encrypted can be vulnerable. Data InTransit should be encrypted.

Evaluation

This control ensures that data in transit in an EMR Cluster will always be encrypted.

Remediation

Using AWS Console:

Note : We cannot modify an existing EMR cluster, new EMR cluster should be created with required configuration.

To configure a security configuration with In Transit data encryption, follow the below procedure:

1. Sign in to AWS Console and navigate to https://console.aws.amazon.com/elasticmapreduce/home.
2. In the Navigation pane, choose **Security configurations**.
3. Click on **Create** button.
4. Check **Data in transit encryption** checkbox, select either**PEM/Custom** option in **Certificate provider type** dropdown.
5. In **Custom key provider location** text box, provide S3 location of Zip file of the **certificate**.
6. Click **Create** button.

To Create and Configure an EMR cluster with Secuirty Configuration having InTransit Data Encryption Enabled.

1. Sign in to AWS Console and navigate to https://console.aws.amazon.com/elasticmapreduce/home.
2. In the Navigation pane, choose **Clusters**.
3. Click on **Create cluster** button.
4. Select **Go to advanced options**.
5. In steps 1 - 3, configure options according to the requirements.

6. In **step 4**, select **Security Configuration**.
7. Select above created security configuration with InTransit data encryption enabled.
8. Click **Create cluster** button.

Using AWS CLI:

```
       aws emr create-security-configuration --name MySecurityConfig --
security-configuration '{
        "EncryptionConfiguration": {
                "EnableInTransitEncryption" : true,
                "EnableAtRestEncryption" : true,
                "InTransitEncryptionConfiguration" : {
                        "TLSCertificateConfiguration" : {
                                "CertificateProviderType" : "PEM/Custom",
                                "S3Object" : <S3BucketCertificateLocation>
                        }
                }
            }
        }'
```

For command usage refer: https://docs.aws.amazon.com/cli/latest/reference/emr/create-security-configuration.html

Reference

- https://docs.aws.amazon.com/emr/latest/ManagementGuide/emr-data-encryption-options.html

Control ID - 236: Ensure that all AWS Systems Manager (SSM) parameters are encrypted

Criticality: MEDIUM

Specification

Parameter Store of AWS Systems Manager, provides secure, hierarchical storage for configuration data management and secrets management. It allows you to store data such as passwords, database strings, Amazon Machine Image (AMI) IDs, and license codes as parameter values. You can store values as plain text or encrypted data.

Parameter Store provides support for three types of parameters: String, StringList, and SecureString.

Rationale

Parameter Store of AWS Systems Manager, provides secure, hierarchical storage for configuration data management and secrets management. It allows you to store data such as passwords, database strings, Amazon Machine Image (AMI) IDs, and license codes as parameter values. You can store values as plain text or encrypted data.

Parameter Store provides support for three types of parameters: String, StringList, and SecureString.

This control ensure that all AWS Systems Manager (SSM) parameters are encrypted

Evaluation

This control ensure that all AWS Systems Manager (SSM) parameters are encrypted

Remediation

Note: You cannot change the parameter type after it has been created. You'll need to recreate the parameter.

Using AWS Console:

1. Sign in to AWS Console and navigate to https://console.aws.amazon.com/systems-manager/
2. Under **Application Manager**, select **Parameter Store** and click on **Create Parameter**
3. Enter values for **Name**, **Description** and **Value** boxes
4. For **Type**, select **SecureString**
5. Select the required KMS key via the **KMS Key ID** dropdown list
6. Click on **Create Parameter**

Using AWS CLI:

```
aws ssm put-parameter --region [REGION] --name "[PARAMETER_NAME]" --type
"SecureString" --value "[PARAMETER_VALUE]" --key-id "[KMS_KEY_ID]"
```

For command usage refer: https://docs.aws.amazon.com/cli/latest/reference/ssm/put-parameter.html

Reference

- https://docs.aws.amazon.com/systems-manager/latest/userguide/systems-manager-parameter-store.html

Control ID - 237: Ensure termination protection is enabled for EMR cluster

Criticality: HIGH

Specification

When you create a cluster using Amazon EMR, you can choose to create a transient cluster that auto-terminates after steps complete, or you can create a long-running cluster that continues to run until you terminate it deliberately. When termination protection is enabled on an EMR cluster, it ensures that EC2 instances are not shut down by an accident or error. Termination protection is especially useful if your cluster might have data stored on local disks that you need to recover before the instances are terminated.

Rationale

When termination protection is enabled on an EMR cluster, it ensures that EC2 instances are not shut down by an accident or error. Termination protection is especially useful if your cluster might have data stored on local disks that you need to recover before the instances are terminated.

An Amazon EMR cluster with termination protection enabled has the disableAPITermination attribute set for all Amazon EC2 instances in the cluster.

Evaluation

This control ensures that termination protection is enabled for EMR clusters.

Remediation

Using AWS Console:

To enable termination protection for EMR clusters, follow the below procedure:

1. Sign in to AWS Console and navigate to https://console.aws.amazon.com/elasticmapreduce/home.
2. In the Navigation pane, choose **Clusters**.
3. Click on **Create cluster** button, Go to **advanced** options.
4. Select all appropriate required settings in step 1 and step 2
5. In step 3 **General Cluster settings**, check the **Termination protection** checkbox
6. Once done setting with all other steps
7. Click **Create cluster** button.

Using AWS CLI:

With below AWS CLI command we can modify an existing EMR cluster termination protection

```
aws emr modify-cluster-attributes --cluster-id <clusterId> --
termination-protected
```

For command usage refer: https://docs.aws.amazon.com/cli/latest/reference/emr/create-cluster.html

Reference

- https://docs.aws.amazon.com/emr/latest/ManagementGuide/UsingEMR_TerminationProtection.html

Control ID - 238: Ensure ACM uses imported certificates only and does not create/issue certificates

Criticality: MEDIUM

Specification

In addition to requesting certificates provided by AWS Certificate Manager (ACM), you can import certificates that you obtained outside of AWS. You might do this because you already obtained a certificate from a third-party issuer, or because the certificates provided by ACM do not meet your requirements.

Rationale

Amazon ACM imported certificates are required as they will be self signed with own keys and not amazon issued. You might do this because you already obtained a certificate from a third-party issuer, or because the certificates provided by ACM do not meet your requirements.

After you import a certificate, you can use it with the AWS services that are integrated with ACM.

Evaluation

This control ensures that ACM certificates are always imported.

Remediation

Using AWS Console:

To ensure ACM uses imported certificates, follow the below procedure:

1. Sign in to AWS Console and navigate to https://console.aws.amazon.com/acm/home.
2. Choose **Import a certificate** button.
3. For **Certificate body**, paste the PEM-encoded certificate to import.
4. For **Certificate private key**, paste the PEM-encoded, unencrypted private key that matches the certificate's public key.
5. (Optional) For **Certificate chain**, paste the PEM-encoded certificate chain.
6. Choose **Review and import**.

Using AWS CLI:

```
aws acm import-certificate --certificate fileb://Certificate.pem \ --
certificate-chain fileb://CertificateChain.pem \ --private-key
fileb://PrivateKey.pem
```

For command usage refer:
https://awscli.amazonaws.com/v2/documentation/api/latest/reference/acm/import-certificate.html

Reference

- https://docs.aws.amazon.com/acm/latest/userguide/import-certificate.html

Control ID - 239: Ensure expired certificates are removed from AWS ACM

Criticality: HIGH

Specification

Expired ACM certificates should be removed or if applicable renewed. So, that services integrated with that certificate will run without failures.

Rationale

Expired ACM certificates should be removed or if applicable renewed. So, that services integrated with that certificates will run without failures.

After you import a certificate, you can use it with the AWS services that are integrated with ACM.

Evaluation

This control ensures that expired ACM certificates are removed.

Remediation

Using AWS Console:

To ensure expired certificates are removed from AWS ACM, follow the below procedure:

1. Sign in to AWS Console and navigate to https://console.aws.amazon.com/acm/home.
2. Check the expired certificate checkbox in the list.
3. Open **Actions** dropdown menu.
4. Select **Delete** option.

Using AWS CLI:

```
aws acm delete-certificate --certificate-arn <certificateARN>
```

For command usage refer: https://docs.aws.amazon.com/cli/latest/reference/acm/delete-certificate.html

Reference

- https://docs.aws.amazon.com/acm/latest/userguide/gs-acm-delete.html

Control ID - 240: Ensure ACM certificates should not have domain with wildcard(*)

Criticality: HIGH

Specification

This control ensures that no ACM certificate should have domain name with wildcard (*) character.

Rationale

ACM certificate with wildcard (*) character in domain name is not a recommended practice. The certificates with wilcard character domain names will be applied all the sub-domains which is not advised as per security guidelines.

For example, *.example.com can be applicable to login.example.com, and test.example.com.

Evaluation

This control ensures that no ACM certificate should have domain name with wildcard (*) character.

Remediation

Using AWS Console:

To ensure ACM certificates should not have domain with wildcard(*) character, follow the below procedure:

1. Sign in to AWS Console and navigate to https://console.aws.amazon.com/acm/home.
2. For **imported** certificates, while creating the self sign certificate please make sure domain name does not contain wilcard(*).
3. For requesting a certificate in ACM, click on **Request a certificate** button.
4. Select either of the two options, **Request a public certificate** or **Request a private certificate**
5. For public certificate, in **step 1** add domain name without a wildcard(*) character
6. Proceed with next steps with required settings.
7. For private certificate On the **Request a certificate** page, choose **Request a private certificate** and **Request a certificate** to continue.
8. On the **Select a certificate authority (CA)** page, click the **Select a CA field** to view the list of available private CAs.
9. Choose a CA from the list. Choose **Next**.
10. On the **Add domain names** page, type your domain name without wildcard (*) character.
11. When you finish with next steps, choose Review and request.

Using AWS CLI:

Requesting a Public certificate

```
aws acm request-certificate \
     --domain-name www.example.com \
     --validation-method DNS \
     --idempotency-token 1234 \
     --options CertificateTransparencyLoggingPreference=DISABLED
```

Requesting a Private certificate

```
aws acm request-certificate \
      --domain-name www.example.com \
      --idempotency-token 12563 \
      --options CertificateTransparencyLoggingPreference=DISABLED \
      --certificate-authority-arn arn:aws:acm-pca:region:account:\
      certificate-authority/12345678-1`234-1234-1234-123456789012
```

To Import a self signed certificate

```
      aws acm import-certificate --certificate fileb://Certificate.pem \ --
certificate-chain fileb://CertificateChain.pem \ --private-key
fileb://PrivateKey.pem
```

For command usage refer: https://docs.aws.amazon.com/cli/latest/reference/acm/request-certificate.html

https://awscli.amazonaws.com/v2/documentation/api/latest/reference/acm/import-certificate.html

Reference

- https://docs.aws.amazon.com/acm/latest/userguide/gs-acm-request-public.html
- https://docs.aws.amazon.com/acm/latest/userguide/gs-acm-request-private.html
- https://docs.aws.amazon.com/acm/latest/userguide/import-certificate.html

Control ID - 241: Ensure that the certificate use appropriate algorithms and key size

Criticality: MEDIUM

Specification

This control ensures that ACM certificate should use appropriate key algorithms.

Rationale

ACM certificate with key algorithms like RSA_2048 and RSA_4096 are advised as a security best practice. ACM certificates with other key algorithms are not recommended.

Evaluation

This control ensures that ACM certificate should use appropriate key algorithms.

Remediation

Using AWS Console:

To ensure ACM certificates use appropriate key size, follow the below procedure:

1. Sign in to AWS Console and navigate to https://console.aws.amazon.com/acm/home.
2. For **imported** certificates, while creating the self sign certificate please make sure key size should be starting with **RSA**
3. For requesting a certificate in ACM, click on **Request a certificate** button.
4. Select either of the two options, **Request a public certificate** or **Request a private certificate**
5. For public certificate, in **step 1** add domain name without a wildcard(*) character
6. Proceed with next steps with required settings. AWS auotmatically assigns a key size for public certificate
7. For private certificate On the **Request a certificate** page, choose **Request a private certificate** and **Request a certificate** to continue.
8. On the **Select a certificate authority (CA)** page, click the **Select a CA field** to view the list of available private CAs.
9. Key size of **private certificate** depends on the key algorithm specified during the setup of the **certificate authority**
10. Choose a CA from the list. Choose **Next**.
11. On the **Add domain names** page, type your domain name without wildcard (*) character.
12. When you finish with next steps, choose Review and request.

Using AWS CLI:

Requesting a Public certificate

```
aws acm request-certificate \
     --domain-name www.example.com \
     --validation-method DNS \
     --idempotency-token 1234 \
     --options CertificateTransparencyLoggingPreference=DISABLED
```

Requesting a Private certificate

```
aws acm request-certificate \
     --domain-name www.example.com \
     --idempotency-token 12563 \
     --options CertificateTransparencyLoggingPreference=DISABLED \
     --certificate-authority-arn arn:aws:acm-pca:region:account:\
     certificate-authority/12345678-1`234-1234-1234-123456789012
```

To Import a self signed certificate

```
aws acm import-certificate --certificate fileb://Certificate.pem \ --
certificate-chain fileb://CertificateChain.pem \ --private-key
fileb://PrivateKey.pem
```

For command usage refer: https://docs.aws.amazon.com/cli/latest/reference/acm/request-certificate.html

https://awscli.amazonaws.com/v2/documentation/api/latest/reference/acm/import-certificate.html

Reference

- https://docs.aws.amazon.com/acm/latest/userguide/gs-acm-request-public.html
- https://docs.aws.amazon.com/acm/latest/userguide/gs-acm-request-private.html
- https://docs.aws.amazon.com/acm/latest/userguide/import-certificate.html
- https://docs.aws.amazon.com/acm-pca/latest/userguide/supported-algorithms.html

Control ID - 242: Ensure logging is not set to OFF for Rest APIs Stage in all regions

Criticality: MEDIUM

Specification

There are two types of API logging in CloudWatch: execution logging and access logging. In execution logging, API Gateway manages the CloudWatch Logs. The process consists of creating log groups and log streams, and reporting to the log streams any caller's requests and responses.

The logged data includes errors or execution traces (such as request or response parameter values or payloads), data used by Lambda authorizers (formerly known as custom authorizers), whether API keys are required, whether usage plans are enabled.

Rationale

There are two types of API logging in CloudWatch: execution logging and access logging. In execution logging, API Gateway manages the CloudWatch Logs. The process consists of creating log groups and log streams, and reporting to the log streams any caller's requests and responses.

The logged data includes errors or execution traces (such as request or response parameter values or payloads), data used by Lambda authorizers (formerly known as custom authorizers), whether API keys are required, whether usage plans are enabled.

This control ensures that logging is not set to OFF for all Rest APIs Stage in API gateway for all regions

Evaluation

This control ensures that logging is not set to OFF for all Rest APIs Stages in API gateway for all regions

Remediation

Using AWS Console:

1. Sign in to the API Gateway console and Click on the API you want to edit
2. On the **Stages** pane, choose the **Logs/Tracing** tab
3. On the Logs/Tracing tab, under **CloudWatch Settings**, do the following to turn on execution logging: Choose the Enable **CloudWatch Logs** check box
4. You can choose from ERROR and INFO according to your logging requirements
5. Click on **Save Changes**

Note: Please repeat these steps for all API Stages

Reference

- https://docs.aws.amazon.com/apigateway/latest/developerguide/set-up-logging.html
- https://aws.amazon.com/premiumsupport/knowledge-center/api-gateway-cloudwatch-logs/

Control ID - 243: Ensure to enable encryption if caching is enabled for Rest API Stage in all regions

Criticality: MEDIUM

Specification

You can enable API caching in AWS API Gateway to cache your endpoint's responses. With caching, you can reduce the number of calls made to your endpoint and also improve the latency of requests to your API. When you enable caching for a stage, API Gateway caches responses from your endpoint for a specified time-to-live (TTL) period, in seconds. API Gateway then responds to the request by looking up the endpoint response from the cache instead of making a request to your endpoint.

If you anticipate that a method that you are caching will receive sensitive data in its responses, in Cache Settings, choose Encrypt cache data.

Rationale

You can enable API caching in AWS API Gateway to cache your endpoint's responses. With caching, you can reduce the number of calls made to your endpoint and also improve the latency of requests to your API. When you enable caching for a stage, API Gateway caches responses from your endpoint for a specified time-to-live (TTL) period, in seconds. API Gateway then responds to the request by looking up the endpoint response from the cache instead of making a request to your endpoint.

If you anticipate that a method that you are caching will receive sensitive data in its responses, in Cache Settings, choose Encrypt cache data.

This control ensures that data encryption is enabled if caching is enabled for all Rest API Stages in all regions

Evaluation

This control ensures that data encryption is enabled if caching is enabled for all Rest API Stages in all regions

Remediation

Using AWS Console:

1. Sign in to the API Gateway console and Click on the API you want to edit
2. On the **Stages** pane, choose the **Logs/Tracing** tab
3. On the **Settings** tab,Under **Cache Settings**, Choose the **Enable API Cache** check box
4. Choose the **Encrypt cache data** check box
5. Click on **Save Changes**

Note: Please repeat these steps for all API Stages

Reference

- https://docs.aws.amazon.com/apigateway/latest/developerguide/data-protection-encryption.html
- https://docs.aws.amazon.com/apigateway/latest/developerguide/api-gateway-caching.html

Control ID - 244: Ensure accessLogSettings exists with the destinationArn and in the json format for Rest API Stage in all regions

Criticality: MEDIUM

Specification

There are two types of API logging in CloudWatch: execution logging and access logging. In access logging, you, as an API developer, wants to log who has accessed your API and how the caller accessed the API. You can create your own log group or choose an existing log group that could be managed by API Gateway. To specify the access details, you select $context variables (expressed in a format of your choosing) and choose a log group as the destination. To preserve the uniqueness of each log, the access log format must include $context.requestId variable.

Choose a log format that is also adopted by your analytic backend, such as Common Log Format (CLF), JSON, XML, or CSV. You can then feed the access logs to it directly to have your metrics computed and rendered.

Rationale

There are two types of API logging in CloudWatch: execution logging and access logging. In access logging, you, as an API developer, wants to log who has accessed your API and how the caller accessed the API. You can create your own log group or choose an existing log group that could be managed by API Gateway. To specify the access details, you select $context variables (expressed in a format of your choosing) and choose a log group as the destination. To preserve the uniqueness of each log, the access log format must include $context.requestId variable.

Choose a log format that is also adopted by your analytic backend, such as Common Log Format (CLF), JSON, XML, or CSV. You can then feed the access logs to it directly to have your metrics computed and rendered.

This control ensures that accessLogSettings exists with the destinationArn and in the json format for all Rest API Stages in all regions

Evaluation

This control ensures that accessLogSettings exists with the destinationArn and in the json format for all Rest API Stages in all regions

Remediation

Using AWS Console:

1. Sign in to the API Gateway console and Click on the API you want to edit
2. On the **Stages** pane, choose the **Logs/Tracing** tab
3. On the **Logs/Tracing** tab,Under **Custom Access Logging**, do the following to turn on access logging: Choose the Enable **Access Logging** check box
4. For **Access Log Destination ARN**, enter the ARN of a CloudWatch log group or an Amazon Kinesis Data Firehose stream
5. under **Log Format**, enter the logging config in JSON format
6. Click on **Save Changes**

Note: Please repeat these steps for all API Stages

Reference

- https://docs.aws.amazon.com/apigateway/latest/developerguide/set-up-logging.html
- https://aws.amazon.com/premiumsupport/knowledge-center/api-gateway-cloudwatch-logs/

Control ID - 245: Ensure there are no Internet facing Network load balancers

Criticality: MEDIUM

Specification

Network load balancers need to be configured in the right scheme for maintaining secure architecture.

Rationale

To maintain a secure load balancing architecture it is essential for Network Load balancers to be configured with the right scheme. An internet-facing load balancer has a publicly resolvable DNS name, so it can route requests from clients over the internet to the EC2 instances that are registered with the load balancer. An internal load balancer routes requests to targets using private IP addresses.

Evaluation

This control ensures no Network Load Balancers is configured for internet facing scheme.

Remediation

1. Go to AWS Management EC2 dashboard at https://console.aws.amazon.com/ec2/
2. In the left navigation panel under **LOAD BALANCING**, select **Load Balancers**.
3. Click Create load balancer from the dashboard top menu, select Network Load Balancer then click create.
4. On Create Network Load Balancer page, provide a unique name for your new AWS NLB then set the load balancer Scheme to internal. Configure the necessary listeners and availability zones then once all these are configured, use the Add tag button, available in the Tags section, to attach tags to your new NLB.

Using AWS CLI:

Run create-load-balancer command to create a internal AWS Network Load balancer

```
aws elbv2 create-load-balancer --region <region> --name <load balancer name> --scheme internal --type network --subnets <subnet id>
```

Note: For more details on command, please refer to https://docs.aws.amazon.com/cli/latest/reference/elbv2/create-load-balancer.html

Reference

- https://docs.aws.amazon.com/elasticloadbalancing/latest/classic/elb-internet-facing-load-balancers.html
- https://docs.aws.amazon.com/elasticloadbalancing/latest/network/create-network-load-balancer.html
- https://docs.aws.amazon.com/cli/latest/reference/elbv2/create-load-balancer.html

Control ID - 246: Ensure NLB using listener type TLS must have SSL Security Policy

Criticality: MEDIUM

Specification

Network load balancers are using latest security policy for SSL configuration.

Rationale

Using insecure and deprecated security policies for SSL negotiation configuration within Load Balancers will expose the connection between the client and the load balancer to various SSL/TLS vulnerabilities. Latest security policy secures SSL negotiation configuration in order to follow security best practices and protect their front-end connections.

Evaluation

This control ensures that TLS listeners have security policies configured for Network Load Balancer.

Remediation

1. Go to AWS Management EC2 dashboard at https://console.aws.amazon.com/ec2/
2. In the left navigation panel under **LOAD BALANCING**, select **Load Balancers**.
3. Select the Load Balancer that you want to reconfigure.
4. Choose the **Listeners** tab from the bottom panel.
5. Select the **Non TLS listener**, click the Actions dropdown button from the panel top menu and select Edit.
6. Under Protocol and Port select TLS, under Default actions choose desired action and target.
7. Under **Security Policy** section select desired **ELBSecurityPolicy** policy.

Using AWS CLI:

Run modify-listener command using the ARN of the TLS listener that you want to reconfigure as identifier to update its predefined security policy

```
aws elbv2 modify-listener --region <region> --listener-arn <listener arn> --
protocol TLS --port 443 --certificates CertificateArn=<Certificate arn> --ssl-
policy  <ELBSecurityPolicy name> --default-actions Type=<Action
Type>,TargetGroupArn=<Target Group Arn>
```

Note: For more details on command, please refer to
https://docs.aws.amazon.com/cli/latest/reference/elbv2/modify-listener.html

Reference

- https://docs.aws.amazon.com/elasticloadbalancing/latest/network/listener-update-certificates.html
- https://docs.aws.amazon.com/elasticloadbalancing/latest/network/create-tls-listener.html

Control ID - 247: Ensure that NLB listeners using TLS have TLS enabled Target Groups configured

Criticality: MEDIUM

Specification

Network Load balancer target group needs to be configured with TLS on port 443.

Rationale

Each target group is used to route requests to one or more registered targets. Each created listener rule is specified with a target group and conditions. When a rule condition is met, traffic is forwarded to the corresponding target group. Multiple target groups can be created per different requirements. These target groups can be configured to perform health check for load balancer instances.

Evaluation

This control ensures that Network Load balancer target group is configured using TLS on port 443.

Remediation

1. Go to AWS Management EC2 dashboard at https://console.aws.amazon.com/ec2/
2. In the left navigation panel under **LOAD BALANCING**, select **Target Groups**.
3. Click on Create Target Group button and follow below steps
    1. Choose a target type from Instance, IP Address or Lambda Function as per requirement.
    2. Give the target group a name.
    3. Choose protocol as TLS and Port as 443.
    4. Select Protocol version and Health check protocol as desired.
    5. Click next button and add Instances as needed and click Create target group to submit.

Using AWS CLI:

Run create-target-group command to create target group with protocol TLS and port 443

```
aws elbv2 create-target-group --name <target group name> --protocol TLS --port
443 --target-type instance --vpc-id <VPC ID>
```

Note: For more details on command, please refer to
https://docs.aws.amazon.com/cli/latest/reference/elbv2/create-target-group.html

Reference

- https://docs.aws.amazon.com/elasticloadbalancing/latest/network/load-balancer-target-groups.html
- https://docs.aws.amazon.com/elasticloadbalancing/latest/network/create-target-group.html

Control ID - 248: Ensure that NLB listeners using default insecure ports are not configured for passthrough

Criticality: MEDIUM

Specification

Network load balancers using insecure ports are not configured for passthrough.

Rationale

Using insecure ports and hence no security policies for SSL negotiation configuration within Load Balancers will expose the connection between the client and the load balancer to various SSL/TLS vulnerabilities. Secure ports with security policy will secure SSL negotiation configuration in order to follow security best practices and protect their front-end connections.

Evaluation

This control ensures that NLB listeners having insecure ports are not configured for passthrough.

Remediation

1. Go to AWS Management EC2 dashboard at https://console.aws.amazon.com/ec2/
2. In the left navigation panel under **LOAD BALANCING**, select **Load Balancers**.
3. Select the Load Balancer that you want to reconfigure.
4. Choose the **Listeners** tab from the bottom panel.

5. Select the **Non TLS listener**, click the Actions dropdown button from the panel top menu and select Edit.
6. Under Protocol and Port select TLS, under Default actions choose desired action and target.
7. Under **Security Policy** section select desired **ELBSecurityPolicy** policy.

Using AWS CLI:

Run modify-listener command using the ARN of the TLS listener that you want to reconfigure as identifier to update its predefined security policy

```
aws elbv2 modify-listener --region <region> --listener-arn <listener arn> --
protocol TLS --port 443 --certificates CertificateArn=<Certificate arn> --ssl-
policy  <ELBSecurityPolicy name> --default-actions Type=<Action
Type>,TargetGroupArn=<Target Group Arn>
```

Note: For more details on command, please refer to
https://docs.aws.amazon.com/cli/latest/reference/elbv2/modify-listener.html

Reference

- https://docs.aws.amazon.com/elasticloadbalancing/latest/network/listener-update-certificates.html
- https://docs.aws.amazon.com/elasticloadbalancing/latest/network/create-tls-listener.html

Control ID - 249: Ensure AWS NLB logging is enabled

Criticality: MEDIUM

Specification

Access logs for a Network Load Balancer contains detailed information about requests sent to the load balancer. These include time the request was received, the client's IP address, latencies, request paths, and server responses. These logs can be monitored and reviewed to analyze traffic patterns and to troubleshoot issues and security incidents.

Rationale

Access logs for Network Load balancer provide visibility on the access requests to the load balancer and can be used anomalous and malicious traffic and calls. These logs can also play a vital role in troubleshooting and forensic analysis.

Access logging is disabled by default.

Evaluation

This control verifies that Access Logs are enabled for all Network Load Balancers.

Remediation

Note:

1. To enable access logs for Network Load balancer, an S3 bucket is required in the same region as the Load Balancer with Amazon S3-Managed Encryption Keys (SSE-S3) encryption.
2. The bucket must have a bucket policy that grants Network Load Balancing permission to write the access logs to your bucket. For more details on required permissions and policy, please refer to https://docs.aws.amazon.com/elasticloadbalancing/latest/application/load-balancer-access-logs.html#enable-access-logging

Using AWS Console:

- To create a new S3 bucket

1. Go to Amazon S3 console at https://console.aws.amazon.com/s3/
2. Select `Create Bucket.`
3. Enter a unique bucket name and select the region where you created your load balancer.
4. Click `Create`.

- To allow write access to Network Load Balancer on S3 bucket, edit the bucket policy

1. Go to Amazon S3 console at https://console.aws.amazon.com/s3/
2. Select the S3 bucket, choose `Permissions`, choose `Bucket Policy`.
3. Edit the policy to allow the required permissions for Load Balancer to write the logs to the bucket. Please refer to https://docs.aws.amazon.com/elasticloadbalancing/latest/network/load-balancer-access-logs.html#enable-access-logging for more details on required permissions and bucket policy.

- To Enable Access Logs for Network Load Balancer

1. Go to Amazon EC2 console at https://console.aws.amazon.com/ec2/v2/home
2. In the navigation pane, under `Load Balancing`, choose `Load Balancers`.
3. Select the load balancer.
4. Under `Description` tab, choose `Edit attributes`.
5. Select `Enable` for `Access logs`.
6. For `S3 location`, select an existing S3 bucket or create a new bucket.
7. Click `Save`.

Using AWS CLI:

- To create a new S3 bucket

```
aws s3api create-bucket --bucket my-bucket --region us-east-1
```

- To allow write access to Network Load Balancer on S3 bucket, edit the bucket policy

1. Create a policy JSON file to allow the required permissions for Load Balancer to write the logs to the bucket. Please refer to https://docs.aws.amazon.com/elasticloadbalancing/latest/network/load-balancer-access-logs.html#enable-access-logging for more details on required permissions and bucket policy.
2. Use the command below to attach the policy to the bucket
   `aws s3api put-bucket-policy --bucket <bucket_name> --policy <path_to_json>`

Note: Using the above command will overwrite the current policy. Please keep the include all statements in the JSON that are required for the policy.

- To Enable Access Logs for existing Network Load Balancer

Use the following command to enable the access logs for existing Load Balancer

```
aws elbv2 modify-load-balancer-attributes --load-balancer-arn
<load_balancer_arn> --attributes Key=access_logs.s3.enabled,Value=true
Key=access_logs.s3.bucket,Value=<bucket_name>
Key=access_logs.s3.prefix,Value=<log_prefix>
```

Reference

- https://docs.aws.amazon.com/elasticloadbalancing/latest/network/load-balancer-access-logs.html
- https://docs.aws.amazon.com/cli/latest/reference/s3api/put-bucket-policy.html
- https://docs.aws.amazon.com/cli/latest/reference/s3api/create-bucket.html
- https://docs.aws.amazon.com/cli/latest/reference/elbv2/modify-load-balancer-attributes.html

Control ID - 252: Ensure to encrypt the data in transit when using NFS between the client and EFS service

Criticality: HIGH

Specification

Amazon Elastic File System (Amazon EFS) provides a simple, serverless, set-and-forget, elastic file system that lets you share file data without provisioning or managing storage.Enabling encryption of data in transit for your Amazon EFS file system is done by enabling Transport Layer Security (TLS) when you mount your file system using the Amazon EFS mount helper.

When encryption of data in transit is declared as a mount option for your Amazon EFS file system, the mount helper initializes a client stunnel process. Stunnel is an open source multipurpose network relay. The client stunnel process listens on a local port for inbound traffic, and the mount helper redirects Network File System (NFS) client traffic to this local port. The mount helper uses TLS version 1.2 to communicate with your file system.

Rationale

Enabling encryption of data in transit for your Amazon EFS file system is done by enabling Transport Layer Security (TLS) when you mount your file system using the Amazon EFS mount helper. For more information, see Mounting file systems using the EFS mount helper.

Evaluation

This control ensures whether the data in transit is encrypted or not.

Remediation

1. Open the Amazon Elastic File System console at https://console.aws.amazon.com/efs/
2. Choose **File Systems**
3. On the **File Systems** page, choose the file system that you want to edit or create a file system policy for. The details page for that file system is displayed.
4. Choose **File system policy**, then choose **Edit**. The **File system policy** page appears.
5. In **Policy options**, choose **Enforce in-transit encryption for all clients** - This option denies access to unencrypted clients.
6. choose **Save**.

Using AWS CLI:

To modify a file system policy, use the following AWS CLI command:

```
aws efs put-file-system-policy --file-system-id <value> --policy '{
            "Version": "2012-10-17",
            "Id": "<value>",
            "Statement": [
                {
                    "Sid": "<value>",
                    "Effect": "Allow",
                    "Principal": {
                        "AWS": "*"
                    },
                    "Action": [
                        "elasticfilesystem:ClientRootAccess",
                        "elasticfilesystem:ClientWrite",
                        "elasticfilesystem:ClientMount"
                    ],
                    "Condition": {
                        "Bool": {
                            "elasticfilesystem:AccessedViaMountTarget": "true"
                        }
                    }
                },
                {
                    "Sid": "<value>",
                    "Effect": "Deny",
                    "Principal": {
                        "AWS": "*"
                    },
                    "Action": "*",
                    "Condition": {
                        "Bool": {
                            "aws:SecureTransport": "false"
                        }
                    }
                }
            ]
        }'
```

Note: For more details on command, please refer to https://docs.aws.amazon.com/cli/latest/reference/efs/put-file-system-policy.html

Reference

- https://docs.aws.amazon.com/AmazonRDS/latest/AuroraUserGuide/Aurora.Managing.Backups.html
- https://docs.aws.amazon.com/efs/latest/ug/encryption-in-transit.html
- https://docs.aws.amazon.com/efs/latest/ug/encryption-in-transit.html

Control ID - 256: Ensure trail is configure on organization level

Criticality: HIGH

Specification

AWS CloudTrail is an AWS service that helps you enable governance, compliance, and operational and risk auditing of your AWS account. Using AWS CloudTrail, a user in a management account can create an organization trail that logs all events for all AWS accounts in that organization. Organization trails are automatically applied to all member accounts in the organization.

Rationale

AWS Organizations is integrated with AWS CloudTrail, a service that provides a record of actions taken by a user, role, or an AWS service in AWS Organizations. CloudTrail captures all API calls for AWS Organizations as events, including calls from the AWS Organizations console and from code calls to the AWS Organizations APIs.

Evaluation

This control ensures that trail is configured on organization level

Remediation

1. Open the Amazon Cloudtrail console at https://console.aws.amazon.com/cloudtrail.
2. Click on the trail name that needs to be remediate.
3. Go to **General details** section and click on edit button.
4. Check the field **Enable for all accounts in my organization** and click on **Save changes**

Note : You only see this option if you are signed in to the console with an IAM user or role in the management account. To successfully create an organization trail, be sure that the user or role has sufficient permissions.

Using AWS CLI:

To update the trail, use the following AWS CLI command:

```
aws cloudtrail update-trail --name [trail_name] --is-organization-trail
```

Note: You can run the update-trail command only from the region in which the trail was created.

Note: For more details on command, please refer to
https://docs.aws.amazon.com/cli/latest/reference/cloudtrail/update-trail.html

Reference

- https://docs.aws.amazon.com/awscloudtrail/latest/userguide/creating-an-organizational-trail-in-the-console.html
- https://docs.aws.amazon.com/organizations/latest/userguide/services-that-can-integrate-cloudtrail.html

Control ID - 264: Ensure each trail includes the global services

Criticality: HIGH

Specification

For most services, events are recorded in the region where the action occurred. For global services such as AWS Identity and Access Management (IAM), AWS STS, and Amazon CloudFront, events are delivered to any trail that includes global services.

Rationale

For most global services, events are logged as occurring in US East (N. Virginia) Region, but some global service events are logged as occurring in other regions, such as US East (Ohio) Region or US West (Oregon) Region.

Evaluation

This control ensures that trail is configured to log global service events.

Remediation

We can not perform this operation using AWS Management Console

Using AWS CLI:

To include global service events logging to trail use the following cli command:

```
aws cloudtrail update-trail --name [trail_name] --include-global-service-events
```

Note: For more details on command, please refer to https://docs.aws.amazon.com/cli/latest/reference/cloudtrail/update-trail.html

Reference

https://docs.aws.amazon.com/awscloudtrail/latest/userguide/cloudtrail-concepts.html#cloudtrail-concepts-global-service-events

Control ID - 272: Ensure to log KMS events to the trail

Criticality: HIGH

Specification

Management events describe management operations that are performed on resources in your AWS account. These are also known as control plane operations. By default, trails are configured to log management events.

AWS KMS actions such as Encrypt, Decrypt, and GenerateDataKey typically generate a large volume (more than 99%) of events. These actions are logged as Read events. Low-volume, relevant AWS KMS actions such as Disable, Delete and ScheduleKey are logged as Write events.

Rationale

It is recommended not to exclude KMS events from Management events logging.

Evaluation

This control ensures to log KMS events to the trail

Remediation

Using AWS management console:

1. Open the Amazon Cloudtrail console at https://console.aws.amazon.com/cloudtrail.
2. Click on the trail name that needs to be remediate.
3. Go to **Management events** section and click on edit button.
4. Unchecked **Exclude AWS KMS events** field.
5. Click on **Save changes**.

Reference

- https://docs.aws.amazon.com/awscloudtrail/latest/userguide/logging-management-events-with-cloudtrail.html

Control ID - 273: Ensure block public access is enabled so that no port should have public access for EMR clusters

Criticality: HIGH

Specification

Amazon EMR block public access prevents a cluster in a public subnet from launching when any security group associated with the cluster has a rule that allows inbound traffic from IPv4 0.0.0.0/0 or IPv6 ::/0 (public access) on a port, unless the port has been specified as an exception. Port 22 is an exception by default. You can configure exceptions to allow public access on a port or range of ports. Block public access does not take effect in private subnets.

Rationale

Block public access is only applicable during cluster creation. Block public access does not block IAM principals with appropriate permissions from updating security group configurations to allow public access on running clusters.

Evaluation

This control ensures that block public access is enabled so that no port should have public access for EMR clusters.

Remediation

Note: This Control evaluates EMR service for all regions collectively, i.e the control will fail if any region fails. If any region is disabled, this would be counted as a region failure.

Using AWS Console:

1. Sign in to AWS Console and navigate to https://console.aws.amazon.com/elasticmapreduce/home.
2. In left navigation pane, Click on **Block public access**.

3. In the next page, Click on **Change** under **Block public access settings** and turn on the **Block public access**.

Using AWS CLI:

```
aws emr put-block-public-access-configuration --block-public-access-
configuration BlockPublicSecurityGroupRules=true --region [Region]
```

For command usage refer: https://docs.aws.amazon.com/cli/latest/reference/emr/put-block-public-access-configuration.html

Reference

- https://docs.aws.amazon.com/emr/latest/ManagementGuide/emr-block-public-access.html

Control ID - 285: Ensure all data stored in the Elasticsearch is securely encrypted at rest

Criticality: HIGH

Specification

Encryption of data at rest is a security feature that helps prevent unauthorized access to your data.
The feature uses AWS Key Management Service (AWS KMS) to store and manage your encryption keys and the Advanced Encryption Standard algorithm with 256-bit keys (AES-256) to perform the encryption.
If enabled, the feature encrypts the domain's: indices, logs, swap files, all data in the application directory, and automated snapshots.

OpenSearch Service domains offer encryption of data at rest, a security feature that helps prevent unauthorized access to your data.
The feature uses AWS Key Management Service (AWS KMS) to store and manage your encryption keys and the Advanced Encryption Standard algorithm with 256-bit keys (AES-256) to perform the encryption.

Rationale

When you use your own Encryption of data at rest helps prevent unauthorized users from reading sensitive information available on your ES domains (clusters) and their storage systems. This includes all data stored on the underlying file systems, primary and replica indices, log files, memory swap files and automated snapshots saved to S3. Amazon ElasticSearch handles the encryption/decryption process seamlessly, so you do not have to modify your applications to access your data. The ElasticSearch at-rest encryption feature uses AWS KMS service to store and manage the encryption keys.

Evaluation

This control ensures that all data stored in the Elasticsearch is securely encrypted at rest.

Remediation

Using AWS Console

To Create Amazon OpenSearch Service (successor to Amazon Elasticsearch Service) Domain

1. Open Amazon OpenSearch Service (successor to Amazon Elasticsearch Service) console at https://console.aws.amazon.com/esv3.
2. In the left navigation panel, choose **Domains**.
3. Click **Create domain** button.
4. Navigate to **Fine-grained access control** section, and disable **Enable fine-grained access control**.
5. Navigate down to **Encryption** section, add check if **Enable encryption of data at rest** is enabled.
6. Click **Create**.

Using AWS CLI

To create AWS KMS key, use the following command:

From the output of this cli copy the key Amazon Resource Name (ARN) as this will be required for next steps.

```
aws kms create-key \

        --region [region]  \

        --description ['key-description']
```

To create AWS ElasticSearch domain and enable at-rest encryption feature using --encryption-at-rest-options, use the following command:

```
aws es create-elasticsearch-domain \

        --region [region] \

        --domain-name [domainName] \

        --elasticsearch-version 5.5 \

        --elasticsearch-cluster-config
InstanceType=r5.large.elasticsearch,InstanceCount=3 \

        --ebs-options EBSEnabled=true,VolumeType=gp2,VolumeSize=10 \

        --vpc-options SubnetIds=[subnet-id],SecurityGroupIds=[security-group-
id] \

        --encryption-at-rest-options Enabled=true,kmsKeyId=[KMS-key-ID]
```

Reference

- Reference Amazon OpenSearch Service
- Encryption of data at rest for Amazon OpenSearch Service
- AWS CLI reference

Control ID - 286: Ensure all data stored in the Launch configuration EBS is securely encrypted

Criticality: HIGH

Specification

Rationale

Amazon Elastic Block Store (EBS) encryption as a straight-forward encryption solution for your EBS resources associated with your EC2 instances. With Amazon EBS encryption, you aren't required to build, maintain, and secure your own key management infrastructure. Amazon EBS encryption uses AWS KMS keys when creating encrypted volumes and snapshots. Encryption operations occur on the servers that host EC2 instances, ensuring the security of both data-at-rest and data-in-transit between an instance and its attached EBS storage.

Evaluation

This control ensures that all data stored in the Launch configuration EBS is securely encrypted

Remediation

Using AWS Console:

1. Note: The launch configuration EBS can be encrypted only at the time of creating an instance.
2. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/v2/.
3. In the left navigation pane, under **Auto Scaling**, choose **Launch Configurations**.
4. From the Launch configurations page, choose **Create launch configuration**.
5. Enter a name for your launch configuration.
6. For **Amazon machine image (AMI)**, choose an encrypted EBS-backed AMI from AMI drop-down menu.
7. For **Instance type**, select a hardware configuration for your instances.
8. For **Storage (volumes)**, to specify volumes to attach to the instances in addition to the volumes specified by the AMI, choose Add new volume.
9. Choose the desired options for snapshot (encrypted) and check the checkbox for **Encrypted**.
10. For **Security groups**, create or select the security group to associate with the group's instances.
11. For **Key pair (login)**, choose an option under **Key pair options**.
12. If you've already configured an Amazon EC2 instance key pair, you can choose **Choose an existing key pair**.
    If you don't have an Amazon EC2 instance key pair, choose **Create a new key pair** and give it a name. Choose **Download key pair** to download the key pair to your computer.
13. Select the acknowledgment check box, and click **Create launch configuration**.

Using AWS CLI:

To launch an instance with additional volumes. Create a JSON file on your local machine with a name such as `mapping.json`, and then paste the following content into it.

```
[
    {
        "DeviceName": "/dev/sdh",
        "Ebs": {
            "VolumeSize": 8,
            "VolumeType": "gp2",
            "Encrypted": true
        }
    }
]
```

To create a launch configuration with additional ebs vloume, use the following command:

```
aws autoscaling create-launch-configuration \

        --launch-configuration-name [launch-config-name] \

        --image-id [ami-id] \

        --instance-type [instance-type] \

        --block-device-mappings file://mapping.json
```

Reference

- Amazon EC2 Auto Scaling and data protection
- Amazon EBS encryption
- Use encryption with EBS-backed AMIs
- Block device mappings
- AWS CLI Command Reference

Control ID - 288: Ensure SageMaker Notebook is encrypted at rest using KMS CMK

Criticality: HIGH

Specification

SageMaker is a fully-managed AWS service that enables developers and data engineers to quickly and easily build, train and deploy machine learning models at any scale. An AWS SageMaker notebook instance is a fully managed ML instance that is running the Jupyter Notebook open-source web application.

Rationale

Data stored on Machine Learning (ML) storage volumes attached to your AWS SageMaker notebook instances is encrypted in order to meet regulatory requirements and protect your SageMaker data at rest.

Evaluation

This control ensures that data stored on AWS SageMaker notebook instances is encrypted.

Remediation

Using AWS Console:

Note : Encryption settings for already created SageMaker notebook instance cannot be changed.

1. Go to AWS Console SageMaker dashboard at https://console.aws.amazon.com/sagemaker
2. In the left navigation pane, under **Notebook** click on **Notebook Instances**.
3. Click on **Create notebook instance**.
4. Under **Permissions and encryption** section, In **Encryption key** field, provide the **KMS key ARN** to be used for encryption.
5. Configure remaining configuration as per requirements.
6. Click on **Create notebook instance**.

Using AWS CLI:

```
aws sagemaker create-notebook-instance --notebook-instance-name
[Notebook_Instance_Name] --instance-type [Instance_Type] --role-arn [Role_Arn]
--kms-key-id [Key_Arn]
```

Note: For more details on command, please refer to https://docs.aws.amazon.com/cli/latest/reference/sagemaker/create-notebook-instance.html

Reference

- https://docs.aws.amazon.com/sagemaker/latest/dg/encryption-at-rest-nbi.html

Control ID - 289: Ensure every security groups rule has a description

Criticality: LOW

Specification

Security Group Rule descriptions help you get an insight onto the rule's configuration. It gives you an idea who has access and what prompted the rule's creation

Rationale

Security Group Rule descriptions help you get an insight onto the rule's configuration. It gives you an idea who has access and what prompted the rule's creation

This control ensures every security groups rules has a description

Evaluation

This control ensures every security groups rules has a description

Remediation

Using AWS Console:

1. Sign in to the AWS Management Console and open the Amazon EC2 console at https://console.aws.amazon.com/ec2/v2/home
2. In the left navigation panel, Select **Security Groups**
3. Select the Security Group you want to edit
4. Click on the **Inbound Rules** tab, Click on **Edit Inbound Rules**
5. Enter a description for all the rules listed
6. Click on **Save Rules**
7. Switch to **Outbound Rules**, Click on **Edit Outbound Rules**
8. Enter a description for all the rules listed
9. Click on **Save Rules**

Using AWS CLI:

To enter a description on Security Group rules, use the below command:

- `aws ec2 update-security-group-rule-descriptions-ingress --group-id [GROUP-ID] --ip-permissions '[{"IpProtocol": "[PROTOCOL]", "FromPort":`

```
[FROM-PORT], "ToPort": [TO-PORT], "IpRanges": [{"CidrIp": "[IP-RANGE]",
"Description": "[DESCRIPTION]"}]}]'
```

- `aws ec2 update-security-group-rule-descriptions-egress --group-id [GROUP-ID] --ip-permissions '[{"IpProtocol": "[PROTOCOL]", "FromPort": [FROM-PORT], "ToPort": [TO-PORT], "IpRanges": [{"CidrIp": "[IP-RANGE]", "Description": "[DESCRIPTION]"}]}]'`

Note: For more details on command, please refer to :

- https://docs.aws.amazon.com/cli/latest/reference/ec2/update-security-group-rule-descriptions-ingress.html
- https://docs.aws.amazon.com/cli/latest/reference/ec2/update-security-group-rule-descriptions-egress.html

Reference

- https://aws.amazon.com/blogs/aws/new-descriptions-for-security-group-rules/

Control ID - 290: Ensure SNS Topics have encryption at rest enabled

Criticality: LOW

Specification

By default, Amazon SNS provides in-transit encryption. Enabling server-side encryption adds at-rest encryption to your topic.

Rationale

Encryption at rest protects the contents of messages in topics using KMS key configured. As soon as the message is received by SNS, it will be encrypted and stored in encrypted form. Without encryption enabled, anyone who gains access will be able to read the message.

Evaluation

This control ensures that SNS topics have encryption at rest enabled.

Remediation

Using AWS Console:

1. Go to AWS Console SNS dashboard at https://console.aws.amazon.com/sns/
2. In the left navigation panel select **Topics**, select effected **topic**.
3. Click on **Edit** button.
4. Under **Encryption** section, select **Enable encryption**.
5. Under **Customer master key**, Select a custom CMK or enter an existing CMK ARN.
6. Click on **Save changes**.

Using AWS CLI:

```
aws sns set-topic-attributes --topic-arn [topic_arn] --attribute-name
KmsMasterKeyId --attribute-value [key-Id]
```

Note: For more details on command, please refer to https://docs.aws.amazon.com/cli/latest/reference/sns/set-topic-attributes.html

Reference

- https://docs.aws.amazon.com/sns/latest/dg/sns-server-side-encryption.html

Control ID - 291: Ensure SQS Queue have encryption at rest enabled

Criticality: LOW

Specification

Amazon SQS Server-side encryption transits sensitive data in encrypted queues.

Rationale

Whenever SQS Queue receives a message, it encrypts the message and stores it in encrypted form. Without encryption enabled, anyone who gains access will be able to read the message.

Evaluation

This control ensures that SQS Queues have encryption at rest enabled.

Remediation

Using AWS Console:

1. Go to AWS Console SQS dashboard at https://console.aws.amazon.com/sqs/
2. In the left navigation panel select **Queues**, select effected **Queue**.
3. Click on **Edit** button.
4. Under **Encryption** section, select **Enable encryption**.
5. Under **Customer master key**, Select a custom CMK or enter an existing CMK ARN.
6. Click on **Save changes**.

Using AWS CLI:

```
aws sqs set-queue-attributes --queue-url [Queue_url] --attributes
"KmsMasterKeyId"="[keyArn]"
```

Note: For more details on command, please refer to https://docs.aws.amazon.com/cli/latest/reference/sqs/set-queue-attributes.html

Reference

- https://docs.aws.amazon.com/AWSSimpleQueueService/latest/SQSDeveloperGuide/sqs-server-side-encryption.html

Control ID - 293: Ensure ECR repository policy is not set to public

Criticality: HIGH

Specification

Amazon Elastic Container Registry (ECR) is a managed Docker registry service that makes it easy for DevOps teams to store, manage and deploy Docker container images. An ECR repository is a collection of Docker images available on AWS cloud.

Rationale

An ECR policy which makes ECR repository publicly accessible is not recommended. Setting the policy and updating the Permissions in order to protect against unauthorized access is recommended.

By default, only the repository owner has access to a repository.

Evaluation

This control ensures that ECR repository is not exposed publicly.

Remediation

Using AWS Console:

1. Go to AWS Console ECR dashboard at https://console.aws.amazon.com/ecr/
2. Select the private repository that you want to change the policy.
3. In left navigation pane, Select **Permissions** option.
4. Select **Edit policy JSON** option.
5. Set below policy
6. Click on **Save**.

Using AWS CLI:

```
aws ecr set-repository-policy --repository-name [Repository_Name] --policy-
text file://[policy.json]
```

Policy:

```
{
    "Version": "2008-10-17",
    "Statement": [
        {
            "Sid": "access-control-policy",
            "Effect": "Allow",
            "Principal": {
                "AWS": "[AWS_USER_IAM_ARN]"
            },
            "Action": [
                "ecr:GetDownloadUrlForLayer",
                "ecr:BatchGetImage",
                "ecr:BatchCheckLayerAvailability",
                "ecr:PutImage",
```

```
                    "ecr:InitiateLayerUpload",
                    "ecr:UploadLayerPart",
                    "ecr:CompleteLayerUpload",
                    "ecr:DescribeRepositories",
                    "ecr:GetRepositoryPolicy",
                    "ecr:ListImages",
                    "ecr:DescribeImages",
                    "ecr:SetRepositoryPolicy",
                    "ecr:GetLifecyclePolicy",
                    "ecr:PutLifecyclePolicy",
                    "ecr:GetLifecyclePolicyPreview",
                    "ecr:StartLifecyclePolicyPreview"
                ]
            }
        ]
    }
```

Note: In above policy, please use actions relevant to your requirements.

For more details on command, please refer to https://docs.aws.amazon.com/cli/latest/reference/ecr/set-repository-policy.html

Reference

- https://docs.aws.amazon.com/AmazonECR/latest/userguide/registry-permissions.html

Control ID - 294: Ensure Customer managed KMS key policy does not contain wildcard (*) principal

Criticality: HIGH

Specification

Key policies are the primary way to control access to AWS KMS keys. The statements in the key policy document determine who has permission to use the KMS key and how they can use it.

Rationale

Key policies are the primary way to control access to AWS KMS keys. The statements in the key policy document determine who has permission to use the KMS key and how they can use it.

This control ensures KMS key policy does not contain wildcard (*) principal

Evaluation

This control ensures KMS key policy does not contain wildcard (*) principal

Remediation

Note: KMS Policy can only changed via the AWS Console

Using AWS Console:

1. Sign in to the AWS Management Console and open the Amazon KMS console at https://console.aws.amazon.com/kms/home
2. Select the KMS Key you want to edit
3. Switch to **Key Policy** tab
4. Click on **Edit**
5. Make the necessary changes to Key policy
6. Click on **Save Changes**

Reference

- https://docs.aws.amazon.com/kms/latest/developerguide/key-policy-services.html

Control ID - 295: Ensure Cloudfront distribution ViewerProtocolPolicy is set to HTTPS

Criticality: HIGH

Specification

Amazon CloudFront is a web service that speeds up distribution of your static and dynamic web content, such as .html, .css, .js, and image files, to your users.

You can configure one or more cache behaviors in your CloudFront distribution to require HTTPS or HTTP and HTTPS for communication between viewers and CloudFront. To enable data in transit encryption, cache behaviorshould set viewer protocol policy to HTTPS only.

Rationale

CloudFront distribution viewer policy should be set to HTTPS only, as communication between user and distribution should be encrypted, which can be accessed by malicious individuals.

Evaluation

This control ensures cloudfront distribution ViewerProtocolPolicy is set to HTTPS.

Remediation

Using AWS Console:

1. Login to the AWS Management Console using https://console.aws.amazon.com/console/home.
2. Navigate to **CloudFront.**.
3. Click on the CloudFront Distribution to be remediated.
4. Go to the **Behaviours** pane.
5. Select the behavior and click on Edit button.
6. Under **Viewer Protocol Policy**, choose the option **HTTPS only or Redirect HTTP to HTTPS**.
7. Click **Save Changes** button.

Using AWS CLI:

To get the Distribution configuration and save the configuration output in `cf_config.json` file, use the following command:

```
aws cloudfront get-distribution-config \

        --id [distribution-id] \

        --output json > cf_config.json
```

To get the Etag for your config, use the following command:

```
ETAG=$(cat cf_config.json | jq -r '.ETag')
```

To get the `DistributionConfig`, use the following command:

```
cat cf_config.json | jq '.DistributionConfig' > cloudfront-config.json
```

Edit the `cloudfront_config.json` at `CallerReference.DefaultCacheBehavior.ViewerProtocolPolicy` & `CallerReference.CacheBehaviors.ViewerProtocolPolicy`.

```
vi cloudfront-config.json
```

- Edit the `cloudfront-config.json` at `ViewerProtocolPolicy`. Set value to 'redirect-to-https' or 'https-only'.
- For Ex.

  ```
  "ViewerProtocolPolicy": "https-only",
  ```

- To enter in edit mode press `"i"` key.
- To save & exit the file press "Esc" and the ":wq".

Update the behavior settings for the ETag value.

```
aws cloudfront update-distribution \

        --distribution-config file://cloudfront-config.json \

        --id [Distribution_Id] \

        --if-match $ETAG
```

Note: For more details on command, please refer to https://docs.aws.amazon.com/cli/latest/reference/cloudfront/update-distribution.html.

Note: Change all the protocol policies for all behaviors(if more than one).

Reference

- Requiring HTTPS for communication between viewers and CloudFront.
- Cache Behavior Settings.

Control ID - 303: Ensure MQ Broker logging is enabled

Criticality: MEDIUM

Specification

Amazon MQ is integrated with AWS CloudWatch Logs, a service that monitors, stores and accesses your log files from a variety of sources within your AWS account. Once the Log Exports feature is enabled, Amazon MQ publish general and audit logs to AWS CloudWatch Logs, allowing you to maintain continuous visibility into your brokers activity and meet compliance requirements when it comes to auditing.

Rationale

Monitoring is an important part of maintaining the reliability, availability, and performance of your AWS solutions. You should collect monitoring data from all of the parts of your AWS solution so that you can more easily debug a multi-point failure if one occurs.

Evaluation

Ensure that MQ Broker logging is enabled.

Remediation

Using AWS Console:

1. Go to AWS console at https://console.aws.amazon.com/amazon-mq/
2. In the navigation panel, under **Amazon MQ** , click Brokers.
3. Choose the **Brokers** that you want to Examine
4. Click the Edit button from the dashboard top menu to access the broker configuration panel
5. Within CloudWatch Logs section, select General and Audit checkboxes to enable log publishing to Amazon CloudWatch Logs.
6. Click on Schedule Modifications button.
7. Select suitable option to apply scheduled modifications.
8. Click Apply to save and apply your configuration changes.

Using AWS CLI:

To list brokers, use following command

```
aws mq list-brokers
```

Run update-broker command (OSX/Linux/UNIX) to enable Log Exports feature

```
aws mq update-broker \

        --broker-id [broker-id]        \

        --logs Audit=true,General=true
```

To reboot broker, use following command

```
aws mq reboot-broker \
```

```
--broker-id [broker-id]
```

Reference

- Logging and monitoring Amazon MQ brokers

Control ID - 305: Ensure ECR Image Tags are immutable

Criticality: MEDIUM

Specification

Amazon ECR image tag immutability is used to prevent image tags from being overwritten by new image tags.

Rationale

Amazon ECR image tag immutability is used to prevent image tags from being overwritten by new image tags.

After a repository is configured with tag immutability, an ImageTagAlreadyExistsException error is returned if you attempt to push an image with a tag that is already in the repository.

Evaluation

This control ensures that ECR image tag immutability is enabled for ECR repositories.

Remediation

Using AWS Console:

1. Go to AWS Console ECR dashboard at https://console.aws.amazon.com/ecr/
2. Select the private repository that you want to reconfigure.
3. Select **Edit** option.
4. Enable **Tag immutability**.
5. Click on **Save**.

Using AWS CLI:

```
aws ecr put-image-tag-mutability --repository-name [REPOSITORY_NAME] --image-tag-mutability IMMUTABLE
```

Note: For more details on command, please refer to https://docs.aws.amazon.com/cli/latest/reference/ecr/put-image-tag-mutability.html

Reference

- https://docs.aws.amazon.com/AmazonECR/latest/userguide/image-tag-mutability.html

Control ID - 312: Ensure container insights are enabled on ECS cluster

Criticality: MEDIUM

Specification

ECS Container Insights uses CloudWatch to collect, aggregate, and summarize metrics and logs from your containerized applications and microservices.

Rationale

CloudWatch collects metrics for many resources, such as CPU, memory, disk, and network. Container Insights also provides diagnostic information, such as container restart failures, to help you isolate issues and resolve them quickly.

You can also set CloudWatch alarms on metrics that Container Insights collects.

Evaluation

This control ensures that AWS ECS has container insights enabled.

Remediation

Using AWS CLI:

```
aws ecs update-cluster-settings --cluster [CLUSTER_NAME] --settings
name=containerInsights,value=enabled
```

For more details on command, please refer to
https://docs.aws.amazon.com/cli/latest/reference/ecs/update-cluster-settings.html

Reference

- https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/deploy-container-insights-ECS-cluster.html
- https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/Container-Insights-metrics-ECS.html

Control ID - 313: Ensure CloudWatch Log Group has a retention period set to 7 days or greater

Criticality: LOW

Specification

This control ensures that Cloudwatch log groups have retention period set.

Rationale

Cloudwatch log groups will store data infinite number of days, but we need to remember that we pay for log storage.While the costs are not high, this is one of those services that can quietly sneak up on you and end up costing a fair amount every month. According to the requirement we can set the retention policy.

Evaluation

This control ensures that Cloudwatch log groups have retention period set.

Remediation

Using AWS Console:

1. Go to AWS Console Cloudwatch dashboard at https://console.aws.amazon.com/cloudwatch/
2. In the left navigation panel under **Logs**, select **Log groups**.
3. Select the Log group that you want to reconfigure.
4. Select **Actions** dropdown.
5. Select the **Edit retention setting**, select retention days from the dropdown.
6. Click on **Save**.

Using AWS CLI:

```
aws logs put-retention-policy --log-group-name [log_group_name] --retention-
in-days [no_of_days]
```

Note: For more details on command, please refer to
https://docs.aws.amazon.com/cli/latest/reference/logs/put-retention-policy.html

Reference

- https://docs.aws.amazon.com/cli/latest/reference/logs/put-retention-policy.html
- https://docs.aws.amazon.com/AmazonCloudWatch/latest/logs/Working-with-log-groups-and-streams.html

Control ID - 314: Ensure that CloudFront Distribution has WAF enabled

Criticality: HIGH

Specification

Amazon CloudFront is a web service that speeds up distribution of your static and dynamic web content, such as .html, .css, .js, and image files, to your users.

AWS WAF is a web application firewall that lets you monitor the HTTP and HTTPS requests that are forwarded to CloudFront, and lets you control access to your content. Based on conditions that you specify, such as the IP addresses that requests originate from or the values of query strings, CloudFront responds to requests either with the requested content or with an HTTP 403 status code.

Rationale

If AWS Cloudfront is WAF enabled, you will be able to block any malicious requests made to your Cloudfront Network based on the conditions defined in the WAF Web Access Control List (ACL).

Evaluation

This control ensures CloudFront Distribution should be WAF enabled.

Remediation

Using AWS Portal:

1. Prerequisite: Create WAF (web ACL) for Cloudfront distributions.
2. Login to the AWS Management Console using https://console.aws.amazon.com/console/home.
3. Navigate to **CloudFront.**.
4. Click on the CloudFront Distribution to be remediated.
5. Go to the **General** tab. Click on edit button in Settings pane.
6. Select web ACL from the dropdown for **Amazon WAF web ACL**.
7. Click **Save Changes** button.

Using AWS CLI:

Update cloudfront distribution using following steps.

```
aws cloudfront get-distribution-config \

        --id [Distribution_Id] \

        --query 'DistributionConfig' > cloudfront_distribution_config.json
```

You need to edit this cloudfront_distribution_config.json file and add web ACL ARN value as follows

- edit file

  ```
  vi cloudfront_distribution_config.json
  ```

- To enter in edit mode press `"i"` key.
- Add web ACL arn value

  ```
  "WebACLId": "[web-ACL-arn-id]",
  ```

- To save & exit the file press "Esc" and the ":wq".

Get ETag for the distribution.

```
aws cloudfront get-distribution-config \

        --id [Distribution_Id] \

        --query 'ETag'
```

Update the settings for the ETag value.

```
aws cloudfront update-distribution \

        --distribution-config file://cloudfront_distribution_config.json \
```

```
--id [Distribution_Id] \

--if-match [ETag-value]
```

Reference

- Distribution Settings.
- Using AWS WAF to control access to your content.

Control ID - 315: Ensure MQ Broker is not publicly exposed

Criticality: HIGH

Specification

Amazon MQ is a managed message broker service that makes it easy to migrate to a message broker in the cloud. A message broker allows software applications and components to communicate using various programming languages, operating systems, and formal messaging protocols. Brokers created without public accessibility cannot be accessed from outside of your VPC. Public Amazon MQ brokers can be accessed directly, outside of a VPC, allowing every EC2 on the Internet to reach your brokers through their public endpoints.

Rationale

Brokers created without public access will greatly reduces your broker's susceptibility to DDoS attacks from the internet. This will also decrease the opportunity for malicious activity such as cross-site scripting and clickjacking attacks.

Evaluation

Ensure that the AWS MQ broker is not publicly exposed.

Remediation

Using AWS Console:

You can restrict Public accessibility only while creating New MQ

1. Go to AWS MQ console at https://console.aws.amazon.com/amazon-mq/
2. In the Dashbord click on **Get started** button for Create Brokers.
3. In the MQ broker Select broker engine, click on Next. Select Deployment mode and storage type.
4. Click on next and got to configure settings screen. Enter broker name. Add Active MQ access details.
5. Expand the Advanced settings.
6. Select **Public accessibility** as No.
7. Click on Create broker to create new broker which in not publicly exposed

Using AWS CLI:

Run create-broker command using --no-publicly-accessible parameter:

```
aws mq create-broker \

        --region [region] \

        --broker-name [broker-name] \

        --configuration Id="[id]",Revision=1  \

        --deployment-mode SINGLE_INSTANCE \

        --engine-type ACTIVEMQ \

        --engine-version [version] \

        --host-instance-type [instance-type] \

        --security-groups "[security-group-id]" \

        --subnet-ids "[subnet-id]" \

        --users ConsoleAccess=true,Username="[username]",Password="[password]"
\

        --no-publicly-accessible
```

Reference

- Security best practices for Amazon MQ
- Create broker cli reference

Control ID - 318: Ensure API Gateway has X-Ray Tracing enabled

Criticality: MEDIUM

Specification

X-Ray is used trace and analyze user requests as they travel through your APIs to the underlying services. API Gateway supports X-Ray tracing for all API Gateway endpoint types: Regional, edge-optimized, and private.

Currently X-Ray Tracing is supported only for REST APIs in Api Gateway

Rationale

X-Ray is used trace and analyze user requests as they travel through your APIs to the underlying services. API Gateway supports X-Ray tracing for all API Gateway endpoint types: Regional, edge-optimized, and private.

Currently X-Ray Tracing is supported only for REST APIs in Api Gateway

This control ensures API Gateway has X-Ray Tracing enabled for API Gateway Apis

Evaluation

This control ensures API Gateway Api stages have X-Ray Tracing enabled

Remediation

Using AWS Console:

1. Sign in to the AWS Management Console and open the Amazon API Gateway console at https://console.aws.amazon.com/apigateway/
2. Click on the name of the API to edit
3. In the left navigation panel, click **Stages**
4. Select the API Stage you want to edit
5. On the **Stage** Editor panel, select **Logs** tab to access the stage settings
6. Under **X-Ray Tracing**, Check **Enable X-Ray Tracing**
7. Click on **Save Changes**

Using AWS CLI:

To enable X-Ray tracing on API Gateway Stages, use the below command:

```
aws apigateway update-stage --rest-api-id [REST_API_ID] --stage-name
[API_STAGE_NAME] --patch-operations op=replace,path=/tracingEnabled,value=true
```

Note: For more details on command, please refer to https://docs.aws.amazon.com/cli/latest/reference/apigateway/update-stage.html

Reference

- https://docs.aws.amazon.com/xray/latest/devguide/aws-xray.html

Control ID - 319: Ensure Global Accelerator has flow logs enabled

Criticality: MEDIUM

Specification

Global Accelerator is a networking service that sends traffic through AWS's global network enabling global access to your web apps. Flow logs enable you to capture information about the IP address traffic going to and from network interfaces in your accelerator in AWS Global Accelerator. Flow log data is published to Amazon S3, where you can retrieve and view your data after you've created a flow log.

Rationale

Enabling flow logs you can troubleshoot why specific traffic is not reaching an endpoint, which in turn helps you diagnose overly restrictive security group rules. You can also use flow logs as a security tool to monitor the traffic that is reaching your endpoints.

Evaluation

This control ensures that Global Accelerator has flow logs enabled

Remediation

Using AWS CLI:

To create an Amazon S3 bucket for your flow logs, use the following command:

```
aws s3api create-bucket \

        --bucket [s3-bucket-name] \

        --region [region]
```

To enable Flow logs in AWS Global Accelerator with the Amazon S3 bucket name and prefix that you want to use for your log files, use the following command:

```
aws globalaccelerator update-accelerator-attributes \

        --accelerator-arn [global-accelerator-arn] \

        --region [region] \

        --flow-logs-enabled \

        --flow-logs-s3-bucket [s3-bucket-name] \

        --flow-logs-s3-prefix [s3-bucket-prefix]
```

Reference

- Flow logs in AWS Global Accelerator
- Logging and monitoring in AWS Global Accelerator

Control ID - 321: Ensure that CodeBuild Project encryption is not disabled

Criticality: HIGH

Specification

Encryption is an important part of CodeBuild security. CodeBuild comes with 3 different types of encryption :

- **Encryption of data at-rest**: Build artifacts, such as a cache, logs, exported raw test report data files, and build results
- **Encryption of data in-transit**: All communication between customers and CodeBuild and between CodeBuild and its downstream dependencies
- **Build artifact encryption**: Build Output artifacts

Rationale

Encryption is an important part of CodeBuild security. CodeBuild comes with 3 different types of encryption :

- **Encryption of data at-rest**: Build artifacts, such as a cache, logs, exported raw test report data files, and build results
- **Encryption of data in-transit**: All communication between customers and CodeBuild and between CodeBuild and its downstream dependencies
- **Build artifact encryption**: Build Output artifacts

This control ensures that CodeBuild Project encryption is not disabled

Evaluation

This control ensures that CodeBuild Project encryption is not disabled

Remediation

Note: This control does not evaluates resource with no artifacts set up

Using AWS Console:

1. Sign in to the AWS Management Console and open the Amazon CodeBuild console at https://console.aws.amazon.com/codesuite/codebuild/home
2. Under **Build projects**, select the project you want to edit
3. Switch **Build Details** tab, scroll down to **Artifacts**
4. Click on **Edit**
5. Uncheck **Disable artifact encryption**
6. Click on **Update Artifacts**

Using AWS CLI:

To enable artifact encryption on CodeBuild Projects, use the below command:

```
aws codebuild update-project --name "[PROJECT-NAME]" --artifacts {"\"type\":
\"S3\", \"location\": \"[BUCKET-NAME]\", \"encryptionDisabled\": false"}
```

Note: For more details on command, please refer to https://docs.aws.amazon.com/cli/latest/reference/ec2/modify-vpc-endpoint-service-configuration.html

Reference

- https://docs.aws.amazon.com/codebuild/latest/userguide/security-encryption.html

Control ID - 322: Ensure Instance Metadata Service Version 1 is not enabled

Criticality: MEDIUM

Specification

It is recommended to not enable Instance Metadata Service Version 1 as it is prone to local misconfigurations like open proxies, open NATs and routers and server-side reflection vulnerabilities.

Rationale

Instance metadata is data that's related to an Amazon Elastic Compute Cloud (Amazon EC2) instance that applications can use to configure or manage the running instance. IMDS, or Instance Metadata Service, is an on-instance component that makes instance metadata accessible to our code. Code can access instance metadata from a running instance using one of two methods: Instance Metadata Service Version 1 (IMDSv1) or Instance Metadata Service Version 2 (IMDSv2).

Evaluation

This control ensures that EC2 Instance Metadata Service Version 1 is not enabled.

Remediation

To disable Instance Metadata Service Version 1 while creating new EC2 instance:

Note: We can not modify instance metadata options for existing instances using AWS console

Using AWS Console:

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. Click on **Launch instances**.
3. Choose **AMI**.
4. Choose **Instance Type** click on **Next: Configure Instance Details**.
5. On the **Configure Instance Details** page, under **Advanced Details**, for **Metadata accessible**, select **Disabled**.
6. Click on **Review and Launch**.
7. Click on **Launch**.

To modify instance metadata options for existing instances using AWS CLI:

```
aws ec2 modify-instance-metadata-options --instance-id [Instance_ID] --http-endpoint disabled
```

Note: For more details on command, please refer to https://docs.aws.amazon.com/cli/latest/reference/ec2/modify-instance-metadata-options.html

Reference

- https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/configuring-instance-metadata-options.html

Control ID - 323: Ensure MSK Cluster logging is enabled

Criticality: HIGH

Specification

Consistent cluster logging helps to determine if a request was made with root or AWS Identity and Access Management (IAM) user credentials and whether the request was made with temporary security credentials for a role or federated user.

Rationale

Amazon MSK enables to build and run applications that use Apache Kafka to process streaming data. It also provides a control-plane for advanced operations, for example, creating, updating, and deleting clusters.

Broker logs enable to troubleshoot Apache Kafka applications and to analyze their communications with MSK cluster. We can configure new or existing MSK cluster to deliver INFO-level broker logs to one or more of the following types of destination resources: a CloudWatch log group, an S3 bucket, a Kinesis Data Firehose delivery stream.

Evaluation

This control ensures that MSK Cluster logging is enabled

Remediation

1. Go to AWS Console MSK dashboard at https://console.aws.amazon.com/msk/home
2. In the left navigation panel select **Clusters** under the **MSK Clusters**.
3. Click on the cluster to be remediated.
4. Click on **Actions** select **Edit log delivery**.
5. Scroll down to **Broker log delivery** section.
6. Specify the destinations to which you want Amazon MSK to deliver your broker logs.
7. click on **Save changes**.

Using AWS CLI:

To specify destination as Amazon CloudWatch Logs

```
aws kafka update-monitoring --cluster-arn [cluster_ARN] --current-version
[Version] --logging-info
BrokerLogs={CloudWatchLogs={Enabled=true,LogGroup=[LogGroup_Name]}}
```

To specify destination as Amazon Kinesis Data Firehose

```
aws kafka update-monitoring --cluster-arn [cluster_ARN] --current-version
[Version] --logging-info
BrokerLogs={Firehose={DeliveryStream=[DeliveryStream_Name],Enabled=true}}
```

To specify destination as Amazon S3

```
aws kafka update-monitoring --cluster-arn [cluster_ARN] --current-version
[Version] --logging-info BrokerLogs={S3={Bucket=[Bucket_Name],Enabled=true}}
```

Note: You need to describe an MSK cluster to find its version

```
aws kafka describe-cluster --cluster-arn [cluster_ARN]
```

Note: For more details on command, please refer to
https://docs.aws.amazon.com/cli/latest/reference/kafka/update-monitoring.html

Reference

- https://docs.aws.amazon.com/msk/latest/developerguide/msk-logging.html
- https://docs.aws.amazon.com/cli/latest/reference/kafka/update-monitoring.html

Control ID - 324: Ensure MSK Cluster encryption at rest and in transit is enabled

Criticality: HIGH

Specification

Amazon MSK provides data encryption options that you can use to meet strict data management requirements.

Amazon MSK uses TLS 1.2. By default, it encrypts data in transit between the brokers of your MSK cluster.

Amazon MSK uses AWS KMS keys to encrypt your data at rest. You can use AWS KMS to create and manage KMS keys.

Rationale

Encrypt data as it travels between brokers within the cluster and as it travels between Apache Kafka clients and the cluster.

Evaluation

This control ensures that MSK Cluster has enabled encryption at rest and in transit

Remediation

1. Go to AWS Console MSK dashboard at https://console.aws.amazon.com/msk/home
2. In the left navigation panel select **Clusters** under the **MSK Clusters**.
3. Click on the affected cluster from list.
4. Under **Details** tab, Go to **Security settings**.
5. Click on **Edit**.
6. Go to **Encryption** section
7. Ensure only **TLS encryption** is checked for Encryption between clients and brokers.
8. Ensure **TLS encryption** is **Enabled** for Encryption within the cluster.
9. click on **Save changes**

Note: Amazon MSK always encrypts your data at rest. When you create an MSK cluster, you can specify the AWS KMS customer master key (CMK) that you want Amazon MSK to use to encrypt your data at rest.

Reference

- https://docs.aws.amazon.com/msk/latest/developerguide/msk-encryption.html
- https://docs.aws.amazon.com/msk/latest/developerguide/msk-working-with-encryption.html

Control ID - 325: Ensure Athena Workgroups enforce configuration to prevent client disabling encryption

Criticality: MEDIUM

Specification

Amazon Athena is a serverless interactive query service that enables users to easily analyze data in Amazon S3 using standard SQL. This new feature not only makes it possible for Athena to provide support for querying encrypted data in Amazon S3, but also enables the encryption of data from Athena's query results. Each user can specify client-side settings in the Settings menu on the console. If Override client-side settings is not selected, workgroup settings are not enforced. In this case, for all queries that run in this workgroup, Athena uses the clients-side settings for query results location and encryption.

If Override client-side settings is selected, Athena uses the workgroup-wide settings for query results location and encryption. It also overrides any other settings that you specified for the query in the console, by using the API operations, or with the drivers.

Rationale

If Override Client-Side Settings is selected, the user is notified on the console that their settings have changed. If workgroup settings are enforced this way, users can omit corresponding client-side settings. Also, if you run queries in this workgroup through the command line interface, API operations, or the drivers, any settings that you specified are overwritten by the workgroup's settings.

Evaluation

This control ensures Athena Workgroup should enforce configuration to prevent client disabling encryption

Remediation

Using AWS Console

To Create Amazon Athena Workgroup

1. Open Amazon Athena Service console at https://console.aws.amazon.com/athena.
2. In the left navigation panel, choose **Workgroups**.
3. Click on the workgroup to be remediated.
4. Click on the edit button to edit the workgroup.
5. Scroll to **Settings** section, check the checkbox for **Override client-side settings**.
6. Click on Save changes button at the bottom.

Using AWS CLI

To to prevent client disabling encryption for Athena workgroup, use the following command:

```
aws athena update-work-group \

        --work-group [workgroup-name] \

        --configuration-updates EnforceWorkGroupConfiguration=true
```

Reference

- Amazon Athena workgroup reference
- Workgroup Settings Override Client-Side Settings

Control ID - 326: Ensure Elasticsearch Domain enforces HTTPS

Criticality: HIGH

Specification

Amazon Elasticsearch Service now lets you configure your domains to require that all traffic be submitted over HTTPS so that you can ensure that communications between your clients and your domain are

encrypted. You can also configure the minimum required TLS version to accept. This option is a useful additional security control to ensure your clients are not misconfigured.

Rationale

When you use Amazon Elasticsearch Service (Amazon ES), it allows you to build applications without setting up and maintaining your own search cluster on Amazon EC2. Amazon ES allows you to configure your domains to require that all traffic be submitted over HTTPS. This ensures communications between your clients and your domain are encrypted.

Evaluation

This control ensures that Elasticsearch Domain enforces HTTPS

Remediation

Using AWS Console

1. Open Amazon OpenSearch Service (successor to Amazon Elasticsearch Service) console at https://console.aws.amazon.com/esv3.
2. In the left navigation panel, choose **Domains**.
3. Click on the **domain** to be remediated.
4. Select Edit security configuration from Actions dropdown
5. Navigate down to **Encryption** section, add check **Require HTTPS for all traffic to the domain** to enable it.
6. Note: If Fine grained access control is disabled, then required HTTPS value can be modified. Enforce HTTPS is automatically enabled if Fine grained access control is enabled.
7. Click on **Save**.

Using AWS CLI

To modify AWS ElasticSearch domain and enforces HTTPS, use the following command:

```
aws es update-elasticsearch-domain-config \

        --domain-name [domain-Name] \

        --domain-endpoint-options EnforceHTTPS=true
```

Reference

- Data protection in Amazon OpenSearch Service
- Amazon Elasticsearch Service provides option to mandate HTTPS

Control ID - 327: Ensure Cloudfront distribution has Access Logging enabled

Criticality: HIGH

Specification

Amazon CloudFront is a web service that speeds up distribution of your static and dynamic web content, such as .html, .css, .js, and image files, to your users.

Monitoring activity is an important part of maintaining the availability and performance of CloudFront and your AWS solutions, so that you can more easily debug a multi-point failure if one occurs. CloudFront standard logs provide detailed records about every request that's made to a distribution.

Rationale

Access logs are useful for many scenarios, including security and access audits. Logging also helps in detecting and investigating potential attacks, malicious activity.

Evaluation

This control ensures AWS Cloudfront distribution has Access Logging enabled.

Remediation

Using AWS Portal:

1. Login to the AWS Management Console using https://console.aws.amazon.com/console/home.
2. Navigate to **CloudFront.**
3. Click on the CloudFront Distribution to be remediated.
4. In the **General** pane, scroll down to **Settings**. Click on the **Edit** button.
5. In the **Settings** window, scroll down to logging.
6. Select **On** radio button for **Standard Logging**.
7. Under the **S3 Bucket** option, select Amazon S3 Bucket to store logs.
8. Optionally specify prefix for logs.
9. Click **Save Changes** button.

Using AWS CLI:

Update cloudfront distribution using following steps.

```
aws cloudfront get-distribution-config \

        --id [Distribution_Id] \

        --query 'DistributionConfig' > cloudfront_distribution_config.json
```

You need to edit this cloudfront_distribution_config.json file using vi and change Logging section as follows

- edit file

  ```
  vi cloudfront_distribution_config.json
  ```

- To enter in edit mode press `"i"` key.
- Change logging as
- ```
  "Logging": {
  ```
- ```
              "Enabled": true,
  ```
- ```
              "IncludeCookies": false,
  ```
- ```
              "Bucket": "[S3_bucket_Id]",
  ```
- ```
              "Prefix": "[Log_prefix]"
              },
  ```
  ```

- To save & exit the file press "Esc" and the ":wq".

Get ETag for the distribution.

```
aws cloudfront get-distribution-config \

        --id [Distribution_Id] \

        --query 'ETag'
```

Update the logging settings for the ETag value.

```
aws cloudfront update-distribution \

        --distribution-config file://cloudfront_distribution_config.json \

        --id [Distribution_Id] \

        --if-match [ETag-value]
```

Reference

- Logging and monitoring in Amazon CloudFront.
- Configuring and using standard logs (access logs).

Control ID - 328: Ensure that EC2 instance have no public IP

Criticality: MEDIUM

Specification

It is recommended to not use public IP for instance as it is reachable from the internet.

Rationale

An Elastic IP address is a public IPv4 address, which is reachable from the internet. We can associate an Elastic IP address with our instance to enable communication with the internet. For example, this allows to connect to instance from local computer.

Evaluation

This control ensures that EC2 instance does not have public IP.

Remediation

To Disassociate Elastic IP address for an existing instance:

Using AWS Console:

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.

2. In the navigation pane, choose **Instances**.
3. Select the instance you want to remediate and choose **Actions, Networking, Disassociate Elastic IP address**.
4. On the **Disassociate Elastic IP address** detail page, click **Disassociate**.
5. To confirm disassociation, type **proceed** in the field provided.
6. Choose **disassociate**.

Using AWS CLI:

```
aws ec2 disassociate-address --association-id [Association_ID]
```

Note: For more details on command, please refer to https://docs.aws.amazon.com/cli/latest/reference/ec2/disassociate-address.html

Reference

- https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/elastic-ip-addresses-eip.html

Control ID - 329: Ensure that DMS replication instance is not publicly accessible

Criticality: HIGH

Specification

AWS Database Migration Service (AWS DMS) is a service for migrating relational databases, data warehouses, NoSQL databases and other data stores. An AWS DMS replication instance can have one public IP address and one private IP address. By disabling public access, the replication instance can communicate with a host that is in the same Amazon Virtual Private Cloud (Amazon VPC) and that can communicate with the private IP address. Or the replication instance can communicate with a host that is connected privately, for example, by VPN, VPC peering, or AWS Direct Connect.

Rationale

When your AWS DMS replication instances are publicly accessible and have public IP addresses, any machine outside the VPC can establish a connection to these instances, increasing the attack surface and the opportunity for malicious activity. Of course, the level of access to your replication instances depends on their use cases, however, for most use cases the instances should be privately accessible only from within your Amazon Virtual Private Cloud (VPC).

Evaluation

Ensure that DMS replication instance should not be publicly accessible.

Remediation

Note: Public access can be disabled for Database Migration Service replication instance, only at the time of creation of instance.

1. Go to AWS Database Migration Service (DMS) dashboard at https://console.aws.amazon.com/dms/.
2. In the navigation panel, choose **Replication instances.**
3. Click the Create replication instance button from the dashboard top menu to initiate the launch process.

4. Enter the unique instance name and select instance class, allocated storage and VPC values.
5. uncheck the checkbox for Publicly accessible.
6. Select the information for Advanced security and network configuration as per your requirement.
7. Select the information for Maintenance and Tags as per your requirements.
8. Click on Create button to launch your new Amazon DMS instance.

Using AWS CLI:

Run create-replication-instance command (OSX/Linux/UNIX) to create your new Amazon DMS replication instance

```
aws dms create-replication-instance \

        --region [region] \

        --replication-instance-identifier [instance-name] \

        --replication-instance-class [instance-class] \

        --allocated-storage [storage-required] \

        --no-publicly-accessible
```

Reference

- How can I disable public access for an AWS DMS replication DB instance?
- Public and private replication instances

Control ID - 332: Ensure Glue Data Catalog Encryption is enabled with SSE-KMS with customer-managed keys

Criticality: HIGH

Specification

When encryption is turned on, all future Data Catalog objects are encrypted. The default key is the AWS Glue AWS KMS key that is created for your account by AWS.

Rationale

If you clear this setting, objects are no longer encrypted when they are written to the Data Catalog. Any encrypted objects in the Data Catalog can continue to be accessed with the key.

Evaluation

This control ensures that AWS Glue Data Catalog Encryption is enabled with SSE-KMS with customer-managed key.

Remediation

Note: This Control evaluates Glue service for all regions collectively, i.e the control will fail if any region fails. If any region is disabled, this would be counted as a region failure.

Using AWS Console:

1. Sign in to AWS Console and navigate to https://console.aws.amazon.com/glue/home.
2. In left navigation pane, Click on **Setting** under **Data catalog** section.
3. In the next page, select **Metadata encryption** box and choose customer-managed key from the AWS KMS key dropdown.

Using AWS CLI:

```
aws glue put-data-catalog-encryption-settings --data-catalog-encryption-
settings EncryptionAtRest={CatalogEncryptionMode='SSE-KMS',SseAwsKmsKeyId=[kms
key arn]} --region [Region]
```

For command usage refer: https://docs.aws.amazon.com/cli/latest/reference/glue/put-data-catalog-encryption-settings.html

Reference

- https://docs.aws.amazon.com/glue/latest/dg/encrypt-glue-data-catalog.html

Control ID - 334: Ensure all data stored in the Sagemaker Endpoint is securely encrypted at rest

Criticality: HIGH

Specification

This control ensures that encryption of data on the storage volume attached to the ML compute instance that hosts the sagemaker endpoint is enabled

Rationale

Using Kms key to encrypt data at rest in the storage volume attached to the ML compute instance that hosts the endpoint is recommended.

Evaluation

This control ensures that encryption of data on the storage volume attached to the ML compute instance that hosts the sagemaker endpoint is enabled.

Remediation

Using AWS Console:

Note: To update an endpoint, you must create a new EndpointConfig. You must not delete an EndpointConfig that is in use by an endpoint that is live or while the UpdateEndpoint or CreateEndpoint operations are being performed on the endpoint.

1. Go to AWS Console SageMaker dashboard at https://console.aws.amazon.com/sagemaker
2. In the left navigation pane, under **Inference** click on **Endpoints**.
3. Select the Endpoint that you want to reconfigure.
4. Select **Update endpoint** option.
5. Under **Change the Endpoint configuration**, select **Create a new endpoint configuration**.
6. Under **Endpoint configuration**, provide Kms key to use for encryption under **Encryption key**.
7. Provide other configurations according to the requirements.
8. Click on **Update endpoint**.

Using AWS CLI:

```
aws sagemaker update-endpoint --endpoint-name [ENDPOINT_NAME] --endpoint-
config-name [ENDPOINT_CONFIG_NAME]
```

For more details on command, please refer to
https://docs.aws.amazon.com/cli/latest/reference/sagemaker/update-endpoint.html

Note: ENDPOINT_CONFIG_NAME is the new configuration which has encryption enabled in endpoint configuration section.

Reference

- https://docs.aws.amazon.com/sagemaker/latest/APIReference/API_CreateEndpointConfig.html#sagemaker-CreateEndpointConfig-request-KmsKeyId

Control ID - 338: Ensure that load balancer is using TLS 1.2 or above

Criticality: HIGH

Specification

Elastic Load Balancing uses a Transport Layer Security (TLS) negotiation configuration, known as a security policy, to negotiate TLS connections between a client and the load balancer. A security policy is a combination of protocols and ciphers. The protocol establishes a secure connection between a client and a server and ensures that all data passed between the client and your load balancer is private.

Rationale

TLS 1.2 is more secure than the previous cryptographic protocols such as TLS 1.1. Essentially, TLS 1.2 keeps data being transferred across the network more secure.

Evaluation

This control ensures that load balancer is using TLS 1.2 or above

Remediation

Using AWS Console

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, under **Load Balancing**, choose **Load Balancers**.
3. Select the load balancer to be remediated.

4. Click **Listeners** tab, below the list.
5. Select the check box for the TLS/HTTPS listener and then choose **Edit**.
6. Under **Secure listener settings**, for **Security policy** choose TLS 1.2 or above security policy.
7. Click **Save changes**.

Using AWS CLI

To modify listener security policy, use the following command:

```
aws elbv2 modify-listener \

        --listener [listener-arn] \

        --ssl-policy [ssl-policy]
```

Reference

- TLS listeners for your Network Load Balancer
- Update a TLS listener for your Network Load Balancer

Control ID - 339: Ensure EBS default encryption is enabled with customer managed key

Criticality: HIGH

Specification

Use Amazon EBS encryption as a straight-forward encryption solution for your EBS resources associated with your EC2 instances. With Amazon EBS encryption, you aren't required to build, maintain, and secure your own key management infrastructure. Amazon EBS encryption uses AWS KMS keys when creating encrypted volumes and snapshots.

Rationale

Encryption operations occur on the servers that host EC2 instances, ensuring the security of both data-at-rest and data-in-transit between an instance and its attached EBS storage.

Evaluation

This control ensures that EBS default encryption is enabled with customer managed key

Remediation

Note: This Control evaluates EC2 service for all regions collectively, i.e the control will fail if any region fails. If any region is disabled, this would be counted as a region failure.

Using AWS Console:

1. Sign in to AWS Console and navigate to https://console.aws.amazon.com/ec2/v2/home.
2. Click on **EBS encryption** under **Account attributes** section.

3. In the next page, **Enable** encryption and select customer-managed key from **Default encryption key** dropdown.

Using AWS CLI:

For enabling default encryption:
```
aws ec2 enable-ebs-encryption-by-default --region [Region]
```
For setting customer-managed kms key:
```
aws ec2 modify-ebs-default-kms-key-id --kms-key-id [kms key ARN] --region [Region]
```
For command usage refer:

- https://docs.aws.amazon.com/cli/latest/reference/ec2/enable-ebs-encryption-by-default.html
- https://docs.aws.amazon.com/cli/latest/reference/ec2/modify-ebs-default-kms-key-id.html

Reference

- https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EBSEncryption.html

Control ID - 342: Ensure that EMR clusters with Kerberos have Kerberos Realm set

Criticality: MEDIUM

Specification

It is recommended to enable Kerberos authentication to provide strong authentication so that passwords or other credentials aren't sent over the network in an unencrypted format.

Rationale

Amazon EMR release version 5.10.0 and later supports Kerberos, which is a network authentication protocol created by the Massachusetts Institute of Technology (MIT). Kerberos uses secret-key cryptography to provide strong authentication so that passwords or other credentials aren't sent over the network in an unencrypted format. In Kerberos, services and users that need to authenticate are known as principals. Principals exist within a Kerberos realm. Within the realm, a Kerberos server known as the key distribution center (KDC) provides the means for principals to authenticate. The KDC does this by issuing tickets for authentication. The KDC maintains a database of the principals within its realm, their passwords, and other administrative information about each principal. A KDC can also accept authentication credentials from principals in other realms, which is known as a cross-realm trust. In addition, an EMR cluster can use an external KDC to authenticate principals.

Evaluation

This control ensures that EMR clusters with Kerberos have Kerberos Realm set

Remediation

Using AWS Console:

Note : We cannot modify an existing EMR cluster, new EMR cluster should be created with required configuration.

To configure a security configuration with Kerberos authentication enabled, follow the below procedure:

1. Sign in to AWS Console and navigate to https://console.aws.amazon.com/elasticmapreduce/home.
2. In the Navigation pane, choose **Security configurations**.
3. Click on **Create** button.
4. Check **Enable Kerberos authentication** checkbox, select **Provider** as **Cluster dedicated KDC** or **External KDC**.
5. Click **Create** button.

To Create and Configure an EMR cluster with Secuirty Configuration having Kerberos authentication enabled.

1. Sign in to AWS Console and navigate to https://console.aws.amazon.com/elasticmapreduce/home.
2. In the Navigation pane, choose **Clusters**.
3. Click on **Create cluster** button.
4. Select **Go to advanced options**.
5. In steps 1 - 3, configure options according to the requirements.
6. In **step 4**, select **Security Configuration**.
7. Select above created security configuration with Kerberos authentication enabled.
8. Set **Realm** and **KDC admin password**.
9. Click **Create cluster** button.

Reference

- https://docs.aws.amazon.com/emr/latest/ManagementGuide/emr-kerberos-configure.html
- https://docs.aws.amazon.com/emr/latest/ManagementGuide/emr-kerberos-configure-settings.html#emr-kerberos-security-configuration

Control ID - 347: Ensure that direct internet access is disabled for an Amazon SageMaker Notebook Instance

Criticality: MEDIUM

Specification

SageMaker is a fully-managed AWS service that enables developers and data engineers to quickly and easily build, train and deploy machine learning models at any scale. An AWS SageMaker notebook instance is a fully managed ML instance that is running the Jupyter Notebook open-source web application.

Rationale

By default, notebook allows direct internet access, which is not recommended. To stop internet access, we can specify a VPC for notebook instance. Now notebook instance cannot train or host models unless we have a NAT gateway or interface endpoint configured in the VPC.

Evaluation

This control ensures that AWS SageMaker notebook instances has direct internet access disabled.

Remediation

Using AWS Console:

Note : Internet Access for already created SageMaker notebook instance cannot be changed.

1. Go to AWS Console SageMaker dashboard at https://console.aws.amazon.com/sagemaker
2. In the left navigation pane, under **Notebook** click on **Notebook Instances**.
3. Click on **Create notebook instance**.
4. Under **Network** section, Please select **VPC, Subnet and Security group** as per requirement.
5. Under **Direct internet access**, please **Disabled** option.
6. Click on **Create notebook instance**.

Using AWS CLI:

```
aws sagemaker create-notebook-instance --notebook-instance-name
[Notebook_Instance_Name] --instance-type [Instance_Type] --role-arn [Role_Arn]
--subnet-id [Subnet_Id] --security-group-ids [Security_Group_Id] --direct-
internet-access Disabled
```

For more details on command, please refer to
https://docs.aws.amazon.com/cli/latest/reference/sagemaker/create-notebook-instance.html

Note: Once direct internet access is disabled, notebook instance cannot train or host models. So please make sure to configure a NAT gateway or an interface endpoint to access notebook.

Reference

- https://docs.aws.amazon.com/sagemaker/latest/dg/appendix-notebook-and-internet-access.html#appendix-notebook-and-internet-access-default
- https://aws.amazon.com/blogs/machine-learning/understanding-amazon-sagemaker-notebook-instance-networking-configurations-and-advanced-routing-options/

Control ID - 348: Ensure that VPC Endpoint Service is configured for Manual Acceptance

Criticality: MEDIUM

Specification

Upon creation of an endpoint service, service users with permission can create an interface endpoint or Gateway Load Balancer endpoint to connect to the service.
If specified that acceptance is required for connection requests, you must manually accept or reject endpoint connection requests to your endpoint service. After an endpoint is accepted, it becomes available. Be aware that it can take time for a validation status change to be completed and the state to be available.

Rationale

Upon creation of an endpoint service, service users with permission can create an interface endpoint or Gateway Load Balancer endpoint to connect to the service.
If specified that acceptance is required for connection requests, you must manually accept or reject endpoint connection requests to your endpoint service. After an endpoint is accepted, it becomes available. Be aware that it can take time for a validation status change to be completed and the state to be available.

This control ensures that VPC Endpoint Service is configured for Manual Acceptance

Evaluation

This control ensures that VPC Endpoint Service is configured for Manual Acceptance

Remediation

Using AWS Console:

1. Sign in to the AWS Management Console and open the Amazon VPC console at https://console.aws.amazon.com/vpc/
2. In the left navigation panel, click **Endpoint services**
3. Select the Service you want to edit
4. Click on the **Actions** drop down, select **Modify endpoint acceptance setting**
5. Under **Modify endpoint acceptance settings**, Check **Acceptance required**
6. Click on **Save Changes**

Using AWS CLI:

To enable manual acceptance on VPC Endpoint Services, use the below command:

```
aws ec2 modify-vpc-endpoint-service-configuration --service-id [VPC-SERVICE-ID] --acceptance-required
```

Note: For more details on command, please refer to https://docs.aws.amazon.com/cli/latest/reference/ec2/modify-vpc-endpoint-service-configuration.html

Reference

- https://docs.aws.amazon.com/vpc/latest/privatelink/endpoint-service.html

Control ID - 349: Ensure that CloudFormation stacks are sending event notifications to an SNS topic

Criticality: MEDIUM

Specification

AWS CloudFormation is a service that helps you model and set up your AWS resources so that you can spend less time managing those resources and more time focusing on your applications that run in AWS.

Amazon Simple Notification Service (Amazon SNS) is a managed service that provides message delivery from publishers to subscribers (also known as producers and consumers). Publishers communicate asynchronously with subscribers by sending messages to a topic, which is a logical access point and communication channel. Clients can subscribe to the SNS topic and receive published messages using a supported endpoint type, such as Amazon Kinesis Data Firehose, Amazon SQS, AWS Lambda, HTTP, email, mobile push notifications, and mobile text messages (SMS).

Rationale

Amazon SNS enables you to modernize your applications and decouple them into smaller, independent components that are easier to develop, deploy, and maintain. Leveraging a pub/sub event-driven architecture for your application improves performance, reliability, and allows each component to scale independently.

Evaluation

This control ensures that CloudFormation stacks are sending event notifications to an SNS topic

Remediation

Using AWS Console

1. Open AWS CloudFormation console at http://console.aws.amazon.com/cloudformation/.
2. In the left navigation panel, choose **Stacks**.
3. Select the active stack that to be remediated.
4. In the stack details pane, click **Update** button.
5. Click **Next** button until you reach **Configure stack options** page.
6. Under **Advanced options**, select **Notification options**.
7. Perform one of the following:

- Select an existing **SNS topic ARN** from the drop-down menu.
- Click **Create new SNS topic**, Enter SNS topic name and Email adderess where you want to receive notifications.

- Click **Next** button and review the stack information and the changes.
- If your template contains IAM resources, select checkbox I acknowledge that this template may create IAM resources with custom names.
- Click **Update stack**.

Using AWS CLI

To create SNS topic, use the following command:

```
aws sns create-topic \

        --name [sns-topic-name] \

        --region [region]
```

To associate CloudFormation stack with SNS topic, use the following command:

```
aws cloudformation update-stack \

        --stack-name [stack-name] \

        --region [region] \

        --use-previous-template \

        --notification-arn "[sns-topic-arn]" \

        --capabilities CAPABILITY_NAMED_IAM
```

Note: For more details on command, please refer to https://docs.aws.amazon.com/cli/latest/reference/cloudformation/update-stack.html

Note: CLI command may vary for stack to stack please prefer remediating through AWS console.

Reference

- Working with stacks
- Amazon Simple Notification Service-backed custom resources
- AWS::CloudFormation::Stack

Control ID - 350: Ensure that detailed monitoring is enabled for EC2 instances

Criticality: HIGH

Specification

By enabling detailed monitoring for instance, data is available in 1-minute periods and we can also get aggregated data across groups of similar instances.

Rationale

By default, instance is enabled for basic monitoring. We can optionally enable detailed monitoring. After enabling detailed monitoring, the Amazon EC2 console displays monitoring graphs with a 1-minute period for the instance.

Basic monitoring: Data is available automatically in 5-minute periods.

Detailed monitoring: Data is available in 1-minute periods. For the instances where we've enabled detailed monitoring, we can also get aggregated data across groups of similar instances.

Evaluation

This control ensures that detailed monitoring is enabled for EC2 instances.

Remediation

To enable detailed monitoring for an existing instance:

Using AWS Console:

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Instances**.
3. Select the instance you want to remediate and choose **Actions, Monitoring, Manage detailed monitoring**.
4. On the **Detailed monitoring** detail page, for **Detailed monitoring,** select the **Enable** check box.
5. Choose **Save**.

Using AWS CLI:

```
aws ec2 monitor-instances --instance-ids [INSTANCE_ID]
```

Note: For more details on command, please refer to https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-cloudwatch-new.html#enable-detailed-monitoring

Reference

- https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-cloudwatch-new.html

Control ID - 351: Ensure that Elastic Load Balancers use SSL certificates provided by AWS Certificate Manager

Criticality: HIGH

Specification

A listener is a process that checks for connection requests, using the protocol and port that you configure. The rules that you define for a listener determine how the load balancer routes requests to its registered targets. You define a listener when you create your load balancer, and you can add listeners to your load balancer at any time.

Rationale

This feature enables traffic encryption between your load balancer and the clients that initiate SSL or TLS sessions.

Evaluation

This control ensures that Elastic Load Balancer(s) uses SSL certificates provided by AWS Certificate Manager

Remediation

Using AWS Console:

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, under **Load Balancing**, choose **Load Balancers**.
3. Select the load balancer, and select **Listeners** tab.
4. Select the check box for the listener and then choose **Edit**.
5. For **Protocol**, select HTTPS or TLS protocol and keep the default port or enter a different port.
6. For Default actions, do one of the following:

- Choose **Add action**, **Forward to** and choose a target group.
- Choose **Add action**, **Redirect to** and provide the URL for the redirect.
- Choose **Add action**, **Return fixed response** and provide a response code and optional response body.

- Select **Security policy**, from the drop-down menu.
- For **Default SSL certificate**, choose **From ACM** and choose the certificate.
- Click **Save changes**.
- To add listener to your load balancer perform the following:

1. Select a load balancer, and select **Listeners**, click **Add listener**.

2. For **Protocol**, select HTTPS or TLS protocol and keep the default port or enter a different port.
3. For Default actions, do one of the following:

- Choose **Add action**, **Forward to** and choose a target group.
- Choose **Add action**, **Redirect to** and provide the URL for the redirect.
- Choose **Add action**, **Return fixed response** and provide a response code and optional response body.

- Select **Security policy**, from the drop-down menu.
- For **Default SSL certificate**, choose **From ACM** and choose the certificate.
- Click **Add**.

Using AWS CLI:

To add SSL certificate for the HTTPS and TLS listeners, use the following command:

```
aws elbv2 modify-listener \

        --listener-arn [listener-arn] \

        --certificates CertificateArn=[certificate-arn]
```

To create HTTPS|TLS listener, use the following command:

```
aws elbv2 create-listener \

        --load-balancer-arn [load-balancer-arn] \

        --protocol [HTTPS|TLS] --port 443 \

        --certificates CertificateArn=[certificate-arn] \

        --ssl-policy [ssl-policy] \

        --default-actions Type=forward,TargetGroupArn=[target-group-arn]
```

Reference

- SSL certificates
- Create an HTTPS listener for your Application Load Balancer
- Update a TLS listener for your Network Load Balancer

Control ID - 354: Ensure that ALB drops HTTP headers

Criticality: HIGH

Specification

Elastic Load Balancing requires that message header names conform to the regular expression [-A-Za-z0-9]+, which describes all registered internet message headers. Each name consists of alphanumeric characters or hyphens. Therefore, Amazon Elastic Load Balancing service introduced `routing.http.drop_invalid_header_fields.enabled` Indicates whether HTTP headers with

header fields that are not valid are removed by the load balancer (true), or routed to targets (false). The default is false.

Rationale

By enabling Drop Invalid Header Fields, HTTP headers with header fields that are not valid are removed by the Application Load Balancer instead of being routed to the associated targets.

Evaluation

This control ensures that ALB drops HTTP headers

Remediation

Using AWS Console:

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, under **Load Balancing**, choose **Load Balancers**.
3. Select the application load balancer to be remediated.
4. Click Description tab, Under **Attributes** section, Click **Edit attributes**.
5. Select **Drop invalid header fields** checkbox to enable the Drop Invalid Header Fields.
6. Click **Save**.

Using AWS CLI:

To enable Drop invalid header fields for the load balancer, use the following command:

```
aws elbv2 modify-load-balancer-attributes \

        --region [region] \

        --load-balancer-arn [load-balancer-arn] \

        --attributes
Key=routing.http.drop_invalid_header_fields.enabled,Value=true
```

Reference

- What is an Application Load Balancer?
- Application Load Balancers
- Elastic Load Balancing features

Control ID - 355: Ensure Trail is configured to log Data events for s3 buckets

Criticality: HIGH

Specification

To log data events for all Amazon S3 buckets in your AWS account, use the default setting, All current and future S3 buckets. You can choose to log Read events, such as GetObject, Write events, such as PutObject, or both. This global setting overrides settings you configure for individual buckets.

Rationale

Choosing All current and future S3 buckets enables data event logging for all buckets currently in your AWS account and any buckets you create after you finish creating the trail. It also enables logging of data event activity performed by any user or role in your account, even if performed on a bucket that belongs to another AWS account.

Note: This control is applicable if you have configured data events using basic event selectors only.

Evaluation

This control ensures that trail is configured to log Data events for s3 buckets

Remediation

Using AWS Console:

Note: This control is applicable if you have configured data events using basic event selectors only.

1. Open the Amazon Cloudtrail console at https://console.aws.amazon.com/cloudtrail.
2. Click on the trail name that needs to be remediate.
3. Go to **Data events** section and click on edit button.
4. Check the **Data events** checkbox and click on **Switch to basic event selectors**
5. Under the **Data events: S3**, select **S3** as a **Data event source**.
6. For **All current and future S3 buckets** select both checkboxes of **Read** & **Write**.
7. Click on **Save changes**.

Reference

- https://docs.aws.amazon.com/awscloudtrail/latest/userguide/logging-data-events-with-cloudtrail.html#logging-data-events

Control ID - 357: Ensure that EC2 is EBS optimized

Criticality: LOW

Specification

It is recommended to use EBS-optimized instance, this optimization provides the best performance for your EBS volumes by minimizing contention between Amazon EBS I/O and other traffic from your instance.

Rationale

An Amazon EBS–optimized instance uses an optimized configuration stack and provides additional, dedicated capacity for Amazon EBS I/O. This optimization provides the best performance for your EBS volumes by minimizing contention between Amazon EBS I/O and other traffic from your instance.

EBS–optimized instances deliver dedicated bandwidth to Amazon EBS. When attached to an EBS–optimized instance, General Purpose SSD (gp2 and gp3) volumes are designed to deliver their baseline and burst performance 99% of the time, and Provisioned IOPS SSD (io1 and io2) volumes are designed to deliver their provisioned performance 99.9% of the time. Both Throughput Optimized HDD (st1) and Cold HDD (sc1)

guarantee performance consistency of 90% of burst throughput 99% of the time. Non-compliant periods are approximately uniformly distributed, targeting 99% of expected total throughput each hour.

Evaluation

This control ensures that EC2 instances are EBS optimized.

Remediation

To enable EBS optimization for an existing instance:

You can enable or disable optimization for an existing instance by modifying its Amazon EBS–optimized instance attribute. If the instance is running, you must stop it first.

Using AWS Console:

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, choose **Instances**.
3. Select the instance you want to remediate.
4. To stop the instance, choose **Actions, Instance state, Stop instance**. It can take a few minutes for the instance to stop.
5. With the instance still selected, choose **Actions, Instance settings, Change instance type**.
6. For **Change Instance Type,** do one of the following:

- If the instance type of your instance is Amazon EBS–optimized by default, **EBS-optimized** is selected and you can't change it. You can choose **Cancel,** because Amazon EBS optimization is already enabled for the instance.
- If the instance type of your instance supports Amazon EBS optimization, choose **EBS-optimized** and then choose **Apply.**
- If the instance type of your instance does not support Amazon EBS optimization, you can't choose **EBS-optimized.** You can select an instance type from Instance type that supports Amazon EBS optimization, choose **EBS-optimized,** and then choose **Apply.**

- Choose **Instance state, Start instance.**.

Note: For details on which instance types support EBS optimization, please refer to https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ebs-optimized.html#ebs-optimization-support

Using AWS CLI:

1. If the instance is running, use the following command to stop it:

   ```
   aws ec2 stop-instances --instance-ids [INSTANCE_ID]
   ```

2. To enable EBS optimization, use the following command:

   ```
   aws ec2 modify-instance-attribute --instance-id [INSTANCE_ID] --ebs-optimized
   ```
   For more details on commands, please refer to:

- https://docs.aws.amazon.com/cli/latest/reference/ec2/stop-instances.html
- https://docs.aws.amazon.com/cli/latest/reference/ec2/modify-instance-attribute.html

Reference

- https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ebs-optimized.html

Control ID - 358: Ensure that ECR repositories are encrypted using KMS

Criticality: HIGH

Specification

By default Amazon ECR encrypts images with Amazon S3-managed encryption keys using server side encryption. For more control over the encryption, KMS keys stored in AWS key management service.

Rationale

Amazon ECR stores images in Amazon S3 buckets that Amazon ECR manages.By default Amazon ECR encrypts images with Amazon S3-managed encryption keys using server side encryption. For more control over the encryption of images stored in S3, KMS keys stored in AWS key management service. When using KMS key, you can either use default key managed by AWS or specify your own KMS key.

When using self managed KMS key, make sure the key is present in same region as ECR repository.

Evaluation

This control ensures that encryption at rest is enabled for ECR repositories.

Remediation

Using AWS Console:

Note: ECR repository encryption cannot be changed once created. You'll need to create a new one.

1. Go to AWS Console ECR dashboard at https://console.aws.amazon.com/ecr/
2. Click on **Create repository**.
3. Select **Private** option under **Visibility settings** section.
4. Under **Encryption settings**, please enable **KMS encryption**.
5. Now, encryption is managed by key that AWS owns and manages for you.
6. To use a key owned by self, select **Customize encryption settings** checkbox.
7. Select a key from the drop-down.
8. Click on **Create repository**.

Using AWS CLI:

```
aws ecr create-repository --repository-name [REPOSITORY_NAME] --encryption-
configuration encryptionType=KMS,kmsKey=[KMS Key Alias/Key ID/Key ARN]
```

Note: For more details on command, please refer to
https://docs.aws.amazon.com/cli/latest/reference/ecr/create-repository.html

Reference

- https://docs.aws.amazon.com/AmazonECR/latest/userguide/encryption-at-rest.html

Control ID - 359: Ensure that Elasticsearch is configured inside a VPC

Criticality: HIGH

Specification

Amazon Elasticsearch Service domains from within an Amazon VPC without the need for NAT instances or Internet gateways.
VPC support for Amazon ES is easy to configure, reliable, and offers an extra layer of security. With VPC support, traffic between other services and Amazon ES stays entirely within the AWS network, isolated from the public Internet. You can manage network access using existing VPC security groups, and you can use AWS Identity and Access Management (IAM) policies for additional protection.

Rationale

When you use AWS Elasticsearch domains that reside within a VPC have an extra layer of security when compared to ES domains that use public endpoints. Launching an Amazon ES cluster within an AWS VPC enables secure communication between the ES cluster (domain) and other AWS services without the need for an Internet Gateway, a NAT device or a VPN connection and all traffic remains secure within the AWS Cloud.

Evaluation

This control ensures that Elasticsearch is configured inside a VPC

Remediation

Using AWS Console

To Create Amazon OpenSearch Service (successor to Amazon Elasticsearch Service) Domain inside a VPC, use following commands.

1. Open Amazon OpenSearch Service (successor to Amazon Elasticsearch Service) console at https://console.aws.amazon.com/esv3.
2. In the left navigation panel, choose **Domains**.
3. Click **Create domain** button.
4. Enter domain name and choose appropriate values for data nodes configuration.
5. In the **Network** section, and Select **VPC access**.
6. Select **VPC**, **Subnets** and **Security groups**.
7. Click on **Create**.

Using AWS CLI

To create Elasticsearch inside a VPC, use following command:

```
aws es create-elasticsearch-domain \

        --region [region] \

        --domain-name [domain-name] \

        --elasticsearch-version [version] \
```

```
        --elasticsearch-cluster-config InstanceType=[instance-
type].elasticsearch,InstanceCount=2 \

        --ebs-options EBSEnabled=true,VolumeType=standard,VolumeSize=100 \

        --vpc-options SubnetIds=[subnet-ID],SecurityGroupIds=[security-group-
id]
```

Reference

- Launching your Amazon OpenSearch Service domains within a VPC
- AWS Elasticsearch Domain In VPC CLI reference

Control ID - 360: Ensure that ELB has cross-zone-load-balancing enabled

Criticality: MEDIUM

Specification

A load balancer distributes workloads across multiple compute resources, such as virtual servers. Using a load balancer increases the availability and fault tolerance of your applications. With cross-zone load balancing, each load balancer node for your Classic Load Balancer distributes requests evenly across the registered instances in all enabled Availability Zones. If cross-zone load balancing is disabled, each load balancer node distributes requests evenly across the registered instances in its Availability Zone only.

Rationale

By enabling Cross-zone load balancing reduces the need to maintain equivalent numbers of instances in each enabled Availability Zone, and improves your application's ability to handle the loss of one or more instances.

Evaluation

This control ensures that ELB is cross-zone-load-balancing enabled

Remediation

Using AWS Console:

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, under **Load Balancing**, choose **Load Balancers**.
3. Select the classic load balancer to be remediated.
4. On the Description tab, Click **Change cross-zone load balancing setting**.
5. On the **Configure Cross-Zone Load Balancing** page, select **Enable**.
6. Click **Save**.

Using AWS CLI:

To enable cross-zone load balancing for the load balancer, use the following command:

```
aws elb modify-load-balancer-attributes \
```

```
        --load-balancer-name [load-balancer-name] \

        --region [region] \

        --load-balancer-attributes
"{\"CrossZoneLoadBalancing\":{\"Enabled\":true}}"
```

Reference

- How Elastic Load Balancing works
- Configure cross-zone load balancing for your Classic Load Balancer

Control ID - 366: Ensure that Secrets Manager secret is encrypted using KMS using a customer managed Key (CMK)

Criticality: HIGH

Specification

AWS Secrets Manager is an AWS service that encrypts and stores your secrets, and transparently decrypts and returns them to you in plaintext. It's designed especially to store application secrets, such as login credentials, that change periodically.

Secrets Manager integrates with AWS Key Management Service (AWS KMS) to encrypt every version of every secret value with a unique data key that is protected by an AWS KMS key. This integration protects your secrets under encryption keys that never leave AWS KMS unencrypted. It also enables you to set custom permissions on the KMS key and audit the operations that generate, encrypt, and decrypt the data keys that protect your secrets.

Rationale

Secrets Manager uses encryption via AWS Key Management Service (AWS KMS) to protect the confidentiality of data at rest. Using your own Amazon KMS Customer Master Key (CMK) to protect the secret data managed by AWS Secrets Manager service, you get full control over the encryption key to access your secrets. Amazon KMS allows you to easily create, rotate, disable and audit Customer Master Keys created for your Secrets Manager secrets.

Evaluation

This control ensures that Secrets Manager secret is encrypted using KMS.

Remediation

Using AWS Console:

1. Sign in to AWS Console and navigate to https://console.aws.amazon.com/secretsmanager.
2. Click on the secret to be modified.
3. From the actions dropdown, select Edit encryption key.
4. If KMS key is not present, click on add new key.

- Click on Create key button.
- Select all the default values and click on Next button

- Enter the Alias/ name of the key and click on Next button.
- Select the Key administrator and click on next button.
- Select the user or role who can use the key and click on Next button
- Review and click on Finish button.

- Or select the existing KMS encryption key from the dropdown.
- Click on Save button.

Using AWS CLI:

To create KMS symmetric key with default values, use following command.

```
aws kms create-key
```

Refer document for Creating Key.

To create alias for KMS key, use following command. Enter Arn value in target-key-id option from the create key command response.

```
aws kms create-alias \

        --region [region] \

        --alias-name [*"alias/*alias-name"] \

        --target-key-id [kms-key-arn-id]
```

To update secret to use KMS key, use following command.

```
aws secretsmanager update-secret \

        --region [region] \

        --secret-id [secret-name] \

        --kms-key-id [kms-key-arn-id]
```

Reference

- How AWS Secrets Manager uses AWS KMS
- Modify a secret

Control ID - 367: Ensure that Load Balancer has deletion protection enabled

Criticality: MEDIUM

Specification

A load balancer distributes workloads across multiple compute resources, such as virtual servers. Using a load balancer increases the availability and fault tolerance of your applications. To prevent your load balancer

from being deleted accidentally, you can enable deletion protection. By default, deletion protection is disabled for your load balancer.

Rationale

By enabling the "Deletion Protection" feature, any request to delete the load balancer is denied, whether operation request is from the console, the CLI, or the API.

With Deletion Protection safety feature enabled, you have the guarantee that your AWS load balancers cannot be accidentally deleted and make sure that your load-balanced environments remain safe.

Evaluation

This control ensures that Load Balancer has deletion protection enabled

Remediation

Using AWS Console:

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, under **Load Balancing**, choose **Load Balancers**.
3. Select the load balancer to be remediated.
4. On the Description tab, Click **Edit attributes**.
5. On the **Edit load balancer attributes** page, select **Enable** checkbox for **Delete Protection**.
6. Click **Save**.

Using AWS CLI:

To enable deletion protection for the load balancer, use the following command:

```
aws elbv2 modify-load-balancer-attributes \

        --load-balancer-arn [load-balancer-arn] \

        --attributes Key=deletion_protection.enabled,Value=true
```

For command, usage refer: https://docs.aws.amazon.com/cli/latest/reference/elbv2/modify-load-balancer-attributes.html

Reference

- Application Load Balancers
- Network Load Balancers
- Gateway Load Balancers

Control ID - 369: Ensure that Load Balancer (Network/Gateway) has cross-zone load balancing enabled

Criticality: MEDIUM

Specification

A load balancer distributes workloads across multiple compute resources, such as virtual servers. Using a load balancer increases the availability and fault tolerance of your applications. With cross-zone load balancing, each load balancer node for your Load Balancer distributes requests evenly across the registered instances in all enabled Availability Zones. If cross-zone load balancing is disabled, each load balancer node distributes requests evenly across the registered instances in its Availability Zone only.

Rationale

By enabling Cross-zone load balancing reduces the need to maintain equivalent numbers of instances in each enabled Availability Zone, and improves your application's ability to handle the loss of one or more instances.

Evaluation

This control ensures that Load Balancer (Network/Gateway) has cross-zone load balancing enabled

Remediation

Using AWS Console:

1. Open the Amazon EC2 console at https://console.aws.amazon.com/ec2/.
2. In the navigation pane, under **Load Balancing**, choose **Load Balancers**.
3. Select the load balancer to be remediated.
4. On the Description tab, Under **Attributes**, click **Edit attributes** button.
5. To enable an Cross-zone load balancing, select the check box for **Cross-zone load balancing**.
6. Click **Save**.

Using AWS CLI:

To enable cross-zone load balancing for the load balancer, use the following command:

```
aws elbv2 modify-load-balancer-attributes \

        --load-balancer-arn [load-balancer-arn] \

        --region [region] \

        --attributes Key=load_balancing.cross_zone.enabled,Value=true
```

Reference

- How Elastic Load Balancing works
- Network Load Balancers
- Gateway Load Balancers

Control ID - 370: Ensure that Auto Scaling Groups supply tags to Launch Configurations

Criticality: MEDIUM

Specification

It is recommended that tags are created for new and existing Auto Scaling Groups. Tags should be propagated from an Auto Scaling Group to the Amazon EC2 instances it launches.

Tags are not propagated to Amazon EBS volumes. To add tags to Amazon EBS volumes, specify the tags in a launch template.

Rationale

Tags serve several purposes. Tags can help you track AWS costs, filter and search, control access, identify and organize resources. It is recommended that tags are created for Auto Scaling groups and propagated to the Amazon EC2 instances it launches.

Evaluation

This control ensures that Autoscaling Groups supply tags to Launch Configurations

Remediation

Using AWS Console:

1. Go to Amazon EC2 Auto Scaling dashboard at https://console.aws.amazon.com/ec2autoscaling/home.
2. Select the Auto Scaling Group to be remediated.
3. Scroll to Tags section and click on edit button
4. On the **Tags** screen, click **Add tag**.
5. Supply values for **Key**, **Value** and make sure that the option **Tag New Instances** is checked.
6. If the **Tag New Instances** is not selected for existing tags, select the option to enable the propagation of tags to launch configurations.

Using AWS CLI:

Create a new tag or update an existing tag for an Auto Scaling Group:

```
aws autoscaling create-or-update-tags \

        --tags ResourceId=[auto-scaling-group-name],ResourceType=auto-scaling-group,Key=[tag-key],Value=[tag-value],PropagateAtLaunch=true
```

Note: When you specify a tag with a key that already exists, the operation overwrites the previous tag definition, and you do not get an error message.

Reference

- Tagging Auto Scaling groups and instances
- AWS CLI Command Reference - create-or-update-tags

Control ID - 373: Ensure to encrypt CloudWatch log groups

Criticality: LOW

Specification

This control ensures that Cloudwatch log groups are encrypted.

Rationale

Cloudwatch log groups will store data indefinitely. After you associate a CMK with a log group, all newly logged data for the log group is encrypted using the key associated. This data is stored in encrypted format throughout its retention. CloudWatch Logs decrypts this data whenever it is required. CloudWatch Logs must have permissions for the CMK whenever encrypted data is requested.

Evaluation

This control ensures that Cloudwatch log groups are encrypted.

Remediation

Note : CloudWatch Log group encryption settings update is only available through AWS CLI.

Using AWS CLI:

```
aws logs associate-kms-key --log-group-name [log_group_name] --kms-key-id
"[key-arn]"
```

Note : Please set the below policy for the KMS key being used

```
            {
                "Version": "2012-10-17",
                "Id": "key-default-1",
                "Statement": [
                    {
                        "Sid": "Enable IAM User Permissions",
                        "Effect": "Allow",
                        "Principal": {
                            "AWS": "arn:aws:iam::ACCOUNT-ID:root"
                        },
                        "Action": "kms:*",
                        "Resource": "*"
                    },
                    {
                        "Effect": "Allow",
                        "Principal": {
                            "Service": "logs.REGION.amazonaws.com"
                        },
                        "Action": [
                            "kms:Encrypt*",
                            "kms:Decrypt*",
                            "kms:ReEncrypt*",
                            "kms:GenerateDataKey*",
                            "kms:Describe*"
                        ],
                        "Resource": "*",
                        "Condition": {
                            "ArnLike": {
                                "kms:EncryptionContext:aws:logs:arn":
"arn:aws:logs:REGION:ACCOUNT-ID:log-
            group:LOG-GROUP-NAME"
                            }
```

```
                        }
                    }
                ]
            }
```

Note: For more details on command, please refer to https://docs.aws.amazon.com/AmazonCloudWatch/latest/logs/encrypt-log-data-kms.html#associate-cmk

Reference

- https://docs.aws.amazon.com/AmazonCloudWatch/latest/logs/encrypt-log-data-kms.html

Control ID - 374: Ensure that Athena Workgroup is encrypted

Criticality: HIGH

Specification

Amazon Athena is a serverless interactive query service that enables users to easily analyze data in Amazon S3 using standard SQL. This new feature not only makes it possible for Athena to provide support for querying encrypted data in Amazon S3, but also enables the encryption of data from Athena's query results. You can run queries in Amazon Athena on encrypted data in Amazon S3 in the same Region and across a limited number of Regions.
You can also encrypt the query results in Amazon S3 and the data in the AWS Glue Data Catalog. AWS Athena supports the following S3 encryption options: Server Side Encryption (SSE) with an Amazon S3-managed key (SSE-S3), SSE with a AWS Key Management Service customer managed key (SSE-KMS) and Client-Side Encryption (CSE) with a AWS KMS customer managed key (CSE-KMS).

Rationale

Encryption at rest is enabled for Amazon Athena query results stored in Amazon S3, in order to secure data and meet compliance requirements for data-at-rest encryption.

Evaluation

This control ensures that athena workgroup is encrypted

Remediation

Using AWS Console

1. Open Amazon Athena Service console at https://console.aws.amazon.com/athena.
2. In the left navigation panel, choose **Workgroups**.
3. Click on the workgroup to be remediated.
4. Click on the edit button to edit the workgroup.
5. Scroll to **Query result location and encryption** section, check the checkbox for **Encrypt query results**.
6. Choose an **Encryption type (CSE_KMS, SSE_KMS, SSE_S3)**.
7. If encryption type is SSE_KMS or CSE_KMS, choose the AWS KMS key.
8. If encryption type is SSE_S3, no need of the AWS KMS key.
9. Click on Save changes button at the bottom.

Using AWS CLI

To encrypt Athena workgroup, use the following command:

```
aws athena update-work-group \

        --work-group [workgroup-name]\

        --configuration-updates
EnforceWorkGroupConfiguration=true,ResultConfigurationUpdates={EncryptionConfi
guration={EncryptionOption=[Option],KmsKey=[key-arn]}}
```

Note: For SSE-KMS and CSE-KMS , there is the KMS key ARN or ID. For SSE_S3, no need of KMS key

Reference

- Amazon Athena workgroup reference
- Amazon Athena workgroup encryption at rest

Control ID - 377: Ensure ECR image scanning on push is enabled

Criticality: MEDIUM

Specification

Amazon ECR image scanning helps in finding out the software vulnerabilities in container images.

Amazon ECR uses the Common Vulnerabilities and Exposures (CVEs) database from the open-source Clair project and provides a list of scan findings.

Rationale

Automatic ECR image scanning is disabled by default whenever we create a repository. If scan on push is enabled for a repository, new images being pushed are scanned automatically and findings are logged in cloudwatch.

If disabled, images has to be manually scanned for vulnerabilities.

Evaluation

This control ensures that image scanning is enabled for ECR repositories.

Remediation

Using AWS Console:

1. Go to AWS Console ECR dashboard at https://console.aws.amazon.com/ecr/
2. Select the private repository that you want to reconfigure.
3. Select **Edit** option.
4. Scroll to **Image scan settings** section.
5. Enable **Scan on push**.

6. Click on **Save**.

Using AWS CLI:

```
aws ecr put-image-scanning-configuration --repository-name [REPOSITORY_NAME] --image-scanning-configuration scanOnPush=true
```

Note: For more details on command, please refer to https://docs.aws.amazon.com/cli/latest/reference/ecr/put-image-scanning-configuration.html

Reference

- https://docs.aws.amazon.com/AmazonECR/latest/userguide/image-scanning.html#scanning-repository

Control ID - 378: Ensure Transfer Server is not exposed publicly.

Criticality: HIGH

Specification

AWS Transfer Family is a secure transfer service that enables you to transfer files into and out of AWS storage services. If you use Amazon Virtual Private Cloud (Amazon VPC) to host your AWS resources, you can establish a private connection between your VPC and a server. You can then use this server to transfer data over your client to and from your Amazon S3 bucket without going over the public internet.

Rationale

This policy allows any resource in the VPC full access to the service behind the endpoint, for example, S3, DynamoDB. Using Amazon VPC, you can launch AWS resources in a custom virtual network. You can use a VPC to control your network settings, such as the IP address range, subnets, route tables, and network gateways

Evaluation

Transfer Server is not exposed publicly.

Remediation

Using AWS Console:

1. **Note : Server endpoint accessible though VPC can be created only while creating a server.**
2. Sign in to the AWS Management Console and open the AWS Transfer Family console at https://console.aws.amazon.com/transfer/.
3. In the left navigation panel, Click on Server.
4. Click on Create server button.
5. Choose a protocol and an identity provider.
6. On the **Choose an endpoint** screen, in Endpoint type select **VPC hosted**.
7. Select Existing VPC from dropdown or create new VPC
8. Select atleast one Availability Zones and subnet. Click on Next
9. Choose a required domain and configure additional details.
10. Click on **Create server**.

Reference

- Create a server endpoint that can be accessed only within your VPC

Control ID - 379: Ensure S3 bucket must not allow WRITE permission for server access logs from everyone on the bucket

Criticality: MEDIUM

Specification

Ensure that your S3 buckets that are used as a target to server access logging access control list does not allow unrestricted public to read or write access. Exposing your S3 buckets to everyone can lead to data leaks, data loss and unexpected charges for the S3 service.

Rationale

Allowing unrestricted access increases opportunities for loss of data. It is recommended to promptly review your S3 buckets and their contents to ensure that you are not accidentally making objects available to users that you don't intend.

Evaluation

Control checks whether buckets that are set as targets for server access logging doesn't have write access to everyone.

Remediation

Perform the following:

1. Sign in to the AWS management console and open the amazon s3 console at https://console.aws.amazon.com/s3/.
2. Select the bucket and click **Permissions**
3. Click on **Edit**, under **Access control list**
4. In **Edit access control list** page, for Grantee **Everyone** uncheck both **Write** checkboxes
5. Click on **Save changes**.

Note: Only S3 buckets used as a target to server access logging will be evaluated by this control

Reference

https://docs.aws.amazon.com/AmazonS3/latest/dev/acl-overview.html

Control ID - 380: Ensure Backup Vault is encrypted at rest using KMS CMK

Criticality: HIGH

Specification

AWS Backup is a centralized backup service that makes it easy and cost-effective for you to back up your application data across AWS services in the AWS Cloud, helping you meet your business and regulatory backup compliance requirements. AWS Backup makes protecting your AWS storage volumes, databases, and file systems simple by providing a central place where you can configure and audit the AWS resources you want to backup, automate backup scheduling, set retention policies, and monitor all recent backup and restore activity.

The KMS encryption key that is configured for a backup vault applies only to the backups created for certain resource types such as Amazon EFS file systems. This adds another layer of protection for your backups. The backups taken for all other resource types are configured using the key that is used to encrypt the source resource.

Rationale

When you use your own AWS KMS Customer Master Keys (CMKs) to protect the backups created with Amazon Backup service, you have full control over who can use the encryption keys to access your backups. Amazon Key Management Service (KMS) service allows you to easily create, rotate, disable and audit the Customer Master Keys used to encrypt AWS Backup data.

The AWS KMS encryption key protects your backups in this backup vault. By default, AWS Backup creates a KMS key with the alias aws/backup for you.

Evaluation

This control ensures that Backup Vault is encrypted at rest using KMS CMK.

Remediation

Note: Encryption at rest using AWS KMS Customer Master Keys, option is only available while creating the AWS Backup vault.

Using AWS Console

To Create AWS KMS key

1. Open AWS Key Management Service (KMS) console at https://console.aws.amazon.com/kms.
2. In the left navigation panel, choose **Customer managed keys**.
3. Click **Create Key** button from the dashboard top menu. Choose **Next**.
4. In the **Alias**, enter a unique name and (optionally) a description for new CMK.
5. (Optional). In the **Add tags**, add tags that identify or categorize your KMS key. Choose **Next**.
6. In the **Key administrators** section, select the IAM users and roles who can manage the KMS key. Choose **Next**.
7. In the **This account** section, select the IAM users and roles in this AWS account who can use the KMS key in cryptographic operations. Choose **Next**.
8. Review the key settings that you chose. You can still go back and change all settings.
9. When you're done, choose **Finish** to create the key.
10. Once the key is created, the KMS dashboard will display a confirmation message: "Your AWS KMS key was created with alias. Alias: "cmk-name" and key ID "key-id".

To Create the Backup vault

1. Open AWS Backup console at https://console.aws.amazon.com/backup.
2. In the left navigation panel, choose **Backup vaults**.
3. Click **Create Backup vault** button from the dashboard top menu.
4. Enter a name for your backup vault.

5. Select the ID of newly created AWS KMS key from the **KMS encryption** dropdown list.
6. Optionally, add tags that will help you search for and identify your backup vault.
7. Choose **Create Backup vault** to create the backup vault.

Using AWS CLI

To create AWS KMS key, use the following command:

From the output of this cli copy the key Amazon Resource Name (ARN) as this will be required for next steps.

```
aws kms create-key \

        --region [region] \

        --description ['key-description']
```

To create alias specify the alias name, use the following command:

The value for **--alias-name** must begin with alias/ followed by a name, such as alias/ExampleAlias.
Specify the key ARN ot key id value for **--target-key-id** that you've copied earlier.

```
aws kms create-alias \

        --region [region] \

        --alias-name [alias-name] \

        --target-key-id [key-arn]
```

To create AWS Backup vault, use the following command:

```
aws backup create-backup-vault \

        --region [region] \

        --backup-vault-name [backup-vault-name] \

        --encryption-key-arn [key-arn]
```

Reference

- Working with backup vaults
- Encryption for backups in AWS Backup
- Managing keys

Control ID - 381: Ensure Glacier Vault access policy is not public by only allowing specific services or principals to access it

Criticality: HIGH

Specification

Amazon S3 Glacier is a secure, durable, and extremely low-cost Amazon S3 storage class for data archiving and long-term backup. Each item stored in Glacier is known as an archive, and can be as large as 40 terabytes. Archives are stored in vaults, each of which can store as many archives as desired. An Amazon S3 Glacier vault access policy is a resource-based policy that you can use to manage permissions to your vault. You can create one vault access policy for each vault to manage permissions. You can modify permissions in a vault access policy at any time.

Rationale

The Glacier Vault Access Policies govern access control to a particular vault and it controls who can access which service or deny a specific service. This manages the requests coming to the vault and ensures secure access to resource.

Evaluation

This control ensures that Glacier Vault access policy is not public by only allowing specific services or principals to access it

Remediation

Using AWS Console:

1. Open the Amazon S3 Glacier console at https://console.aws.amazon.com/glacier/.
2. Select the glacier vault, and select **Permissions** tab.
3. Click **Edit policy document** and click **Add a permission**.
4. For **Effect:** Select **Allow** to allow access to a resource. Or Select **Deny** to deny access to a resource.
5. For **Principals:** Specify Amazon Resource Name (ARN) of AWS account user, IAM roles and IAM users.
6. For **Actions:** Select the specific action or actions that will be allowed.
7. Click **Add Permission**, **Save** the policy.

Using AWS CLI:

Create a JSON file on your local machine with a name such as **vault_access_policy.json**, and then paste the following content into it.

```
{"Policy": "{\"Version\":\"2012-10-
17\",\"Statement\":[{\"Effect\":\"[Allow|Deny]\",\"Principal\":{\"AWS\":\"[use
r-arn]\"},\"Action\":\"glacier:[action]\",\"Resource\":\"[vault-arn]\"}]}"}
```

To set glacier vault access policy, use the following command:

```
aws glacier set-vault-access-policy \

        --account-id [account-id] \

        --vault-name [vault-name] \

        --policy=file://vault_access_policy.json
```

Reference

- Amazon S3 Glacier Access Control with Vault Access Policies
- Overview of Managing Access Permissions to Your Amazon S3 Glacier Resources
- Identity and Access Management in Amazon S3 Glacier

Control ID - 382: Ensure SQS queue policy is not public by only allowing specific services or principals to access it

Criticality: HIGH

Specification

Amazon SQS is a reliable, highly-scalable hosted queue for storing messages as they travel between applications or microservices. Amazon SQS moves data between distributed application components and helps you decouple these components. The Action element describes the specific action or actions that will be allowed or denied. Statements must include either an Action or NotAction element. Each AWS service has its own set of actions that describe tasks that can be performed with that service. Specify a value using a namespace that identifies a service, for example, iam, ec2 sqs, sns, s3, followed by the name of the action to be allowed or denied

Rationale

SQS access policies should not have global "*:*" access. For security reasons, you can write permissions policies that you can attach to an IAM identity and you can grant permissions to perform the action.

Evaluation

This control ensures that SQS queue policy is not public by only allowing specific services or principals to access it.

Remediation

Using AWS Console:

1. Sign in to AWS Console and navigate to https://console.aws.amazon.com/sqs/.
2. Select the SQS queue that you want to examine.
3. Select the Access policy tab from the panel.
4. Click on Edit to Edit the Permissions
5. In Access policy pane, click on policy generator link.
6. On the policy generator page, enter specific IAM user in Principal section. Choose an action to be permitted.
7. Click on Add statement button and then click on generate policy button.
8. Copy the policy and paste it in the access policy pane.
9. Click on save to save the changes

Reference

- Amazon SQS API permissions: Actions and resource reference
- Overview of managing access in Amazon SQS

Control ID - 383: Ensure SNS topic policy is not public by only allowing specific services or principals to access it

Criticality: HIGH

Specification

Amazon Simple Notification Service (Amazon SNS) is a managed service that provides message delivery from publishers to subscribers (also known as producers and consumers). Publishers communicate asynchronously with subscribers by sending messages to a topic, which is a logical access point and communication channel. Clients can subscribe to the SNS topic and receive published messages using a supported endpoint type, such as Amazon Kinesis Data Firehose, Amazon SQS, AWS Lambda, HTTP, email, mobile push notifications, and mobile text messages (SMS).

Rationale

When an SNS topic policy grants permission to "Everyone" by using a wildcard, i.e. "*", as the Principal value, the topic security can be at risk as any unauthenticated entity can subscribe and receive messages from the topic publishers, messages that usually should be destined only to known subscribers.

Evaluation

Ensure SNS topic policy is not public by only allowing specific services or principals to access it

Remediation

Using AWS Console:

1. Sign in to AWS Console and navigate to https://console.aws.amazon.com/sns/v2/.
2. In the left navigation panel, select Topics.
3. Select the Amazon SNS topic that you want to examine.
4. Click on Edit to Edit the resource.
5. Expand the Access policy option to edit the policy
6. Under Principal section remove * and add specific AWS accounts, IAM users and roles.
7. Click on save to save the changes

Using AWS CLI:

Run list-queues command (OSX/Linux/UNIX) to expose all SNS queues available in the selected region and their URLs:

```
aws sns list-topics \

        --region [region]
```

Run set-topic-attributes command (OSX/Linux/UNIX) using the ARN of the SNS topic that you want to reconfigure.

```
aws sns set-topic-attributes \

        --region [region] \

        --topic-arn [topic-arn]        \

        --attribute-name Policy        \

        --attribute-value file://secure-publish-policy.json
```

File - secure-publish-policy.json

Remove * from principal and add IAM user arn for specific user.

```
{
  "Version": "2008-10-17",
  "Id": "__default_policy_ID",
  "Statement": [
    {
      "Sid": "__default_statement_ID",
      "Effect": "Allow",
      "Principal": {
        "AWS": "*"
      },
      "Action": [
        "SNS:Publish",
        "SNS:RemovePermission",
        "SNS:SetTopicAttributes",
        "SNS:DeleteTopic",
        "SNS:ListSubscriptionsByTopic",
        "SNS:GetTopicAttributes",
        "SNS:Receive",
        "SNS:AddPermission",
        "SNS:Subscribe"
      ],
      "Resource": "[topic-arn]",
      "Condition": {
        "StringEquals": {
          "AWS:SourceOwner": "[source-id]"
        }
      }
    }
   ]
  }
```

For command usage refer: set-topic-attributes

Reference

- Permissions for the Amazon SNS Topic

Control ID - 385: Ensure that EMR Cluster security configuration encryption is using SSE-KMS

Criticality: HIGH

Specification

AWS EMR cluster security configuration encryption helps to prevent unauthorized users from reading data on a cluster and associated data storage systems.

Rationale

Data encryption helps to prevent unauthorized users from reading data on a cluster and associated data storage systems. This includes data saved to persistent media, known as data at rest, and data that may be intercepted as it travels the network, known as data in transit.

Evaluation

This control ensures that cluster security configuration encryption is using SSE-KMS

Remediation

Using AWS Console:

Note : We cannot modify an existing EMR cluster, new EMR cluster should be created with required configuration.

To configure a security configuration with Encryption Mode set to SSE-KMS, follow the below procedure:

1. Sign in to AWS Console and navigate to https://console.aws.amazon.com/elasticmapreduce/home.
2. In the Navigation pane, choose **Security configurations**.
3. Click on **Create** button.
4. Check **S3 encryption** checkbox, select **SSE-KMS** option in **Default encryption mode** dropdown.
5. In **AWS KMS customer master key** text box, provide **key ARN**.
6. Click **Create** button.

To Create and Configure an EMR cluster with Secuirty Configuration having Encryption Mode set to SSE-KMS.

1. Sign in to AWS Console and navigate to https://console.aws.amazon.com/elasticmapreduce/home.
2. In the Navigation pane, choose **Clusters**.
3. Click on **Create cluster** button.
4. Select **Go to advanced options**.
5. In steps 1 - 3, configure options according to the requirements.
6. In **step 4**, select **Security Configuration**.
7. Select above created security configuration with Encryption Mode set to SSE-KMS.
8. Click **Create cluster** button.

Reference

- https://docs.aws.amazon.com/emr/latest/ManagementGuide/emr-data-encryption-options.html

Control ID - 386: Ensure that all NACLs are attached to subnets

Criticality: MEDIUM

Specification

A network access control list (NACL) is an optional layer of security for your VPC that acts as a firewall for controlling traffic in and out of one or more subnets. You can set up network ACLs with rules similar to your security groups in order to add an additional layer of security to your VPC.

Rationale

Network ACLs control traffic flow in and out of the subnets with which they are associated. NACL helps in providing a firewall thereby helping secure the VPCs and subnets.

Network ACL is attached to subnets
Network ACL is not attached to subnets

Evaluation

This control ensures that all NACLs are attached to subnets

Remediation

Using AWS Console:

1. Go to Amazon VPC dashboard at https://console.aws.amazon.com/vpc/home.
2. Select the **Network ACLs** tab in the **Security** section from the menu.
3. For every NACL displayed in the table, ensure that there is at least 1 subnet in the **Associated with** column.
4. Identify the NACLs which are not associated with any subnets.
5. Select the **Subnets** tab in the **Virtual Private Cloud** section from the menu.
6. Select the subnet to be associated wth the NACL identified in step(4) above and click the **Edit network ACL association** option from the **Actions** menu.
7. Select the NACL identified in step(4) in the **Network ACL ID** dropdown and click **Save**.

Using AWS CLI:

To replace the default network ACL associated with a subnet:

```
aws ec2 replace-network-acl-association \

        ---association-id [association-id] \

        --network-acl-id [network-acl-id]
```

Reference

- Network ACLs
- AWS CLI Command Reference - replace-network-acl-association

Control ID - 387: Ensure GuardDuty is enabled to specific org/region

Criticality: MEDIUM

Specification

Amazon GuardDuty uses threat intelligence feeds to detect malicious activity in your AWS environment.

Rationale

Amazon GuardDuty is a continuous security monitoring service that analyzes and processes the following Data sources: VPC Flow Logs, AWS CloudTrail management event logs, CloudTrail S3 data event logs, and DNS

logs. It uses threat intelligence feeds, such as lists of malicious IP addresses and domains, and machine learning to identify detect unexpected and potentially unauthorized malicious activity within your AWS environment. This can include issues like escalations of privileges, uses of exposed credentials, or communication with malicious IP addresses, or domains.

GuardDuty informs you of the status of your AWS environment by producing security findings that you can view in the GuardDuty console or through Amazon CloudWatch events.

Evaluation

This control ensures that AWS guardduty is enabled.

Remediation

Note: This Control evaluates Guardduty service for all regions collectively, i.e the control will fail if any region fails. If any region is disabled, this would be counted as a region failure.

Using AWS Console:

1. Sign in to AWS Console and navigate to https://console.aws.amazon.com/guardduty/home.
2. If you are enabling the Guardduty in particular region for the first time :
    - Click **Get Started** button.
    - In the next page, click **Enable GuardDuty** button.
3. If detector is already present in particular region :
    - Go to **Settings**
    - Click on **Re-enable GuardDuty** button under the **Suspend GuardDuty** section.

Using AWS CLI:

If you are enabling the Guardduty in particular region for the first time

```
aws guardduty create-detector --enable --region [Region]
```

If detector is already present in particular region

- Get the detector-id

  ```
  aws guardduty list-detectors --region [Region]
  ```

  copy the detector-id from the output of above command

- Enable the detector

  ```
  aws guardduty update-detector --detector-id [Detector-id] --enable --
  region [Region]
  ```

For command usage refer:

- https://awscli.amazonaws.com/v2/documentation/api/latest/reference/guardduty/create-detector.html
- https://docs.aws.amazon.com/cli/latest/reference/guardduty/update-detector.html

Reference

- https://docs.aws.amazon.com/guardduty/latest/ug/guardduty_settingup.html

Control ID - 388: Ensure API Gateway stage have logging level defined as appropriate and have metrics enabled

Criticality: HIGH

Specification

Enabling Logging and Metrics allows a user to order to track and analyze execution behavior at the API stage level. API gateway allows to record logs at INFO and ERROR levels.

Rationale

Enabling Logging and Metrics allows a user to order to track and analyze execution behavior at the API stage level. API gateway allows to record logs at INFO and ERROR levels.

This control ensures API Gateway stage have logging level defined as Error or Info and metrics enabled

Evaluation

This control ensures API Gateway stage have logging level defined as Error or Info and metrics enabled

Remediation

Note: Repeat these steps for all API Gateway Stages.

Using AWS Console:

1. Sign in to the AWS Management Console and open the Amazon API Gateway console at https://console.aws.amazon.com/apigateway/
2. Click on the name of the API to edit
3. In the left navigation panel, click **Stages**
4. Select the API Stage you want to edit
5. On the **Stage** Editor panel, select **Logs** tab to access the stage settings
6. Under **CloudWatch Settings**, Check **Enable CloudWatch Logs**
7. For **Log level**, select the appropriate value for **INFO or ERROR**
8. Check **Enable Detailed CloudWatch Metrics**
9. Click on **Save Changes**

Reference

- https://docs.aws.amazon.com/apigateway/latest/developerguide/set-up-logging.html

Control ID - 393: Ensure the option group attached to the RDS Oracle Instance have TLSv1.2 and the required ciphers configured

Criticality: MEDIUM

Specification

SSL option will be used to encrypt communication to and from db instance. If SSL encryption is not enabled, data is at risk of exposure.

Rationale

To enable SSL encryption, SSL option needs to be added with proper cipher suite and TLS version to the option group associated with the db instance

Evaluation

This control ensures latest TLS/SSL encryption is used with proper cipher suites

Remediation

1. Sign in to the AWS Management Console and open the Amazon RDS console at https://console.aws.amazon.com/rds/
2. Click on the affected oracle instance.
3. Go to **Configuration** tab
4. Under the configuration tab go to **Option group** and note associated option group name.
5. In left navigation panel, select **Option groups**.
6. In the list of groups, select the checkbox against the option group assocaited with the instance.
7. Click on **Modify option** on top right corner.
8. Select **SSL** option in **Installed options** dropdown.
9. Under **Option settings** section, set **SQLNET.CIPHER_SUITE** option with **SSL_RSA_WITH_AES_256_GCM_SHA384** value
10. Set **SQLNET.SSL_VERSION** option with **1.2** value
11. Select **Yes** option at **Apply immediately** section.
12. Click on **Modify option**.

Note: We cannot modify a default option group, instead custom option group with required options has to be created and linked with RDS instance.

Using AWS CLI:

To modify a DB option group, use the following AWS CLI command:

```
aws rds add-option-to-option-group --option-group-name
[Instance_Associated_Group] --options
OptionName=SSL,Port=[PORT_USED_FOR_CONNECTION],VpcSecurityGroupMemberships=[In
stance_Associated_SecurityGroup],OptionSettings=[{Name=SQLNET.SSL_VERSION,Valu
e=1.0},{Name=SQLNET.CIPHER_SUITE,Value=SSL_RSA_WITH_AES_256_CBC_SHA}] --apply-
immediately
```

Note: For more details on command, please refer to https://docs.aws.amazon.com/cli/latest/reference/rds/add-option-to-option-group.html

Reference

- https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/Appendix.Oracle.Options.SSL.html#Appendix.Oracle.Options.SSL.TLS

Control ID - 395: Ensure that Auto Scaling Groups that are associated with a Load Balancer are using Elastic Load Balancing health checks

Criticality: MEDIUM

Specification

Auto Scaling group can determine instance health based on additional load balancer tests. You can configure the Auto Scaling group to use Elastic Load Balancing (ELB) health checks. The load balancer periodically sends pings, attempts connections, or sends requests to test the EC2 instances and determines if an instance is unhealthy. If any one of the checks reports an instance as unhealthy, the Auto Scaling group replaces the instance immediately. If you configure the Auto Scaling group to use Elastic Load Balancing health checks, it considers the instance unhealthy if it fails either the EC2 status checks or the Elastic Load Balancing health checks. If you attach multiple load balancer target groups or Classic Load Balancers to the group, all of them must report that an instance is healthy in order for it to consider the instance healthy.

Rationale

When you use a load balancer registered with an Auto Scaling group, it can be configured to use the results of the ELB health check in addition to the EC2 instance status checks to determine the health of the EC2 instances in the Auto Scaling group.

Evaluation

This control ensures that Auto Scaling groups that are associated with a load balancer, are using Elastic Load Balancing health checks.

Remediation

Using AWS Console

1. Open EC2 Auto Scaling groups console at https://console.aws.amazon.com/ec2autoscaling
2. Click on the **Auto Scaling group** to be remediated.
3. Scroll down to **Health Check** section. Click Edit button.
4. Select ELB checkbox for **Health check type**.
5. Enter seconds value for **Health check grace period**.
6. Click on **Update** button.

Using AWS CLI

To configure ELB health check for Auto Scaling groups, use the following command:

```
aws autoscaling update-auto-scaling-group \

        --auto-scaling-group-name [autoscaling-group-name] \

        --health-check-type ELB \

        --health-check-grace-period 600
```

Note: For more details on command, please refer to https://docs.aws.amazon.com/cli/latest/reference/autoscaling/update-auto-scaling-group.html

Reference

- Adding Elastic Load Balancing health checks to an Auto Scaling group
- Auto Scaling cli reference

Control ID - 396: Ensure that Auto Scaling is enabled on your DynamoDB tables

Criticality: MEDIUM

Specification

Amazon DynamoDB auto scaling uses the AWS Application Auto Scaling service to dynamically adjust provisioned throughput capacity on your behalf, in response to actual traffic patterns. This enables a table or a global secondary index to increase its provisioned read and write capacity to handle sudden increases in traffic, without throttling. When the workload decreases, Application Auto Scaling decreases the throughput so that you don't pay for unused provisioned capacity.

Rationale

When you use the AWS Management Console to create a new table, Amazon DynamoDB auto scaling is enabled for that table by default.
You can also use the console to enable auto scaling for existing tables, modify auto scaling settings, or disable auto scaling.

Evaluation

This control ensures that Auto Scaling is enabled on your DynamoDB tables

Remediation

Using AWS Console

To create and enable Auto Scaling on your DynamoDB tables

1. Open AWS DynamoDB console at https://console.aws.amazon.com/dynamodbv2.
2. In the left navigation panel, choose **Tables**.
3. Click on the table to be remediated.
4. Go to Additional settings tab and click on Edit button.
5. Under **Edit Read/write capacity**, select option **Provisioned**.
6. In the **Read capacity** option, select **On** to enable **Auto scaling**
7. In the **Write capacity** option, select **On** to enable **Auto scaling**
8. When you're done, click **Save changes**.

Reference

- Using the AWS Management Console with DynamoDB Auto Scaling
- Using the AWS CLI to Manage DynamoDB Auto Scaling

Control ID - 398: Ensure that all EIP addresses allocated to a VPC are attached to EC2 instances

Criticality: MEDIUM

Specification

An Elastic IP address is a static public IPv4 address associated with your AWS account in a specific Region. Unlike an auto-assigned public IP address, an Elastic IP address is preserved after you stop and start your instance in a virtual private cloud (VPC).

To use an Elastic IP address, you first allocate one to your account, and then associate it with your instance or a network interface.

Rationale

The Elastic IP should be assigned to an instance, if it is not assigned you may incur additional billing for idle time usage. So if you have any unassigned Elastic IP address, make sure to remove them.

Evaluation

This control ensures that all Elastic IP addresses allocated to a VPC are attached to EC2 instances

Remediation

Using AWS Console:

1. Go to Amazon EC2 dashboard at https://console.aws.amazon.com/ec2/v2/home.
2. Select **Elastic IPs** in the **Network & Security** section from the menu.
3. For every Elastic IP displayed in the table, ensure that the **Association ID** column has a value.
4. Perform the following steps for each Elastic IP which has no association id.
5. Select the Elastic IP and click the **Associate Elastic IP address** option from the **Actions** menu.
6. Select the **Instance** option, and provide the values for **Instance** and **Private IP address** and click **Associate**.
7. Alternatively, you can select the **Network Interface** option, and provide the values for **Network Interface** and **Private IP address** and click **Associate**.
8. To release (remove) the unassociated/unused **Elastic IPs**, perform following steps:

- Select unused IP address to be removed.
- Click on actions and choose **Release Elastic IP address**.
- Click on **Release**.

Using AWS CLI:

To associate an Elastic IP address with an instance in a VPC:

```
aws ec2 associate-address \

--allocation-id [allocation-id] \

--instance-id [instance-id]
```

To associate an Elastic IP address with a network interface:

```
aws ec2 associate-address \

 --allocation-id [allocation-id] \

 --network-interface-id [network-interface-id]
```

To release an unassociated Elastic IP address:

```
aws ec2 release-address \

 --region [region] \

 --allocation-id [allocation-id]
```

Reference

- Elastic IP addresses

Control ID - 399: Ensure that all IAM users are members of at least one IAM group.

Criticality: MEDIUM

Specification

Ensure that all IAM users are members of at least one IAM group.

Rationale

It is recommended to avoid assigning identity-based policies to individual IAM users or defining inline policies when creating an IAM user. Instead, you can assign policies to a group of IAM users or write inline policies when creating an IAM group. All the IAM users within your group will inherit the permissions assigned to the group. This streamlines the process of making changes to multiple user permissions and decreases the risk of accidentally giving individual IAM users excessive permissions. As people move around in your organization, you can simply change what IAM group their IAM user belongs to.

Evaluation

Ensure that all IAM users are members of at least one IAM group.

Remediation

1. Login to the AWS Management Console: https://console.aws.amazon.com/iam/
2. Click Services
3. Click IAM
4. Click on Users
5. Click on Groups
6. Click on **Add user to groups**
7. Select the group from the list of groups

8. Click **Add to Groups**

Perform following commands Via CLI to add user to an group:

```
aws iam add-user-to-group --user-name [USER_NAME] --group-name [GROUP_NAME]
```

For command usage refer: https://docs.aws.amazon.com/cli/latest/reference/iam/add-user-to-group.html

Reference

- https://docs.aws.amazon.com/IAM/latest/UserGuide/id_groups.html

Control ID - 400: Ensure an IAM User does not have access to the console

Criticality: MEDIUM

Specification

Ensure that your existing IAM users(except for administrators) are using for API access(access keys) in order to reduce the risk of unauthorized access in case their credentials are compromised.

Rationale

It is recommended administrators who need console access should use only passwords to manage AWS resources. Other users should use access keys to programmatically access data in AWS.

Evaluation

Ensure that an IAM User does not have access to the console.

Remediation

Perform the following via AWS management console to disable password:

1. Login to the AWS Management Console: https://console.aws.amazon.com/iam/
2. Click **Services**
3. Click **IAM**
4. Click on **Users**
5. Select the **user** to remediate which has console password enabled
6. Click on **Security Credentials**
7. Click on **Manage** at console password section.
8. Click on **Disable** as per business requirement.
9. Click on **Apply**.

To create access keys to programmatically access data in AWS for an IAM user

1. Login to the AWS Management Console: https://console.aws.amazon.com/iam/
2. Click Services
3. Click IAM
4. Click on Users
5. Click on Security Credentials
6. Click on Create Access Key

7. Update programmatic call with new Access Key credentials

Perform following commands Via CLI to disable password:

```
aws iam delete-login-profile --user-name [USER_NAME]
```

For command usage refer: https://docs.aws.amazon.com/cli/latest/reference/iam/delete-login-profile.html

Note: Changes in account credentials may take upto 4 hours to get reflected in the AWS IAM evaluations. The time taken depends on when the last credential report was fetched by the Cloud View service and the time when changes were made in AWS IAM

Reference

- https://docs.aws.amazon.com/IAM/latest/UserGuide/console_controlling-access.html

Control ID - 401: Route53 A Record has Attached Resource

Criticality: MEDIUM

Specification

Amazon Route 53 is a highly available and scalable Domain Name System (DNS) web service. You can use Route 53 to perform three main functions in any combination: domain registration, DNS routing, and health checking.
Using alias records you can use Amazon Route 53 to map your zone apex (example.com versus www.example.com) to your Elastic Load Balancing instance, Amazon CloudFront distribution, AWS Elastic Beanstalk environment, API Gateway, VPC endpoint, or Amazon S3 website bucket.

Rationale

Amazon Route 53 alias records provide a Route 53–specific extension to DNS functionality. Alias records let you route traffic to selected AWS resources, such as CloudFront distributions and Amazon S3 buckets. They also let you route traffic from one record in a hosted zone to another record.

Evaluation

This control ensures that Route 53 A Record has Attached Resource.

Remediation

Using AWS Console:

1. Open the Amazon Route 53 console at https://console.aws.amazon.com/route53/v2.
2. In the left navigation panel, choose **Hosted zones**.
3. On the Hosted zones page, Choose hosted zone to be remediated.
4. Under Records tab, select the row for the Type **A** record that you want to edit.
5. Under **Record details**, Click on **Edit record** button.
6. Enable **Alias** option to route traffic to selected AWS resources.
7. Choose resource type from the dropdown menu.
8. Choose the region and select the resource from the region.
9. In the **Routing policy** choose routing policy from the dropdown menu.

10. Click **Save**.

Using AWS CLI:

Modify the following JSON syntax for pointing to an AWS resources. Following is an example for pointing to an AWS CloudFront distribution.

```
                    {
                        "Comment": "Swaps the Policy Record for a simple
routing policy",
                        "Changes": [
                            {
                                "Action": "UPSERT",
                                "ResourceRecordSet": {
                                    "Name": "[record-name]",
                                    "Type": "A",
                                    "AliasTarget": {
                                        "HostedZoneId": "Z2FDTNDATAQYW2",
                                        "DNSName": "xxxxxxxx.cloudfront.net",
                                        "EvaluateTargetHealth": false
                                    }
                                }
                            }
                        ]
                    }
```

To attach other types of resources please refer this link: AWS::Route53::RecordSetGroup AliasTarget

To create your resource record set in a hosted zone, use the following command:

```
aws route53 change-resource-record-sets \

                    --hosted-zone-id [hosted-zone-id] \

                    --change-batch file://sample.json
```

Reference

- Working with records
- Choosing between alias and non-alias records
- How do I create alias resource record sets in Route 53 using the AWS CLI?

Control ID - 403: Ensure public facing ALB are protected by WAF

Criticality: HIGH

Specification

AWS Web Application Firewall (WAF) helps protect your web applications or APIs against common web exploits and bots that may affect availability, compromise security, or consume excessive resources. AWS WAF gives you control over how traffic reaches your applications by enabling you to create security rules that

control bot traffic and block common attack patterns, such as SQL injection or cross-site scripting. You can also customize rules that filter out specific traffic patterns.

AWS Application Load Balancer (ALB) option for the Elastic Load Balancing service runs at the application layer. It allows you to define routing rules that are based on content that can span multiple containers or EC2 instances. Application Load Balancers support HTTP/2 and WebSocket, and give you additional visibility into the health of the target containers and instances.

Rationale

AWS Web Application Firewall (WAF) lets you monitor the HTTP(S) requests that are forwarded to an Application Load Balancer. AWS WAF also lets you control access to your content. Based on conditions that you specify, such as the IP addresses that requests originate from or the values of query strings, the service associated with your protected resource responds to requests either with the requested content or with an HTTP 403 status code (Forbidden).

Evaluation

This control ensures that public facing ALB is protected by WAF

Remediation

Using AWS Console:

To associate load balancer with existing Web ACL

1. Open the Amazon WAF console at https://console.aws.amazon.com/wafv2/.
2. From the AWS WAF home page, In the left navigation pane, choose **Web ACLs**.
3. Choose the web ACL that you want to associate with a resource.
4. On the **Associated AWS resources** tab, choose **Add AWS resources**.
5. Select **Resource type** - **Application Load Balancer**.
6. Select the load balancers you want to associate with the web ACL. Choose **Add**.
7. Choose **Save**.

Using AWS CLI:

To create WebACL, use the following command:

```
aws wafv2 create-web-acl \

        --name [web-acl-name] \

        --scope REGIONAL \

        --default-action Allow={} \

        --visibility-config
SampledRequestsEnabled=true,CloudWatchMetricsEnabled=true,MetricName=TestWebAc
lMetrics \

        --region [region]
```

To associate load balancer with WebACL, use the following command:

```
aws wafv2 associate-web-acl \

        --web-acl-arn [web-acl-arn] \

        --resource-arn [load-balancer-arn] \

        --region [region]
```

Reference

- Application Load Balancers and AWS WAF
- AWS WAF
- AWS Web Application Firewall (WAF) for Application Load Balancers
- AWS WAF2 CLI Reference

Control ID - 407: Ensure all data stored in the Elasticache Replication Group is securely encrypted at transit and has auth token

Criticality: MEDIUM

Specification

Amazon ElastiCache is a web service that makes it easy to set up, manage, and scale a distributed in-memory data store or cache environment in the cloud. It provides a high-performance, scalable, and cost-effective caching solution.Amazon ElastiCache in-transit encryption is an optional feature that allows you to increase the security of your data at its most vulnerable points. Using the Redis AUTH feature, the server can authenticate the clients while connecting to server.

Rationale

The in-transit encryption option in Amazon ElastiCache helps protect your data when it is moving from one location to another. To help keep your data secure, Amazon ElastiCache and Amazon EC2 provide Auth token mechanisms to guard against unauthorized access of your data on the server.

Evaluation

Ensure that all data stored in the Elasticache Replication Group has auth token.

Remediation

Using AWS Console:

Encryption in transit option is only available while creating the ElastiCache Redis Cluster. So to modify the option we have to first backup the ElastiCache Redis Cluster and delete the Cluster. After this operation, we can restore the backup with encryption in transit option enabled. Once the encryption in transit is enabled, cluster can be modified for user AUTH token.

1. Open AWS ElastiCache console at https://console.aws.amazon.com/elasticache/home.
2. In the navigation pane, choose Redis.
3. Choose the box to the left of the name of the Redis cluster you want to modify.
4. Go to the Actions dropdown and choose **Modify**.
5. In Modify cluster popup, select **Redis AUTH Default user** in **Access Control Option** dropdown.

6. For **AUTH token**, select **Set Auth** radio button.
7. Enter the token value in the text box for **Redis AUTH**.
8. Click on Modify button.

Using AWS CLI:

Run modify-replication-instance command to set Auth token for Redis replication group.

```
aws elasticache modify-replication-group \

        --replication-group-id [redis-cluster-arn] \

        --auth-token [token-value] \

        --auth-token-update-strategy SET \

        --apply-immediately
```

Reference

- Authenticating users with the Redis AUTH command
- ElastiCache for Redis in-transit encryption (TLS)
- Modify replication group CLI

Control ID - 411: Ensure that a log driver has been defined for each active Amazon ECS task definition

Criticality: HIGH

Specification

To debug and audit container for any production related issues, container logs should be available.

Rationale

Amazon ECS provides logging options to send container logs to either cloudwatch or splunk, these log drivers provide additional functionalities to debug and monitor logs. Containers have to be configured with logging to send logs to cloudwatch.

Evaluation

Control checks whether logging is configured for containers of an ECS task definition.

Remediation

Perform the following:

1. Sign in to the AWS management console and open the amazon ecs console at https://console.aws.amazon.com/ecs/.
2. In the left navigation panel, select **Task Definitions**
3. Click on the task definition to be remediated.
4. Select the latest active revision for selected task definition and choose **Create new revision**

5. On **Create new revision of Task Definition** page, under **container definitions** select the container to be reconfigured.
6. In **Edit container** page, under **Storage and Logging/Log configuration** section select **Auto-configure CloudWatch Logs** checkbox to configure logging automatically.
7. To configure values manually, select log driver value under **Log driver** value.
8. Add log groups in **Log options** section to store the logs
9. Click on **Update** to save changes
10. Click on **Create** to create new task definition version
11. Please repeat above steps for all containers

Using AWS CLI:

To create a task definition with logging configuration, use the following AWS CLI command:

```
aws ecs register-task-definition --cli-input-json
file://[path_to_json_file].json
```

Add below block to each and every container being created for the new task definition to configure logging

```
{
  "containerDefinitions": [
    {
      "name": "[CONTAINER_NAME]",
      "image": "[IMAGE_REPOSITORY]",
      "cpu": "CPU_CONFIGURATION",
      "command": [
        "sleep",
        "360"
      ],
      "memory": [
        "MEMORY_REQUIREMENT"
      ],
      "essential": true,
      "logConfiguration": {
        "logDriver": "awslogs",
        "options": {
          "awslogs-region": "[LOGS_REGION]",
          "awslogs-stream-prefix": "[LOGS_STREAM_PREFIX]",
          "awslogs-group": "[CLOUDWATCH_LOGGROUP]"
        }
      }
    }
  ],
  "family": "[TASK_DEFINITION_NAME]"
}
```

For more details on command, please refer to
https://docs.aws.amazon.com/cli/latest/reference/ecs/register-task-definition.html

Note: Exisiting version of task definition cannot be modified, new version has to be created. If the selected task definition is used in a service(ECS Cluster), you must update that service to use the new version of the task definition.

Reference

https://docs.aws.amazon.com/AmazonECS/latest/developerguide/using_awslogs.html

Control ID - 419: Ensure that AWS CloudFront distribution origins do not use insecure SSL protocols

Criticality: MEDIUM

Specification

Amazon CloudFront is a web service that speeds up distribution of your static and dynamic web content, such as .html, .css, .js, and image files, to your users. CloudFront delivers your content through a worldwide network of data centers called edge locations. An origin is the location where content is stored, and from which CloudFront gets content to serve to viewers.

You can configure CloudFront to require that viewers use HTTPS so that connections are encrypted when CloudFront communicates with viewers. You also can configure CloudFront to use HTTPS with your origin so that connections are encrypted when CloudFront communicates with your origin.

When your origin is an Amazon S3 bucket, your options for using HTTPS for communications with CloudFront depend on how you're using the bucket. If your Amazon S3 bucket is configured as a website endpoint, you can't configure CloudFront to use HTTPS to communicate with your origin because Amazon S3 doesn't support HTTPS connections in that configuration.

Rationale

Using insecure and deprecated SSL protocols could make the connection between the Cloudfront CDN and the origin server vulnerable to exploits.

Evaluation

This control ensures that AWS CloudFront distributions origin do not use insecure SSL protocols

Remediation

Using AWS Console:

1. Open the Amazon CloudFront console at https://console.aws.amazon.com/cloudfront/v3/home.
2. In the top pane of the CloudFront console, choose the ID for the distribution to be remediated.
3. Click the **Origins** tab, choose the origin that you want to update, and then choose **Edit**.
4. For **Minimum origin SSL protocol**, choose **TLSv1.2** SSL protocol that CloudFront uses with the origin.
5. Click **Save changes**.

Using AWS CLI:

To get the Distribution configuration and save the configuration output in `cf_config.json` file, use the following command:

```
aws cloudfront get-distribution-config \

        --id [distribution-id] \
```

```
        --output json > cf_config.json
```

To get the Etag for your config, use the following command:

```
ETAG=$(cat cf_config.json | jq -r '.ETag')
```

To get the `DistributionConfig`, use the following command:

```
cat cf_config.json | jq '.DistributionConfig' > cloudfront-config.json
```

To edit the `cloudfront-config.json` and update the OriginSslProtocols config for TLSv1.2, use the following command:

```
vi cloudfront-config.json
```

- Edit the `cloudfront-config.json` at `OriginSslProtocols`.
- For Ex.
-
-
```
                        "OriginSslProtocols": {
                            "Quantity": 1,
                            "Items": [
                                "TLSv1.2"
                          ]
                         },
```

To update the cloudfront distribution using `cloudfront-config.json` file, use the following command:

```
aws cloudfront update-distribution \

        --id [distribution-id] \

        --if-match $ETAG \

        --distribution-config file://cloudfront-config.json
```

For more details on command, please refer to
https://docs.aws.amazon.com/cli/latest/reference/cloudfront/update-distribution.html

Note: Update all the custom origins (if more than one).

Reference

- Using HTTPS with CloudFront
- Requiring HTTPS for communication between CloudFront and your Amazon S3 origin
- Requiring HTTPS for communication between CloudFront and your custom origin

Control ID - 426: Ensure Amazon API Gateway REST APIs are protected by AWS WAF

Criticality: HIGH

Specification

API Gateway provides a number of ways to protect your API from certain threats, like malicious users or spikes in traffic. You can protect your API using strategies like generating SSL certificates, configuring a web application firewall, setting throttling targets, and only allowing access to your API from a Virtual Private Cloud (VPC).

AWS WAF is a web application firewall that helps protect web applications and APIs from attacks. It enables you to configure a set of rules (called a web access control list (web ACL)) that allow, block, or count web requests based on customizable web security rules and conditions that you define.

Rationale

You can use AWS WAF to protect your API Gateway API from common web exploits, such as SQL injection and cross-site scripting (XSS) attacks. These could affect API availability and performance, compromise security, or consume excessive resources. For example, you can create rules to allow or block requests from specified IP address ranges, requests from CIDR blocks, requests that originate from a specific country or region, requests that contain malicious SQL code, or requests that contain malicious script.

Evaluation

This control ensures use of Amazon API Gateway REST APIs are protected by AWS WAF

Remediation

Using AWS Console:

1. Open the Amazon API Gateway console at https://console.aws.amazon.com/apigateway/.
2. In the left navigation panel, Select **APIs**.
3. Choose the **API**, and then choose **Stages**.
4. Under **Stages**, Select the API stage that you want to reconfigure in order to enable AWS WAF.
5. In the **Stage Editor** pane, Choose the **Settings** tab.
6. To associate a Web ACL with the API stage:

- Under **Web Application Firewall (WAF)**, In the **Web ACL** dropdown list, choose the web ACL that you want to associate with this stage.
- If the web ACL you need doesn't exist, Choose **Create WebACL**. Then go to AWS WAF console and create a web ACL. Then return to the API Gateway console to associate the web ACL with the stage. Repeat the steps no. 2 to 6.

- Click **Save changes**.

Using AWS CLI:

To associate an AWS WAF regional Web ACL with an API Gateway API stage , use the following command:

```
aws wafv2 associate-web-acl \

        --web-acl-arn [web-acl-arn] \

        --resource-arn arn:aws:apigateway:[region]::/restapis/[rest-api-
id]/stages/[stage-name] \

        --region [region]
```

Reference

- Using AWS WAF to protect your APIs
- Amazon API Gateway adds support for AWS WAF
- Protecting your API using Amazon API Gateway and AWS WAF — Part 2

Control ID - 427: Ensure client-side SSL certificates are used for HTTP backend authentication in AWS API Gateway REST APIs

Criticality: MEDIUM

Specification

API Gateway provides a number of ways to protect your API from certain threats, like malicious users or spikes in traffic. You can protect your API using strategies like generating SSL certificates, configuring a web application firewall, setting throttling targets, and only allowing access to your API from a Virtual Private Cloud (VPC).

You can use API Gateway to generate an SSL certificate and then use its public key in the backend to verify that HTTP requests to your backend system are from API Gateway. This allows your HTTP backend to control and accept only requests that originate from Amazon API Gateway, even if the backend is publicly accessible. The SSL certificates that are generated by API Gateway are self-signed, and only the public key of a certificate is visible in the API Gateway console or through the APIs.

Rationale

You can use SSL certificate to verify that HTTP requests to your backend system are from API Gateway. This allows your HTTP backend to control and accept only requests that originate from Amazon API Gateway, even if the backend is publicly accessible.

Evaluation

This control ensures that client-side SSL certificates are used for HTTP backend authentication in AWS API Gateway REST APIs

Remediation

Using AWS Console:

To generate a client-side SSL certificate:

1. Open the Amazon API Gateway console at https://console.aws.amazon.com/apigateway/.
2. Choose a REST API.
3. In the left navigation panel, Select **Client Certificates**.
4. From the dashboard top menu click **+ Generate Client Certificate** to create a new client-side SSL certificate.
5. Optional, click **Edit**, and add a descriptive title for the generated certificate and click **Save** to save the description.

To configure an API to use SSL certificates:

1. In the API Gateway console, Select **APIs**.
2. Choose the **API**, and then choose **Stages** in the left navigation menu.
3. Under **Stages**, Select the API stage that you want to reconfigure in order to use the client certificate.
4. In the **Stage Editor** pane, Choose the **Settings** tab.
5. Select a newly created certificate under the **Client Certificate** section.
6. Click **Save changes**.

Using AWS CLI:

To generate a client-side SSL certificate, use the following command:

```
aws apigateway generate-client-certificate \

        --region [region] \

        --description ["certificate-description"]
```

To configure an API to use client-side SSL certificate, use the following command:

```
aws apigateway update-stage \

        --region [region] \

        --rest-api-id [rest-api-id] \

        --stage-name [stage-name] \

        --patch-operations
op=replace,path=/clientCertificateId,value=[certificate-id]
```

Reference

- Controlling and managing access to a REST API in API Gateway
- Generate and configure an SSL certificate for backend authentication
- API Gateway-supported certificate authorities for HTTP and HTTP proxy integrations

Control ID - 428: Ensure that SSL certificates associated with API Gateway REST APIs are rotated periodically

Criticality: MEDIUM

Specification

API Gateway provides a number of ways to protect your API from certain threats, like malicious users or spikes in traffic. You can protect your API using strategies like generating SSL certificates, configuring a web application firewall, setting throttling targets, and only allowing access to your API from a Virtual Private Cloud (VPC).

You can use API Gateway to generate an SSL certificate and then use its public key in the backend to verify that HTTP requests to your backend system are from API Gateway. This allows your HTTP backend to control and accept only requests that originate from Amazon API Gateway, even if the backend is publicly accessible. The SSL certificates that are generated by API Gateway are self-signed, and only the public key of a certificate is visible in the API Gateway console or through the APIs.

Rationale

The SSL certificates used by Amazon API Gateway service are valid for 365 days. To avoid any downtime for your Amazon API Gateway REST APIs, rotate the associated certificates before they expire.

Evaluation

This control ensures that SSL certificates associated with API Gateway REST APIs are rotated periodically

Remediation

Using AWS Console:

To generate a client-side SSL certificate:

1. Open the Amazon API Gateway console at https://console.aws.amazon.com/apigateway/.
2. In the API Gateway console, Select **APIs**.
3. Choose the Rest API.
4. In the left navigation panel, Select **Client Certificates**.
5. From the dashboard top menu click **+ Generate Client Certificate** to create a new client-side SSL certificate.
6. Optional, click **Edit**, and add a descriptive title for the generated certificate and click **Save** to save the description.

To replace/rotate the old certificate:

1. In the navigation panel, Select **APIs**.
2. Choose the **API**, and then choose **Stages**.
3. Under **Stages**, Select the API stage that you want to reconfigure in order to use the client certificate.
4. In the **Stage Editor** pane, Choose the **Settings** tab.
5. Select a newly created certificate under the **Client Certificate** section.
6. Click **Save changes**.

Using AWS CLI:

To generate a client-side SSL certificate, use the following command:

```
aws apigateway generate-client-certificate \

        --region [region] \

        --description ["certificate-description"]
```

To configure an API to use client-side SSL certificate, use the following command:

```
aws apigateway update-stage \

        --region [region] \

        --rest-api-id [rest-api-id] \

        --stage-name [stage-name] \

        --patch-operations
op=replace,path=/clientCertificateId,value=[certificate-id]
```

Reference

- Controlling and managing access to a REST API in API Gateway
- Generate and configure an SSL certificate for backend authentication
- API Gateway-supported certificate authorities for HTTP and HTTP proxy integrations

Control ID - 429: Ensure AWS CloudFront distributions use improved security policies for HTTPS connections

Criticality: MEDIUM

Specification

Amazon CloudFront is a web service that speeds up distribution of your static and dynamic web content, such as .html, .css, .js, and image files, to your users. When you require HTTPS between viewers and your CloudFront distribution, you must choose a security policy. The security policy setting has 1. Minimum SSL/TLS protocol that CloudFront uses to communicate with viewers. 2. The ciphers that CloudFront can use to encrypt the communication with viewers.

Rationale

Using TLS protocols version less than 1.2 for your Cloudfront distributions could make the connection between the Cloudfront CDN and the viewer vulnerable which allows an attacker to retrieve sensitive data by implementing tactics like man-in-the-middle.

Evaluation

This control ensures that AWS CloudFront distributions are using improved security policies for HTTPS connections.

Remediation

Using AWS Console:

1. Open the Amazon CloudFront console at https://console.aws.amazon.com/cloudfront/v3/home.
2. In the top pane of the CloudFront console, choose the ID for the distribution to be remediated.
3. In the **General** tab, scroll to Settings section and click on Edit button.
4. Select the custom SSL certificate, if not selected.
5. Under Security policy, choose any TLS version from TLSv1.2_2018, TLSv1.2_2019, TLSv1.2_2021
6. Click **Save changes**.

Using AWS CLI:

To get the Distribution configuration and save the configuration output in `cf_config.json` file, use the following command:

```
aws cloudfront get-distribution-config \

        --id [distribution-id] \

        --output json > cf_config.json
```

To get the Etag for your config, use the following command:

```
ETAG=$(cat cf_config.json | jq -r '.ETag')
```

To get the `DistributionConfig`, use the following command:

```
cat cf_config.json | jq '.DistributionConfig' > cloudfront-config.json
```

To edit the `cloudfront-config.json` and update the OriginProtocolPolicy to Https-only, use the following command:

```
vi cloudfront-config.json
```

- Edit the `cloudfront-config.json` at `MinimumProtocolVersion and Certificate`.
- For Ex.
-
-
-
-
-
-
-
-

```
                        "ViewerCertificate": {
                                "SSLSupportMethod": "sni-only",
                                "ACMCertificateArn": "[certificate-
arn]",
                                "MinimumProtocolVersion":
"TLSv1.2_2018",
                                "Certificate": "[certificate-arn]",
                                "CertificateSource": "acm"
                        },
```

To update the cloudfront distribution using `cloudfront-config.json` file, use the following command:

```
aws cloudfront update-distribution \

        --id [distribution-id] \

        --if-match $ETAG \

        --distribution-config file://cloudfront-config.json
```

Note: Cli reference - [aws cloudfront update-distribution]

Reference

- [Supported protocols and ciphers between viewers and CloudFront]
- [Update distribution CLI]

Control ID - 430: Ensure the traffic between the AWS CloudFront distributions and their origins is encrypted

Criticality: HIGH

Specification

Amazon CloudFront is a web service that speeds up distribution of your static and dynamic web content, such as .html, .css, .js, and image files, to your users. CloudFront delivers your content through a worldwide network of data centers called edge locations. An origin is the location where content is stored, and from which CloudFront gets content to serve to viewers.

You can configure CloudFront to use HTTPS with your origin by setting origin protocol policy to HTTPS only, so that connections are encrypted when CloudFront communicates with your origin. When your origin is an Amazon S3 bucket, your options for using HTTPS for communications with CloudFront depend on how you're using the bucket. If your Amazon S3 bucket is configured as a website endpoint, you can't configure CloudFront to use HTTPS to communicate with your origin because Amazon S3 doesn't support HTTPS connections in that configuration.

Rationale

Using insecure SSL protocols for your Cloudfront distributions could make the connection between the Cloudfront CDN and the origin server vulnerable which allows an attacker to retrieve sensitive data by implementing tactics like man-in-the-middle.

Evaluation

This control ensures that the traffic between the AWS CloudFront distributions and their origins is encrypted. (Origin protocol policy)

Remediation

Using AWS Console:

1. Open the Amazon CloudFront console at https://console.aws.amazon.com/cloudfront/v3/home.
2. In the top pane of the CloudFront console, choose the ID for the distribution to be remediated.
3. Click the **Origins** tab, choose the origin that you want to update, and then choose **Edit**.
4. if the origin domain is other than Amazon S3, then follow the steps
5. On the Origin settings pane, for **Origin protocol policy**, choose **HTTPS only**.
6. Click **Save changes**.

Using AWS CLI:

To get the Distribution configuration and save the configuration output in `cf_config.json` file, use the following command:

```
aws cloudfront get-distribution-config \

        --id [distribution-id] \

        --output json > cf_config.json
```

To get the Etag for your config, use the following command:

```
ETAG=$(cat cf_config.json | jq -r '.ETag')
```

To get the `DistributionConfig`, use the following command:

```
cat cf_config.json | jq '.DistributionConfig' > cloudfront-config.json
```

To edit the `cloudfront-config.json` and update the OriginProtocolPolicy to Https-only, use the following command:

```
vi cloudfront-config.json
```

- Edit the `cloudfront-config.json` at `OriginSslProtocols`.
- For Ex.
-

```
"CustomOriginConfig": {
"OriginProtocolPolicy": "https-only",
"HTTPSPort": 443,
"OriginSslProtocols": {
    "Items": [
        "TLSv1.2"
    ],
    "Quantity": 1
},
```

To update the cloudfront distribution using `cloudfront-config.json` file, use the following command:

```
aws cloudfront update-distribution \

        --id [distribution-id] \

        --if-match $ETAG \

        --distribution-config file://cloudfront-config.json
```

Reference

- Data protection in Amazon CloudFront
- Requiring HTTPS for communication between CloudFront and your custom origin

Control ID - 431: Ensure your AWS Cloudfront distributions are using an origin access identity for their origin S3 buckets

Criticality: MEDIUM

Specification

When you first set up an Amazon S3 bucket as the origin for a CloudFront distribution, you grant everyone permission to read the files in your bucket. This allows anyone to access your files either through CloudFront or using the Amazon S3 URL. CloudFront doesn't expose Amazon S3 URLs, but your users might have those URLs if your application serves any files directly from Amazon S3 or if anyone gives out direct links to specific files in Amazon S3.

If you use CloudFront signed URLs or signed cookies to restrict access to files in your Amazon S3 bucket, you probably also want to prevent users from accessing your Amazon S3 files by using Amazon S3 URLs. If users access your files directly in Amazon S3, they bypass the controls provided by CloudFront signed URLs or signed cookies. This includes control over the date and time that a user can no longer access your content, and control over which IP addresses can be used to access content. In addition, if users access files both

through CloudFront and directly by using Amazon S3 URLs, CloudFront access logs are less useful because they're incomplete.

Rationale

By enabling origin access identity feature for Cloudfront distribution S3 origins restrict any direct access to your objects through Amazon S3 URLs.

Evaluation

This control ensures that your AWS Cloudfront distributions are using an origin access identity for their origin S3 buckets

Remediation

Using AWS Console:

1. Open the Amazon CloudFront console at https://console.aws.amazon.com/cloudfront/v3/home.
2. In the top pane of the CloudFront console, choose the ID for the distribution to be remediated.
3. Click the **Origins** tab, Select the S3 origin, and then choose **Edit**.
4. If the origin domain is amazon S3, then follow the next steps.
5. On the origin settings page, For **S3 bucket access**, select **Yes use OAI (bucket can restrict access to only CloudFront)**.
6. For **Origin access identity**, select an existing identity from the dropdown list or choose **Create new OAI**.
7. For **Bucket policy**, select **Yes, update the bucket policy**.
   Note: This step updates the bucket policy of your S3 origin to grant the OAI access for `s3:GetObject`.
8. Click **Save changes**.

Using AWS CLI:

To get the Distribution configuration and save the configuration output in `cf_config.json` file, use the following command:

```
aws cloudfront get-distribution-config \

        --id [distribution-id] \

        --output json > cf_config.json
```

To get the Etag for your config, use the following command:

```
ETAG=$(cat cf_config.json | jq -r '.ETag')
```

To get the `DistributionConfig`, use the following command:

```
cat cf_config.json | jq '.DistributionConfig' > cloudfront-config.json
```

To list the origin access identitities, use the following command:

```
aws cloudfront list-cloud-front-origin-access-identities
```

Copy the origin access identity id at `CloudFrontOriginAccessIdentityList.Items.Id` from command output.

To edit the `cloudfront-config.json` and update the `OriginAccessIdentity`, use the following command:

```
vi cloudfront-config.json
```

Edit the `cloudfront-config.json` at `OriginAccessIdentity`.

```
                    "S3OriginConfig": {
                        "OriginAccessIdentity": "origin-access-
identity/cloudfront/[ID-of-origin-access-identity]"
                        },
```

To update the cloudfront distribution origin access identity for their origin S3 buckets using `cloudfront-config.json` file, use the following command:

```
aws cloudfront update-distribution \

        --id [distribution-id] \

        --if-match $ETAG \

        --distribution-config file://cloudfront-config.json
```

For more details on command, please refer to
https://docs.aws.amazon.com/cli/latest/reference/cloudfront/update-distribution.html

Note: Update all the S3 origins (if more than one).

Reference

- S3 origin with CloudFront
- Restricting access to Amazon S3 content by using an origin access identity (OAI)
- cloudfront-origin-access-identity-enabled
- How do I use CloudFront to serve a static website hosted on Amazon S3?
- Identity and access management in Amazon S3

Control ID - 433: Ensure EC2 Instances are using IAM Roles

Criticality: MEDIUM

Specification

Use an IAM role to manage temporary credentials for applications that run on an EC2 instance.
When you use a role, you don't have to distribute long-term credentials (such as a user name and password or access keys) to an EC2 instance.
Instead, the role supplies temporary permissions that applications can use when they make calls to other AWS resources.
When you launch an EC2 instance, you specify an IAM role to associate with the instance.

Applications that run on the instance can then use the role-supplied temporary credentials to sign API requests.

Rationale

Using IAM Roles over IAM Access Keys to sign AWS API requests has multiple benefits.
For example, once enabled, you or your administrators don't have to manage credentials anymore as the credentials provided by the IAM roles are temporary and rotated automatically behind the scenes.
You can use a single role for multiple EC2 instances within your stack, manage its access policies in one place and allow these to propagate automatically to all instances.
Also, you can easily restrict which role a IAM user can assign to an EC2 instance during the launch process in order to stop the user from trying to gain elevated (overly permissive) privileges.

Evaluation

This control ensures that EC2 Instances are using IAM Roles

Remediation

Using AWS Console

1. Open AWS EC2 console at https://console.aws.amazon.com/ec2/v2/home.
2. In the left navigation panel, under **Instances** select **Instances**.
3. Click on the instance to be modified.
4. On the Instance summary screen, select **Security -> Modify IAM role** from **Actions** dropdown.
5. On the modify IAM role choose the IAM role to be used or create a new IAM role.
6. Choose **Save** to modify IAM role.

Using AWS CLI

To associate instance-profile to an EC2 Instance, use the following command:

```
aws ec2 associate-iam-instance-profile \

               --iam-instance-profile Name=[IAM-role-name]\

               --instance-id [EC2-instance-id]
```

Reference

- IAM roles for Amazon EC2
- Using an IAM role to grant permissions to applications running on Amazon EC2 instances

Control ID - 434: Ensure no backend EC2 instances are running in public subnets

Criticality: HIGH

Specification

An Amazon EC2 instance is a virtual server in Amazon's Elastic Compute Cloud (EC2) for running applications on the Amazon Web Services (AWS) infrastructure. These EC2 instances are provisioned inside a VPC and subnets.

Subnets enable you to control traffic to your instance, including the kind of traffic that can reach your instance. To enable network access to your instance, you must allow inbound traffic to your instance through public or private subnet. A route table contains a set of rules, called routes, that are used to determine where network traffic from your subnet or gateway is directed. You can set rules and enable access through specific IP address range, thus making subnet a private subnet. If the GatewayId value set to igw-xxxxxxxx and the DestinationCidrBlock value set to 0.0.0.0/0 in the routes table, the instance is provisioned in a public subnet.

Rationale

By provisioning EC2 instances within a private subnet, you will prevent these instances being exposed to the internet, therefore no malicious requests can reach your backend instances.

Evaluation

This control ensures no backend EC2 instances are running in public subnets.

Remediation

Using AWS Console:

1. Open the Amazon EC2 console.
2. In the navigation pane, choose **Instances**
3. To create an AMI for the current instance, follow given steps.

- Select your instance and, select **Image and Templates** from Actions dropdown.
- Click on Create Image option.
- Enter the image name and click on create image button at the bottom.

- launch a new instance using the AMI created

- In the navigation pane, go to the AMIs option
- Choose the newly created AMI and click on Launch instance from image.
- Choose the instance type and click on Configure instance details.
- Select the network and the private subnet (the GatewayId value is not set to igw-xxxxxxxx and the DestinationCidrBlock value is not set to 0.0.0.0/0 in the routes table)
- Click on Review and launch button.
- Choose Launch
- Select an existing keypair for authorization.
- Click on launch instances.

- Terminate the previous instance when new instance is available.

Using AWS CLI:

Create an AMI for the instance using the following command.

```
aws ec2 create-image \

        --region [region] \

        --instance-id [instance-id] \

        --name [AMI-name] \
```

```
        --description [AMI-description] \

        --no-reboot
```

Launch a new instance using the image created, using following command.

```
aws ec2 run-instances \

        --region [region] \

        --image-id [AMI-id] \

        --subnet-id [private-subnet-id] \

        --count [count] \

        --instance-type [instance-type] \

        --key-name [keypair]
```

Reference

- VPC with public and private subnets (NAT)
- Cli to create EC2 instance.

Control ID - 436: Ensure to encrypt data in transit for SNS topic

Criticality: LOW

Specification

AWS recommends that you use HTTPS instead of HTTP while publishing messages to a topic. When you use HTTPS, messages are automatically encrypted during transit, even if the SNS topic itself isn't encrypted. Without HTTPS, a network-based attacker can eavesdrop on network traffic or manipulate it using an attack such as man-in-the-middle.

Rationale

To enable only encrypted connections over HTTPS , add the aws:SecureTransport condition in access policy attached to SNS topics, which will make publishers to use HTTPS.

Evaluation

This control ensures that SNS topics encryption in transit enabled.

Remediation

Using AWS Console:

1. Go to AWS Console SNS dashboard at https://console.aws.amazon.com/sns/
2. In the left navigation panel select **Topics**, select effected **topic**.
3. Under **Access policy** section, add below policy to the existing policy,

```
{
    "Id": "ExamplePolicy",
    "Version": "2012-10-17",
    "Statement": [
      {
        [EXISTING_POLICY]
      },
      {
        "Sid": "AllowPublishThroughSSLOnly",
        "Action": "SNS:Publish",
        "Effect": "Deny",
        "Resource": "[TOPIC_ARN]",
        "Condition": {
          "Bool": {
            "aws:SecureTransport": "false"
          }
        },
        "Principal": "*"
      }
    ]
  }
```

1. Click on **Save changes**.

Using AWS CLI:

```
aws sns set-topic-attributes --topic-arn [topic_arn] --attribute-name Policy --attribute-value file://[Policy.json]
```

Note: For more details on command, please refer to https://docs.aws.amazon.com/cli/latest/reference/sns/set-topic-attributes.html

Reference

- https://docs.aws.amazon.com/sns/latest/dg/sns-security-best-practices.html#enforce-encryption-data-in-transit

Control ID - 437: Ensure unused AWS EC2 key pairs are decommissioned

Criticality: HIGH

Specification

You can use Amazon EC2 to launch as many or as few virtual servers as you need, configure security and networking, and manage storage. A key pair, consisting of a public key and a private key, is a set of security credentials that you use to prove your identity when connecting to an Amazon EC2 instance. Amazon EC2 stores the public key on your instance, and you store the private key.

Rationale

Removing unused SSH key pairs can significantly reduce the risk of unauthorized access to your AWS EC2 instances as these key pairs can be reassociated at any time, providing access to the wrong users. Ideally, you

will want to restrict access to your EC2 resources for all individuals who leave your organization and still possess the private key from the SSH key pair used.

Evaluation

This control ensures that unused AWS EC2 key pairs are decommissioned.

Remediation

Using AWS Console

1. Open AWS EC2 console at https://console.aws.amazon.com/ec2/v2/home.
2. In the left navigation panel, under **Network & Security** section.
3. Click on **Key Pairs** option and copy the name of the **Key pair** you want to examine.
4. In the left navigation panel, under **Instances** section. Click on **Instances** option.
5. In the top search box, search for **keypair=[name of keypair]**. If no instance is listed, then key pair is unused.
6. In the left navigation panel, under **Network & Security** section. Go to the **Keypairs** option.
7. Select the Key pair and in the **Actions** dropdown, select **delete.**.
8. Confirm the deletion and click on delete.

Using AWS CLI

To delete a keypair use following command:

```
aws ec2 delete-key-pair \

        --region [region] \

        --key-name [keypair-name]
```

Reference

- Security in Amazon EC2
- Amazon EC2 key pairs reference

Control ID - 438: Ensure AWS SNS topics do not allow HTTP subscriptions

Criticality: LOW

Specification

AWS recommends that you use HTTPS instead of HTTP while subscribing to a topic. When you use HTTPS, messages are automatically encrypted during transit, even if the SNS topic itself isn't encrypted. Without HTTPS, a network-based attacker can eavesdrop on network traffic or manipulate it using an attack such as man-in-the-middle.

Rationale

To enable only encrypted connections over HTTPS , add the SNS:Protocol condition in access policy attached to SNS topic and assign https value, which will make publishers to use HTTPS.

Evaluation

This control ensures that SNS topics do not allow Http subscriptions.

Remediation

Using AWS Console:

1. Go to AWS Console SNS dashboard at https://console.aws.amazon.com/sns/
2. In the left navigation panel select **Topics**, select effected **topic**.
3. Under **Access policy** section, add below policy to the existing policy,

```
{
    "Id": "ExamplePolicy",
    "Version": "2012-10-17",
    "Statement": [
      {
        [EXISTING_POLICY]
      },
      {
        "Sid": "AllowPublishThroughSSLOnly",
        "Action": "SNS:Subscribe",
        "Effect": "Deny",
        "Resource": "[TOPIC_ARN]",
        "Condition": {
            "StringEquals": {
                "SNS:Protocol": "http"
            }
        },
        "Principal": "*"
      }
    ]
  }
```

4. Click on **Save changes**.

Using AWS CLI:

```
aws sns set-topic-attributes --topic-arn [topic_arn] --attribute-name Policy --attribute-value file://[Policy.json]
```

Note: For more details on command, please refer to https://docs.aws.amazon.com/cli/latest/reference/sns/set-topic-attributes.html

Reference

- https://docs.aws.amazon.com/sns/latest/dg/sns-access-policy-use-cases.html#sns-limit-subscriptions-to-https

Control ID - 439: Ensure that Elastic File System does not have the default access policy

Criticality: MEDIUM

Specification

Amazon Elastic File System is a cloud storage service provided by Amazon Web Services designed to provide scalable, elastic, concurrent with some restrictions, and encrypted file storage. Like every AWS resource, Amazon EFS, also support attaching permission policies to resources. A permissions policy describes who has access to what. Policies attached to an IAM identity are referred to as identity-based policies (IAM policies) and policies attached to a resource are referred to as resource-based policies. Amazon EFS supports both identity-based policies and resource-based policies.

Rationale

By creating file system policy for EFS, you decide who is getting the permissions, the resources they get permissions for, and the specific actions that you want to allow on those resources. This helps to secure your resources stored in the file system.

Evaluation

This control ensures that Elastic File System does not have any default access policy.

Remediation

Using AWS Console:

1. Open the File systems on the EFS console.
2. Choose the filesystem for which you want to remediate.
3. Scroll down and click on the **File system policy** tab.
4. Click on Edit button to open File system policy editor.
5. In Policy options, you can choose any combination of the preconfigured file system policies (Prevent root access by default, Enforce read-only access by default, Prevent anonymous access, Enforce in-transit encryption for all clients)
6. Click on Grant additional permissions to grant file system permissions to additional IAM principals.
7. Click on Add button. Enter the principal ARN of the entity that you are granting permissions to. Then choose the Permissions that you want to grant.
8. Policy editor can be used to customise the predefined policy.
9. Choose **Save** button.

Using AWS CLI:

To create a file system policy for the EFS which gives read only access to a AWS account, use following command.

```
aws efs put-file-system-policy \

        --file-system-id [file-system-id] \

        --policy "{\"Id\": \"1\", \"Statement\": [{\"Effect\": \"Allow\",
\"Action\": [\"elasticfilesystem:ClientMount\" ], \"Principal\": {\"AWS\":
\"[IAM-user-arn]\"}}]}"
```

Note: Refer CLI documentation put-file-system-policy

Reference

- Overview of managing access permissions to your Amazon EFS resources
- Using IAM to control file system data access
- Creating file system policies

Control ID - 440: Ensure that the latest version of Memcached is used for AWS ElastiCache clusters

Criticality: MEDIUM

Specification

Amazon ElastiCache is a web service that makes it easy to set up, manage, and scale a distributed in-memory data store or cache environment in the cloud. Amazon ElastiCache works with both the Redis and Memcached engines. You initiate engine version upgrades to your cluster or replication group by modifying it and specifying a new engine version as version upgrades might involve some compatibility risks.

Rationale

ElastiCache cluster should use a updated to latest stable version of the cache engine as to get benefit from better security by having the recent vulnerability patches, receive the latest software features and get the latest performance optimizations.

Evaluation

This control ensures that the latest version of Memcached is used for your AWS ElastiCache clusters.

Remediation

Using AWS Console:

1. Go to https://console.aws.amazon.com/elasticache/home
2. In the left navigation panel, under ElastiCache Dashboard, click Memcached to access the cache clusters created with the Memcached.
3. Select the ElastiCache Memcached cache cluster that you want to Modify.
4. Click on the Modify button from the dashboard.
5. Inside Modify Cluster dialog box, select the latest stable version available from the Engine Version Compatibility dropdown list
6. Click on the Modify to save the changes.

Using AWS CLI:

```
aws elasticache modify-cache-cluster \

    --region [region] \

    --cache-cluster-id [cluster-name] \
```

```
--engine-version [latest-version] \

--apply-immediately
```

For command usage refer: https://docs.aws.amazon.com/cli/latest/reference/elasticache/modify-cache-cluster.html

Reference

- Supported ElastiCache for Memcached versions
- Comparing Memcached and Redis

Control ID - 443: Ensure that Route 53 Hosted Zone has configured logging for DNS queries

Criticality: MEDIUM

Specification

Query logs contain only the queries that DNS resolvers forward to Route 53. If a DNS resolver has already cached the response to a query (such as the IP address for a load balancer for example.com), the resolver will continue to return the cached response without forwarding the query to Route 53 until the TTL for the corresponding record expires.

Amazon Route 53 sends query logs directly to CloudWatch Logs, the logs are never accessible through Route 53. Instead, you use CloudWatch Logs to view logs in near real-time, search and filter data, and export logs to Amazon S3. Route 53 creates one CloudWatch Logs log stream for each Route 53 edge location that responds to DNS queries for the specified hosted zone and sends query logs to the applicable log stream.

Rationale

You can configure Amazon Route 53 Hosted zones to log information about the public DNS queries that Route 53 receives, such as the domain or subdomain that was requested, the date and time of the request, Route 53 edge location that responded to the DNS query, and the DNS record type, such as A or AAAA.

Once you configure query logging, Route 53 will send logs to CloudWatch Logs. You use CloudWatch Logs tools to access the query logs.

Evaluation

This control ensures that Route 53 Hosted Zone has configured logging for DNS queries

Remediation

Using AWS Console:

1. Open the Amazon Route 53 console at https://console.aws.amazon.com/route53/.
2. In the left navigation pane, choose **Hosted zones**.
3. Choose the hosted zone that you want to configure query logging for.
4. In the **Hosted zone details** pane, choose **Configure query logging**.
5. Choose an existing log group or create a new log group.

6. If you receive an alert about permissions (this happens if you haven't configured query logging with the new console before), do one of the following:

- If you have 10 resource policies already, you can't create any more. Select any of your resource policies, and select **Edit**. Editing will give Route 53 permissions to write logs to your log groups. Choose **Save**. The alert goes away and you can continue to the next step.
- If you have never configured query logging before (or if you haven't created 10 resource policies already), you need to grant permissions to Route 53 to write logs to your CloudWatch Logs groups. Choose **Grant permissions**. The alert goes away and you can continue to the next step.

- Choose **Permissions - optional** to see a table that shows whether the resource policy matches the CloudWatch log group, and whether the Route 53 has the permission to publish logs to CloudWatch.
- Choose **Create**.

Using AWS CLI:

To configure query logging for AWS Route 53 hosted zone, use the following command:

```
aws route53 create-query-logging-config \

        --hosted-zone-id [hosted-zone-id] \

        --cloud-watch-logs-log-group-arn [cloud-watch-log-group-arn]
```

To get the ARN for a log group, you can use the CloudWatch console.

Reference

- Public DNS query logging
- Logging and monitoring in Amazon Route 53
- Logging Amazon Route 53 API calls with AWS CloudTrail

Control ID - 444: Ensure that DNSSEC Signing is enabled for Route 53 Hosted Zones

Criticality: MEDIUM

Specification

DNS Security Extensions (DNSSEC) signing on public zones for Amazon Route 53 and validation for Amazon Route 53 Resolver. DNSSEC is a specification that provides data integrity assurance for DNS and helps customers meet compliance mandates (for example, FedRAMP and security standards such as NIST).

Rationale

When you enable DNSSEC signing for a hosted zone, Route 53 adds cryptographic signatures to your DNS records. However, signing, by itself, is not sufficient to secure your zone. You also must establish a chain of trust to the zone. When you enable DNSSEC validation for your VPC in Route 53 Resolver, the resolver validates those signatures, confirming that no one tampered with the record. This helps protect you against DNS spoofing, cache poisoning, or other DNS-related man-in-the-middle attacks.

Evaluation

This control ensures that DNSSEC Signing is enabled for Route 53 Hosted Zones

Remediation

Using AWS Console:

To enable DNSSEC signing and create a KSK:

1. Open the Amazon Route 53 console at https://console.aws.amazon.com/route53/.
2. In the left navigation pane, choose **Hosted zones**.
3. Choose a public hosted zone that you want to enable DNSSEC signing.
4. Click **DNSSEC signing** tab and choose **Enable DNSSEC signing**.
5. **Note:-** If the option in this section is **Disable DNSSEC signing**, you have already completed the first step in enabling DNSSEC signing. Be sure that you establish, or that there already exists, a chain of trust for the hosted zone for DNSSEC, and then you're done.
6. On the **Enable DNSSEC signing** page, For **Provide KSK name**, enter an name for the KSK that Amazon Route 53 will create for you.
7. Under **Customer managed CMK in AWS KMS**, Perform one of the following:

- Select **Choose customer managed CMK** to use an existing customer-managed CMK that applies to DNSSEC signing.
- Select **Create customer managed CMK** to create a new customer-managed CMK for DNSSEC signing. Enter an alias for the new key in the **Create customer managed CMK** box.

- Click **Create KSK and enable signing** to enable DNSSEC signing for the selected hosted zone.

Using AWS CLI:

To create new key-signing key (KSK) to associate with a hosted zone, use the following command:

```
aws route53 create-key-signing-key \

        --hosted-zone-id [hosted-zone-id] \

        --key-management-service-arn [customer-managed-key-arn] \

        --name [ksk-name] \

        --status ACTIVE \

        --caller-reference ["unique-string-that-identifies-request"] \

        --region [region]
```

To enable DNSSEC signing for hosted zone, use the following command:

```
aws route53 enable-hosted-zone-dnssec \

        --hosted-zone-id [hosted-zone-id] \

        --region [region]
```

Reference

- Configuring DNSSEC signing in Amazon Route 53

- Enabling DNSSEC signing and establishing a chain of trust
- Configuring DNSSEC signing and validation with Amazon Route 53

Control ID - 445: Ensure that Route 53 domains have Privacy Protection enabled

Criticality: MEDIUM

Specification

When AWS Route 53 Privacy Protection is disabled for a domain, anyone can send a WHOIS query for the domain and, for most top-level domains (TLDs), might be able to get all the contact information that you provided when you registered or transferred the domain, including name, address, phone number, and email address. The WHOIS command is widely available, it's included in many operating systems, and it's also available as a web application on many websites.

Rationale

Enabling Privacy Protection feature hides most of your contact information from WHOIS ("Who is") queries and reduces the amount of spam that you receive. Your contact information is replaced either with contact information for the registrar or with the phrase "REDACTED FOR PRIVACY".

Evaluation

This control ensures that Route 53 domains has Privacy Protection enabled

Remediation

Using AWS Console:

1. Open the Amazon Route 53 console at https://console.aws.amazon.com/route53/.
2. In the left navigation panel, under **Domains**, click **Registered Domains**.
3. Choose the domain name that you want to enable privacy protection.
4. Click **Edit Contacts**.
5. In the bottom of the page, For **Privacy Protection** settings choose **Enable** for all three contacts: administrative, registrant, and technical.
6. Click **Save** to apply the changes.

Using AWS CLI:

To enable privacy protection for AWS Route 53 domain, use the following command:

```
aws route53domains update-domain-contact-privacy \

        --domain-name [domain-name] \

        --admin-privacy \

        --registrant-privacy \

        --tech-privacy
```

Reference

- Updating contact information and ownership for a domain
- Enabling or disabling privacy protection for contact information for a domain

Control ID - 446: Ensure a loggroup is created to upload logs of datasync task to the cloudwatch log group

Criticality: LOW

Specification

Any transfer related information or errors have to be logged for further processing and analysis of transfer errors that might happen while task execution.

Rationale

It is recommended to enable logging of transfer related information and errors into CloudWatch log group for all data sync tasks created.

Evaluation

This control ensures logging is enabled for data sync tasks into CloudWatch log group.

Remediation

Perform the following via AWS management console:

1. Sign in to AWS Console and navigate to https://console.aws.amazon.com/datasync.
2. Click on **Tasks** in navigation pane
3. Click on the task to be remediated
4. Click on **Edit**
5. Under **Task logging** section, select any value other than Do not send logs to CloudWatch under **Log level** dropdown according to the requirement.
6. Click on **Save changes**.

Perform following commands via CLI :

```
aws datasync update-task --task-arn
arn:aws:datasync:[REGION]:[ACCOUNT_ID]:task/[TASK_ID] --options LogLevel=BASIC
--cloud-watch-log-group-arn arn:aws:logs:[REGION]:[ACCOUNT_ID]:log-
group:[LOG_GROUP] --region [REGION]
```

For command usage refer: https://docs.aws.amazon.com/cli/latest/reference/datasync/update-task.html

Reference

- https://docs.aws.amazon.com/datasync/latest/userguide/create-task.html#configure-logging

Control ID - 447: Ensure to enable data integrity checks for only files transferred in datasync task

Criticality: LOW

Specification

As DataSync transfers data, it always performs data integrity checks during the transfer. You can enable additional verification to compare source and destination at the end of a transfer.

Rationale

For most use cases, we recommend verifying only the files transferred.

Evaluation

This control ensures to enable data integrity checks for only files transferred.

Remediation

Perform the following via AWS management console:

1. Sign in to AWS Console and navigate to https://console.aws.amazon.com/datasync.
2. Click on **Tasks** in navigation pane
3. Click on the task to be remediated
4. Click on **Edit**
5. Under **Task execution configuration** section, select **Verify only the data transferred** under **Verify data** dropdown according to the requirement.
6. Click on **Save changes**.

Perform following commands via CLI :

```
aws datasync update-task --task-arn
arn:aws:datasync:[REGION]:[ACCOUNT_ID]:task/[TASK_ID] --options
VerifyMode=ONLY_FILES_TRANSFERRED --region [REGION]
```

For command usage refer: https://docs.aws.amazon.com/cli/latest/reference/datasync/update-task.html

Reference

- https://docs.aws.amazon.com/datasync/latest/userguide/create-task.html#configure-data-verification-options

Control ID - 448: Ensure that all your SSL/TLS IAM certificates are using 2048 or higher bit RSA keys

Criticality: LOW

Specification

Server certificates managed by AWS IAM, have a strong key length of 2048 or 4096 bit in order to adhere to security best practices and protect them from cryptographic algorithm hacking attacks using brute-force methods.

Rationale

Cloudview recommends upgrading your 1024-bit server certificates to 2048-bit or 4096-bit RSA certificates which are using stronger encryption algorithms. For example, a 2048-bit key is much harder to crack than a 1024-bit key.

Deleting the certificate could have implications for your application and services. One has to make configurations at respective services to ensure there is no interruption in application.

Evaluation

This control checks that there are no server certificates stored in AWS IAM with RSA key less than or equal to 1024 bits.

Remediation

IAM server certificate with RSA key length less than or equal to 1024 have to be deleted

Using AWS Console:

Removing certificates via AWS Management Console is not currently supported. To delete SSL/TLS certificates stored in IAM via the AWS API use the Command Line Interface (CLI).

Using AWS CLI:

To delete certificate with RSA key length less than or equal to 1024, run following command by replacing [CERTIFICATE_NAME] with the name of the certificate to delete:

```
aws iam delete-server-certificate --server-certificate-name [CERTIFICATE_NAME]
```

For command usage refer: https://docs.aws.amazon.com/cli/latest/reference/iam/delete-server-certificate.html

Please create new certificate with key length more than or equal to 2048.

To upload an IAM server certificate run following command

```
aws iam upload-server-certificate --server-certificate-name [CERTIFICATE_NAME]
--certificate-body file://[FILE_PATH].pem --private-key file://[FILE_PATH].pem
--certificate-chain file://[FILE_PATH].pem
```

Note: --certificate-chaining is an optional field only used when certificates are chained together.

For command usage refer: https://docs.aws.amazon.com/cli/latest/reference/iam/upload-server-certificate.html

Reference

- https://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_server-certs.html

Control ID - 449: Ensure to disable default endpoint for all the APIs

Criticality: HIGH

Specification

By disabling the default, auto-generated REST API endpoint ensures that all traffic to API only goes through the custom domain name and not the default endpoint.

Rationale

Amazon API Gateway now supports disabling the default, auto-generated REST API endpoint. This feature is intended for customers who use custom domain names for REST APIs and want to ensure that all traffic to their API only goes through the custom domain name and not the default endpoint.

Evaluation

This control ensures that default endpoint for all the APIs is disabled

Remediation

Using AWS Console:

1. Sign in to the AWS Management Console and open the Amazon API Gateway console at https://console.aws.amazon.com/apigateway/
2. In the API Gateway console, Select **APIs**.
3. Click on the name of the API to edit
4. In the left navigation panel, click **Settings**.
5. Scroll down to **Default Endpoint**section.
6. Select **Disabled** under **Default endpoint**.
7. Click on **Save Changes**

Using AWS CLI:

To enable X-Ray tracing on API Gateway Stages, use the below command:

```
aws apigateway update-rest-api --rest-api-id [REST_API_ID] --patch-operations
op=replace,path=/disableExecuteApiEndpoint,value='True'
```

Note: For more details on command, please refer to https://docs.aws.amazon.com/apigateway/latest/developerguide/rest-api-disable-default-endpoint.html

Reference

- https://docs.aws.amazon.com/apigateway/latest/developerguide/rest-api-disable-default-endpoint.html

Control ID - 450: Ensure that Microsoft AD directory forward domain controller security event logs to cloudwatch logs

Criticality: HIGH

Specification

Enabling log forwarding helps to meet security monitoring, audit, and log retention policy requirements by providing transparency of the security events in directory.

Rationale

Forwarding domain controller security event logs to Amazon CloudWatch Logs helps to meet security monitoring, audit, and log retention policy requirements by providing transparency of the security events in directory.

CloudWatch Logs can also forward these events to other AWS accounts, AWS services, or third party applications. This makes it easier to centrally monitor and configure alerts to detect and respond proactively to unusual activities in near real time.

Evaluation

This control ensures that Microsoft AD directory forward domain controller security event logs to cloudwatch logs.

Remediation

Using AWS Console:

1. Go to AWS Console Directory Service dashboard at https://console.aws.amazon.com/directoryservicev2/
2. In the left navigation panel select **Directories**, select effected **Microsoft AD**.
3. Under **Networking and security** tab scroll down to **Log forwarding** section, choose **Enable**.
4. On the **Enable log forwarding to CloudWatch** dialog, choose either of the following options:
    a. Select **Create a new CloudWatch log group,** under **CloudWatch Log group name,** specify a name that you can refer to in CloudWatch Logs.
    b. Select **Choose an existing CloudWatch log group,** and under **Existing CloudWatch log groups,** select a log group from the menu.
5. Click on **Enable**.

Using AWS CLI:

CLI to create a log group in CloudWatch Logs

```
aws logs create-log-group --log-group-name [Log_Group_Name]
```

CLI to enable log forwarding

```
aws ds create-log-subscription --directory-id [Directory_ID] --log-group-name
[Log_Group_Name]
```

Note: For more details on command, please refer to https://docs.aws.amazon.com/directoryservice/latest/admin-guide/ms_ad_enable_log_forwarding.html#enable_log_forwarding_with_cli

Reference

- https://docs.aws.amazon.com/directoryservice/latest/admin-guide/ms_ad_enable_log_forwarding.html

Control ID - 451: Ensure SQS queues uses KMS customer managed master key

Criticality: HIGH

Specification

Amazon SQS Server-side encryption transits sensitive data in encrypted queues.Enabling server-side encryption (SSE) for a queue protect the data in a queue's messages.

Rationale

Whenever SQS Queue receives a message, it encrypts the message and stores it in encrypted form.Without encryption enabled, anyone who gains access will be able to read the message.

To protect the data in a queue's messages, we can enable server-side encryption (SSE) for a queue. Amazon SQS integrates with the Amazon Web Services Key Management Service (Amazon Web Services KMS) to manage KMS keys for server-side encryption (SSE).

Evaluation

This control ensures that SQS queues uses KMS customer managed master key.

Remediation

Using AWS Console:

1. Go to AWS Console SQS dashboard at https://console.aws.amazon.com/sqs/
2. In the left navigation panel select **Queues**, select effected **Queue**.
3. Click on **Edit** button.
4. Scroll down to **Encryption** section, under **Server-side encryption** select, **Enabled** if not enabled already.
5. Select **Encryption key type** as **AWS Key Management Service key (SSE-KMS)**.
6. Under **Customer master key**, dropdown select customer managed key.
7. Ensure **Customer master key** is not selected as **alias/aws/sqs**.
8. Click on **Save**.

Using AWS CLI:

```
aws sqs set-queue-attributes --queue-url [Queue_url] --attributes
"KmsMasterKeyId"="[keyArn]"
```

Note: For more details on command, please refer to https://docs.aws.amazon.com/cli/latest/reference/sqs/set-queue-attributes.html

Reference

- https://docs.aws.amazon.com/AWSSimpleQueueService/latest/SQSDeveloperGuide/sqs-server-side-encryption.html
- https://docs.aws.amazon.com/kms/latest/developerguide/create-keys.html
- https://docs.aws.amazon.com/AWSSimpleQueueService/latest/SQSDeveloperGuide/sqs-configure-sse-existing-queue.html

Control ID - 452: Ensure SQS queues are encrypted in transit

Criticality: HIGH

Specification

Without HTTPS (TLS), a network-based attacker can eavesdrop on network traffic or manipulate it, using an attack such as man-in-the-middle. Allow only encrypted connections over HTTPS (TLS) using the aws:SecureTransport condition in the queue policy to force requests to use SSL.

Rationale

Data protection refers to protecting data while in-transit (as it travels to and from Amazon SQS) and at rest (while it is stored on disks in Amazon SQS data centers). Amazon SQS provides in-transit encryption by default. We can also protect data in transit using Secure Sockets Layer (SSL) or client-side encryption.

Evaluation

This control ensures that SQS queues are encrypted in transit.

Remediation

Using AWS Console:

1. Go to AWS Console SQS dashboard at https://console.aws.amazon.com/sqs/
2. In the left navigation panel select **Queues**, select effected **Queue**.
3. Click on **Edit** button.
4. Scroll down to **Access policy** section.
5. If there is no existing policy for SQS define one with json:

```
{
   "Id": "Policy1648531238349",
   "Version": "2012-10-17",
   "Statement": [
     {
       "Sid": "Stmt1648531234085",
       "Action": "sqs:*",
       "Effect": "Deny",
       "Resource": "[Resource_ARN]",
       "Condition": {
         "Bool": {
           "aws:SecureTransport": "false"
         }
```

```
      },
      "Principal": "*"
    }
  ]
}
```

6. If there is already a policy, in Statement section append json mentioned below:

```
{
   "Action": "sqs:*",
   "Effect": "Deny",
   "Resource": "[Resource_ARN]",
   "Condition": {
     "Bool": {
       "aws:SecureTransport": "false"
     }
   },
   "Principal": "*"
 }
```

7. Click **Save**.

Reference

- https://docs.aws.amazon.com/AWSSimpleQueueService/latest/SQSDeveloperGuide/sqs-configure-add-permissions.html

Control ID - 453: Ensure to block public access to Amazon EFS file systems

Criticality: MEDIUM

Specification

Amazon EFS file systems by default don't allow public access.Amazon EFS block public access feature provides settings to manage public access to Amazon EFS file systems.

Rationale

By default, new Amazon EFS file systems don't allow public access. However, we can modify file system policies to allow public access.When evaluating whether a file system allows public access, Amazon EFS assumes that the file system policy is public. It then evaluates the file system policy to determine if it qualifies as non-public. To be considered non-public, a file system policy must grant access only to fixed values (values that don't contain a wild card).

Evaluation

This control ensures to block public access to EFS file systems.

Remediation

Using AWS Console:

1. Open the File systems on the EFS console.
2. Choose the filesystem for which you want to remediate.
3. Scroll down and click on the **File system policy** tab.
4. Click on Edit button to open File system policy editor.
5. Add, EFS condition key **elasticfilesystem:AccessedViaMountTarget** set to **true** to the File system policy.

```
"Condition":
 {
     "Bool":
     {
         "elasticfilesystem:AccessedViaMountTarget": "true"
     }
 }
```

6. Choose **Save** button.

Using AWS CLI:

To update file system policy for the EFS use following command:

Note: You can use custom policy but the you need to set elasticfilesystem:AccessedViaMountTarget to true.

```
aws efs put-file-system-policy \

        --file-system-id [file-system-id] \

        --policy "{\"Version\": \"2012-10-17\", \"Id\": \"efs-policy-wizard-
15ad9567-2546-4bbb-8168-5541b6fc0e55\", \"Statement\": [{\"Sid\": \"efs-
statement-14a7191c-9401-40e7-a388-6af6cfb7dd9c\", \"Effect\": \"Allow\",
\"Action\": [\"elasticfilesystem:ClientMount\",
\"elasticfilesystem:ClientWrite\", \"elasticfilesystem:ClientRootAccess\" ],
\"Principal\": {\"AWS\": \"*\"}, \"Condition\": {\"Bool\":
{\"elasticfilesystem:AccessedViaMountTarget\": \"true\"}}}]}"
```

Note: For more details on command, please refer to
https://docs.aws.amazon.com/cli/latest/reference/efs/put-file-system-policy.html

Reference

- https://docs.aws.amazon.com/efs/latest/ug/access-control-block-public-access.html

Control ID - 458: Ensure connection draining is enabled for AWS ELB

Criticality: LOW

Specification

When you enable connection draining, you can specify a maximum time for the load balancer to keep connections alive before reporting the instance as de-registered. The maximum timeout value can be set between 1 and 3,600 seconds (the default is 300 seconds). When the maximum time limit is reached, the load balancer forcibly closes connections to the de-registering instance.

If your instances are part of an Auto Scaling group and connection draining is enabled for your load balancer, Auto Scaling waits for the in-flight requests to complete, or for the maximum timeout to expire, before terminating instances due to a scaling event or health check replacement.

Rationale

To ensure that a Classic Load Balancer stops sending requests to instances that are de-registering or unhealthy, while keeping the existing connections open, use connection draining. This enables the load balancer to complete in-flight requests made to instances that are de-registering or unhealthy.

Evaluation

This control ensures that connection draining is enabled for AWS ELB.

Remediation

Using AWS Console:

Note: This configuration can be performed only on Classic Load balancer.

1. Open the Amazon EC2 console
2. On the navigation pane, under **Load Balancing**, choose **Load Balancers**.
3. Select your load balancer.
4. On the **Instances** tab, for **Connection Draining**, choose **(Edit)**.
5. On the **Configure Connection Draining** page, select **Enable Connection Draining**.
6. (Optional) For Timeout, type a value between 1 and 3,600 seconds.
7. Choose **Save**.

Using AWS CLI:

Use the following modify-load-balancer-attributes command:

```
aws elb modify-load-balancer-attributes \
        --load-balancer-name [loadbalancer-name]
        --load-balancer-attributes
'{"ConnectionDraining":{"Enabled":true,"Timeout":300}}'
```

Reference

- Configure connection draining for your Classic Load Balancer

Control ID - 460: Ensure that content encoding is enabled for API Gateway Rest API

Criticality: MEDIUM

Specification

For an existing API, you must deploy the API after enabling the compression in order for the change to take effect.

The client can submit an API request with a compressed payload and an appropriate Content-Encoding header for API Gateway to decompress the method request payload and apply applicable mapping templates, before passing the request to the integration endpoint. After the compression is enabled and the API is deployed, the client can receive an API response with a compressed payload if it specifies an appropriate Accept-Encoding header in the method request.

Rationale

Content encoding for Rest API helps receive faster responses and improves performance at the expense of easily and abundantly available processing power.

Evaluation

This control ensures that content encoding is enabled for API Gateway Rest API.

Remediation

Using AWS Console:

1. Sign in to the API Gateway console.
2. Choose an existing API.
3. In the primary navigation pane, choose **Settings** under the API you chose or the one you created.
4. Under the **Content Encoding** section, select the **Content Encoding enabled** option to enable payload compression. Enter a number for the minimum compression size (in bytes) next to **Minimum body size required for compression**. To disable the compression, clear the **Content Encoding enabled** option.
5. Choose **Save Changes**.

Using AWS CLI:

To use the AWS CLI to enable compression on an existing API, call the update-rest-api command as follows:

```
aws apigateway update-rest-api \
    --rest-api-id [rest-api-id] \
    --patch-operations op=replace,path=/minimumCompressionSize,value=0
```

The minimumCompressionSize property has a non-negative integer value between 0 and 10485760 (10M bytes). It measures the compression threshold. If the payload size is smaller than this value, compression or decompression are not applied on the payload. Setting it to zero allows compression for any payload size.

Reference

- Enable payload compression for an API

Control ID - 461: Ensure to configure idle session timeout in all regions

Criticality: MEDIUM

Specification

Session Manager, a capability of AWS Systems Manager, allows you to specify the amount of time to allow a user to be inactive before the system ends a session. By default, sessions time out after 20 minutes of inactivity. You can modify this setting to specify that a session times out between 1 and 60 minutes of inactivity. Some professional computing security agencies recommend setting idle session timeouts to a maximum of 15 minutes.

Rationale

Idle session timeout feature is used for security and resource management purposes that will require user to authenticate again after specified idle time (in minutes).

Evaluation

This control ensures to configure idle session timeout in all the production accounts in the regions.

Remediation

Note - This control evaluates idle session timeout feature for all regions together i.e. failure in any one region will fail the control. If any region is not configured for idle session timeout then it is considered as region failure.

Using AWS Console:

- Open the AWS Systems Manager console
- In the navigation pane, choose **Session Manager**.
- Click the **Configure Preferences** option.
- Specify the amount of time to allow a user to be inactive before a session ends in the **minutes (max. 15 minutes)** field under **Idle session timeout**.
- Choose **Save**.

Reference

- Specify an idle session timeout value

Control ID - 462: Ensure session logs for system manager are stored in CloudWatch log groups or S3 buckets

Criticality: MEDIUM

Specification

With Amazon CloudWatch Logs, you can monitor, store, and access log files from various AWS services. You can send session log data to a CloudWatch Logs log group for debugging and troubleshooting purposes. You can choose to store session log data in a specified Amazon Simple Storage Service (Amazon S3) bucket for debugging and troubleshooting purposes.

Rationale

Logging session activity allows you to do following:

- Create and store session logs for archival purposes.
- Generate a report showing details of every connection made to your managed nodes using Session Manager over the past 30 days.
- Generate notifications of session activity in your AWS account, such as Amazon Simple Notification Service (Amazon SNS) notifications.
- Automatically initiate another action on an AWS resource as the result of session activity, such as running an AWS Lambda function, starting an AWS CodePipeline pipeline, or running an AWS Systems Manager Run Command document.

Evaluation

This control ensures that session logs for system manager are stored in CloudWatch log groups or S3 buckets.

Remediation

Using AWS Console:

1. Open the AWS Systems Manager console
2. In the navigation pane, choose **Session Manager**.
3. Click the **Configure Preferences** button.
4. Select the checkbox next to **Enable** under **CloudWatch logging** and/or **Send session logs to S3**.
5. For **CloudWatch logging**, choose **Stream session logs** option.
6. For **CloudWatch logs**, to specify the existing CloudWatch Logs log group in your AWS accounts to upload session logs to, select one of the following:

- **Choose a log group name from the list:** Select a log group that has already been created in your account to store session log data.
- **Enter the name of a log group:** Enter the name of a log group in the text box that has already been created in your account to store session log data.

- For **S3 bucket name, select one of the following:**

- **Choose a bucket name from the list:** Select an Amazon S3 bucket that has already been created in your account to store session log data.
- **Enter a bucket name in the text box:** Enter the name of an Amazon S3 bucket that has already been created in your account to store session log data.
- (Optional) For S3 key prefix, enter the name of an existing or new folder to store logs in the selected bucket.

- Choose **Save**.

Using AWS CLI:

Create a JSON file on your local machine with a name such as SessionManagerRunShell.json, and then paste the following content into it.

```
{ \
  "schemaVersion": "1.0", \
  "description": "Document to hold regional settings for Session Manager", \
  "sessionType": "Standard_Stream", \
  "inputs": { \
    "s3BucketName": [s3BucketName], \
    "s3KeyPrefix": "-", \
    "s3EncryptionEnabled": false, \
    "cloudWatchLogGroupName": [cloudWatchLogGroupName], \
```

```
    "cloudWatchEncryptionEnabled": false, \
    "cloudWatchStreamingEnabled": true, \
    "kmsKeyId": "-", \
    "runAsEnabled": true, \
    "runAsDefaultUser": [myDefaultUser], \
    "idleSessionTimeout": "20", \
    "maxSessionDuration": "60", \
    "shellProfile": { \
        "windows": "MyCommands", \
        "linux": "MyCommands" \
    } \
  } \
}
```

Save the file.

In the directory where you created the JSON file, run the following command.

```
aws ssm update-document \
                        --name "SSM-SessionManagerRunShell" --content
"file://SessionManagerRunShell.json"  \
                        --document-version "\$LATEST"
```

Reference

- Logging session data using Amazon S3 (console)
- Logging session data using Amazon CloudWatch Logs (console)
- Update Session Manager preferences (command line)

Control ID - 463: Ensure session logs for system manager are stored in only Encrypted CloudWatch log groups or S3 buckets

Criticality: HIGH

Specification

The default option is for log data to be sent with encryption using your KMS key, but you can send the data to your log group with or without encryption. Encryption is performed using the key specified for the bucket, either an AWS KMS key or an Amazon S3 Server-Side Encryption (SSE) key (AES-256).

Rationale

Storing System manager session logs in an encrypted Cloudwatch log groups or S3 buckets ensures data protection at rest.

Evaluation

This control ensures that session logs for system manager are stored in only Encrypted Cloudwatch log groups or S3 Buckets.

Remediation

Using AWS Console:

1. Open the AWS Systems Manager console
2. In the navigation pane, choose **Session Manager**.
3. Click the **Configure Preferences** button.
4. Select the checkbox next to **Enable** under **CloudWatch logging** and/or **Send session logs to S3**.
5. For **CloudWatch logging**, choose **Stream session logs** option.
6. Select the checkbox next to **Allow only encrypted CloudWatch log groups** and/or **Allow only encrypted S3 buckets**.
7. For **CloudWatch logs**, to specify the existing CloudWatch Logs log group in your AWS accounts to upload session logs to, select one of the following:

- **Choose a log group name from the list:** Select a log group that has already been created in your account to store session log data.
- **Enter the name of a log group:** Enter the name of a log group in the text box that has already been created in your account to store session log data.

- For **S3 bucket name, select one of the following:**

  - **Choose a bucket name from the list:** Select an Amazon S3 bucket that has already been created in your account to store session log data.
  - **Enter a bucket name in the text box:** Enter the name of an Amazon S3 bucket that has already been created in your account to store session log data.
  - (Optional) For S3 key prefix, enter the name of an existing or new folder to store logs in the selected bucket.

- Choose **Save**.

Using AWS CLI:

Create a JSON file on your local machine with a name such as SessionManagerRunShell.json, and then paste the following content into it.

```
{
  "schemaVersion": "1.0",
  "description": "Document to hold regional settings for Session Manager",
  "sessionType": "Standard_Stream",
  "inputs": {
    "s3BucketName": [s3BucketName],
    "s3KeyPrefix": "-",
    "s3EncryptionEnabled": true,
    "cloudWatchLogGroupName": [cloudWatchLogGroupName],
    "cloudWatchEncryptionEnabled": true,
    "cloudWatchStreamingEnabled": true,
        "runAsEnabled": true,
    "runAsDefaultUser": "myDefaultUser",
    "idleSessionTimeout": "20",
    "maxSessionDuration": "60",
    "shellProfile": {
        "windows": "windowsCommands",
        "linux": "linuxCommands"
    }
  }
}
```

Save the file.

In the directory where you created the JSON file, run the following command.

```
aws ssm update-document \
        --name "SSM-SessionManagerRunShell" \
        --content "file://SessionManagerRunShell.json"  \
        --document-version "\$LATEST"
```

Reference

- Logging session data using Amazon S3 (console)
- Logging session data using Amazon CloudWatch Logs (console)
- Update Session Manager preferences (command line)

Control ID - 464: Ensure Block public sharing setting is ON for the documents in all regions

Criticality: HIGH

Specification

You can share AWS Systems Manager (SSM) documents privately or publicly with accounts in the same AWS Region. To privately share a document, you modify the document permissions and allow specific individuals to access it according to their AWS account ID. To publicly share an SSM document, you modify the document permissions and specify All. It is recommended that you block the public sharing for all documents to prevent unwanted access. The block public sharing setting is an account level setting that can differ for each AWS Region.

Rationale

Turning on Block public sharing setting prevents unwanted access to your SSM documents.

Evaluation

This control ensures to enable "Block public sharing setting" is ON for the documents in all the production accounts in the region.

Remediation

Note - This control evaluates public access setting for all regions together i.e. failure in any one region will fail the control. If any region is not configured for public access setting then it is considered as region failure.

Using AWS Console:

1. Open the AWS console.
2. Navigate to Amazon System manager
3. On the left side menu, select **Documents** under Shared resources
4. Select All documents tab and click on the Preferences button.
5. Click on the Edit button.
6. Select the checkbox for block public sharing.
7. Select Save button to save the changes.

Using AWS CLI:

To edit block public access settings for an account

```
aws ssm update-service-setting \
                        --setting-id /ssm/documents/console/public-sharing-
permission \
                        --setting-value Disable \
                        --region [region]
```

Reference

- Configuring block public access settings for your account

Control ID - 465: Ensure stage caching is enabled for AWS API Gateway Method Settings

Criticality: MEDIUM

Specification

You can override stage-level cache settings by enabling or disabling caching for a specific method.

Rationale

You can enable API caching in Amazon API Gateway to cache your endpoint's responses. With caching, you can reduce the number of calls made to your endpoint and also improve the latency of requests to your API.

Evaluation

This control ensures that stage caching is enabled for AWS API Gateway Method Settings.

Remediation

Using AWS Console:

1. Sign in to the API Gateway console
2. Go to the API Gateway console.
3. Select the API Gateway to be updated.
4. From left side, choose **Stages**.
5. In the **Stages** list for the API, choose the stage.
6. Choose the Settings tab.
7. Under "Cache Settings" page, Check "Enable API cache".
8. Click save changes.

Using AWS CLI:

To override the stage settings and disable caching for a specific resource and method in an API's stage

```
aws apigateway update-stage \
                --rest-api-id [rest-api-id] \
                --stage-name [stage-name] \
```

```
                --patch-operations
op=replace,path=/{resource_path}/{http_method}/caching/enabled,value=true
```

Reference

- Enabling API caching to enhance responsiveness
- update-stage

Control ID - 466: Ensure transit encryption is enabled for EFS volumes in AWS ECS Task Definition

Criticality: HIGH

Specification

Enabling encryption of data in transit for your Amazon EFS file system is done by enabling Transport Layer Security (TLS) when you mount your file system using the Amazon EFS mount helper.Transit encryption must be enabled if Amazon EFS IAM authorization is used. If this parameter is omitted, the default value of DISABLED is used.

Rationale

Amazon EFS file systems are encrypted in order to meet security and compliance requirements of the customer. Data in transit can be securely transmitted to EFS to reduce the security attack. Your data is transparently encrypted in transit while being written and transparently decrypted while being read from your file system.

Evaluation

This control ensures that transit encryption is enabled for EFS volumes in AWS ECS Task Definition.

Remediation

Note - Resource Can be configured only during resource creation.

Using AWS Management Console:

1. Login to AWS Management Console.
2. Navigate to Elastic Container Service
3. From left side select Task Definitions tab.
4. Click "Create new Task Definition".
5. Select any launch type compatibility and click "Next step".
6. Fill all the required fields.
7. Scroll down to "Container definitions" section & click on "Add container" button.
8. Enter the Container name.
9. For "Image" field use the image tag as "public.ecr.aws/ecs/amazon-ecs-agent:latest".
10. Fill "Memory limits".
11. Click "Add" button.
12. Scroll down to "Volumes" section & click on "Add volume".
13. Fill Name, from "Volume type" field select "EFS","File system ID".
14. Use the "Encryption in transit" checkbox for "Enable transit encryption".
15. Click Create button.

Reference

- Encrypting data in transit
- Amazon EFS volumes

Control ID - 467: Ensure to disable root access for all notebook instance users

Criticality: HIGH

Specification

To update root access attribute of a notebook instance, its status must be stopped.

Rationale

Removing root access prevents notebook users from deleting system-level software, installing new software, and modifying essential environment components.

Evaluation

This control ensures that root access for all notebook instance users is disabled in the regions.

Remediation

Using AWS Console:

1. Open the SageMaker console.
2. Navigate to Amazon SageMaker.
3. Choose **Notebook instances**.
4. Choose the notebook instance that you want to update by selecting the notebook instance **Name** from the list.
5. If your notebook **Status** is not Stopped, select the **Stop** button to stop the notebook instance.
6. When you do this, the notebook instance status changes to Stopping. Wait until the status changes to Stopped to complete the following steps.
7. Select **Update settings** from **Actions** dropdown.
8. Under **Permissions and encryption**, select **Disable** under **Root access** and then click **Update notebook instance** button at the bottom of the page when you are done to return to the notebook instances page. Your notebook instance status changes to **Updating**.
9. When the notebook instance update is complete, the status changes to Stopped.

Using AWS CLI:

```
aws sagemaker update-notebook-instance \
        --notebook-instance-name [notebook-instance-name] \
        --root-access Disabled
```

Reference

- Update a Notebook Instance
- update-notebook-instance

Control ID - 468: Ensure to enable inter-container traffic encryption for Processing jobs(if configured)

Criticality: MEDIUM

Specification

To analyze data and evaluate machine learning models on Amazon SageMaker, use Amazon SageMaker Processing. With Processing, you can use a simplified, managed experience on SageMaker to run your data processing workloads, such as feature engineering, data validation, model evaluation, and model interpretation. You can also use the Amazon SageMaker Processing APIs during the experimentation phase and after the code is deployed in production to evaluate performance.

Rationale

Enabling inter-container traffic encryption can increase training time, especially if you are using distributed deep learning algorithms. Enabling inter-container traffic encryption doesn't affect training jobs with a single compute instance. However, for training jobs with several compute instances, the effect on training time depends on the amount of communication between compute instances. For affected algorithms, adding this additional level of security also increases cost. The training time for most SageMaker built-in algorithms, such as XGBoost, DeepAR, and linear learner, typically aren't affected.

Evaluation

This control ensures to enable inter-container traffic encryption for Processing jobs(if configured) in all the production accounts in the regions

Remediation

Using AWS Console

You need to Create AWS SageMaker Processing job to enable inter-container traffic encryption

1. Open AWS SageMaker console at https://us-east-1.console.aws.amazon.com/sagemaker/home.
2. In the left navigation panel, under **Processing**, choose **Processing jobs**.
3. Click **Create Processing job** button from the dashboard top menu. Choose **Next**.
4. In the **Network** section, choose **Enable inter-container traffic encryption**.
5. When you're done, choose **Submit** to create the **SageMaker Processing Job**.

Using AWS CLI

To create AWS SageMaker Processing Job to enable inter-container traffic encryption, use the following command:

```
aws sagemaker create-processing-job \
    --processing-job-name [job-name] \
    --role-arn [role-arn] \
    --app-specification '{"ImageUri":"[image-uri]","ContainerEntrypoint":["-
"],"ContainerArguments":["-"]}' \
    --processing-resources
'{"ClusterConfig":{"InstanceCount":1,"InstanceType":"[instance-
type]","VolumeSizeInGB":1,"VolumeKmsKeyId":"[kms-key]"}}' \
```

```
    --network-config '{ "EnableInterContainerTrafficEncryption": true,
"EnableNetworkIsolation": true, "VpcConfig": {"SecurityGroupIds": [security-
group-id], "Subnets": [subnet-name]}}'
```

Reference

- AWS SageMaker processing-Job reference
- AWS SageMaker processing-Job inter-container traffic encryption reference

Control ID - 469: Ensure processing jobs(if configured) are running inside a VPC

Criticality: MEDIUM

Specification

To control access to your data and processing jobs, we recommend that you create a private Amazon VPC and configure it so that your jobs aren't accessible over the public internet.
For information about creating and configuring a VPC, see Getting Started With Amazon VPC in the Amazon VPC User Guide.
Using a VPC helps to protect your processing containers and data because you can configure your VPC so that it is not connected to the internet.
Using a VPC also allows you to monitor all network traffic in and out of your processing containers by using VPC flow logs.

Rationale

A VPC endpoint enables connections between a virtual private cloud (VPC) and supported services, without requiring that you use an internet gateway, NAT device, VPN connection, or AWS Direct Connect connection. Therefore, you control the specific API endpoints, sites, and services that are reachable from your VPC.
Amazon Virtual Private Cloud (Amazon VPC) enables you to launch AWS resources into a virtual network that you've defined.
This virtual network closely resembles a traditional network that you'd operate in your own data center, with the benefits of using the scalable infrastructure of AWS.

Evaluation

This control ensures processing jobs are running inside a VPC in all the production accounts in the regions

Remediation

Using AWS Console

To Create AWS SageMaker Processing job running inside a VPC

1. Open AWS SageMaker console at https://us-east-1.console.aws.amazon.com/sagemaker/home.
2. In the left navigation panel, under **Processing**, choose **Processing jobs**.
3. Click **Create Processing job** button from the dashboard top menu. Choose **Next**.
4. In the **Network** section, choose a **VPC**.
5. When you're done, choose **Submit** to create the **SageMaker Processing Job**.

Using AWS CLI

To create AWS SageMaker Processing Job running inside a VPC, use the following command:

```
aws sagemaker create-processing-job \
        --processing-job-name [job-name] \
        --role-arn [role-arn] \
        --app-specification '{"ImageUri":"[image-
uri]","ContainerEntrypoint":["-"],"ContainerArguments":["-"]}' \
        --processing-resources
'{"ClusterConfig":{"InstanceCount":1,"InstanceType":"[instance-
type]","VolumeSizeInGB":1,"VolumeKmsKeyId":"[kms-key]"}}' \
        --network-config '{ "EnableInterContainerTrafficEncryption": true,
"EnableNetworkIsolation": true, "VpcConfig": {"SecurityGroupIds": [security-
group], "Subnets": [subnet-name]}}'
```

Reference

- AWS SageMaker processing-Job reference
- Configure a Processing Job for Amazon VPC Access

Control ID - 470: Ensure to enable network isolation for processing jobs(if configured)

Criticality: HIGH

Specification

SageMaker training and deployed inference containers are internet-enabled by default. This allows containers to access external services and resources on the public internet as part of your training and inference workloads. However, this could provide an avenue for unauthorized access to your data.

Rationale

If you enable network isolation, the containers can't make any outbound network calls, even to other AWS services such as Amazon S3. Additionally, no AWS credentials are made available to the container runtime environment. In the case of a training job with multiple instances, network inbound and outbound traffic is limited to the peers of each training container. SageMaker still performs download and upload operations against Amazon S3 using your SageMaker execution role in isolation from the training or inference container.

Evaluation

This control ensures to enable network isolation for processing jobs

Remediation

Using AWS Console

To Create AWS SageMaker Processing job to enable network isolation

1. Open AWS SageMaker console at https://us-east-1.console.aws.amazon.com/sagemaker/home.
2. In the left navigation panel, under **Processing**, choose **Processing jobs**.
3. Click **Create Processing job** button from the dashboard top menu. Choose **Next**.
4. In the **Network** section, choose **Enable network isolation**.

5. When you're done, choose **Submit** to create the **SageMaker Processing Job**.

Using AWS CLI

To create AWS SageMaker Processing job to enable network isolation, use the following command:

```
aws sagemaker create-processing-job \
        --processing-job-name [job-name] \
        --role-arn [role-arn] \
        --app-specification '{"ImageUri":"[image-
uri]","ContainerEntrypoint":["-"],"ContainerArguments":["-"]}' \
        --processing-resources
'{"ClusterConfig":{"InstanceCount":1,"InstanceType":"[instance-
type]","VolumeSizeInGB":1,"VolumeKmsKeyId":"[kms-key]"}}' \
        --network-config '{ "EnableInterContainerTrafficEncryption": true,
"EnableNetworkIsolation": true, "VpcConfig": {"SecurityGroupIds": [security-
group], "Subnets": [subnet-name]}}'
```

Reference

- AWS SageMaker processing-Job reference
- AWS Sagemaker Network Isolation reference

Control ID - 471: Ensure ML storage volume attached to training jobs are encrypted

Criticality: HIGH

Specification

To encrypt the machine learning (ML) storage volume that is attached to notebooks, processing jobs, training jobs, hyperparameter tuning jobs, batch transform jobs, and endpoints, you can pass a AWS KMS key to SageMaker. If you don't specify a KMS key, SageMaker encrypts storage volumes with a transient key and discards it immediately after encrypting the storage volume. For training jobs, if you don't specify a KMS key, SageMaker encrypts both OS volumes and ML data volumes with a system-managed KMS key.

Rationale

Sensitive data that needs to be encrypted with a KMS key for compliance reasons should be stored in the ML storage volume or in Amazon S3, both of which can be encrypted using a KMS key you specify for security reasons.

Evaluation

This control ensures ML storage volume attached to training jobs are encrypted in all the production accounts in the regions

Remediation

Using AWS Console

You need to create AWS sagemaker training job to encrypt attached ML storage volume.

1. Open AWS SageMaker console at https://us-east-1.console.aws.amazon.com/sagemaker/home?region=us-east-1#/jobs.
2. In the left navigation panel, under **Training**, choose **Training jobs**.
3. Click **Create Training job** button from the dashboard top menu.
4. In the **Resource configuration** section, choose an **Encryption key**.
5. When you're done, choose **Create training job** to create the **SageMaker Training Job**.

Reference

- AWS SageMaker training-Job reference
- AWS SageMaker training job ML storage volume encryption reference

Control ID - 472: Ensure ML storage volume attached to training jobs are encrypted with customer managed master key

Criticality: HIGH

Specification

To encrypt the machine learning (ML) storage volume that is attached to notebooks, processing jobs, training jobs, hyperparameter tuning jobs, batch transform jobs, and endpoints, you can pass a AWS KMS key to SageMaker.
If you don't specify a KMS key, SageMaker encrypts storage volumes with a transient key and discards it immediately after encrypting the storage volume.
For training jobs, if you don't specify a KMS key, SageMaker encrypts both OS volumes and ML data volumes with a system-managed KMS key.

Rationale

Sensitive data that needs to be encrypted with a KMS key for compliance reasons should be stored in the ML storage volume or in Amazon S3, both of which can be encrypted using a KMS key you specify for security reasons.

Evaluation

This control ensures ML storage volume attached to training jobs are encrypted with customer managed master key.

Remediation

Using AWS Console

You need to create AWS sagemaker training job to encrypt attached ML storage volume.

1. Open AWS SageMaker console at https://us-east-1.console.aws.amazon.com/sagemaker/home?region=us-east-1#/jobs.
2. In the left navigation panel, Under **Training**, choose **Training jobs**.
3. Click **Create Training job** button from the dashboard top menu.
4. In the **Resource configuration** section, use a **Customer managed Key** for **Encryption key**.
5. When you're done, choose **Create training job** to create the **SageMaker Training Job**.

Reference

- AWS SageMaker training-Job reference
- AWS SageMaker training job ML storage volume encryption with customer managed master key reference

Control ID - 473: Ensure to encrypt the output of the training jobs in s3 with customer managed master key

Criticality: HIGH

Specification

Server-side encryption is the encryption of data at its destination by the application or service that receives it. AWS Key Management Service (AWS KMS) is a service that combines secure, highly available hardware and software to provide a key management system scaled for the cloud. Amazon S3 uses AWS KMS keys to encrypt your Amazon S3 objects. AWS KMS encrypts only the object data. Any object metadata is not encrypted.

Rationale

When you use server-side encryption with AWS KMS (SSE-KMS), you can use the default AWS managed key, or you can specify a customer managed key that you have already created. AWS KMS uses envelope encryption to further protect your data. Envelope encryption is the practice of encrypting your plaintext data with a data key, and then encrypting that data key with a root key. If you don't specify a customer managed key, Amazon S3 automatically creates an AWS KMS key in your AWS account the first time that you add an object encrypted with SSE-KMS to a bucket. By default, Amazon S3 uses this KMS key for SSE-KMS.

Evaluation

This control ensures to encrypt the output of the training jobs in s3 with customer managed master key in all regions

Remediation

Using AWS Console

1. Open AWS SageMaker console at https://us-east-1.console.aws.amazon.com/sagemaker/home?region=us-east-1#/jobs.
2. In the left navigation panel, under **Training**, choose **Training jobs**.
3. Click **Create Training job** button from the dashboard top menu.
4. In the **Output data configuration** section, give S3 output path under **S3 output path** and use **Customer managed master key** in the **Encryption key**.
5. When you're done, choose **Create training job** to create the **SageMaker Training Job**.

Reference

- AWS SageMaker training-Job reference
- AWS SageMaker training job encrypt the output of the training jobs in s3 with customer managed master key

Control ID - 474: Ensure to enable inter-container traffic encryption for training jobs

Criticality: HIGH

Specification

By default, Amazon SageMaker runs training jobs in an Amazon Virtual Private Cloud (Amazon VPC) to help keep your data secure. You can add another level of security to protect your training containers and data by configuring a private VPC. Distributed ML frameworks and algorithms usually transmit information that is directly related to the model such as weights, not the training dataset. When performing distributed training, you can further protect data that is transmitted between instances. This can help you to comply with regulatory requirements. To do this, use inter-container traffic encryption.

Rationale

Enabling inter-container traffic encryption can increase training time, especially if you are using distributed deep learning algorithms. Enabling inter-container traffic encryption doesn't affect training jobs with a single compute instance. However, for training jobs with several compute instances, the effect on training time depends on the amount of communication between compute instances. The training time for most SageMaker built-in algorithms, such as XGBoost, DeepAR, and linear learner, typically aren't affected.

Evaluation

This control ensures to enable inter-container traffic encryption for training jobs in all the production accounts in the regions

Remediation

Using AWS Console

1. Open AWS SageMaker console at https://us-east-1.console.aws.amazon.com/sagemaker/home?region=us-east-1#/jobs.
2. In the left navigation panel, under **Training**, choose **Training jobs**.
3. Click **Create Training job** button from the dashboard top menu.
4. In the **Network** section, choose a **VPC** and then select **Enable inter-container traffic encryption**.
5. When you're done, choose **Create training job** to create the **SageMaker Training Job**.

Reference

- AWS SageMaker training-Job reference
- AWS SageMaker training-Job inter-container traffic encryption reference

Control ID - 475: Ensure to enable network isolation for training jobs

Criticality: HIGH

Specification

SageMaker training and deployed inference containers are internet-enabled by default. This allows containers to access external services and resources on the public internet as part of your training and inference workloads. However, this could provide an avenue for unauthorized access to your data.

Rationale

If you enable network isolation, the containers can't make any outbound network calls, even to other AWS services such as Amazon S3. Additionally, no AWS credentials are made available to the container runtime environment. In the case of a training job with multiple instances, network inbound and outbound traffic is limited to the peers of each training container. SageMaker still performs download and upload operations against Amazon S3 using your SageMaker execution role in isolation from the training or inference container.

Evaluation

This control ensures to enable network isolation for training jobs in the regions.

Remediation

Using AWS Console

Note: Resource can be configured during Creation Only.

1. Open AWS SageMaker console at https://console.aws.amazon.com.
2. Navigate to Amazon SageMaker.
3. In the left navigation panel, under **Training**, choose **Training jobs**.
4. Click **Create Training job** button from the dashboard top menu.
5. In the **Network** section, choose **Enable network isolation**.
6. When you're done, choose **Create training job** to create the **SageMaker Training Job**.

Reference

- AWS SageMaker training-Job reference
- AWS SageMaker training-Job network isolation reference

Control ID - 476: Ensure ML storage volume attached to Hyperparameter Tuning jobs are encrypted

Criticality: HIGH

Specification

You can encrypt all ML data volumes for all SageMaker instances with an AWS KMS key that you specify.

Rationale

Sensitive data needs to be protected at rest and in-transit. By configuring ML storage volume encryption, data protection at rest is ensured.

Evaluation

This control ensures that ML storage volumes attached to hyperparameter tuning jobs are encrypted in the regions.

Remediation

Note: Resource can be configured only at the time of Creation.

Using AWS Console:

1. Open the Amazon Sagemaker console
2. Navigate to SageMaker.
3. In the navigation pane, choose **Training**, then choose **Hyperparameter tuning jobs**.
4. Choose **Create hyperparameter tuning job**.
5. In **Step 2 - Create training job definition**, go to **Configure resources**.
6. Under **Resource configurations**, type or select AWS KMS Key ID/Key ARN from the dropdown below **Encryption key**.
7. After configuring encryption, click on create hyperparameter tuning job.

Reference

- Notebook instances and SageMaker jobs
- create-hyper-parameter-tuning-job

Control ID - 477: Ensure ML storage volume attached to Hyperparameter Tuning jobs (if configured) are encrypted with customer managed master key

Criticality: HIGH

Specification

You can encrypt all ML data volumes for all SageMaker instances with a customer managed key that you specify.

Rationale

Sensitive data needs to be protected at rest and in-transit. By configuring ML storage volume encryption with customer managed key, data protection at rest is ensured.

Evaluation

This control ensures that ML storage volume attached to hyperparameter tuning jobs (if configured) are encrypted with customer managed master key.

Remediation

Note: Resource can be configured only at the time of Creation.

Using AWS Console:

1. Open the Amazon Sagemaker console
2. Navigate to SageMaker.
3. In the navigation pane, choose **Training**, then choose **Hyperparameter tuning jobs**.

4. Choose **Create hyperparameter tuning job**.
5. In **Step 2 - Create training job definition**, go to **Configure resources**.
6. Under **Resource configurations**, type or select an existing Customer Managed Key ID/Key ARN from the dropdown below **Encryption key**.
7. After configuring encryption, click on create hyperparameter tuning job.

Reference

- Notebook instances and SageMaker jobs
- create-hyper-parameter-tuning-job

Control ID - 478: Ensure to encrypt the output of Hyperparameter tuning jobs in s3

Criticality: HIGH

Specification

You can encrypt S3 output data of hyperparameter tuning jobs with an AWS KMS key that you specify.

Rationale

Sensitive data needs to be protected at rest and in-transit. By configuring S3 output data of Hyperparameter tuning jobs, data protection at rest is ensured.

Evaluation

This control ensures that S3 output data of hyperparameter tuning jobs is encrypted in the regions.

Remediation

Note: Resource can be configured only at the time of Creation.

Using AWS Console:

1. Open the Amazon Sagemaker console
2. Navigate to SageMaker.
3. In the navigation pane, choose **Training**, then choose **Hyperparameter tuning jobs**.
4. Choose **Create hyperparameter tuning job**.
5. In **Step 2 - Create training job definition**, go to **Define data input and output**.
6. Under **Output data configuration**, type or select AWS KMS Key ID/Key ARN from the dropdown below **Encryption key** for S3 output path.
7. After configuring encryption, click on create hyperparameter tuning job.

Reference

- Notebook instances and SageMaker jobs
- create-hyper-parameter-tuning-job

Control ID - 479: Ensure to encrypt the output of Hyperparameter tuning jobs(if configured) in s3 with customer managed master key

Criticality: HIGH

Specification

You can encrypt S3 output data of hyperparameter tuning jobs with a Customer Managed Key that you specify.

Rationale

Sensitive data needs to be protected at rest and in-transit. By configuring S3 output data of Hyperparameter tuning jobs encryption with Customer Managed Key, data protection at rest is ensured.

Evaluation

This control ensures that S3 output data of hyperparameter tuning jobs (if configured) is encrypted with customer managed master key in the regions.

Remediation

Note: Resource can be configured only at the time of Creation.

Using AWS Console:

1. Open the Amazon Sagemaker console
2. Navigate to SageMaker.
3. In the navigation pane, choose **Training**, then choose **Hyperparameter tuning jobs**.
4. Choose **Create hyperparameter tuning job**.
5. In **Step 2 - Create training job definition**, go to **Define data input and output**.
6. Under **Output data configuration**, type or select an existing Customer Managed Key ID/Key ARN from the dropdown below **Encryption key**.
7. After configuring encryption, click on create hyperparameter tuning job.

Reference

- Notebook instances and SageMaker jobs
- create-hyper-parameter-tuning-job

Control ID - 480: Ensure to enable inter-container traffic encryption for Hyperparameter tuning jobs(if configured)

Criticality: HIGH

Specification

By default, Amazon SageMaker runs training jobs in an Amazon Virtual Private Cloud (Amazon VPC) to help keep your data secure. You can add another level of security to protect your training containers and data by configuring a private VPC. Distributed ML frameworks and algorithms usually transmit information that is

directly related to the model such as weights, not the training dataset. When performing distributed training, you can further protect data that is transmitted between instances. This can help you to comply with regulatory requirements. To do this, use inter-container traffic encryption.

Rationale

Enabling inter-container traffic encryption can increase training time, especially if you are using distributed deep learning algorithms. Enabling inter-container traffic encryption doesn't affect training jobs with a single compute instance. However, for training jobs with several compute instances, the effect on training time depends on the amount of communication between compute instances. For affected algorithms, adding this additional level of security also increases cost. The training time for most SageMaker built-in algorithms, such as XGBoost, DeepAR, and linear learner, typically aren't affected.

Evaluation

This control ensures to enable inter-container traffic encryption for Hyperparameter tuning jobs(if configured) in all the regions

Remediation

Using AWS Console

1. Open AWS SageMaker console at https://us-east-1.console.aws.amazon.com/sagemaker/home?region=us-east-1#/hyper-tuning-jobs.
2. In the left navigation panel, under **Training**, choose **Hyperparameter tuning jobs**.
3. Click **Create Hyperparameter tuning jobs** button from the dashboard top menu.
4. In the **Create training job definition** section, choose a **VPC** and then select **Enable inter-container traffic encryption**.
5. When you're done, choose **Create hyperparameter tuning job** to create the **SageMaker hyperparameter tuning job**.

Reference

- AWS SageMaker hyperparameter tuning reference
- AWS SageMaker hyperparameter tuning Job inter-container traffic encryption reference

Control ID - 481: Ensure Hyperparameter tuning jobs(if configured) are running inside a VPC

Criticality: MEDIUM

Specification

Amazon Virtual Private Cloud (VPC) is a service that lets you launch AWS resources in a logically isolated virtual network that you define.
You have complete control over your virtual networking environment, including selection of your own IP address range, creation of subnets, and configuration of route tables and network gateways.
You can use both IPv4 and IPv6 for most resources in your VPC, helping to ensure secure and easy access to resources and applications. As one of AWS's foundational services, Amazon VPC makes it easy to customize your VPC's network configuration.
You can create a public-facing subnet for your web servers that have access to the internet.
It also lets you place your backend systems, such as databases or application servers, in a private-facing

subnet with no internet access.
Amazon VPC lets you to use multiple layers of security, including security groups and network access control lists, to help control access to Amazon Elastic Compute Cloud (Amazon EC2) instances in each subnet.

Rationale

When you use Amazon Virtual Private Cloud (Amazon VPC) enables you to launch AWS resources into a virtual network that you've defined.
This virtual network closely resembles a traditional network that you'd operate in your own data center, with the benefits of using the scalable infrastructure of AWS.

Evaluation

This control ensures that Hyperparameter tuning jobs(if configured) are running inside a VPC in all the regions

Remediation

Using AWS Console

To Create AWS SageMaker Hyperparameter tuning job inside a VPC

1. Open AWS SageMaker console at https://us-east-1.console.aws.amazon.com/sagemaker/home?region=us-east-1#/hyper-tuning-jobs.
2. In the left navigation panel, under **Training**, choose **Hyperparameter tuning jobs**.
3. Click **Create Hyperparameter tuning jobs** button from the dashboard top menu.
4. In the **Create training job definition** section, choose a **VPC**.
5. When you're done, choose **Create hyperparameter tuning job** to create the **SageMaker hyperparameter tuning job**.

Reference

- AWS SageMaker hyperparameter tuning reference
- Access to resource in a VPC

Control ID - 482: Ensure to enable network isolation for Hyperparameter tuning jobs(if configured)

Criticality: HIGH

Specification

SageMaker training and deployed inference containers are internet-enabled by default. This allows containers to access external services and resources on the public internet as part of your training and inference workloads. However, this could provide an avenue for unauthorized access to your data.

Rationale

If you enable network isolation, the containers can't make any outbound network calls, even to other AWS services such as Amazon S3. Additionally, no AWS credentials are made available to the container runtime environment. In the case of a training job with multiple instances, network inbound and outbound traffic is

limited to the peers of each training container. SageMaker still performs download and upload operations against Amazon S3 using your SageMaker execution role in isolation from the training or inference container.

Evaluation

This control ensures to enable network isolation for Hyperparameter tuning jobs(if configured) in all the production accounts in the regions

Remediation

Using AWS Console

1. Open AWS SageMaker console at https://us-east-1.console.aws.amazon.com/sagemaker/home?region=us-east-1#/hyper-tuning-jobs.
2. In the left navigation panel, under **Training**, choose **Hyperparameter tuning jobs**.
3. Click **Create Hyperparameter tuning jobs** button from the dashboard top menu.
4. In the **Create training job definition** section, choose **Enable network isolation**.
5. When you're done, choose **Create hyperparameter tuning job** to create the **SageMaker hyperparameter tuning job**.

Reference

- AWS SageMaker hyperparameter tuning reference
- AWS SageMaker hyperparameter tuning Job network isolation reference

Control ID - 483: Ensure to enable network isolation for models

Criticality: HIGH

Specification

SageMaker training and deployed inference containers are internet-enabled by default. This allows containers to access external services and resources on the public internet as part of your training and inference workloads. However, this could provide an avenue for unauthorized access to your data.

Rationale

If you enable network isolation, the containers can't make any outbound network calls, even to other AWS services such as Amazon S3. Additionally, no AWS credentials are made available to the container runtime environment. This would restrict unauthorized access to your data. SageMaker still performs download and upload operations against Amazon S3 using your SageMaker execution role in isolation from the training or inference container.

Evaluation

This control ensures to enable network isolation for models in the regions.

Remediation

Using AWS Console

Note :- Resource can be configured at the time of creation.

1. Open AWS SageMaker console at https://us-east-1.console.aws.amazon.com/sagemaker/home?region=us-east-1#/jobs.
2. In the left navigation panel, under **Inference** select **Models**.
3. Click **Create Model** button from the dashboard top menu.
4. In the **Network** section, choose **Enable network isolation**.
5. When you're done, choose **Create Model** to create the **SageMaker Model**.

Using AWS CLI

To create AWS SageMaker Model and enable network isolation, use the following command:

```
aws sagemaker create-model \
        --model-name [model-name] \
        --execution-role-arn [IAM-role-arn] \
        --primary-container ContainerHostname=[container-host],Image=[image] \
        --enable-network-isolation
```

Reference

- Secure deployment of Amazon SageMaker resources
- Run Training and Inference Containers in Internet-Free Mode

Control ID - 485: Ensure to enable CloudWatch logging in the audit logging account

Criticality: MEDIUM

Specification

CloudWatch Logs enables you to centralize the logs from all of your systems, applications, and AWS services that you use, in a single, highly scalable service. You can then easily view them, search them for specific error codes or patterns, filter them based on specific fields, or archive them securely for future analysis. CloudWatch Logs enables you to see all of your logs, regardless of their source, as a single and consistent flow of events ordered by time, and you can query them and sort them based on other dimensions, group them by specific fields, create custom computations with a powerful query language, and visualize log data in dashboards.

Rationale

Monitoring is an important part of maintaining the reliability, availability, and performance of Amazon Kinesis Data Analytics and your Kinesis Data Analytics applications. You should collect monitoring data from all of the parts of your AWS solution so that you can more easily debug a multipoint failure if one occurs.

Evaluation

This control ensures that CloudWatch logging in the audit logging account in the regions is enabled.

Remediation

Using AWS Console:

1. Login to AWS Console.
2. Navigate to Amazon Kinesis Dashboard.
3. In the left navigation panel, under Kinesis Dashboard, click on **Delivery streams**.
4. Select the Delivery streams that you want to Modify.
5. In selected Delivery streams go to **configuration** tab.
6. Under **Destination error logs** Click on edit to modify Amazon CloudWatch error logging.
7. Select **Enabled** to enable CloudWatch logging.
8. Click on **Save changes** to save the changes.

Reference

- Logging and Monitoring in Amazon Kinesis Data Analytics

Control ID - 489: Ensure multi-az is enabled for AWS DMS instances

Criticality: MEDIUM

Specification

You can change this option to create a standby replica of your replication instance in another Availability Zone for failover support or remove this option. If you intend to use change data capture (CDC), ongoing replication, you should enable this option.

Rationale

In a Multi-AZ deployment, AWS DMS automatically provisions and maintains a synchronous standby replica of the replication instance in a different Availability Zone. The primary replication instance is synchronously replicated across Availability Zones to a standby replica. This approach provides data redundancy, eliminates I/O freezes, and minimizes latency spikes.

Evaluation

This control ensures that multi-az is enabled for AWS DMS instances.

Remediation

Using AWS Console:

1. Sign in to the AWS Management Console and open the AWS DMS console.
2. In the navigation pane, choose **Replication instances**.
3. Choose the replication instance you want to modify. Select **Modify** from **Actions** dropdown.
4. Under **Replication instance configuration** section, select **Production workload (Multi-AZ)** for **Multi AZ** setting.
5. Click **Save**.

Using AWS CLI:

```
aws dms modify-replication-instance \
        --replication-instance-arn [replication-instance-arn] \
        --multi-az
```

Note - When you modify a replication instance, you can apply the changes immediately. To apply changes immediately, choose the **Apply changes immediately** option in the AWS Management Console. Or use the **--apply-immediately** parameter when calling the AWS CLI

If you don't choose to apply changes immediately, the changes are put into the pending modifications queue. During the next maintenance window, any pending changes in the queue are applied.

Reference

- Modifying a replication instance
- modify-replication-instance

Control ID - 490: Ensure auto minor version upgrade is enabled for AWS DMS instances

Criticality: MEDIUM

Specification

Auto minor version upgrade indicates that minor version upgrades are applied automatically to the replication instance during the maintenance window. Changing this parameter doesn't result in an outage, except in the case described following. The change is asynchronously applied as soon as possible.

An outage does result if these factors apply:

- This parameter is set to true during the maintenance window.
- A newer minor version is available.
- AWS DMS has enabled automatic patching for the given engine version.

Rationale

Auto Minor Version Upgrade is a feature that you can enable to have your database automatically upgraded when a new minor database engine version is available.

Evaluation

This control ensures that auto minor version upgrade is enabled for AWS DMS instances.

Remediation

Using AWS Console:

1. Sign in to the AWS Management Console and open the AWS DMS console.
2. In the navigation pane, choose **Replication instances**.
3. Choose the replication instance you want to modify. Select **Modify** from **Actions** dropdown.
4. Under **Maintenance** section, select **Yes** for **Minor version automatic upgrade** setting.
5. Click **Save**.

Using AWS CLI:

```
aws dms modify-replication-instance \
```

```
        --replication-instance-arn [replication-instance-arn] \
        --auto-minor-version-upgrade
```

Note - When you modify a replication instance, you can apply the changes immediately. To apply changes immediately, choose the **Apply changes immediately** option in the AWS Management Console. Or use the **--apply-immediately** parameter when calling the AWS CLI

If you don't choose to apply changes immediately, the changes are put into the pending modifications queue. During the next maintenance window, any pending changes in the queue are applied.

Reference

- Modifying a replication instance
- modify-replication-instance

Control ID - 491: Ensure auto minor version upgrade is enabled for AWS MQ Brokers

Criticality: MEDIUM

Specification

Automatic minor version upgrades occur only if the broker is running a minor engine version that is lower than the new recommended minor version.

Rationale

When you activate the automatic minor version upgrade option, Amazon MQ upgrades your broker to new minor versions as they become available.

Evaluation

This control ensures that auto minor version upgrade is enabled for AWS MQ brokers.

Remediation

Using AWS Console:

1. Sign in to the Amazon MQ console
2. In the left navigation pane, choose **Brokers**, and then choose the broker that you want to upgrade from the list.
3. On the broker details page, choose **Edit**.
4. Under **Maintenance**, choose **Enable automatic minor version upgrades**.
5. Choose **Save** at the bottom of the page.

Using AWS CLI:

```
aws mq update-broker \
        --broker-id [broker-id] \
        --auto-minor-version-upgrade
```

Reference

- Upgrading an Amazon MQ broker engine version

Control ID - 492: Ensure active/standby deployment mode is used for AWS MQ Brokers

Criticality: MEDIUM

Specification

A broker is a message broker environment running on Amazon MQ. It is the basic building block of Amazon MQ. The combined description of the broker instance class (m5, t3) and size (large, micro) is a broker instance type (for example, mq.m5.large)

Rationale

An Active/standby broker for high availability comprises two brokers in two different Availability Zones, configured in a redundant pair. These brokers communicate synchronously with your application, and with Amazon EFS.

Evaluation

This control ensures that active/standby deployment mode is used for AWS MQ Brokers.

Remediation

Note - Deployment mode for AWS MQ Broker can be configured only at the time of creation. Remediation is not possible for this control once it is created. You have to delete the broker instance and create new with desired deployment mode.

Using AWS Console:

1. Sign in to the Amazon MQ console
2. Click Create Broker.
3. On the Select broker engine page, click Next.
4. In the **Deployment mode and storage type** section, choose the deployment mode.
5. Select **Active/standby broker**
6. Choose the **Storage type**.
7. Choose **Next**.
8. On the **Configure settings** page, in the Details section, do the following:
9. Choose the **Broker instance type** (for example, **mq.m5.large**). For more information, see Broker instance types.
10. In the **ActiveMQ Web Console access** section, provide a **Username** and **Password**.
11. Leave/change any other optional settings for the broker.
12. Choose **Next**.
13. Review your settings, edit and then click on **Create broker**.
14. When your broker is created successfully, Amazon MQ displays the **Running** status.

Using AWS CLI:

```
aws mq create-broker \
```

```
        --auto-minor-version-upgrade \
        --broker-name [broker-name] \
        --engine-type "ACTIVEMQ" --engine-version [version-number] \
        --host-instance-type "mq.t3.micro" --publicly-accessible \
        --users '{"ConsoleAccess" : true, "Groups" : [group-name], "Password"
: [password] , "Username" : [username]}' --deployment-mode
"ACTIVE_STANDBY_MULTI_AZ"
```

Reference

- Creating and configuring an ActiveMQ broker
- create-broker

Control ID - 495: Ensure advanced security options are enabled for AWS ElasticSearch Domain

Criticality: HIGH

Specification

Fine-grained access control offers additional ways of controlling access to your data on Amazon OpenSearch Service. For example, depending on who makes the request, you might want a search to return results from only one index. You might want to hide certain fields in your documents or exclude certain documents altogether.

Fine-grained access control requires OpenSearch or Elasticsearch 6.7 or later. It also requires HTTPS for all traffic to the domain, **Encryption of data at rest**, and **node-to-node encryption**. After you enable fine-grained access control, you can't disable it.

Rationale

Enabling fine-grained access control(Advanced security options) to protect the data on your domain. Fine-grained access control provides security at the cluster, index, document, and field levels.

Evaluation

This control ensures that advanced security options are enabled for AWS ElasticSearch Domain

Remediation

Using AWS Console

Note: You can enable fine-grained access control on existing domains running OpenSearch or Elasticsearch 6.7 or later.

1. Open the Amazon OpenSearch Service console at https://us-east-1.console.aws.amazon.com/esv3/home.
2. From Left side select Domain.
3. Select your domain to be remediated.
4. On the right side at the top choose **Actions** and **Edit security configuration**.
5. Select **Enable fine-grained access control**.
6. For Master user:

- If you want to use IAM for user management, choose **Set IAM ARN as master user** and specify the ARN for an IAM role.
- If you want to use the internal user database, choose **Create master user** and specify a user name and password.

- Click **Save changes**.

Using AWS CLI

To enable fine-grained access control(advanced security options) using command line. You must enable these security options: Required HTTPS, Node-to-node encryption, Encryption at rest, using the following command:

```
aws opensearch update-domain-config \
        --domain-name [domain-name] \
        --region [region] \
        --node-to-node-encryption-options '{ "Enabled": true }'  \
        --encryption-at-rest-options '{ "Enabled": true, "KmsKeyId": ["kms-
key-arn"] }' \
        --domain-endpoint-options EnforceHTTPS=true
```

Note: If all three security options are already enabled, you can skip the first step.

To enable fine-grained access control on OpenSearch or ElasticSearch domain, use the following command:

```
aws opensearch update-domain-config \
        --domain-name [domain-name] \
        --region [region] \
        --advanced-security-options '{ "Enabled": true,
"InternalUserDatabaseEnabled":true, "MasterUserOptions":
{"MasterUserName":["master-username"],"MasterUserPassword":["master-
password"], "MasterUserARN":["master-user-arn"]} }'
```

Note: For more details on command, please refer to
https://docs.aws.amazon.com/cli/latest/reference/opensearch/update-domain-config.html

Reference

- Fine-grained access control
- Get started with fine-grained access control in Amazon Elasticsearch Service
- Enabling fine-grained access control

Control ID - 496: Ensure general purpose SSD node type is used for AWS ElasticSearch Domains

Criticality: LOW

Specification

General Purpose SSD (gp2) volumes offer cost-effective storage that is ideal for a broad range of workloads. These volumes deliver single-digit millisecond latencies and the ability to burst to 3,000 IOPS for extended periods of time. Between a minimum of 100 IOPS (at 33.33 GiB and below) and a maximum of 16,000 IOPS (at 5,334 GiB and above), baseline performance scales linearly at 3 IOPS per GiB of volume size. AWS designs gp2

volumes to deliver their provisioned performance 99% of the time. A gp2 volume can range in size from 1 GiB to 16 TiB.

Rationale

Using EBS General Purpose (SSD) storage allows storing more data and to run on less costly instances. For gp2-type nodes you only pay for the storage compared to io1 nodes where you pay for both storage and I/O operations.

Evaluation

This control ensures that general purpose SSD node type is used for AWS ElasticSearch Domains.

Remediation

Note - We get General purpose SSD option only when we select instance type, for example, when we select instance type as "i2 xlarge search".

Using AWS Console

1. Open the Amazon OpenSearch Service console.
2. From Left side select Domain.
3. Select your domain to be remediated.
4. On the right side at the top choose **Actions** and **Edit cluster configuration**.
5. Select **General Purpose (SSD)** from the **EBS volume type** dropdown-list.
6. Click **Save changes**.

Using AWS CLI

To convert the storage type to General Purpose SSD (gp2), use the following command:

```
aws opensearch update-domain-config \
                    --domain-name [domain-name] \
                    --region [region] \
                    --ebs-options
EBSEnabled=true,VolumeType="gp2",VolumeSize=[volume-size]
```

Note: For more details on command, please refer to update-domain-config.

Reference

- Sizing Amazon OpenSearch Service domains
- Get Started with Amazon Elasticsearch Service: How Many Data Instances Do I Need?
- Amazon EBS volume types

Control ID - 497: Ensure KMS customer managed keys are used for encryption for AWS ElasticSearch Domains

Criticality: HIGH

Specification

Amazon Elasticsearch Service allows you to encrypt your data using keys that can be managed using AWS Key Management Service (KMS). You can choose to bring your own master key or leverage the one provided by the service. On an Amazon Elasticsearch Service domain with encryption enabled, all data stored on the underlying file systems are encrypted, including primary and replica indices, log files, memory swap files, and automated Amazon S3 snapshots. Encryption at rest supports both Amazon Elastic Block Store (EBS) and instance storage.

Rationale

When you use your own KMS Customer Master Keys to protect your ElasticSearch domains (clusters) and their storage systems, you have full control over who can use these keys to access the clusters data. The AWS KMS service allows you to easily create, rotate, disable and audit CMK encryption keys for your ES domains.

Note: Encryption of data at rest on new domains requires either OpenSearch or Elasticsearch 5.1 or later. Enabling it on existing domains requires either OpenSearch or Elasticsearch 6.7 or later.

Evaluation

This control ensures that KMS customer managed keys are used for encryption for AWS ElasticSearch Domains

Remediation

Using AWS Console

1. Note: Resource Can be configured only during resource creation.
2. Open the Amazon OpenSearch Service console at https://us-east-1.console.aws.amazon.com/esv3/home.
3. From Left Side Select "Domains" field.
4. Click Create Domain.
5. Fill the Required information.
6. From **Fine-grained access control** section Uncheck the **Enable fine-grained access control** field.
7. Under **Encryption**, select **Enable encryption of data at rest** and choose any AWS KMS key.
8. Click **Create** button.

Reference

- Encryption of data at rest for Amazon OpenSearch Service
- Amazon Elasticsearch Service extends encryption at rest and node-to-node encryption to existing domains
- AWS KMS concepts

Control ID - 498: Ensure Zone Awareness is enabled for AWS ElasticSearch Domain

Criticality: MEDIUM

Specification

To prevent data loss and minimize Amazon OpenSearch Service cluster downtime in the event of a service disruption, you can distribute nodes across two or three Availability Zones in the same Region, a configuration known as Multi-AZ. Availability Zones are isolated locations within each AWS Region.

Rationale

Enabling ES Zone Awareness provides better failure tolerance and availability because even if one zone is not available, you still have a complete copy of the data in the other zone. Availability Zones are isolated locations within each AWS Region.

Note: For a setup of three Availability Zones, use two replicas of your index. If there is a single zone failure, the two replicas afford 100% data redundancy.

Evaluation

This control ensures that Zone Awareness is enabled for AWS ElasticSearch Domain

Remediation

Using AWS Console

1. Open the Amazon OpenSearch Service console at https://us-east-1.console.aws.amazon.com/esv3/home.
2. From Left side select Domain.
3. Select your domain to be remediated.
4. On the right side at the top choose **Actions** and **Edit cluster configuration**.
5. Under **Availability Zones**, select **2-AZ** or **3-AZ**.
6. On the **Number of nodes** configuration:

- For two Availability Zones, you must choose instances in multiples of two.
- For three Availability Zones, we recommend instances in multiples of three for equal distribution across the Availability Zones.

- Click **Save changes**.

Using AWS CLI

To enable zone awareness for ElasticSearch domain, use the following command:

```
aws opensearch update-domain-config \
                    --domain-name [domain-name] \
                    --region [region]  \
                    --cluster-config InstanceCount=[Instance count/Number of
nodes],ZoneAwarenessEnabled=true,ZoneAwarenessConfig={AvailabilityZoneCount=[2
/3]}
```

Note: For more details on command, please refer to https://docs.aws.amazon.com/cli/latest/reference/opensearch/update-domain-config.html

Reference

- What is Amazon OpenSearch Service?
- Creating and managing Amazon OpenSearch Service domains
- Increase availability for Amazon OpenSearch Service by deploying in three Availability Zones

- How do I make my Amazon OpenSearch Service domain more fault tolerant?
- Amazon OpenSearch Service FAQs

Control ID - 499: Ensure Amazon cognito authentication is enabled for AWS ElasticSearch Domain

Criticality: MEDIUM

Specification

Amazon Cognito provides authentication, authorization, and user management for your web and mobile apps. Your users can sign in directly with a user name and password, or through a third party such as Facebook, Amazon, Google or Apple.

The two main components of Amazon Cognito are user pools and identity pools. User pools are user directories that provide sign-up and sign-in options for your app users. Identity pools enable you to grant your users access to other AWS services. You can use identity pools and user pools separately or together.

Rationale

Amazon Cognito helps you create unique identifiers for your end users that are kept consistent across devices and platforms. Cognito also delivers temporary, limited-privilege credentials to your application to access AWS resources.

Note: Amazon Cognito authentication requires either OpenSearch or Elasticsearch 5.1 or later.

Evaluation

This control ensures that Amazon cognito authentication is enabled for AWS ElasticSearch Domain

Remediation

Using AWS Console

1. Open the Amazon OpenSearch Service console at https://us-east-1.console.aws.amazon.com/esv3/home.
2. Select your domain to be remediated.
3. On the right side at the top choose **Actions** and **Edit security configuration**.
4. Under **Amazon Cognito authentication**, select **Enable Amazon Cognito authentication**.
5. For **Region**, select the Region that contains your Amazon Cognito user pool and identity pool.
6. For **Cognito user pool**, select a user pool or create one. For guidance, see About the user pool.
7. For **Cognito identity pool**, select an identity pool or create one. For guidance, see About the identity pool.
8. For **IAM role name**, use the default value of `CognitoAccessForAmazonOpenSearch` (recommended) or enter a new name.
9. Click **Save changes**.

After your domain finishes processing, see Allowing the authenticated role and Configuring identity providers for additional configuration steps.

Using AWS CLI

To list User pools, use the following command:

```
aws cognito-idp list-user-pools \
                --region [region] \
                --max-results 20
```

To list Identity pools , use the following command:

```
aws cognito-identity list-identity-pools \
                    --region [region] \
                    --max-results 20
```

Copy the **User pool Id** and **Identity pool Id** form the above commands output, to use in next commamd.

To enable Amazon Cognito authentication on ElasticSearch domain, use the following command:

```
aws opensearch update-domain-config \
                    --domain-name [domain-name] \
                    --region [region] \
                    --cognito-options Enabled=true,UserPoolId=["user-pool-
id"],IdentityPoolId=["identity-pool-id"],RoleArn=["iam-role-arn"]
```

Note: For more details on command, please refer to
https://docs.aws.amazon.com/cli/latest/reference/opensearch/update-domain-config.html

Reference

- What is Amazon Cognito?
- Get started with Amazon Elasticsearch Service: Use Amazon Cognito for Kibana access control
- Understanding Amazon Cognito Authentication
- Configuring Amazon Cognito authentication for OpenSearch Dashboards

Control ID - 500: Ensure dedicated master nodes are enabled for AWS ElasticSearch Domains

Criticality: MEDIUM

Specification

Amazon OpenSearch Service uses dedicated master nodes to increase cluster stability. A dedicated master node performs cluster management tasks, but does not hold data or respond to data upload requests. This offloading of cluster management tasks increases the stability of your domain. Just like all other node types, you pay an hourly rate for each dedicated master node.
For production domains, three is recommended.

Rationale

Using AWS Elasticsearch dedicated master nodes to separate management tasks from index and search requests will improve the clusters ability to manage easily different types of workload and make them more resilient in production.

Evaluation

This control ensures that Dedicated master nodes are enabled for AWS ElasticSearch Domains

Remediation

Using AWS Console

1. Open the Amazon OpenSearch Service console at https://us-east-1.console.aws.amazon.com/esv3/home.
2. Select your domain to be remediated.
3. On the right side at the top choose **Actions** and **Edit cluster configuration**.
4. Under **Dedicated master nodes** , select **Enable dedicated master nodes**.
5. Select **Instance type** and **Number of master nodes**.
6. Click **Save changes**.

Using AWS CLI

To enable Dedicated master nodes for ElasticSearch domain, use the following command:

```
aws opensearch update-domain-config \
                    --domain-name [domain-name] \
                    --region [region]  \
                    --cluster-config
DedicatedMasterEnabled=true,DedicatedMasterType=["instance-
type"],DedicatedMasterCount=[3/5]
```

Note: For more details on command, please refer to https://docs.aws.amazon.com/cli/latest/reference/opensearch/update-domain-config.html

Reference

- What is Amazon OpenSearch Service?
- Creating and managing Amazon OpenSearch Service domains
- Dedicated master nodes in Amazon OpenSearch Service
- Amazon OpenSearch Service FAQs

Control ID - 501: Ensure policies are used for AWS CloudFormation Stacks

Criticality: MEDIUM

Specification

You can define only one stack policy per stack, but, you can protect multiple resources within a single policy. A stack policy applies to all AWS CloudFormation users who attempt to update the stack.

Rationale

A stack policy defines the resources that you want to protect from unintentional updates during a stack update.

Evaluation

This control ensures that policies are used for AWS CloudFormation Stacks.

Remediation

Note - It is possible to **Update** only stacks with specific stack statuses. For example: CREATE_COMPLETE, UPDATE_COMPLETE, UPDATE_ROLLBACK_COMPLETE etc. For statuses like CREATE_FAILED, ROLLBACK_COMPLETE, DELETE_COMPLETE etc., **Update** option will be disabled. Hence, for stacks with such statuses, remediation won't be possible. Please find reference for stack status codes here.

Using AWS Console:

1. Log in to the AWS CloudFormation console
2. On the **AWS CloudFormation** dashboard, choose a stack that needs to be remediated and then choose **Update**.
3. In the **Update Stack wizard**, on the **Specify template step**, select appropriate template option. Click **Next**.
4. On the next step **Specify stack details**, update any parameters you want to and then click **Next** on the bottom of the page.
5. On the third step, **Configure stack options**, go to **Advanced options**.
6. Here you must select **Enter stack policy** or **Upload a file** option.
7. Click **Next**.
8. On the **Review** screen, verify that all the settings are as you want them, and then choose **Update stack**.

Using AWS CLI:

1. Check if you have a desired policy file by listing all files

```
ls -all
```

2. If the policy file already exists at the home directory, you can use **aws cloudformation set-stack-policy** command to set the desired policy, or you can create a file named **stack-policy.json** using following command.

```
vi stack-policy.json

{
  "Statement" : [
          {
                  "Effect" : "Deny",
                  "Action" : "Update:*",
                  "Principal" : "*",
                  "Resource" : "*",
                  "Condition" : {
                  "StringEquals" : {
                                  "ResourceType" :
["AWS::RDS::DBInstance"]
                          }
                  }
          },
          {
                  "Effect" : "Allow",
                  "Action" : "Update:*",
                  "Principal" : "*",
                  "Resource" : "*"
          }
  ]
```

```
}
```

3. After adding the JSON in the StackPolicy.json file save the file using **esc :wq** command
4. Following command sets above stack policy for a specified stack.

```
aws cloudformation set-stack-policy \
  --stack-name [stack-name] \
  --stack-policy-body file://stack-policy.json
```

Reference

- Modifying a stack policy
- set-stack-policy
- Updating protected resources

Control ID - 502: Ensure termination protection is enabled for AWS CloudFormation Stack

Criticality: MEDIUM

Specification

You can enable termination protection on a stack when you create it. Termination protection on stacks is disabled by default. You can set termination protection on a stack with any status except DELETE_IN_PROGRESS or DELETE_COMPLETE.

Rationale

You can prevent a stack from being accidentally deleted by enabling termination protection on the stack. If a user attempts to delete a stack with termination protection enabled, the deletion fails and the stack, including its status, remains unchanged.

Evaluation

This control ensures that termination protection is enabled for AWS CloudFormation Stack.

Remediation

Using AWS Console:

1. Log in to the AWS CloudFormation console
2. Select the stack you want.
3. In the stack details pane, select **Stack actions** and then click **Edit termination protection**.
4. Cloudformation displays the **Edit termination protection** dialog box.
5. Choose **Enable**, and then select **Save**.

Using AWS CLI:

To enable termination protection for AWS CloudFormation Stack, use following command:

```
aws cloudformation update-termination-protection \
```

```
--enable-termination-protection \
--stack-name [stack-name]
```

Reference

- Protecting a stack from being deleted
- update-termination-protection

Control ID - 503: Ensure TLS security policy is using 1.2 version for the custom domains

Criticality: HIGH

Specification

It is recommended to use Transport Layer Security (TLS) 1.2 protocol it helps in addressing network security problems such as tampering and eavesdropping between a client and server.

Rationale

A security policy is a predefined combination of minimum TLS version and cipher suite offered by Amazon API Gateway. You can choose TLS version 1.2 security policy. The TLS protocol addresses network security problems such as tampering and eavesdropping between a client and server. When clients establish a TLS handshake to your API through the custom domain, the security policy enforces the TLS version and cipher suite options clients can choose to use.

Evaluation

This control ensures that TLS security policy is using 1.2 version for the custom domains

Remediation

Using AWS Console:

1. Sign in to the AWS Management Console and open the Amazon API Gateway console at https://console.aws.amazon.com/apigateway/
2. In the API Gateway console, Select **Custom domain names**.
3. Select the Custom domain to be remediated
4. Click **Edit**.
5. Under **Minimum TLS version** select **TLS 1.2**.
6. Click on **Save Changes**

Using AWS CLI:

```
aws apigateway update-domain-name --domain-name [DOMAIN_NAME] --patch-
operations op='replace',path='/securityPolicy',value='TLS_1_2'
```

Note: For more details on command, please refer to https://docs.aws.amazon.com/cli/latest/reference/apigateway/update-domain-name.html

Reference

- https://docs.aws.amazon.com/apigateway/latest/developerguide/apigateway-custom-domain-tls-version.html

Control ID - 504: Ensure there is a Dead Letter Queue configured for each Amazon SQS queue

Criticality: MEDIUM

Specification

Simple Queue Service (SQS) queue is configured to use a Dead Letter Queue (DLQ) in order to maintain the queue flow and avoid losing data by detecting and mitigating failures and service disruptions on time. A Dead Letter Queue is an SQS queue useful for debugging application or messaging system, that isolate messages that can't be processed successfully for later analysis.

Rationale

Dead Letter Queues (DLQs) for SQS queues help to troubleshoot incorrect message transmission operations that can lead to data loss. Use DLQs to decrease the number of unprocessed messages and reduce the possibility of exposing queues to poison pill messages (i.e. messages that are received but can't be processed for some reason).

Evaluation

This control ensures that each AWS Simple Queue Service (SQS) queue is configured to use a Dead Letter Queue (DLQ).

Remediation

Using AWS Console:

To set up the necessary Dead Letter Queue:

1. Go to AWS Console SQS dashboard at https://console.aws.amazon.com/sqs/
2. Click on **Create Queue** button.
3. Select **Type** for your dead-letter queue with respect to source queue type.

   Note: The dead-letter queue of a FIFO queue must also be a FIFO queue. Similarly, the dead-letter queue of a standard queue must also be a standard queue.

4. Enter a unique name for the queue in the **Queue Name** box and leave the queue default parameters unchanged.
5. Click **Create Queue**.

To edit exiting Queue:

1. Go to AWS Console SQS dashboard at https://console.aws.amazon.com/sqs/
2. In the left navigation panel select **Queues**, select effected **Queue**.
3. Click on **Edit** button.
4. Scroll to the **Dead-letter queue** section and choose **Enabled**.
5. Under **Choose queue**, enter the above created dead-letter queue ARN.

Note: The dead-letter queue of a FIFO queue must also be a FIFO queue. Similarly, the dead-letter queue of a standard queue must also be a standard queue.

6. In the **Maximum Receives** box, enter the maximum number of times an SQS message can be received before it is sent to the Dead Letter Queue (DLQ). The value must be between 1 and 1000.
7. Click on **Save**.

Using AWS CLI:

To create Standard queue"

```
aws sqs create-queue --queue-name MyQueue
```

To create FIFO queue:

```
aws sqs create-queue --queue-name MyQueue --attributes "FifoQueue"
```

To enable Dead-letter queue:

```
aws sqs set-queue-attributes --queue-url [Queue_URL] --attributes
file://[RedrivePolicy.json]
```

RedrivePolicy.json file:

```
            {
               "RedrivePolicy":
"{\"deadLetterTargetArn\":\"[DeadQueue_ARN]\",\"maxReceiveCount\":\"[VALUE]\"}
"
            }
```

Note: For more details on command, please refer to
https://docs.aws.amazon.com/cli/latest/reference/sqs/set-queue-attributes.html

Reference

- https://docs.aws.amazon.com/AWSSimpleQueueService/latest/SQSDeveloperGuide/sqs-dead-letter-queues.html
- https://docs.aws.amazon.com/AWSSimpleQueueService/latest/SQSDeveloperGuide/sqs-configure-dead-letter-queue.html

Control ID - 505: Ensure that EMR cluster is configured with security configuration

Criticality: MEDIUM

Specification

Security configuration specify data encryption, authentication, amazon S3 authorization for EMRFS and ec2 instance metadata service settings when you create an Amazon EMR cluster.

Rationale

With Amazon EMR release version 4.8.0 or later, you can use security configurations to configure data encryption, Kerberos authentication (available in release version 5.10.0 and later), and Amazon S3 authorization for EMRFS (available in release version 5.10.0 or later).

After you create a security configuration, you specify it when you create a cluster, and you can re-use it for any number of clusters.

Evaluation

This control ensures that EMR cluster is configured with security configuration

Remediation

Using AWS Console:

Note : We cannot modify an existing EMR cluster, new EMR cluster should be created with required configuration.

To create a security configuration, follow the below procedure:

1. Sign in to AWS Console and navigate to https://console.aws.amazon.com/elasticmapreduce/home.
2. In the Navigation pane, choose **Security configurations**.
3. Click on **Create** button.
4. Enter a unique name for the Security configuration.
5. Configure the **Security configuration** according to the requirement.
6. Click **Create** button.

To Create and Configure an EMR cluster with Secuirty Configuration

1. Sign in to AWS Console and navigate to https://console.aws.amazon.com/elasticmapreduce/home.
2. In the Navigation pane, choose **Clusters**.
3. Click on **Create cluster** button.
4. Select **Go to advanced options**.
5. In steps 1 - 3, configure options according to the requirements.
6. In **step 4**, select **Security Configuration**.
7. Select above created security configuration.
8. Click **Create cluster** button.

Using AWS CLI:

AWS CLI command to create new Secuirty Configuration

```
aws emr create-security-configuration --name [SECURITY_CONFIGURATION_NAME] --
security-configuration [SECURITY_CONFIGURATIONS]
```

For command usage and for flexibility of adding more settings to the command refer:
https://docs.aws.amazon.com/cli/latest/reference/emr/create-security-configuration.html

AWS CLI command to create new EMR cluster with Secuirty Configuration

```
aws emr create-cluster --release-label [RELEASE_LABEL] --instance-type
[INSTANCE_TYPE] --ec2-attributes InstanceProfile=[EC2_INSTANCE_PROFILE] --
service-role [SERVICE_ROLE] --security-configuration
[SECURITY_CONFIGURATION_NAME]
```

For command usage and for flexibility of adding more settings to the command refer:
https://docs.aws.amazon.com/cli/latest/reference/emr/create-cluster.html

Reference

- https://docs.aws.amazon.com/emr/latest/ManagementGuide/emr-create-security-configuration.html
- https://docs.aws.amazon.com/emr/latest/ManagementGuide/emr-specify-security-configuration.html

Control ID - 506: Ensure AWS Elastic MapReduce (EMR) clusters capture detailed log data to Amazon S3

Criticality: MEDIUM

Specification

By enabling logging, EMR uploads the log files from the cluster master instance(s) to Amazon S3 so the logging data can be utilized later for troubleshooting or compliance purposes.

Rationale

By default, all EMR log files are automatically deleted from the clusters after the retention period ends. By enabling logging, Elastic MapReduce uploads the log files from the cluster master instance(s) to Amazon S3 so the logging data (step logs, Hadoop logs, instance state logs, etc) can be utilized later for troubleshooting or compliance purposes. Once active, the EMR service archives and sends the log files to Amazon S3 at 5 minute intervals.

Evaluation

This control ensures that AWS Elastic MapReduce (EMR) clusters capture detailed log data to Amazon S3

Remediation

Using AWS Console:

Note : We cannot modify an existing EMR cluster, new EMR cluster should be created with required configuration.

To Create and Configure an EMR cluster with logging enabled:

1. Sign in to AWS Console and navigate to https://console.aws.amazon.com/elasticmapreduce/home.
2. In the Navigation pane, choose **Clusters**.
3. Click on **Create cluster** button.
4. In **General Configuration** section select **Logging** checkbox.
5. Select S3 folder.
6. Configure rest of the options according to the requirements.
7. Click **Create cluster** button.

Using AWS CLI:

With below AWS CLI command we can create new EMR cluster with logging enabled

```
aws emr create-cluster --release-label [RELEASE_LABEL] --instance-type
[INSTANCE_TYPE] --ec2-attributes InstanceProfile=[EC2_INSTANCE_PROFILE] --
service-role [SERVICE_ROLE] --log-uri [S3_FOLDER]
```

For command usage and for flexibility of adding more settings to the command refer:
https://docs.aws.amazon.com/cli/latest/reference/emr/create-cluster.html

Reference

- https://docs.aws.amazon.com/emr/latest/ManagementGuide/emr-plan-debugging.html

Control ID - 508: Ensure AWS EBS Volume has a corresponding AWS EBS Snapshot

Criticality: HIGH

Specification

EBS is a block storage service provided by AWS, you can back up the data on your Amazon EBS volumes to Amazon S3 by taking point-in-time snapshots. Snapshots are incremental backups, which means that only the blocks on the device that have changed after your most recent snapshot are saved. This minimizes the time required to create the snapshot and saves on storage costs by not duplicating data. Each snapshot contains all the information that is needed to restore your data (from the moment when the snapshot was taken) to a new EBS volume.

When you create an EBS volume based on a snapshot, the new volume begins as an exact replica of the original volume that was used to create the snapshot. The replicated volume loads data in the background so that you can begin using it immediately. If you access data that hasn't been loaded yet, the volume immediately downloads the requested data from Amazon S3, and then continues loading the rest of the volume's data in the background.

Rationale

Amazon Elastic Block Store (EBS) Snapshots provide a simple and secure data protection solution that is designed to protect your block storage data such as EBS volumes, boot volumes, as well as on-premises block data. EBS Snapshots are a point-in-time copy of your data, and can be used to enable disaster recovery, migrate data across regions and accounts, and improve backup compliance.

Evaluation

This control ensures that AWS EBS Volume has a corresponding AWS EBS Snapshot.

Remediation

Using AWS Console:

1. Login to AWS Management console.
2. Navigate to EC2.
3. In the left navigation panel, under **Elastic Block Store**, click **Snapshots**.
4. Choose **Create snapshot**.
5. For **Resource type**, select **Volume**.

6. For **Volume ID**, select the volume from which to create the snapshot.
7. (Optional) For **Description**, enter a brief description for the snapshot.
8. Click **Create snapshot.**

Using AWS CLI:

```
aws ec2 create-snapshot --volume-id [volume-id] \
    --description [description]
```

Reference

- Amazon Elastic Block Store (Amazon EBS)
- Amazon EBS snapshots
- Create Amazon EBS snapshots
- Create snapshot

Control ID - 509: Ensure egress filter is set as DROP_ALL for AWS Application Mesh

Criticality: MEDIUM

Specification

By default, the type is DROP_ALL, which allows egress only from virtual nodes to other defined resources in the service mesh (and any traffic to *.amazonaws.com for AWS API calls).

Rationale

Egress filter controls how Envoy proxies participating in the mesh will forward traffic to non-AWS resources that are not defined in the mesh. With egress filter set to 'DROP_ALL', proxies will not forward traffic to external services that are not defined in the mesh.

Evaluation

This control ensures egress filter is set as 'DROP_ALL' for AWS Application Mesh.

Remediation

Using AWS Console:

1. Login to AWS Management console.
2. Navigate to AWS App Mesh.
3. Click on **Meshes** from left navigation bar.
4. Select the mesh you want to edit.
5. Click on **Edit**.
6. Select option **Deny external traffic** under Egress filter setting.
7. Click **Save**.

Using AWS CLI:

```
aws appmesh update-mesh \
```

```
--mesh-name [mesh-name] \
--spec '{ "egressFilter": { "type": "DROP_ALL" } }'
```

Reference

- update-mesh

Control ID - 510: Ensure secrets should be auto rotated after not more than 90 days

Criticality: HIGH

Specification

Secrets Manager supports many types of secrets. However, Secrets Manager can natively rotate credentials for supported AWS databases without any additional programming. Rotating the secrets for other databases or services requires creating a custom Lambda function to define how Secrets Manager interacts with the database or service. You need some programming skill to create the function.

Rationale

Secrets Manager enables you to replace hardcoded credentials in your code, including passwords, with an API call to Secrets Manager to retrieve the secret programmatically. This helps ensure the secret can't be compromised by someone examining your code, because the secret no longer exists in the code. Also, you can configure Secrets Manager to automatically rotate the secret for you according to a specified schedule. This enables you to replace long-term secrets with short-term ones, significantly reducing the risk of compromise.

Evaluation

This control ensures that secrets should be auto-rotated after not more than 90 days.

Remediation

Using AWS Console:

1. Login to AWS Management console.
2. Navigate to Secrets Manager console
3. On the **Secrets** page, choose your secret.
4. On the **Secret details** page, in the **Rotation configuration** section, choose **Edit rotation**. The **Edit rotation configuration** dialog box opens.
5. Turn on **Automatic rotation**.
6. Choose **Schedule expression builder** to build a schedule in a form. Select 'Days' as Time Unit and set its value to **90 days or less**.
7. Under **Rotation function**, do one of the following:

- If you already created a rotation function for this type of secret, choose it.
- Otherwise, choose **Create function**.

Using AWS CLI:

```
aws secretsmanager rotate-secret \
```

```
--secret-id [secret-name] \
--rotation-lambda-arn [rotation-lambda-arn] \
--rotation-rules "{\"ScheduleExpression\": \"rate(10 days)\"}"
```

Reference

- AWS Secrets Manager
- Automatically rotate a secret
- rotate-secret

Control ID - 511: Ensure CORS is configured to prevent sharing across all domains for AWS API Gateway V2 API

Criticality: MEDIUM

Specification

You can use API Gateway to generate an SSL certificate and then use its public key in the backend to verify that HTTP requests to your backend system are from API Gateway. This allows your HTTP backend to control and accept only requests that originate from Amazon API Gateway, even if the backend is publicly accessible. The SSL certificates that are generated by API Gateway are self-signed, and only the public key of a certificate is visible in the API Gateway console or through the APIs.

Note: If you configure CORS for an API, API Gateway automatically sends a response to preflight OPTIONS requests, even if there isn't an OPTIONS route configured for your API. For a CORS request, API Gateway adds the configured CORS headers to the response from an integration. If you configure CORS for an API, API Gateway ignores CORS headers returned from your backend integration.

Rationale

Cross-origin resource sharing (CORS) is a browser security feature that restricts HTTP requests that are initiated from scripts running in the browser. CORS is typically required to build web applications that access APIs hosted on a different domain or origin. You can enable CORS to allow requests to your API from a web application hosted on a different domain. For example, if your API is hosted on `https://{api_id}.execute-api.{region}.amazonaws.com/` and you want to call your API from a web application hosted on example.com, your API must support CORS.

Evaluation

This control ensures that CORS is configured to prevent sharing across all domains for AWS API Gateway V2 API.

Remediation

Using AWS Console:

1. Login to AWS Management console.
2. Navigate to Amazon API Gateway console.
3. In the API Gateway console, Select **APIs**.
4. Choose the API to be remediated from the APIs list.
5. Choose **CORS** under **Develop**.
6. Choose **Configure**.

7. In the **CORS Configure** do the following:

- In the **Access-Control-Allow-Origin** input field, you can specify * (to allow all origins) or specify origins to be permitted to access the resource. Click **Add**.
- In the **Access-Control-Allow-Headers** input field, type a static string of a comma-separated list of headers. For example, **Authorization, ***. Click **Add**.
- From the **Access-Control-Allow-Methods** drop-down, choose the allowed methods.
- In the **Access-Control-Expose-Headers** input field, type a static string of a comma-separated list of exposed headers. For example, **Date, x-api-id**. Click **Add**.
- In the **Access-Control-Max-Age** specify the number of seconds that the browser should cache preflight request results.
- Enable **Access-Control-Allow-Credentials** to include credentials in the CORS request.

- Click **Save**.

Using AWS CLI:

```
aws apigatewayv2 update-api --api-id [api-id] \
    --cors-configuration AllowOrigins=[allowed-origins],AllowHeaders=[allowed-
headers-strings],AllowMethods=[allowed-methods],ExposeHeaders=[expose-
headers],MaxAge=[max-age-number],AllowCredentials=[true|false]
```

Note: For more details on command, please refer to update-api

Reference

- What is Amazon API Gateway?
- Working with HTTP APIs
- Configuring CORS for an HTTP API
- AWS::ApiGatewayV2::Api Cors

Control ID - 513: Ensure IMDv1 is disabled for AWS EC2 instances

Criticality: HIGH

Specification

Instance metadata is data that is related to an Amazon Elastic Compute Cloud (Amazon EC2) instance that applications can use to configure or manage the running instance. The instance metadata service (IMDS) is an on-instance component that code on the instance uses to securely access instance metadata. This code can be Elastic Beanstalk platform code on your environment instances, the AWS SDK that your application might be using, or even your application's own code.

Code can access instance metadata from a running instance using one of two methods: Instance Metadata Service Version 1 (IMDSv1) or Instance Metadata Service Version 2 (IMDSv2).

Rationale

Instance metadata is data about your instance that you can use to configure or manage the running instance. Instance metadata is divided into categories, for example, host name, events, and security groups. You can also use instance metadata to access user data that you specified when launching your instance. For example, you can specify parameters for configuring your instance, or include a simple script.

Evaluation

This control ensures that IMDSv1 is disabled for AWS EC2 instances.

Remediation

Using AWS Console:

1. Login to AWS Management console.
2. Navigate to EC2.
3. From left side select **"Instances under instances field"**.
4. Click **Launch instance**.
5. Go to **Advanced details**.
6. Select "Metadata version" as V2 Only & **"Allow tags in metadata"** as **Disable**.
7. Click **Launch instance**.

Using AWS CLI:

```
aws ec2 modify-instance-metadata-options --instance-id [instance-id] \
    --http-tokens required \
    --http-endpoint enabled \
    --instance-metadata-tags disabled
```

Reference

- AWS EC2 configure instance metadata option reference
- AWS EC2 configure instance metadata option cli reference

Control ID- 514: Ensure sufficient data retention period is set for AWS Kinesis Streams (7 days or More)

Criticality: MEDIUM

Specification

Amazon Kinesis Data Streams supports changes to the data record retention period of your data stream. A Kinesis data stream is an ordered sequence of data records meant to be written to and read from in real time. Data records are therefore stored in shards in your stream temporarily. The time period from when a record is added to when it is no longer accessible is called the retention period. A Kinesis data stream stores records from 24 hours by default, up to 8760 hours (365 days).

Rationale

A sufficient data retention period allows more time for your Kinesis stream data consumers to recover. The default retention period for an AWS Kinesis stream is 24 hours. To ensure that your consumers are able to read stream data before it expires if any problems occur, you can extend your data retention period up to 168 hours (7 days).

Evaluation

This control ensures that sufficient data retention period is set for AWS Kinesis Streams (7 days or More).

Remediation

Using AWS Console:

1. Login to AWS Management console.
2. Navigate to Amazon Kinesis Dashboard.
3. In the left navigation panel, Select the **Data streams**.
4. Select the **Data streams** that you want to modify.
5. Under **Configuration** tab. Select **Data retention** click in Edit to modify its value.
6. In **Edit data retention period** Select Data retention period 7 or more than 7 days.
7. Click on **Save** to save your changes.

Using AWS CLI:

```
aws kinesis increase-stream-retention-period --region [region] \
    --stream-name [stream-name] \
    --retention-period-hours 168
```

Reference

- Changing the Data Retention Period

Control ID - 516: Ensure AWS ACM certificates are renewed 7 days before expiration date

Criticality: MEDIUM

Specification

Secure Sockets Layer/Transport Layer Security (SSL/TLS) certificates are used to secure network communication and establish the identity of websites over the internet. Certificates have a defined lifetime and for continued use need to be renewed before they expire. These new metrics and events help administrators keep track of certificate expiration dates and take necessary action or configure automation to prevent certificate expiry and related outages.

ACM lets you easily provision, manage, and deploy public and private SSL/TLS certificates. ACM provides managed renewal to automatically renew certificates in most cases. However, there are exceptions where user action is needed for certificate renewal. For example, ACM does not attempt to renew third-party certificates that are imported. Also, an administrator needs to reconfigure missing DNS records for certificates that use DNS validation if the record was removed for any reason after the certificate was issued. Metrics and events provides you visibility into such certificates that require intervention to continue the renewal process. A certificate that isn't renewed expires, which can lead to a website or an application being unavailable.

Rationale

When Secure Sockets Layer/Transport Layer Security (SSL/TLS) certificates are not renewed prior to their expiration date become invalid. Invalid certificates make communication between the client and AWS resources insecure.

Note: ACM provides managed renewal for your Amazon-issued SSL/TLS certificates. This means that ACM will either renew your certificates automatically (if you are using DNS validation), or it will send you email

notices when expiration is approaching. These services are provided for both public and private ACM certificates.

Evaluation

This control ensures that AWS ACM certificates are renewed 7 days before expiration date.

Remediation

Using AWS Console:

1. Login to AWS Management console.
2. Navigate to Amazon ACM console.
3. Select the certificate to re-import.
4. On the details pane of the certificate and choose the **Reimport** certificate button.
5. On the import certificate page, do the following:
    - For **Certificate body**, paste the PEM-encoded certificate to import. It should begin with `-----BEGIN CERTIFICATE-----` and end with **-----END CERTIFICATE-----**.
    - For **Certificate private key**, paste the certificate's PEM-encoded, unencrypted private key. It should begin with **-----BEGIN PRIVATE KEY-----** and end with **-----END PRIVATE KEY-----**.
    - (Optional) For **Certificate chain**, paste the PEM-encoded certificate chain.
6. Click **Next**.
7. On the **Review and import** page, review the imported certificate details then click **Import** to confirm the action and complete the renewal process.

Using AWS CLI:

```
aws acm import-certificate --certificate-arn [certificate-arn] \
    --certificate fileb://Certificate.pem \
    --certificate-chain fileb://CertificateChain.pem \
    --private-key fileb://PrivateKey.pem
```

Note: For more details on command, please refer to Reimport (AWS CLI)

Reference

- What Is AWS Certificate Manager?
- Managed renewal for ACM certificates
- AWS Certificate Manager now provides certificate expiry monitoring through Amazon CloudWatch
- Importing certificates into AWS Certificate Manager

Control ID- 517: Ensure customer master key (CMK) is not disabled for AWS Key Management Service (KMS)

Criticality: HIGH

Specification

AWS KMS keys (KMS keys) are the primary resource in AWS KMS. You can use a KMS key to encrypt, decrypt, and re-encrypt data. It can also generate data keys that you can use outside of AWS KMS. Typically, you'll use symmetric encryption KMS keys, but you can create and use asymmetric KMS keys for encryption or signing.

Rationale

AWS Key Management Service (AWS KMS) is a managed service that makes it easy for you to create and control the cryptographic keys that are used to protect your data.

Evaluation

This control ensures that customer managed key (CMK) is not disabled for AWS Key Management Service (KMS).

Remediation

Using AWS Console:

1. Login to AWS Management console.
2. Navigate to Amazon AWS Key Management Service Dashboard.
3. In the left navigation panel, Select the **Customer managed keys** .
4. Select the **Customer managed keys** that you want to modify.
5. In key Action dropdown option Select **Enable** Option to enable KMS key.

Using AWS CLI:

```
aws kms enable-key --key-id [key-id]
```

Reference

- To enable a customer master key (CMK)

Control ID - 518: Ensure SNS Topics are encrypted with customer managed master key

Criticality: HIGH

Specification

By default, Amazon SNS provides in-transit encryption. Enabling server-side adds encryption at rest encryption to your topic.

Rationale

Encryption at rest protects the contents of messages in topics using KMS key configured. As soon as the message is received by SNS, it will be encrypted and stored in encrypted form. Without encryption enabled, anyone who gains access will be able to read the message.

Evaluation

This control ensures that SNS Topics are encrypted with customer managed master key in the regions.

Remediation

Using AWS Console:

1. Login to AWS Management console.
2. Navigate to AWS Console SNS Dashboard
3. In the left navigation panel, select **Topics**, select effected **topic**.
4. Click on **Edit** button.
5. Under **Encryption** section, select **Enable encryption**.
6. Under **Customer master key**, Select a custom CMK or enter an existing CMK ARN.
7. Click on **Save changes**.

Using AWS CLI:

```
aws sns set-topic-attributes --topic-arn [topic-arn] \
    --attribute-name KmsMasterKeyId \
    --attribute-value [key-id]
```

Note: For more details on command, please refer to set-topic-attributes.

Reference

- Encryption at rest

Control ID - 519: Ensure ML storage volume attached to notebooks are encrypted

Criticality: HIGH

Specification

To encrypt the machine learning (ML) storage volume that is attached to notebooks, processing jobs, training jobs, hyperparameter tuning jobs, batch transform jobs, and endpoints, you can pass a AWS KMS key to SageMaker. If you don't specify a KMS key, SageMaker encrypts storage volumes with a transient key and discards it immediately after encrypting the storage volume. For notebook instances, if you don't specify a KMS key, SageMaker encrypts both OS volumes and ML data volumes with a system-managed KMS key.

Rationale

Sensitive data needs to be protected at-rest and in-transit. By encrypting ML storage volume data of a notebook instance, data protection at-rest is ensured.

Evaluation

This control ensures that ML storage volumes attached to notebooks are encrypted either with AWS managed key or CMK in the regions.

Remediation

Using AWS Console:

1. Login to AWS Management console.
2. Navigate to AWS SageMaker console.
3. In the left navigation panel, under **Images**, select **Notebook** and choose **Notebook instances**.
4. Click **Create notebook instances** button from the dashboard top menu.
5. In the **Permissions and encryption** section, choose an **Encryption key**.
6. When you're done, choose **Create notebook instance** to create the **SageMaker notebook instance**.

Reference

- AWS SageMaker Documentation
- Use Amazon SageMaker Notebook Instances
- Notebook instances and SageMaker jobs

Control ID - 520: Ensure ML storage volume attached to notebooks are encrypted with customer managed master key

Criticality: HIGH

Specification

To encrypt the machine learning (ML) storage volume that is attached to notebooks, processing jobs, training jobs, hyperparameter tuning jobs, batch transform jobs, and endpoints, you can pass a AWS KMS key to SageMaker. If you don't specify a KMS key, SageMaker encrypts storage volumes with a transient key and discards it immediately after encrypting the storage volume. For notebook instances, if you don't specify a KMS key, SageMaker encrypts both OS volumes and ML data volumes with a system-managed KMS key.

Rationale

Sensitive data needs to be protected at-rest and in-transit. By encrypting ML storage volume data with CMK of a notebook instance, data protection at-rest is ensured.

Evaluation

This control ensures that ML storage volume attached to notebooks are encrypted with customer managed master key in the regions.

Remediation

Using AWS Console:

1. Login to AWS Management console.
2. Navigate to AWS SageMaker console.
3. In the left navigation panel, under **Images**, select **Notebook** and choose **Notebook instances**.
4. Click **Create notebook instances** button from the dashboard top menu.
5. In the **Permissions and encryption** section, use a **Customer managed Key** for **Encryption key**.
6. When you're done, choose **Create notebook instance** to create the **SageMaker notebook instance**.

Reference

- AWS SageMaker Documentation
- Use Amazon SageMaker Notebook Instances
- Notebook instances and SageMaker jobs

Control ID - 521: Ensure ML storage volume attached to processing jobs are encrypted

Criticality: HIGH

Specification

To encrypt the machine learning (ML) storage volume that is attached to notebooks, processing jobs, training jobs, hyperparameter tuning jobs, batch transform jobs, and endpoints, you can pass a AWS KMS key to SageMaker. If you don't specify a KMS key, SageMaker encrypts storage volumes with a transient key and discards it immediately after encrypting the storage volume. For notebook instances, if you don't specify a KMS key, SageMaker encrypts both OS volumes and ML data volumes with a system-managed KMS key.

Rationale

Sensitive data needs to be protected at-rest and in-transit. By encrypting ML storage volume attached to processing jobs, data protection at-rest is ensured.

Evaluation

The control ensures that ML storage volume attached to processing jobs are encrypted in the regions.

Remediation

Using AWS Console:

1. Login to AWS Management console.
2. Navigate to AWS SageMaker console.
3. In the left navigation panel, under **Images**, select **Processing** and choose **Processing jobs**.
4. Click **Create Processing job** button from the dashboard top menu.
5. In the **Resource configuration** section, choose an **Encryption key**.
6. When you're done, choose **Create processing job** to create **SageMaker Processing Job**.

Reference

- AWS SageMaker Documentation
- Process Data
- Notebook instances and SageMaker jobs

Control ID - 522: Ensure ML storage volume attached to processing jobs(if configured) are encrypted with customer managed master key

Criticality: HIGH

Specification

To encrypt the machine learning (ML) storage volume that is attached to notebooks, processing jobs, training jobs, hyperparameter tuning jobs, batch transform jobs, and endpoints, you can pass a AWS KMS key to SageMaker. If you don't specify a KMS key, SageMaker encrypts storage volumes with a transient key and

discards it immediately after encrypting the storage volume. For notebook instances, if you don't specify a KMS key, SageMaker encrypts both OS volumes and ML data volumes with a system-managed KMS key.

Rationale

Sensitive data needs to be protected at-rest and in-transit. By encrypting ML storage volume attached to processing jobs with CMK, data protection at-rest is ensured.

Evaluation

This control ensures that ML storage volume attached to processing jobs(if configured) are encrypted with customer managed key in all the regions.

Remediation

Using AWS Console:

1. Login to AWS Management console.
2. Navigate to AWS SageMaker console.
3. In the left navigation panel, under **Images**, select **Processing** and choose **Processing jobs**.
4. Click **Create Processing job** button from the dashboard top menu.
5. In the **Resource configuration** section, use a **Customer Managed Key** for **Encryption key**.
6. When you're done, choose **Create processing job** to create **SageMaker Processing Job**.

Reference

- AWS SageMaker Documentation
- Process Data
- Notebook instances and SageMaker jobs

Control ID - 523: Ensure to encrypt the output of processing jobs

Criticality: HIGH

Specification

You can encrypt S3 output data of processing jobs with an AWS KMS key that you specify.

Rationale

Sensitive data needs to be protected at-rest and in-transit. By encrypting S3 output data of processing jobs, data protection at-rest is ensured.

Evaluation

This control ensures that S3 output data of processing jobs is encrypted in all the regions.

Remediation

Note: This control evaluates whether S3 output data of processing jobs is encrypted for all regions together i.e. failure in any one region will fail the control. If any region is not configured for S3 output data encryption, then it is considered as region failure.

Using AWS Console:

1. Login to AWS Management console.
2. Navigate to Amazon Sagemaker console
3. In the navigation pane, choose **Processing**, then choose **Processing jobs**.
4. Choose **Create Processing job**.
5. Under **Output data configuration**, type or select AWS KMS keyID/CMK KeyID ARN from the dropdown below **Encryption key** for S3 output path.
6. After enabling S3 output data encryption, finish creating the Processing job.

Reference

- Notebook instances and SageMaker jobs
- create-processing-job

Control ID - 524: Ensure to encrypt the output of processing jobs(if configured) in s3 with customer managed master key

Criticality: HIGH

Specification

You can encrypt S3 output data of processing jobs with a customer managed key that you specify.

Rationale

Sensitive data needs to be protected at-rest and in-transit. By encrypting S3 output data of processing jobs encryption with CMK, data protection at-rest is ensured.

Evaluation

This control ensures that output of processing jobs(if configured) in S3 is encrypted with customer managed master key in all the regions.

Remediation

Note: This control evaluates whether S3 output data of processing jobs is encrypted with Customer Managed Key for all regions together i.e. failure in any one region will fail the control. If any region is not configured for S3 output data encryption with Customer Managed Key then it is considered as region failure.

Using AWS Console:

1. Login to AWS Management console.
2. Navigate to Amazon Sagemaker console.
3. In the navigation pane, choose **Processing**, then choose **Processing jobs**.
4. Choose **Create Processing job**.

5. Under **Output data configuration**, type or select an existing Customer Managed Key ID/Key ARN from the dropdown below **Encryption key**.
6. After enabling S3 output data encryption, finish creating the Processing job.

Reference

- Notebook instances and SageMaker jobs
- create-processing-job

Control ID - 527: Ensure to encrypt the destination bucket in s3 in the audit logging account

Criticality: HIGH

Specification

Amazon Kinesis Data Firehose is a fully managed service for delivering real-time streaming data to destinations such as Amazon Simple Storage Service (Amazon S3), Amazon Redshift, Amazon OpenSearch Service, Splunk, and any custom HTTP endpoint or HTTP endpoints owned by supported third-party service providers. As part of Security Best Practices for Kinesis Data Firehose, it's recommended that data at rest and data in transit be encrypted in Kinesis Data Firehose.

Kinesis Data Firehose can compress records before delivering them to your S3 bucket. Compressed records can also be encrypted in the S3 bucket using an AWS Key Management Service (KMS) master key.

Rationale

As part of Security Best Practices for Kinesis Data Firehose it's recommended that data at rest and data in transit be encrypted in Kinesis Data Firehose.

Evaluation

This control ensures that AWS Kinesis firehose delivery stream data records delivered to destination S3 Buckets are encrypted with a KMS key.

Remediation

Using AWS Console:

1. Login to AWS Management console.
2. Navigate to Amazon Kinesis Dashboard.
3. In the left navigation panel, select **Delivery Streams**.
4. Click the **Kinesis Delivery Stream** instance to be remediated.
5. Select the **Configuration** tab.
6. In the **Destination settings** section, click the **Edit** button.
7. Expand the **Buffer hints, compression and encryption** section.
8. Select the **Enabled** option for the **Encryption for data records** field.
9. Select either AWS managed key or customer managed key option to encrypt resource.
10. Click **Save changes**.

Using AWS CLI:

Retrieve the delivery stream version-id and destination id:

```
aws firehose describe-delivery-stream --delivery-stream-name [delivery-stream-
name] \
    --query
[DeliveryStreamDescription.VersionId,DeliveryStreamDescription.Destinations[*]
.DestinationId]
```

Update the encryption configuration for the delivery stream destination:

```
aws firehose update-destination --delivery-stream-name [delivery-stream-name]
\
    --current-delivery-stream-version-id [current-delivery-stream-version-id]
\
    --destination-id [destination-id] \
    --extended-s3-destination-update '{"EncryptionConfiguration":
{"KMSEncryptionConfig": {"AWSKMSKeyARN": "[aws-kmz-key-arn]"}}}'
```

Reference

- Security Best Practices for Kinesis Data Firehose
- Data Protection in Amazon Kinesis Data Firehose

Control ID - 528: Ensure to encrypt the destination bucket in s3 with customer managed master keys in the audit logging account

Criticality: HIGH

Specification

Amazon Kinesis Data Firehose is a fully managed service for delivering real-time streaming data to destinations such as Amazon Simple Storage Service (Amazon S3), Amazon Redshift, Amazon OpenSearch Service, Splunk, and any custom HTTP endpoint or HTTP endpoints owned by supported third-party service providers.

Kinesis Data Firehose can compress records before delivering them to your S3 bucket. Compressed records can also be encrypted in the S3 bucket using an AWS Key Management Service (KMS) master key. As part of Security Best Practices for Kinesis Data Firehose, it's recommended that data at rest and data in transit be encrypted in Kinesis Data Firehose. Customer-managed keys are recommended because they offer greater flexibility to manage access control.

Rationale

As part of Security Best Practices for Kinesis Data Firehose it's recommended that data at rest and data in transit be encrypted in Kinesis Data Firehose. Customer-managed keys are recommended because they offer greater flexibility to manage access control.

Evaluation

This control ensures that AWS Kinesis Delivery Streams Data records delivered to destination S3 Buckets are encrypted with customer-managed KMS keys in the audit logging account in the regions.

Remediation

Using AWS Console:

1. Login to AWS Management console.
2. Navigate to Amazon Kinesis Dashboard.
3. In the left navigation panel, select **Delivery Streams**.
4. Click the **Kinesis Delivery Stream** instance to be remediated.
5. Select the **Configuration** tab.
6. In the **Destination settings** section, click the **Edit** button.
7. Expand the **Buffer hints, compression and encryption** section.
8. Ensure that the **Enabled** option is selected for the **Encryption for data records** field.
9. Select the **Use customer managed CMK** option for the **Encryption type** field.
10. Select a CMK from the dropdown. If there are no customer-managed keys available in the dropdown, create one by clicking the **Create key** option.
11. Click **Save changes**.

Using AWS CLI:

Retrieve the delivery stream version-id and destination id:

```
aws firehose describe-delivery-stream --delivery-stream-name [delivery-stream-
name] \
    --query
[DeliveryStreamDescription.VersionId,DeliveryStreamDescription.Destinations[*]
.DestinationId]
```

Update the encryption configuration for the delivery stream destination to use a customer-managed key:

```
aws firehose update-destination --delivery-stream-name [delivery-stream-name]
\
    --current-delivery-stream-version-id [current-delivery-stream-version-id]
\
    --destination-id [destination-id] \
    --extended-s3-destination-update '{"EncryptionConfiguration":
{"KMSEncryptionConfig": {"AWSKMSKeyARN": "[aws-kmz-key-arn]"}}}'
```

Reference

- Security Best Practices for Kinesis Data Firehose
- Data Protection in Amazon Kinesis Data Firehose

Control ID - 529: Ensure detailed monitoring is enabled for AWS Launch Configuration

Criticality: HIGH

Specification

Amazon EC2 can enable detailed monitoring when it is launching EC2 instances in your Auto Scaling group. You configure monitoring for Auto Scaling instances using a launch template or launch configuration. Monitoring is enabled whenever an instance is launched, either basic monitoring (five-minute granularity) or detailed monitoring (one-minute granularity).

Rationale

After you enable detailed monitoring, when you use this launch configuration, the Auto Scaling group can have associated scaling policies that scale on Amazon EC2 instance metrics with a 1-minute frequency.

Note: By default, basic monitoring is enabled when you use the AWS Management Console to create a launch template or launch configuration.

Evaluation

This control ensures that detailed monitoring is enabled for AWS Launch Configuration.

Remediation

Using AWS Console:

1. Login to AWS Management console.
2. Navigate to Amazon EC2 dashboard.
3. On the navigation pane, under **Auto Scaling**, choose **Launch Configurations**.
4. Choose **Create launch configuration**, and enter a name for your launch configuration.
5. For **Amazon machine image (AMI)**, choose an AMI.
6. For **Instance type**, select a hardware configuration for your instances.
7. In the **Additional configuration** section, select **Enable EC2 instance detailed monitoring within CloudWatch**.
8. Choose other configurations as per your requirement.
9. Click **Create launch configuration**.

Using AWS CLI:

```
aws autoscaling create-launch-configuration --launch-configuration-name
[launch-configuration-name]
    --image-id [ami-id]
    --instance-type [instance-type]
    --instance-monitoring Enabled=true
```

Note : We cannot modify existing AWS launch configuration

Reference

- Configure monitoring for Auto Scaling instances
- Create a launch configuration
- Monitor CloudWatch metrics for your Auto Scaling groups and instances

Control ID - 533: Ensure that ACM Certificate is validated

Criticality: MEDIUM

Specification

Certificates new/renewed has to be validated within 72 hours after the request is made, if not validated certificates will be invalid and new certificates have to requested which might cause interruption to applications/services.

Rationale

Based on the validaition method selected, certificates has to be validated for approval of issue/renewal requests which will make sure there will be no interruption to applications/services using those certificates.

Evaluation

This control ensures that ACM certificates are validated.

Remediation

Using AWS Console:

1. Sign in to AWS Console and navigate to https://console.aws.amazon.com/acm/home.
2. Click on the certificate to be remediated.
3. If the validation method selected during certificate creation is **Email validation**, then click on **Resend validation email** under **Domains** section.
4. On **Resend validation emails** page, select the domains to resend the validation email
5. This action will send email to domain registrant, administrative and technical contacts
6. Once approved by clicking **I Approve** link in the email sent, certificate will be renewed/issued
7. If the validation method selected during certificate creation is **DNS validation**, then click on **Export to CSV** under **Domains** section.
8. Downloaded csv will have key value **CNAME Records** to be added to your DNS configuration to validate that you control the domain to issue/renew ACM certificates

Using AWS CLI for certificate validation through Email validaition:

```
      aws acm resend-validation-email --certificate-arn [CERTIFICATE_ARN] --
domain [CERTIFICATE_DOMAIN] --validation-domain
[SUPER_DOMAIN/CERTIFICATE_DOMAIN]
```

For command usage refer: https://docs.aws.amazon.com/cli/latest/reference/acm/resend-validation-email.html

Note : To renew existing certificates, automatic renewal will be in place untill CNAME records are in place. If automatic renewal fails, have to follow above steps to renew the certificate before certificate becomes invalid.

Reference

- https://docs.aws.amazon.com/acm/latest/userguide/domain-ownership-validation.html