CIS Center for Internet Security®

CIS Benchmarks™

CIS Amazon Web Services Foundations Benchmark

v5.0.0 - 03-31-2025

Terms of Use

Please see the below link for our current terms of use:

https://www.cisecurity.org/cis-securesuite/cis-securesuite-membership-terms-of-use/

For information on referencing and/or citing CIS Benchmarks in 3rd party documentation (including using portions of Benchmark Recommendations) please contact CIS Legal (legalnotices@cisecurity.org) and request guidance on copyright usage.

NOTE: It is ***NEVER*** acceptable to host a CIS Benchmark in ***ANY*** format (PDF, etc.) on a 3rd party (non-CIS owned) site.

Table of Contents

Overview

All CIS Benchmarks™ (Benchmarks) focus on technical configuration settings used to maintain and/or increase the security of the addressed technology, and they should be used in **conjunction** with other essential cyber hygiene tasks like:

- Monitoring the base operating system and applications for vulnerabilities and quickly updating with the latest security patches.
- End-point protection (Antivirus software, Endpoint Detection and Response (EDR), etc.).
- Logging and monitoring user and system activity.

In the end, the Benchmarks are designed to be a key **component** of a comprehensive cybersecurity program.

Important Usage Information

All Benchmarks are available free for non-commercial use from the CIS Website. They can be used to manually assess and remediate systems and applications. In lieu of manual assessment and remediation, there are several tools available to assist with assessment:

- CIS Configuration Assessment Tool (CIS-CAT® Pro Assessor)
- CIS Benchmarks™ Certified 3rd Party Tooling

These tools make the hardening process much more scalable for large numbers of systems and applications.

NOTE: Some tooling focuses only on the Benchmark Recommendations that can be fully automated (skipping ones marked **Manual**). It is important that ***ALL*** Recommendations (**Automated** and **Manual**) be addressed since all are important for properly securing systems and are typically in scope for audits.

Key Stakeholders

Cybersecurity is a collaborative effort, and cross functional cooperation is imperative within an organization to discuss, test, and deploy Benchmarks in an effective and efficient way. The Benchmarks are developed to be best practice configuration guidelines applicable to a wide range of use cases. In some organizations, exceptions to specific Recommendations will be needed, and this team should work to prioritize the problematic Recommendations based on several factors like risk, time, cost, and labor. These exceptions should be properly categorized and documented for auditing purposes.

Apply the Correct Version of a Benchmark

Benchmarks are developed and tested for a specific set of products and versions and applying an incorrect Benchmark to a system can cause the resulting pass/fail score to be incorrect. This is due to the assessment of settings that do not apply to the target systems. To assure the correct Benchmark is being assessed:

- **Deploy the Benchmark applicable to the way settings are managed in the environment:** An example of this is the Microsoft Windows family of Benchmarks, which have separate Benchmarks for Group Policy, Intune, and Stand-alone systems based upon how system management is deployed. Applying the wrong Benchmark in this case will give invalid results.

- **Use the most recent version of a Benchmark**: This is true for all Benchmarks, but especially true for cloud technologies. Cloud technologies change frequently and using an older version of a Benchmark may have invalid methods for auditing and remediation.

Exceptions

The guidance items in the Benchmarks are called recommendations and not requirements, and exceptions to some of them are expected and acceptable. The Benchmarks strive to be a secure baseline, or starting point, for a specific technology, with known issues identified during Benchmark development are documented in the Impact section of each Recommendation. In addition, organizational, system specific requirements, or local site policy may require changes as well, or an exception to a Recommendation or group of Recommendations (e.g. A Benchmark could Recommend that a Web server not be installed on the system, but if a system's primary purpose is to function as a Webserver, there should be a documented exception to this Recommendation for that specific server).

In the end, exceptions to some Benchmark Recommendations are common and acceptable, and should be handled as follows:

- The reasons for the exception should be reviewed cross-functionally and be well documented for audit purposes.
- A plan should be developed for mitigating, or eliminating, the exception in the future, if applicable.
- If the organization decides to accept the risk of this exception (not work toward mitigation or elimination), this should be documented for audit purposes.

It is the responsibility of the organization to determine their overall security policy, and which settings are applicable to their unique needs based on the overall risk profile for the organization.

Remediation

CIS has developed Build Kits for many technologies to assist in the automation of hardening systems. Build Kits are designed to correspond to Benchmark's "Remediation" section, which provides the manual remediation steps necessary to make that Recommendation compliant to the Benchmark.

When remediating systems (changing configuration settings on deployed systems as per the Benchmark's Recommendations), please approach this with caution and test thoroughly.

The following is a reasonable remediation approach to follow:

- CIS Build Kits, or internally developed remediation methods should never be applied to production systems without proper testing.
- Proper testing consists of the following:
  - Understand the configuration (including installed applications) of the targeted systems. Various parts of the organization may need different configurations (e.g., software developers vs standard office workers).
  - Read the Impact section of the given Recommendation to help determine if there might be an issue with the targeted systems.
  - Test the configuration changes with representative lab system(s). If issues arise during testing, they can be resolved prior to deploying to any production systems.
  - When testing is complete, initially deploy to a small sub-set of production systems and monitor closely for issues. If there are issues, they can be resolved prior to deploying more broadly.
  - When the initial deployment above is completes successfully, iteratively deploy to additional systems and monitor closely for issues. Repeat this process until the full deployment is complete.

Summary

Using the Benchmarks Certified tools, working as a team with key stakeholders, being selective with exceptions, and being careful with remediation deployment, it is possible to harden large numbers of deployed systems in a cost effective, efficient, and safe manner.

NOTE: As previously stated, the PDF versions of the CIS Benchmarks™ are available for free, non-commercial use on the CIS Website. All other formats of the CIS Benchmarks™ (MS Word, Excel, and Build Kits) are available for CIS SecureSuite® members.

CIS-CAT® Pro is also available to CIS SecureSuite® members.

Target Technology Details

This document provides prescriptive guidance for configuring security options for a subset of Amazon Web Services with an emphasis on foundational, testable, and architecture agnostic settings. Some of the specific Amazon Web Services in scope for this document include:

- AWS Identity and Access Management (IAM)
- IAM Access Analyzer
- AWS Config
- AWS CloudTrail
- AWS CloudWatch
- AWS Simple Notification Service (SNS)
- AWS Simple Storage Service (S3)
- Elastic Compute Cloud (EC2)
- Relational Database Service (RDS)
- AWS VPC

To obtain the latest version of this guide, please visit https://benchmarks.cisecurity.org. If you have questions, comments, or have identified ways to improve this guide, please write us at BenchmarkInfo@cisecurity.org.

Intended Audience

This document is intended for system and application administrators, security specialists, auditors, help desk, platform deployment, and/or DevOps personnel who plan to develop, deploy, assess, or secure solutions in Amazon Web Services.

Consensus Guidance

This CIS Benchmark™ was created using a consensus review process comprised of a global community of subject matter experts. The process combines real world experience with data-based information to create technology specific guidance to assist users to secure their environments. Consensus participants provide perspective from a diverse set of backgrounds including consulting, software development, audit and compliance, security research, operations, government, and legal.

Each CIS Benchmark undergoes two phases of consensus review. The first phase occurs during initial Benchmark development. During this phase, subject matter experts convene to discuss, create, and test working drafts of the Benchmark. This discussion occurs until consensus has been reached on Benchmark recommendations. The second phase begins after the Benchmark has been published. During this phase, all feedback provided by the Internet community is reviewed by the consensus team for incorporation in the Benchmark. If you are interested in participating in the consensus process, please visit https://workbench.cisecurity.org/.

Typographical Conventions

The following typographical conventions are used throughout this guide:

Convention	Meaning
`Stylized Monospace font`	Used for blocks of code, command, and script examples. Text should be interpreted exactly as presented.
`Monospace font`	Used for inline code, commands, UI/Menu selections or examples. Text should be interpreted exactly as presented.
`<Monospace font in brackets>`	Text set in angle brackets denote a variable requiring substitution for a real value.
Italic font	Used to reference other relevant settings, CIS Benchmarks and/or Benchmark Communities. Also, used to denote the title of a book, article, or other publication.
Bold font	Additional information or caveats things like **Notes**, **Warnings**, or **Cautions** (usually just the word itself and the rest of the text normal).

Recommendation Definitions

The following defines the various components included in a CIS recommendation as applicable. If any of the components are not applicable it will be noted, or the component will not be included in the recommendation.

Title

Concise description for the recommendation's intended configuration.

Assessment Status

An assessment status is included for every recommendation. The assessment status indicates whether the given recommendation can be automated or requires manual steps to implement. Both statuses are equally important and are determined and supported as defined below:

Automated

Represents recommendations for which assessment of a technical control can be fully automated and validated to a pass/fail state. Recommendations will include the necessary information to implement automation.

Manual

Represents recommendations for which assessment of a technical control cannot be fully automated and requires all or some manual steps to validate that the configured state is set as expected. The expected state can vary depending on the environment.

Profile

A collection of recommendations for securing a technology or a supporting platform. Most benchmarks include at least a Level 1 and Level 2 Profile. Level 2 extends Level 1 recommendations and is not a standalone profile. The Profile Definitions section in the benchmark provides the definitions as they pertain to the recommendations included for the technology.

Description

Detailed information pertaining to the setting with which the recommendation is concerned. In some cases, the description will include the recommended value.

Rationale Statement

Detailed reasoning for the recommendation to provide the user a clear and concise understanding on the importance of the recommendation.

Impact Statement

Any security, functionality, or operational consequences that can result from following the recommendation.

Audit Procedure

Systematic instructions for determining if the target system complies with the recommendation.

Remediation Procedure

Systematic instructions for applying recommendations to the target system to bring it into compliance according to the recommendation.

Default Value

Default value for the given setting in this recommendation, if known. If not known, either not configured or not defined will be applied.

References

Additional documentation relative to the recommendation.

CIS Critical Security Controls® (CIS Controls®)

The mapping between a recommendation and the CIS Controls is organized by CIS Controls version, Safeguard, and Implementation Group (IG). The Benchmark in its entirety addresses the CIS Controls safeguards of (v7) “5.1 - Establish Secure Configurations” and (v8) '4.1 - Establish and Maintain a Secure Configuration Process” so individual recommendations will not be mapped to these safeguards.

Additional Information

Supplementary information that does not correspond to any other field but may be useful to the user.

Profile Definitions

The following configuration profiles are defined by this Benchmark:

- **Level 1**

  Items in this profile intend to:

  - be practical and prudent;
  - provide security focused best practice hardening of a technology; and
  - limit impact to the utility of the technology beyond acceptable means.

- **Level 2**

  This profile extends the "Level 1" profile. Items in this profile exhibit one or more of the following characteristics:

  - are intended for environments or use cases where security is more critical than manageability and usability
  - acts as defense in depth measure
  - may impact the utility or performance of the technology
  - may include additional licensing, cost, or addition of third party software

Acknowledgements

This Benchmark exemplifies the great things a community of users, vendors, and subject matter experts can accomplish through consensus collaboration. The CIS community thanks the entire consensus team with special recognition to the following individuals who contributed greatly to the creation of this guide:

Michael Wicks, Chantel Duckworth

Contributor
Amol Pathak
Rob Witoff
John Martinez
Darwin Sanoy
Ionut Dragoi
John Robel
Mike Wicks
Jeremy Phillips
Maril Vernon
Paul Campbell
Ankit Rao
Steve Laino
Lawrence Sica
Nick Gibbon
Lewis Hardy
Logan McMillan
Darren Joyce
Bhushan Bhat
Sagar Chhatrala
Nirbhay Kumar
Ian McRee
Jason Kao
Cody Bruno
Lawrence Grim
SnowWolf Wagner
Gareth Boyes
Chantel Duckworth
Austin Songer
Zan Liffick
Raphael Durner
Vidush Kandpal

Editor
Iben Rodriguez
Gregory Carpenter
Zan Liffick
Rachel Rice

Recommendations

1 Identity and Access Management

This section contains recommendations for configuring identity and access management related options.

1.1 Maintain current contact details (Manual)

Profile Applicability:

- Level 1

Description:

Ensure contact email and telephone details for AWS accounts are current and map to more than one individual in your organization.

An AWS account supports a number of contact details, and AWS will use these to contact the account owner if activity judged to be in breach of the Acceptable Use Policy or indicative of a likely security compromise is observed by the AWS Abuse team. Contact details should not be for a single individual, as circumstances may arise where that individual is unavailable. Email contact details should point to a mail alias which forwards email to multiple individuals within the organization; where feasible, phone contact details should point to a PABX hunt group or other call-forwarding system.

Rationale:

If an AWS account is observed to be behaving in a prohibited or suspicious manner, AWS will attempt to contact the account owner by email and phone using the contact details listed. If this is unsuccessful and the account behavior needs urgent mitigation, proactive measures may be taken, including throttling of traffic between the account exhibiting suspicious behavior and the AWS API endpoints and the Internet. This will result in impaired service to and from the account in question, so it is in both the customers' and AWS's best interests that prompt contact can be established. This is best achieved by setting AWS account contact details to point to resources which have multiple individuals as recipients, such as email aliases and PABX hunt groups.

Audit:

This activity can only be performed via the AWS Console, with a user who has permission to read and write Billing information (aws-portal:*Billing).

1. Sign in to the AWS Management Console and open the `Billing and Cost Management` console at https://console.aws.amazon.com/billing/home#/.
2. On the navigation bar, choose your account name, and then choose `Account`.
3. On the `Account Settings` page, review and verify the current details.
4. Under `Contact Information`, review and verify the current details.

Remediation:

This activity can only be performed via the AWS Console, with a user who has permission to read and write Billing information (aws-portal:*Billing).

1. Sign in to the AWS Management Console and open the `Billing and Cost Management` console at https://console.aws.amazon.com/billing/home#/.
2. On the navigation bar, choose your account name, and then choose `Account`.
3. On the `Account Settings` page, next to `Account Settings`, choose `Edit`.
4. Next to the field that you need to update, choose `Edit`.
5. After you have entered your changes, choose `Save changes`.
6. After you have made your changes, choose `Done`.
7. To edit your contact information, under `Contact Information`, choose `Edit`.
8. For the fields that you want to change, type your updated information, and then choose `Update`.

References:

1. https://docs.aws.amazon.com/awsaccountbilling/latest/aboutv2/manage-account-payment.html#contact-info

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	17.2 Establish and Maintain Contact Information for Reporting Security Incidents Establish and maintain contact information for parties that need to be informed of security incidents. Contacts may include internal staff, third-party vendors, law enforcement, cyber insurance providers, relevant government agencies, Information Sharing and Analysis Center (ISAC) partners, or other stakeholders. Verify contacts annually to ensure that information is up-to-date.	●	●	●
v7	19.3 Designate Management Personnel to Support Incident Handling Designate management personnel, as well as backups, who will support the incident handling process by acting in key decision-making roles.	●	●	●

1.2 Ensure security contact information is registered (Manual)

Profile Applicability:

- Level 1

Description:

AWS provides customers with the option of specifying the contact information for account's security team. It is recommended that this information be provided.

Rationale:

Specifying security-specific contact information will help ensure that security advisories sent by AWS reach the team in your organization that is best equipped to respond to them.

Audit:

Perform the following to determine if security contact information is present:
From Console:

1. Click on your account name at the top right corner of the console
2. From the drop-down menu Click `My Account`
3. Scroll down to the `Alternate Contacts` section
4. Ensure contact information is specified in the `Security` section

From Command Line:

1. Run the following command:

```
aws account get-alternate-contact --alternate-contact-type SECURITY
```

2. Ensure proper contact information is specified for the `Security` contact.

Remediation:

Perform the following to establish security contact information:
From Console:

1. Click on your account name at the top right corner of the console.
2. From the drop-down menu Click `My Account`
3. Scroll down to the `Alternate Contacts` section
4. Enter contact information in the `Security` section

From Command Line:
Run the following command with the following input parameters:
--email-address, --name, and --phone-number.

```
aws account put-alternate-contact --alternate-contact-type SECURITY
```

Note: Consider specifying an internal email distribution list to ensure emails are regularly monitored by more than one individual.

References:

1. CCE-79200-2

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	17.2 Establish and Maintain Contact Information for Reporting Security Incidents Establish and maintain contact information for parties that need to be informed of security incidents. Contacts may include internal staff, third-party vendors, law enforcement, cyber insurance providers, relevant government agencies, Information Sharing and Analysis Center (ISAC) partners, or other stakeholders. Verify contacts annually to ensure that information is up-to-date.	●	●	●
v8	17.6 Define Mechanisms for Communicating During Incident Response Determine which primary and secondary mechanisms will be used to communicate and report during a security incident. Mechanisms can include phone calls, emails, or letters. Keep in mind that certain mechanisms, such as emails, can be affected during a security incident. Review annually, or when significant enterprise changes occur that could impact this Safeguard.		●	●
v7	19 Incident Response and Management Incident Response and Management			
v7	19.2 Assign Job Titles and Duties for Incident Response Assign job titles and duties for handling computer and network incidents to specific individuals and ensure tracking and documentation throughout the incident through resolution.		●	●

1.3 Ensure no 'root' user account access key exists (Automated)

Profile Applicability:

- Level 1

Description:

The 'root' user account is the most privileged user in an AWS account. AWS Access Keys provide programmatic access to a given AWS account. It is recommended that all access keys associated with the 'root' user account be deleted.

Rationale:

Deleting access keys associated with the 'root' user account limits vectors by which the account can be compromised. Additionally, deleting the 'root' access keys encourages the creation and use of role based accounts that are least privileged.

Audit:

Perform the following to determine if the 'root' user account has access keys:
From Console:

1. Login to the AWS Management Console.
2. Click `Services`.
3. Click `IAM`.
4. Click on `Credential Report`.
5. This will download a `.csv` file which contains credential usage for all IAM users within an AWS Account - open this file.
6. For the `<root_account>` user, ensure the `access_key_1_active` and `access_key_2_active` fields are set to `FALSE`.

From Command Line:
Run the following command:

```
aws iam get-account-summary | grep "AccountAccessKeysPresent"
```

If no 'root' access keys exist the output will show `"AccountAccessKeysPresent": 0,`.
If the output shows a "1", then 'root' keys exist and should be deleted.

Remediation:

Perform the following to delete active 'root' user access keys.
From Console:

1. Sign in to the AWS Management Console as 'root' and open the IAM console at https://console.aws.amazon.com/iam/.
2. Click on `<root_account>` at the top right and select `My Security Credentials` from the drop down list.
3. On the pop out screen Click on `Continue to Security Credentials`.

4. Click on `Access Keys` (Access Key ID and Secret Access Key).
5. If there are active keys, under `Status`, click `Delete` (Note: Deleted keys cannot be recovered).

Note: While a key can be made inactive, this inactive key will still show up in the CLI command from the audit procedure, and may lead to the root user being falsely flagged as being non-compliant.

References:

1. http://docs.aws.amazon.com/general/latest/gr/aws-access-keys-best-practices.html
2. http://docs.aws.amazon.com/general/latest/gr/managing-aws-access-keys.html
3. http://docs.aws.amazon.com/IAM/latest/APIReference/API_GetAccountSummary.html
4. CCE-78910-7
5. https://aws.amazon.com/blogs/security/an-easier-way-to-determine-the-presence-of-aws-account-access-keys/

Additional Information:

- IAM User account "root" for us-gov cloud regions is not enabled by default. However, on request to AWS support enables 'root' access only through access-keys (CLI, API methods) for us-gov cloud region.
- Implement regular checks and alerts for any creation of new root access keys to promptly address any unauthorized or accidental creation.

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	5.4 Restrict Administrator Privileges to Dedicated Administrator Accounts Restrict administrator privileges to dedicated administrator accounts on enterprise assets. Conduct general computing activities, such as internet browsing, email, and productivity suite use, from the user's primary, non-privileged account.	●	●	●
v8	6.8 Define and Maintain Role-Based Access Control Define and maintain role-based access control, through determining and documenting the access rights necessary for each role within the enterprise to successfully carry out its assigned duties. Perform access control reviews of enterprise assets to validate that all privileges are authorized, on a recurring schedule at a minimum annually, or more frequently.			●
v7	4.3 Ensure the Use of Dedicated Administrative Accounts Ensure that all users with administrative account access use a dedicated or secondary account for elevated activities. This account should only be used for administrative activities and not internet browsing, email, or similar activities.	●	●	●

1.4 Ensure MFA is enabled for the 'root' user account (Automated)

Profile Applicability:

- Level 1

Description:

The 'root' user account is the most privileged user in an AWS account. Multi-factor Authentication (MFA) adds an extra layer of protection on top of a username and password. With MFA enabled, when a user signs in to an AWS website, they will be prompted for their username and password as well as for an authentication code from their AWS MFA device.

Note: When virtual MFA is used for 'root' accounts, it is recommended that the device used is NOT a personal device, but rather a dedicated mobile device (tablet or phone) that is kept charged and secured, independent of any individual personal devices ("non-personal virtual MFA"). This lessens the risks of losing access to the MFA due to device loss, device trade-in, or if the individual owning the device is no longer employed at the company.

Where an AWS Organization is using centralized root access, root credentials can be removed from member accounts. In that case it is neither possible nor necessary to configure root MFA in the member account.

Rationale:

Enabling MFA provides increased security for console access as it requires the authenticating principal to possess a device that emits a time-sensitive key and have knowledge of a credential.

Audit:

Perform the following to determine if the 'root' user account is enabled and has MFA setup:
From Console:

1. Login to the AWS Management Console
2. Click `Services`
3. Click `IAM`
4. Click on `Credential Report`
5. This will download a `.csv` file which contains credential usage for all IAM users within an AWS Account - open this file
6. For the `<root_account>` user, ensure the `mfa_active` field is set to `TRUE` or the `password_enabled` field is set to `FALSE`

From Command Line:

1. Run the following command:

```
aws iam get-account-summary | grep "AccountMFAEnabled"
aws iam get-account-summary | grep "AccountPasswordPresent"
```

2. Ensure the AccountMFAEnabled property is set to 1 or the AccountPasswordPresent property is set to 0

Remediation:

Note: To manage MFA devices for the 'root' AWS account, you must use your 'root' account credentials to sign in to AWS. You cannot manage MFA devices for the 'root' account using other credentials.
Perform the following to establish MFA for the 'root' user account:

1. Sign in to the AWS Management Console and open the IAM console at https://console.aws.amazon.com/iam/.
2. Choose `Dashboard` , and under `Security Status` , expand `Activate MFA` on your root account.
3. Choose `Activate MFA`
4. In the wizard, choose `A virtual MFA` device and then choose `Next Step` .
5. IAM generates and displays configuration information for the virtual MFA device, including a QR code graphic. The graphic is a representation of the 'secret configuration key' that is available for manual entry on devices that do not support QR codes.
6. Open your virtual MFA application. (For a list of apps that you can use for hosting virtual MFA devices, see Virtual MFA Applications.) If the virtual MFA application supports multiple accounts (multiple virtual MFA devices), choose the option to create a new account (a new virtual MFA device).
7. Determine whether the MFA app supports QR codes, and then do one of the following:
    - Use the app to scan the QR code. For example, you might choose the camera icon or choose an option similar to Scan code, and then use the device's camera to scan the code.
    - In the Manage MFA Device wizard, choose Show secret key for manual configuration, and then type the secret configuration key into your MFA application.

When you are finished, the virtual MFA device starts generating one-time passwords.
In the Manage MFA Device wizard, in the Authentication Code 1 box, type the one-time password that currently appears in the virtual MFA device. Wait up to 30 seconds for the device to generate a new one-time password. Then type the second one-time password into the Authentication Code 2 box. Choose Assign Virtual MFA.

References:

1. CCE-78911-5

2. https://docs.aws.amazon.com/IAM/latest/UserGuide/id_root-user.html#id_root-user_manage_mfa
3. https://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_mfa_enable_virtual.html#enable-virt-mfa-for-root
4. https://docs.aws.amazon.com/IAM/latest/UserGuide/id_root-enable-root-access.html

Additional Information:

IAM User account "root" for us-gov cloud regions does not have console access. This recommendation is not applicable for us-gov cloud regions.

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	6.5 Require MFA for Administrative Access Require MFA for all administrative access accounts, where supported, on all enterprise assets, whether managed on-site or through a third-party provider.	●	●	●
v7	4.5 Use Multifactor Authentication For All Administrative Access Use multi-factor authentication and encrypted channels for all administrative account access.		●	●

1.5 Ensure hardware MFA is enabled for the 'root' user account (Manual)

Profile Applicability:

- Level 2

Description:

The 'root' user account is the most privileged user in an AWS account. MFA adds an extra layer of protection on top of a username and password. With MFA enabled, when a user signs in to an AWS website, they will be prompted for their username and password as well as for an authentication code from their AWS MFA device. For Level 2, it is recommended that the 'root' user account be protected with a hardware MFA.

Where an AWS Organization is using centralized root access, root credentials can be removed from member accounts. In that case it is neither possible nor necessary to configure root MFA in the member account.

Rationale:

A hardware MFA has a smaller attack surface than a virtual MFA. For example, a hardware MFA does not suffer the attack surface introduced by the mobile smartphone on which a virtual MFA resides.

Note: Using hardware MFA for numerous AWS accounts may create a logistical device management issue. If this is the case, consider implementing this Level 2 recommendation selectively for the highest security AWS accounts, while applying the Level 1 recommendation to the remaining accounts.

Audit:

Perform the following to determine if the 'root' user account has a hardware MFA setup:

1. Run the following command to determine if the 'root' account has MFA setup:

```
aws iam get-account-summary | grep "AccountMFAEnabled"
aws iam get-account-summary | grep "AccountPasswordPresent"
```

The `AccountMFAEnabled` property is set to `1` will ensure that the 'root' user account has MFA (Virtual or Hardware) Enabled. `AccountPasswordPresent` set to `0` indicates that the `root` console credential has been removed.
If `AccountMFAEnabled` property is set to `0` and `AccountPasswordPresent` is set to `1` the account is not compliant with this recommendation.

2. If `AccountMFAEnabled` property is set to `1`, determine 'root' account has Hardware MFA enabled.
   Run the following command to list all virtual MFA devices:

```
aws iam list-virtual-mfa-devices
```

If the output contains one MFA with the following Serial Number, it means the MFA is virtual, not hardware and the account is not compliant with this recommendation:
`"SerialNumber": "arn:aws:iam::_<aws_account_number>_:mfa/root-account-mfa-device"`

Remediation:

Note: To manage MFA devices for the AWS 'root' user account, you must use your 'root' account credentials to sign in to AWS. You cannot manage MFA devices for the 'root' account using other credentials.
Perform the following to establish a hardware MFA for the 'root' user account:

1. Sign in to the AWS Management Console and open the IAM console at https://console.aws.amazon.com/iam/.
2. Choose `Dashboard`, and under `Security Status`, expand `Activate MFA` on your root account.
3. Choose `Activate MFA`.
4. In the wizard, choose `A hardware MFA` device and then choose `Next Step`.
5. In the `Serial Number` box, enter the serial number that is found on the back of the MFA device.
6. In the `Authentication Code 1` box, enter the six-digit number displayed by the MFA device. You might need to press the button on the front of the device to display the number.
7. Wait 30 seconds while the device refreshes the code, and then enter the next six-digit number into the `Authentication Code 2` box. You might need to press the button on the front of the device again to display the second number.
8. Choose `Next Step`. The MFA device is now associated with the AWS account. The next time you use your AWS account credentials to sign in, you must type a code from the hardware MFA device.

Remediation for this recommendation is not available through AWS CLI.

References:

1. CCE-78911-5
2. https://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_mfa_enable_virtual.html
3. https://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_mfa_enable_physical.html#enable-hw-mfa-for-root
4. https://docs.aws.amazon.com/IAM/latest/UserGuide/id_root-enable-root-access.html

Additional Information:

IAM User account 'root' for us-gov cloud regions does not have console access. This control is not applicable for us-gov cloud regions.

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	6.5 Require MFA for Administrative Access Require MFA for all administrative access accounts, where supported, on all enterprise assets, whether managed on-site or through a third-party provider.	●	●	●
v7	4.5 Use Multifactor Authentication For All Administrative Access Use multi-factor authentication and encrypted channels for all administrative account access.		●	●

1.6 Eliminate use of the 'root' user for administrative and daily tasks (Manual)

Profile Applicability:

- Level 1

Description:

With the creation of an AWS account, a 'root user' is created that cannot be disabled or deleted. That user has unrestricted access to and control over all resources in the AWS account. It is highly recommended that the use of this account be avoided for everyday tasks.

Rationale:

The 'root user' has unrestricted access to and control over all account resources. Use of it is inconsistent with the principles of least privilege and separation of duties, and can lead to unnecessary harm due to error or account compromise.

Audit:

From Console:

1. Login to the AWS Management Console at `https://console.aws.amazon.com/iam/`.
2. In the left pane, click `Credential Report`.
3. Click on `Download Report`.
4. Open or Save the file locally.
5. Locate the `<root account>` under the user column.
6. Review `password_last_used, access_key_1_last_used_date, access_key_2_last_used_date` to determine when the 'root user' was last used.

From Command Line:
Run the following CLI commands to provide a credential report for determining the last time the 'root user' was used:

```
aws iam generate-credential-report
aws iam get-credential-report --query 'Content' --output text | base64 -d |
cut -d, -f1,5,11,16 | grep -B1 '<root_account>'
```

Review `password_last_used`, `access_key_1_last_used_date`, `access_key_2_last_used_date` to determine when the *root user* was last used.
Note: There are a few conditions under which the use of the 'root' user account is required. Please see the reference links for all of the tasks that require use of the 'root' user.

Remediation:

If you find that the 'root' user account is being used for daily activities, including administrative tasks that do not require the 'root' user:

1. Change the 'root' user password.
2. Deactivate or delete any access keys associated with the 'root' user.

Remember, anyone who has 'root' user credentials for your AWS account has unrestricted access to and control of all the resources in your account, including billing information.

References:

1. https://docs.aws.amazon.com/IAM/latest/UserGuide/best-practices.html
2. https://docs.aws.amazon.com/IAM/latest/UserGuide/id_root-user.html
3. https://docs.aws.amazon.com/general/latest/gr/aws_tasks-that-require-root.html

Additional Information:

The 'root' user for us-gov cloud regions is not enabled by default. However, on request to AWS support, they can enable the 'root' user and grant access only through access-keys (CLI, API methods) for us-gov cloud region. If the 'root' user for us-gov cloud regions is enabled, this recommendation is applicable.

Monitoring usage of the 'root' user can be accomplished by implementing recommendation 3.3 Ensure a log metric filter and alarm exist for usage of the 'root' user.

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	5.4 Restrict Administrator Privileges to Dedicated Administrator Accounts Restrict administrator privileges to dedicated administrator accounts on enterprise assets. Conduct general computing activities, such as internet browsing, email, and productivity suite use, from the user's primary, non-privileged account.	●	●	●
v7	4.3 Ensure the Use of Dedicated Administrative Accounts Ensure that all users with administrative account access use a dedicated or secondary account for elevated activities. This account should only be used for administrative activities and not internet browsing, email, or similar activities.	●	●	●

1.7 Ensure IAM password policy requires minimum length of 14 or greater (Automated)

Profile Applicability:

- Level 1

Description:

Password policies are, in part, used to enforce password complexity requirements. IAM password policies can be used to ensure passwords are at least a given length. It is recommended that the password policy require a minimum password length 14.

Rationale:

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

Impact:

Enforcing a minimum password length of 14 characters enhances security by making passwords more resistant to brute force attacks. However, it may require users to create longer and potentially more complex passwords, which could impact user convenience.

Audit:

Perform the following to ensure the password policy is configured as prescribed:
From 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. Ensure "Minimum password length" is set to 14 or greater.

From Command Line:

```
aws iam get-account-password-policy
```

Ensure the output of the above command includes "MinimumPasswordLength": 14 (or higher)

Remediation:

Perform the following to set the password policy as prescribed:
From 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. Set "Minimum password length" to 14 or greater.
5. Click "Apply password policy"

From Command Line:

```
aws iam update-account-password-policy --minimum-password-length 14
```

Note: All commands starting with "aws iam update-account-password-policy" can be combined into a single command.

References:

1. CCE-78907-3
2. https://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_passwords_account-policy.html
3. https://docs.aws.amazon.com/IAM/latest/UserGuide/best-practices.html#configure-strong-password-policy

Additional Information:

Ensure the password policy also includes requirements for password complexity, such as the inclusion of uppercase letters, lowercase letters, numbers, and special characters:

```
aws iam update-account-password-policy --require-uppercase-characters --
require-lowercase-characters --require-numbers --require-symbols
```

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	5.2 Use Unique Passwords Use unique passwords for all enterprise assets. Best practice implementation includes, at a minimum, an 8-character password for accounts using MFA and a 14-character password for accounts not using MFA.	●	●	●
v7	16.1 Maintain an Inventory of Authentication Systems Maintain an inventory of each of the organization's authentication systems, including those located onsite or at a remote service provider.		●	●

1.8 Ensure IAM password policy prevents password reuse (Automated)

Profile Applicability:

- Level 1

Description:

IAM password policies can prevent the reuse of a given password by the same user. It is recommended that the password policy prevent the reuse of passwords.

Rationale:

Preventing password reuse increases account resiliency against brute force login attempts.

Audit:

Perform the following to ensure the password policy is configured as prescribed:
From 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. Ensure "Prevent password reuse" is checked
5. Ensure "Number of passwords to remember" is set to 24

From Command Line:

```
aws iam get-account-password-policy
```

Ensure the output of the above command includes "PasswordReusePrevention": 24

Remediation:

Perform the following to set the password policy as prescribed:
From 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. Check "Prevent password reuse"
5. Set "Number of passwords to remember" is set to 24

From Command Line:

```
aws iam update-account-password-policy --password-reuse-prevention 24
```

Note: All commands starting with "aws iam update-account-password-policy" can be combined into a single command.

References:

1. CCE-78908-1
2. https://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_passwords_account-policy.html
3. https://docs.aws.amazon.com/IAM/latest/UserGuide/best-practices.html#configure-strong-password-policy

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	5.2 Use Unique Passwords Use unique passwords for all enterprise assets. Best practice implementation includes, at a minimum, an 8-character password for accounts using MFA and a 14-character password for accounts not using MFA.	●	●	●
v7	4.4 Use Unique Passwords Where multi-factor authentication is not supported (such as local administrator, root, or service accounts), accounts will use passwords that are unique to that system.		●	●

1.9 Ensure multi-factor authentication (MFA) is enabled for all IAM users that have a console password (Automated)

Profile Applicability:

- Level 1

Description:

Multi-Factor Authentication (MFA) adds an extra layer of authentication assurance beyond traditional credentials. With MFA enabled, when a user signs in to the AWS Console, they will be prompted for their user name and password as well as for an authentication code from their physical or virtual MFA token. It is recommended that MFA be enabled for all accounts that have a console password.

Rationale:

Enabling MFA provides increased security for console access as it requires the authenticating principal to possess a device that displays a time-sensitive key and have knowledge of a credential.

Impact:

AWS will soon end support for SMS multi-factor authentication (MFA). New customers are not allowed to use this feature. We recommend that existing customers switch to an alternative method of MFA.

Audit:

Perform the following to determine if a MFA device is enabled for all IAM users having a console password:

From Console:

1. Open the IAM console at https://console.aws.amazon.com/iam/.
2. In the left pane, select `Users`
3. If the `MFA` or `Password age` columns are not visible in the table, click the gear icon at the upper right corner of the table and ensure a checkmark is next to both, then click `Close`.
4. Ensure that for each user where the `Password age` column shows a password age, the `MFA` column shows `Virtual`, `U2F Security Key`, or `Hardware`.

From Command Line:

1. Run the following command (OSX/Linux/UNIX) to generate a list of all IAM users along with their password and MFA status:

```
  aws iam generate-credential-report
  aws iam get-credential-report --query 'Content' --output text | base64 -d |
cut -d, -f1,4,8
```

2. The output of this command will produce a table similar to the following:

```
  user,password_enabled,mfa_active
  elise,false,false
  brandon,true,true
  rakesh,false,false
  helene,false,false
  paras,true,true
  anitha,false,false
```

3. For any column having `password_enabled` set to `true` , ensure `mfa_active` is also set to `true.`

Remediation:

Perform the following to enable MFA:
From Console:

1. Sign in to the AWS Management Console and open the IAM console at 'https://console.aws.amazon.com/iam/'
2. In the left pane, select `Users`.
3. In the `User Name` list, choose the name of the intended MFA user.
4. Choose the `Security Credentials` tab, and then choose `Manage MFA Device`.
5. In the `Manage MFA Device wizard`, choose `Virtual MFA` device, and then choose `Continue`.

IAM generates and displays configuration information for the virtual MFA device, including a QR code graphic. The graphic is a representation of the 'secret configuration key' that is available for manual entry on devices that do not support QR codes.

6. Open your virtual MFA application. (For a list of apps that you can use for hosting virtual MFA devices, see Virtual MFA Applications at https://aws.amazon.com/iam/details/mfa/#Virtual_MFA_Applications). If the virtual MFA application supports multiple accounts (multiple virtual MFA devices), choose the option to create a new account (a new virtual MFA device).
7. Determine whether the MFA app supports QR codes, and then do one of the following:

- Use the app to scan the QR code. For example, you might choose the camera icon or choose an option similar to Scan code, and then use the device's camera to scan the code.

- In the Manage MFA Device wizard, choose Show secret key for manual configuration, and then type the secret configuration key into your MFA application.

When you are finished, the virtual MFA device starts generating one-time passwords.

8. In the `Manage MFA Device wizard`, in the `MFA Code 1 box`, type the `one-time password` that currently appears in the virtual MFA device. Wait up to 30 seconds for the device to generate a new one-time password. Then type the second `one-time password` into the `MFA Code 2 box`.
9. Click `Assign MFA`.

References:

1. https://tools.ietf.org/html/rfc6238
2. https://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_mfa.html
3. https://docs.aws.amazon.com/IAM/latest/UserGuide/best-practices.html#enable-mfa-for-privileged-users
4. https://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_mfa_enable_virtual.html
5. CCE-78901-6
6. https://blogs.aws.amazon.com/security/post/Tx2SJJYE082KBUK/How-to-Delegate-Management-of-Multi-Factor-Authentication-to-AWS-IAM-Users

Additional Information:

Forced IAM User Self-Service Remediation

Amazon has published a pattern that requires users to set up MFA through self-service before they gain access to their complete set of permissions. Until they complete this step, they cannot access their full permissions. This pattern can be used for new AWS accounts. It can also be applied to existing accounts; it is recommended that users receive instructions and a grace period to complete MFA enrollment before active enforcement on existing AWS accounts.

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	6.5 Require MFA for Administrative Access Require MFA for all administrative access accounts, where supported, on all enterprise assets, whether managed on-site or through a third-party provider.	●	●	●
v7	4.5 Use Multifactor Authentication For All Administrative Access Use multi-factor authentication and encrypted channels for all administrative account access.		●	●

1.10 Do not create access keys during initial setup for IAM users with a console password (Manual)

Profile Applicability:

- Level 1

Description:

AWS console defaults to no check boxes selected when creating a new IAM user. When creating the IAM User credentials you have to determine what type of access they require.

Programmatic access: The IAM user might need to make API calls, use the AWS CLI, or use the Tools for Windows PowerShell. In that case, create an access key (access key ID and a secret access key) for that user.

AWS Management Console access: If the user needs to access the AWS Management Console, create a password for the user.

Rationale:

Requiring the additional steps to be taken by the user for programmatic access after their profile has been created will provide a stronger indication of intent that access keys are [a] necessary for their work and [b] that once the access key is established on an account, the keys may be in use somewhere in the organization.

Note: Even if it is known the user will need access keys, require them to create the keys themselves or put in a support ticket to have them created as a separate step from user creation.

Audit:

Perform the following steps to determine if unused access keys were created upon user creation:
From Console:

1. Login to the AWS Management Console
2. Click `Services`
3. Click `IAM`
4. Click on a User where column `Password age` and `Access key age` is not set to `None`
5. Click on `Security credentials` Tab
6. Compare the user `Creation time` to the Access Key `Created` date.
7. For any that match, the key was created during initial user setup.

- Keys that were created at the same time as the user profile and do not have a last used date should be deleted. Refer to the remediation below.

From Command Line:

1. Run the following command (OSX/Linux/UNIX) to generate a list of all IAM users along with their access keys utilization:

```
  aws iam generate-credential-report
  aws iam get-credential-report --query 'Content' --output text | base64 -d |
cut -d, -f1,4,9,11,14,16
```

2. The output of this command will produce a table similar to the following:

```
user,password_enabled,access_key_1_active,access_key_1_last_used_date,access_
key_2_active,access_key_2_last_used_date
  elise,false,true,2015-04-16T15:14:00+00:00,false,N/A
  brandon,true,true,N/A,false,N/A
  rakesh,false,false,N/A,false,N/A
  helene,false,true,2015-11-18T17:47:00+00:00,false,N/A
  paras,true,true,2016-08-28T12:04:00+00:00,true,2016-03-04T10:11:00+00:00
  anitha,true,true,2016-06-08T11:43:00+00:00,true,N/A
```

3. For any user having `password_enabled` set to `true` AND `access_key_last_used_date` set to `N/A` refer to the remediation below.

Remediation:

Perform the following to delete access keys that do not pass the audit:
From Console:

1. Login to the AWS Management Console:
2. Click `Services`
3. Click `IAM`
4. Click on `Users`
5. Click on `Security Credentials`
6. As an Administrator

- Click on the X `(Delete)` for keys that were created at the same time as the user profile but have not been used.

7. As an IAM User

- Click on the X `(Delete)` for keys that were created at the same time as the user profile but have not been used.

From Command Line:

```
aws iam delete-access-key --access-key-id <access-key-id-listed> --user-name
<users-name>
```

References:

1. https://docs.aws.amazon.com/cli/latest/reference/iam/delete-access-key.html
2. https://docs.aws.amazon.com/IAM/latest/UserGuide/id_users_create.html

Additional Information:

Credential report does not appear to contain "Key Creation Date"

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	6.1 Establish an Access Granting Process Establish and follow a process, preferably automated, for granting access to enterprise assets upon new hire, rights grant, or role change of a user.	●	●	●
v8	6.2 Establish an Access Revoking Process Establish and follow a process, preferably automated, for revoking access to enterprise assets, through disabling accounts immediately upon termination, rights revocation, or role change of a user. Disabling accounts, instead of deleting accounts, may be necessary to preserve audit trails.	●	●	●
v7	16.1 Maintain an Inventory of Authentication Systems Maintain an inventory of each of the organization's authentication systems, including those located onsite or at a remote service provider.		●	●

1.11 Ensure credentials unused for 45 days or more are disabled (Automated)

Profile Applicability:

- Level 1

Description:

AWS IAM users can access AWS resources using different types of credentials, such as passwords or access keys. It is recommended that all credentials that have been unused for 45 days or more be deactivated or removed.

Rationale:

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

Audit:

Perform the following to determine if unused credentials exist:
From Console:

1. Login to the AWS Management Console
2. Click `Services`
3. Click `IAM`
4. Click on `Users`
5. Click the `Settings` (gear) icon.
6. Select `Console last sign-in`, `Access key last used`, and `Access Key Id`
7. Click on `Close`
8. Check and ensure that `Console last sign-in` is less than 45 days ago.

Note - `Never` means the user has never logged in.

9. Check and ensure that `Access key age` is less than 45 days and that `Access key last used` does not say `None`

If the user hasn't signed into the Console in the last 45 days or Access keys are over 45 days old refer to the remediation.

From Command Line:
Download Credential Report:

1. Run the following commands:

```
  aws iam generate-credential-report

  aws iam get-credential-report --query 'Content' --output text | base64 -d |
cut -d, -f1,4,5,6,9,10,11,14,15,16 | grep -v '^<root_account>'
```

Ensure unused credentials do not exist:

2. For each user having `password_enabled` set to `TRUE` , ensure `password_last_used_date` is less than `45` days ago.

- When `password_enabled` is set to `TRUE` and `password_last_used` is set to `No_Information` , ensure `password_last_changed` is less than 45 days ago.

3. For each user having an `access_key_1_active` or `access_key_2_active` to `TRUE` , ensure the corresponding `access_key_n_last_used_date` is less than `45` days ago.

- When a user having an `access_key_x_active` (where x is 1 or 2) to `TRUE` and corresponding access_key_x_last_used_date is set to `N/A`, ensure `access_key_x_last_rotated` is less than 45 days ago.

Remediation:

From Console:
Perform the following to manage Unused Password (IAM user console access)

1. Login to the AWS Management Console:
2. Click `Services`
3. Click `IAM`
4. Click on `Users`
5. Click on `Security Credentials`
6. Select user whose `Console last sign-in` is greater than 45 days
7. Click `Security credentials`
8. In section `Sign-in credentials`, `Console password` click `Manage`
9. Under Console Access select `Disable`
10. Click `Apply`

Perform the following to deactivate Access Keys:

1. Login to the AWS Management Console:
2. Click `Services`
3. Click `IAM`

4. Click on `Users`
5. Click on `Security Credentials`
6. Select any access keys that are over 45 days old and that have been used and

- Click on `Make Inactive`

7. Select any access keys that are over 45 days old and that have not been used and

- Click the X to `Delete`

References:

1. CCE-78900-8
2. https://docs.aws.amazon.com/IAM/latest/UserGuide/best-practices.html#remove-credentials
3. https://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_finding-unused.html
4. https://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_passwords_admin-change-user.html
5. https://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_access-keys.html

Additional Information:

<root_account> is excluded in the audit since the root account should not be used for day-to-day business and would likely be unused for more than 45 days.

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	5.3 Disable Dormant Accounts Delete or disable any dormant accounts after a period of 45 days of inactivity, where supported.	●	●	●
v7	16.9 Disable Dormant Accounts Automatically disable dormant accounts after a set period of inactivity.	●	●	●

1.12 Ensure there is only one active access key for any single IAM user (Automated)

Profile Applicability:

- Level 1

Description:

Access keys are long-term credentials for an IAM user or the AWS account 'root' user. You can use access keys to sign programmatic requests to the AWS CLI or AWS API (directly or using the AWS SDK)

Rationale:

One of the best ways to protect your account is to not allow users to have multiple access keys.

Audit:

From Console:

1. Sign in to the AWS Management Console and navigate to IAM dashboard at `https://console.aws.amazon.com/iam/`.
2. In the left navigation panel, choose `Users`.
3. Click on the IAM user name that you want to examine.
4. On the IAM user configuration page, select `Security Credentials` tab.
5. Under `Access Keys` section, in the Status column, check the current status for each access key associated with the IAM user. If the selected IAM user has more than one access key activated, then the user's access configuration does not adhere to security best practices, and the risk of accidental exposures increases.

- Repeat steps 3-5 for each IAM user in your AWS account.

From Command Line:

1. Run `list-users` command to list all IAM users within your account:

```
aws iam list-users --query "Users[*].UserName"
```

The command output should return an array that contains all your IAM user names.

2. Run `list-access-keys` command using the IAM user name list to return the current status of each access key associated with the selected IAM user:

```
aws iam list-access-keys --user-name <user-name>
```

The command output should expose the metadata `("Username", "AccessKeyId", "Status", "CreateDate")` for each access key on that user account.

3. Check the `Status` property value for each key returned to determine each key's current state. If the `Status` property value for more than one IAM access key is set to `Active`, the user access configuration does not adhere to this recommendation; refer to the remediation below.

- Repeat steps 2 and 3 for each IAM user in your AWS account.

Remediation:

From Console:

1. Sign in to the AWS Management Console and navigate to IAM dashboard at `https://console.aws.amazon.com/iam/`.
2. In the left navigation panel, choose `Users`.
3. Click on the IAM user name that you want to examine.
4. On the IAM user configuration page, select `Security Credentials` tab.
5. In `Access Keys` section, choose one access key that is less than 90 days old. This should be the only active key used by this IAM user to access AWS resources programmatically. Test your application(s) to make sure that the chosen access key is working.
6. In the same `Access Keys` section, identify your non-operational access keys (other than the chosen one) and deactivate it by clicking the `Make Inactive` link.
7. If you receive the `Change Key Status` confirmation box, click `Deactivate` to switch off the selected key.
8. Repeat steps 3-7 for each IAM user in your AWS account.

From Command Line:

1. Using the IAM user and access key information provided in the `Audit CLI`, choose one access key that is less than 90 days old. This should be the only active key used by this IAM user to access AWS resources programmatically. Test your application(s) to make sure that the chosen access key is working.
2. Run the `update-access-key` command below using the IAM user name and the non-operational access key IDs to deactivate the unnecessary key(s). Refer to the Audit section to identify the unnecessary access key ID for the selected IAM user

Note - the command does not return any output:

```
aws iam update-access-key --access-key-id <access-key-id> --status Inactive --user-name <user-name>
```

3. To confirm that the selected access key pair has been successfully `deactivated` run the `list-access-keys` audit command again for that IAM User:

```
aws iam list-access-keys --user-name <user-name>
```

- The command output should expose the metadata for each access key associated with the IAM user. If the non-operational key pair(s) `Status` is set to `Inactive`, the key has been successfully deactivated and the IAM user access configuration adheres now to this recommendation.

4. Repeat steps 1-3 for each IAM user in your AWS account.

References:

1. https://docs.aws.amazon.com/general/latest/gr/aws-access-keys-best-practices.html
2. https://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_access-keys.html

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	5.1 Establish and Maintain an Inventory of Accounts Establish and maintain an inventory of all accounts managed in the enterprise. The inventory must include both user and administrator accounts. The inventory, at a minimum, should contain the person's name, username, start/stop dates, and department. Validate that all active accounts are authorized, on a recurring schedule at a minimum quarterly, or more frequently.	●	●	●
v7	16.1 Maintain an Inventory of Authentication Systems Maintain an inventory of each of the organization's authentication systems, including those located onsite or at a remote service provider.		●	●

1.13 Ensure access keys are rotated every 90 days or less (Automated)

Profile Applicability:

- Level 1

Description:

Access keys consist of an access key ID and secret access key, which are used to sign programmatic requests that you make to AWS. AWS users need their own access keys to make programmatic calls to AWS from the AWS Command Line Interface (AWS CLI), Tools for Windows PowerShell, the AWS SDKs, or direct HTTP calls using the APIs for individual AWS services. It is recommended that all access keys be rotated regularly.

Rationale:

Rotating access keys will reduce the window of opportunity for an access key that is associated with a compromised or terminated account to be used.

Access keys should be rotated to ensure that data cannot be accessed with an old key which might have been lost, cracked, or stolen.

Audit:

Perform the following to determine if access keys are rotated as prescribed:
From Console:

1. Go to the Management Console (https://console.aws.amazon.com/iam)
2. Click on `Users`
3. For each user, go to `Security Credentials`
4. Review each key under `Access Keys`
5. For each key that shows `Active` for status, ensure that `Created` is less than or equal to `90 days ago`.

From Command Line:

```
aws iam generate-credential-report
aws iam get-credential-report --query 'Content' --output text | base64 -d
```

The `access_key_1_last_rotated` and the `access_key_2_last_rotated` fields in this file notes the date and time, in ISO 8601 date-time format, when the user's access key was created or last changed. If the user does not have an active access key, the value in this field is N/A (not applicable).

Remediation:

Perform the following to rotate access keys:
From Console:

1. Go to the Management Console (https://console.aws.amazon.com/iam)
2. Click on `Users`
3. Click on `Security Credentials`
4. As an Administrator
    - Click on `Make Inactive` for keys that have not been rotated in `90` Days
5. As an IAM User
    - Click on `Make Inactive` or `Delete` for keys which have not been rotated or used in `90` Days
6. Click on `Create Access Key`
7. Update programmatic calls with new Access Key credentials

From Command Line:

1. While the first access key is still active, create a second access key, which is active by default. Run the following command:

```
aws iam create-access-key
```

At this point, the user has two active access keys.

2. Update all applications and tools to use the new access key.
3. Determine whether the first access key is still in use by using this command:

```
aws iam get-access-key-last-used
```

4. One approach is to wait several days and then check the old access key for any use before proceeding.

Even if step 3 indicates no use of the old key, it is recommended that you do not immediately delete the first access key. Instead, change the state of the first access key to Inactive using this command:

```
aws iam update-access-key
```

5. Use only the new access key to confirm that your applications are working. Any applications and tools that still use the original access key will stop working at this point because they no longer have access to AWS resources. If you find such an application or tool, you can switch its state back to Active to reenable the first access key. Then return to step 2 and update this application to use the new key.
6. After you wait some period of time to ensure that all applications and tools have been updated, you can delete the first access key with this command:

```
aws iam delete-access-key
```

References:

1. CCE-78902-4

2. https://docs.aws.amazon.com/IAM/latest/UserGuide/best-practices.html#rotate-credentials
3. https://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_finding-unused.html
4. https://docs.aws.amazon.com/general/latest/gr/managing-aws-access-keys.html
5. https://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_access-keys.html

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	5.1 Establish and Maintain an Inventory of Accounts Establish and maintain an inventory of all accounts managed in the enterprise. The inventory must include both user and administrator accounts. The inventory, at a minimum, should contain the person's name, username, start/stop dates, and department. Validate that all active accounts are authorized, on a recurring schedule at a minimum quarterly, or more frequently.	●	●	●
v7	16.1 Maintain an Inventory of Authentication Systems Maintain an inventory of each of the organization's authentication systems, including those located onsite or at a remote service provider.		●	●

1.14 Ensure IAM users receive permissions only through groups (Automated)

Profile Applicability:

- Level 1

Description:

IAM users are granted access to services, functions, and data through IAM policies. There are four ways to define policies for a user: 1) Edit the user policy directly, also known as an inline or user policy; 2) attach a policy directly to a user; 3) add the user to an IAM group that has an attached policy; 4) add the user to an IAM group that has an inline policy.

Only the third implementation is recommended.

Rationale:

Assigning IAM policies solely through groups unifies permissions management into a single, flexible layer that is consistent with organizational functional roles. By unifying permissions management, the likelihood of excessive permissions is reduced.

Audit:

Perform the following to determine if an inline policy is set or a policy is directly attached to users:

1. Run the following to get a list of IAM users:

```
aws iam list-users --query 'Users[*].UserName' --output text
```

2. For each user returned, run the following command to determine if any policies are attached to them:

```
aws iam list-attached-user-policies --user-name <iam_user>
aws iam list-user-policies --user-name <iam_user>
```

3. If any policies are returned, the user has an inline policy or direct policy attachment.

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. 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. For each user:
   - Select the user
   - Click on the `Permissions` tab
   - Expand `Permissions policies`
   - Click `X` for each policy; then click Detach or Remove (depending on policy type)

References:

1. http://docs.aws.amazon.com/IAM/latest/UserGuide/best-practices.html
2. http://docs.aws.amazon.com/IAM/latest/UserGuide/access_policies_managed-vs-inline.html
3. CCE-78912-3

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	6.8 Define and Maintain Role-Based Access Control Define and maintain role-based access control, through determining and documenting the access rights necessary for each role within the enterprise to successfully carry out its assigned duties. Perform access control reviews of enterprise assets to validate that all privileges are authorized, on a recurring schedule at a minimum annually, or more frequently.			●

Controls Version	Control	IG 1	IG 2	IG 3
v7	16.1 Maintain an Inventory of Authentication Systems Maintain an inventory of each of the organization's authentication systems, including those located onsite or at a remote service provider.		●	●

1.15 Ensure IAM policies that allow full ":*" administrative privileges are not attached (Automated)*

Profile Applicability:

- Level 1

Description:

IAM policies are the means by which privileges are granted to users, groups, or roles. It is recommended and considered standard security advice to grant least privilege—that is, granting only the permissions required to perform a task. Determine what users need to do, and then craft policies for them that allow the users to perform only those tasks, instead of granting full administrative privileges.

Rationale:

It's more secure to start with a minimum set of permissions and grant additional permissions as necessary, rather than starting with permissions that are too lenient and then attempting to tighten them later.

Providing full administrative privileges instead of restricting access to the minimum set of permissions required for the user exposes resources to potentially unwanted actions.

IAM policies that contain a statement with `"Effect": "Allow"` and `"Action": "*"` over `"Resource": "*"` should be removed.

Audit:

Perform the following to determine existing policies:
From Command Line:

1. Run the following to get a list of IAM policies:

```
aws iam list-policies --only-attached --output text
```

2. For each policy returned, run the following command to determine if any policy is allowing full administrative privileges on the account:

```
aws iam get-policy-version  --policy-arn <policy_arn> --version-id
<version>
```

3. In the output, the policy should not contain any Statement block with `"Effect": "Allow"` and `Action` set to `"*"` and `Resource` set to `"*"`.

Remediation:

From Console:
Perform the following to detach the policy that has full administrative privileges:

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 Policies and then search for the policy name found in the audit step.
3. Select the policy that needs to be deleted.
4. In the policy action menu, select `Detach`.
5. Select all Users, Groups, Roles that have this policy attached.
6. Click `Detach Policy`.
7. Select the newly detached policy and select `Delete`.

From Command Line:
Perform the following to detach the policy that has full administrative privileges as found in the audit step:

1. Lists all IAM users, groups, and roles that the specified managed policy is attached to.

```
aws iam list-entities-for-policy --policy-arn <policy_arn>
```

2. Detach the policy from all IAM Users:

```
aws iam detach-user-policy --user-name <iam_user> --policy-arn <policy_arn>
```

3. Detach the policy from all IAM Groups:

```
aws iam detach-group-policy --group-name <iam_group> --policy-arn
<policy_arn>
```

4. Detach the policy from all IAM Roles:

```
aws iam detach-role-policy --role-name <iam_role> --policy-arn <policy_arn>
```

References:

1. https://docs.aws.amazon.com/IAM/latest/UserGuide/best-practices.html
2. https://docs.aws.amazon.com/IAM/latest/UserGuide/access_policies_managed-vs-inline.html
3. CCE-78912-3
4. https://docs.aws.amazon.com/cli/latest/reference/iam/index.html#cli-aws-iam

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	5.4 Restrict Administrator Privileges to Dedicated Administrator Accounts Restrict administrator privileges to dedicated administrator accounts on enterprise assets. Conduct general computing activities, such as internet browsing, email, and productivity suite use, from the user's primary, non-privileged account.	●	●	●
v7	4.1 Maintain Inventory of Administrative Accounts Use automated tools to inventory all administrative accounts, including domain and local accounts, to ensure that only authorized individuals have elevated privileges.		●	●

1.16 Ensure a support role has been created to manage incidents with AWS Support (Automated)

Profile Applicability:

- Level 1

Description:

AWS provides a support center that can be used for incident notification and response, as well as technical support and customer services. Create an IAM Role, with the appropriate policy assigned, to allow authorized users to manage incidents with AWS Support.

Rationale:

By implementing least privilege for access control, an IAM Role will require an appropriate IAM Policy to allow Support Center Access in order to manage Incidents with AWS Support.

Impact:

All AWS Support plans include an unlimited number of account and billing support cases, with no long-term contracts. Support billing calculations are performed on a per-account basis for all plans. Enterprise Support plan customers have the option to include multiple enabled accounts in an aggregated monthly billing calculation. Monthly charges for the Business and Enterprise support plans are based on each month's AWS usage charges, subject to a monthly minimum, billed in advance.

When assigning rights, keep in mind that other policies may grant access to Support as well. This may include AdministratorAccess and other policies including customer managed policies. Utilizing the AWS managed 'AWSSupportAccess' role is one simple way of ensuring that this permission is properly granted.

To better support the principle of separation of duties, it would be best to only attach this role where necessary.

Audit:

From Command Line:

1. List IAM policies, filter for the 'AWSSupportAccess' managed policy, and note the "Arn" element value:

```
aws iam list-policies --query "Policies[?PolicyName == 'AWSSupportAccess']"
```

2. Check if the 'AWSSupportAccess' policy is attached to any role:

```
aws iam list-entities-for-policy --policy-arn
arn:aws:iam::aws:policy/AWSSupportAccess
```

3. In the output, ensure `PolicyRoles` does not return empty. 'Example: Example: PolicyRoles: [ ]'

If it returns empty refer to the remediation below.

Remediation:

From Command Line:

1. Create an IAM role for managing incidents with AWS:

- Create a trust relationship policy document that allows <iam_user> to manage AWS incidents, and save it locally as /tmp/TrustPolicy.json:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "<iam_user>"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

2. Create the IAM role using the above trust policy:

```
aws iam create-role --role-name <aws_support_iam_role> --assume-role-policy-
document file:///tmp/TrustPolicy.json
```

3. Attach 'AWSSupportAccess' managed policy to the created IAM role:

```
aws iam attach-role-policy --policy-arn
arn:aws:iam::aws:policy/AWSSupportAccess --role-name <aws_support_iam_role>
```

References:

1. https://docs.aws.amazon.com/IAM/latest/UserGuide/access_policies_managed-vs-inline.html
2. https://aws.amazon.com/premiumsupport/pricing/
3. https://awscli.amazonaws.com/v2/documentation/api/latest/reference/iam/list-policies.html
4. https://awscli.amazonaws.com/v2/documentation/api/latest/reference/iam/attach-role-policy.html

5. https://awscli.amazonaws.com/v2/documentation/api/latest/reference/iam/list-entities-for-policy.html

Additional Information:

AWSSupportAccess policy is a global AWS resource. It has same ARN as `arn:aws:iam::aws:policy/AWSSupportAccess` for every account.

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	17.1 Designate Personnel to Manage Incident Handling Designate one key person, and at least one backup, who will manage the enterprise's incident handling process. Management personnel are responsible for the coordination and documentation of incident response and recovery efforts and can consist of employees internal to the enterprise, third-party vendors, or a hybrid approach. If using a third-party vendor, designate at least one person internal to the enterprise to oversee any third-party work. Review annually, or when significant enterprise changes occur that could impact this Safeguard.	●	●	●
v7	14.1 Segment the Network Based on Sensitivity Segment the network based on the label or classification level of the information stored on the servers, locate all sensitive information on separated Virtual Local Area Networks (VLANs).		●	●

1.17 Ensure IAM instance roles are used for AWS resource access from instances (Automated)

Profile Applicability:

- Level 2

Description:

AWS access from within AWS instances can be done by either encoding AWS keys into AWS API calls or by assigning the instance to a role which has an appropriate permissions policy for the required access. "AWS Access" means accessing the APIs of AWS in order to access AWS resources or manage AWS account resources.

Rationale:

AWS IAM roles reduce the risks associated with sharing and rotating credentials that can be used outside of AWS itself. Compromised credentials can be used from outside the AWS account to which they provide access. In contrast, to leverage role permissions, an attacker would need to gain and maintain access to a specific instance to use the privileges associated with it.

Additionally, if credentials are encoded into compiled applications or other hard-to-change mechanisms, they are even less likely to be properly rotated due to the risks of service disruption. As time passes, credentials that cannot be rotated are more likely to be known by an increasing number of individuals who no longer work for the organization that owns the credentials.

Audit:

First, check if the instance has any API secrets stored using Secret Scanning. Currently, AWS does not have a solution for this. You can use open-source tools like TruffleHog to scan for secrets in the EC2 instance. If a secret is found, then assign the role to the instance.
From Console:

1. Sign in to the AWS Management Console and navigate to the EC2 dashboard at `https://console.aws.amazon.com/ec2/`.
2. In the left navigation panel, choose `Instances`.
3. Select the EC2 instance you want to examine.
4. Select `Actions`.
5. Select `View details`.
6. Select `Security` in the lower panel.

- If the value for **Instance profile arn** is an instance profile ARN, then an instance profile (that contains an IAM role) is attached.
- If the value for **IAM Role** is blank, no role is attached.
- If the value for **IAM Role** contains a role, a role is attached.

- If the value for **IAM Role** is "No roles attached to instance profile: <Instance-Profile-Name>", then an instance profile is attached to the instance, but it does not contain an IAM role.

7. Repeat steps 3 to 6 for each EC2 instance in your AWS account.

From Command Line:

1. Run the `describe-instances` command to list all EC2 instance IDs in the selected AWS region:

```
aws ec2 describe-instances --region <region-name> --query
'Reservations[*].Instances[*].InstanceId'
```

2. Run the `describe-instances` command again for each EC2 instance using the `IamInstanceProfile` identifier in the query filter to check if an IAM role is attached:

```
aws ec2 describe-instances --region <region-name> --instance-id <Instance-ID>
--query 'Reservations[*].Instances[*].IamInstanceProfile'
```

3. If an IAM role is attached, the command output will show the IAM instance profile ARN and ID.
4. Repeat steps 2 and 3 for each EC2 instance in your AWS account.

Remediation:

From Console:

1. Sign in to the AWS Management Console and navigate to the EC2 dashboard at `https://console.aws.amazon.com/ec2/`.
2. In the left navigation panel, choose `Instances`.
3. Select the EC2 instance you want to modify.
4. Click `Actions`.
5. Click `Security`.
6. Click `Modify IAM role`.
7. Click `Create new IAM role` if a new IAM role is required.
8. Select the IAM role you want to attach to your instance in the `IAM role` dropdown.
9. Click `Update IAM role`.
10. Repeat steps 3 to 9 for each EC2 instance in your AWS account that requires an IAM role to be attached.

From Command Line:

1. Run the `describe-instances` command to list all EC2 instance IDs in the selected AWS region:

```
aws ec2 describe-instances --region <region-name> --query
'Reservations[*].Instances[*].InstanceId'
```

2. Run the `associate-iam-instance-profile` command to attach an instance profile (which is attached to an IAM role) to the EC2 instance:

```
aws ec2 associate-iam-instance-profile --region <region-name> --instance-id
<Instance-ID> --iam-instance-profile Name="Instance-Profile-Name"
```

3. Run the `describe-instances` command again for the recently modified EC2 instance. The command output should return the instance profile ARN and ID:

```
aws ec2 describe-instances --region <region-name> --instance-id <Instance-ID>
--query 'Reservations[*].Instances[*].IamInstanceProfile'
```

4. Repeat steps 2 and 3 for each EC2 instance in your AWS account that requires an IAM role to be attached.

References:

1. https://docs.aws.amazon.com/IAM/latest/UserGuide/id_roles_use_switch-role-ec2.html
2. https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/iam-roles-for-amazon-ec2.html

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	6.8 Define and Maintain Role-Based Access Control Define and maintain role-based access control, through determining and documenting the access rights necessary for each role within the enterprise to successfully carry out its assigned duties. Perform access control reviews of enterprise assets to validate that all privileges are authorized, on a recurring schedule at a minimum annually, or more frequently.			●
v7	14.1 Segment the Network Based on Sensitivity Segment the network based on the label or classification level of the information stored on the servers, locate all sensitive information on separated Virtual Local Area Networks (VLANs).		●	●

1.18 Ensure that all expired SSL/TLS certificates stored in AWS IAM are removed (Automated)

Profile Applicability:

- Level 1

Description:

To enable HTTPS connections to your website or application in AWS, you need an SSL/TLS server certificate. You can use AWS Certificate Manager (ACM) or IAM to store and deploy server certificates. Use IAM as a certificate manager only when you must support HTTPS connections in a region that is not supported by ACM. IAM securely encrypts your private keys and stores the encrypted version in IAM SSL certificate storage. IAM supports deploying server certificates in all regions, but you must obtain your certificate from an external provider for use with AWS. You cannot upload an ACM certificate to IAM. Additionally, you cannot manage your certificates from the IAM Console.

Rationale:

Removing expired SSL/TLS certificates eliminates the risk that an invalid certificate will be deployed accidentally to a resource such as AWS Elastic Load Balancer (ELB), which can damage the credibility of the application/website behind the ELB. As a best practice, it is recommended to delete expired certificates.

Impact:

Deleting the certificate could have implications for your application if you are using an expired server certificate with Elastic Load Balancing, CloudFront, etc. You must make configurations in the respective services to ensure there is no interruption in application functionality.

Audit:

From Console:
Getting the certificate expiration information via the AWS Management Console is not currently supported. To request information about the SSL/TLS certificates stored in IAM through the AWS API, use the Command Line Interface (CLI).
From Command Line:
Run the `list-server-certificates` command to list all the IAM-stored server certificates:

```
aws iam list-server-certificates
```

The command output should return an array that contains all the SSL/TLS certificates currently stored in IAM and their metadata (name, ID, expiration date, etc):

```
{
    "ServerCertificateMetadataList": [
        {
            "ServerCertificateId": "EHDGFRW7EJFYTE88D",
            "ServerCertificateName": "MyServerCertificate",
            "Expiration": "2018-07-10T23:59:59Z",
            "Path": "/",
            "Arn": "arn:aws:iam::012345678910:server-
certificate/MySSLCertificate",
            "UploadDate": "2018-06-10T11:56:08Z"
        }
    ]
}
```

Verify the `ServerCertificateName` and `Expiration` parameter value (expiration date) for each SSL/TLS certificate returned by the list-server-certificates command and determine if there are any expired server certificates currently stored in AWS IAM. If so, use the AWS API to remove them.
If this command returns:

```
{ { "ServerCertificateMetadataList": [] }
```

This means that there are no expired certificates; it **does not** mean that no certificates exist.

Remediation:

From Console:
Removing expired certificates via AWS Management Console is not currently supported. To delete SSL/TLS certificates stored in IAM through the AWS API, use the Command Line Interface (CLI).
From Command Line:
To delete an expired certificate, run the following command by replacing <CERTIFICATE_NAME> with the name of the certificate to delete:

```
aws iam delete-server-certificate --server-certificate-name
<CERTIFICATE_NAME>
```

When the preceding command is successful, it does not return any output.

Default Value:

By default, expired certificates will not be deleted.

References:

1. https://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_server-certs.html
2. https://awscli.amazonaws.com/v2/documentation/api/latest/reference/iam/delete-server-certificate.html

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.1 Establish and Maintain a Data Management Process Establish and maintain a data management process. In the process, address data sensitivity, data owner, handling of data, data retention limits, and disposal requirements, based on sensitivity and retention standards for the enterprise. Review and update documentation annually, or when significant enterprise changes occur that could impact this Safeguard.	●	●	●
v7	13.1 Maintain an Inventory Sensitive Information Maintain an inventory of all sensitive information stored, processed, or transmitted by the organization's technology systems, including those located onsite or at a remote service provider.	●	●	●

1.19 Ensure that IAM External Access Analyzer is enabled for all regions (Automated)

Profile Applicability:

- Level 1

Description:

Enable the IAM External Access Analyzer regarding all resources in each active AWS region.

IAM Access Analyzer is a technology introduced at AWS reinvent 2019. After the Analyzer is enabled in IAM, scan results are displayed on the console showing the accessible resources. Scans show resources that other accounts and federated users can access, such as KMS keys and IAM roles. The results allow you to determine whether an unintended user is permitted, making it easier for administrators to monitor least privilege access. Access Analyzer analyzes only the policies that are applied to resources in the same AWS Region.

Rationale:

AWS IAM External Access Analyzer helps you identify the resources in your organization and accounts, such as Amazon S3 buckets or IAM roles, that are shared with external entities. This allows you to identify unintended access to your resources and data. Access Analyzer identifies resources that are shared with external principals by using logic-based reasoning to analyze the resource-based policies in your AWS environment. IAM External Access Analyzer continuously monitors all policies for S3 buckets, IAM roles, KMS (Key Management Service) keys, AWS Lambda functions, Amazon SQS (Simple Queue Service) queues and more

Audit:

From Console:

1. Open the IAM console at `https://console.aws.amazon.com/iam/`
2. Under `Access analyzer` choose `External Access`
3. Ensure that at least one analyzer is present
4. Ensure that the `STATUS` is set to `Active`
5. Repeat these steps for each active region

From Command Line:

1. Run the following command:

```
aws accessanalyzer list-analyzers type --<Account|Organization> | grep status
```

2. Ensure that at least one Analyzer's `status` is set to `ACTIVE`.
3. Repeat the steps above for each active region.

If an Access Analyzer is not listed for each region or the status is not set to active refer to the remediation procedure below.

Remediation:

From Console:
Perform the following to enable IAM Access Analyzer for IAM policies:

1. Open the IAM console at `https://console.aws.amazon.com/iam/.`
2. Choose `Access analyzer`.
3. Choose `Create external access analyzer`.
4. On the `Create analyzer` page, confirm that the `Region` displayed is the Region where you want to enable Access Analyzer.
5. Optionally enter a name for the analyzer.
6. Optionally add any tags that you want to apply to the analyzer.
7. Choose `Create Analyzer`.
8. Repeat these step for each active region.

From Command Line:
Run the following command:

```
aws accessanalyzer create-analyzer --analyzer-name <NAME> --type
<ACCOUNT|ORGANIZATION>
```

Repeat this command for each active region.
Note: The IAM Access Analyzer is successfully configured only when the account you use has the necessary permissions.

References:

1. https://docs.aws.amazon.com/IAM/latest/UserGuide/what-is-access-analyzer.html
2. https://docs.aws.amazon.com/IAM/latest/UserGuide/access-analyzer-getting-started.html
3. https://awscli.amazonaws.com/v2/documentation/api/latest/reference/accessanalyzer/get-analyzer.html
4. https://awscli.amazonaws.com/v2/documentation/api/latest/reference/accessanalyzer/create-analyzer.html

Additional Information:

Some regions in AWS are enabled by default, while others are disabled by default. Regions introduced prior to March 20, 2019, are enabled by default and cannot be disabled. Regions introduced afterward can be disabled by default. For more information on managing AWS Regions, please see AWS's documentation on managing AWS Regions.

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.3 Configure Data Access Control Lists Configure data access control lists based on a user's need to know. Apply data access control lists, also known as access permissions, to local and remote file systems, databases, and applications.	●	●	●
v7	14.6 Protect Information through Access Control Lists Protect all information stored on systems with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities.	●	●	●

1.20 Ensure IAM users are managed centrally via identity federation or AWS Organizations for multi-account environments (Manual)

Profile Applicability:

- Level 2

Description:

In multi-account environments, IAM user centralization facilitates greater user control. User access beyond the initial account is then provided via role assumption. Centralization of users can be accomplished through federation with an external identity provider or through the use of AWS Organizations.

Rationale:

Centralizing IAM user management to a single identity store reduces complexity and thus the likelihood of access management errors.

Audit:

For multi-account AWS environments with an external identity provider:

1. Determine the master account for identity federation or IAM user management
2. Login to that account through the AWS Management Console
3. Click `Services`
4. Click `IAM`
5. Click `Identity providers`
6. Verify the configuration

For multi-account AWS environments with an external identity provider, as well as for those implementing AWS Organizations without an external identity provider:

1. Determine all accounts that should not have local users present
2. Log into the AWS Management Console
3. Switch role into each identified account
4. Click `Services`
5. Click `IAM`
6. Click `Users`
7. Confirm that no IAM users representing individuals are present

Remediation:

The remediation procedure will vary based on each individual organization's implementation of identity federation and/or AWS Organizations, with the acceptance criteria that no non-service IAM users and non-root accounts are present outside the account providing centralized IAM user management.

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	5.6 Centralize Account Management Centralize account management through a directory or identity service.		●	●
v7	16.2 Configure Centralized Point of Authentication Configure access for all accounts through as few centralized points of authentication as possible, including network, security, and cloud systems.		●	●

1.21 Ensure access to AWSCloudShellFullAccess is restricted (Manual)

Profile Applicability:

- Level 1

Description:

AWS CloudShell is a convenient way of running CLI commands against AWS services; a managed IAM policy ('AWSCloudShellFullAccess') provides full access to CloudShell, which allows file upload and download capability between a user's local system and the CloudShell environment. Within the CloudShell environment, a user has sudo permissions and can access the internet. Therefore, it is feasible to install file transfer software, for example, and move data from CloudShell to external internet servers.

Rationale:

Access to this policy should be restricted, as it presents a potential channel for data exfiltration by malicious cloud admins who are given full permissions to the service. AWS documentation describes how to create a more restrictive IAM policy that denies file transfer permissions.

Audit:

From Console

1. Open the IAM console at https://console.aws.amazon.com/iam/
2. In the left pane, select Policies
3. Search for and select AWSCloudShellFullAccess
4. On the Entities attached tab, ensure that there are no entities using this policy

From Command Line

1. List IAM policies, filter for the 'AWSCloudShellFullAccess' managed policy, and note the "Arn" element value:

```
aws iam list-policies --query "Policies[?PolicyName ==
'AWSCloudShellFullAccess']"
```

2. Check if the 'AWSCloudShellFullAccess' policy is attached to any role:

```
aws iam list-entities-for-policy --policy-arn
arn:aws:iam::aws:policy/AWSCloudShellFullAccess
```

3. In the output, ensure PolicyRoles returns empty. 'Example: Example: PolicyRoles: [ ]'

If it does not return empty, refer to the remediation below.
Note: Keep in mind that other policies may grant access.

Remediation:

From Console

1. Open the IAM console at https://console.aws.amazon.com/iam/
2. In the left pane, select Policies
3. Search for and select AWSCloudShellFullAccess
4. On the Entities attached tab, for each item, check the box and select Detach

References:

1. https://docs.aws.amazon.com/cloudshell/latest/userguide/sec-auth-with-identities.html

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	6.1 Establish an Access Granting Process Establish and follow a process, preferably automated, for granting access to enterprise assets upon new hire, rights grant, or role change of a user.	●	●	●
v7	14.1 Segment the Network Based on Sensitivity Segment the network based on the label or classification level of the information stored on the servers, locate all sensitive information on separated Virtual Local Area Networks (VLANs).		●	●

2 Storage

This section contains recommendations for configuring AWS Storage.

2.1 Simple Storage Service (S3)

This section contains recommendations for configuring AWS Simple Storage Service (S3) Buckets

2.1.1 Ensure S3 Bucket Policy is set to deny HTTP requests (Automated)

Profile Applicability:

- Level 2

Description:

At the Amazon S3 bucket level, you can configure permissions through a bucket policy, making the objects accessible only through HTTPS.

Rationale:

By default, Amazon S3 allows both HTTP and HTTPS requests. To ensure that access to Amazon S3 objects is only permitted through HTTPS, you must explicitly deny HTTP requests. Bucket policies that allow HTTPS requests without explicitly denying HTTP requests will not comply with this recommendation.

Audit:

To allow access to HTTPS, you can use a bucket policy with the effect `allow` and a condition that checks for the key `"aws:SecureTransport": "true"`. This means that HTTPS requests are allowed, but it does not deny HTTP requests. To explicitly deny HTTP access, ensure that there is also a bucket policy with the effect `deny` that contains the key `"aws:SecureTransport": "false"`. You may also require TLS by setting a policy to deny any version lower than the one you wish to require, using the condition `NumericLessThan` and the key `"s3:TlsVersion": "1.2"`.
From Console:

1. Login to the AWS Management Console and open the Amazon S3 console using https://console.aws.amazon.com/s3/.
2. Select the check box next to the Bucket.
3. Click on 'Permissions', then click on `Bucket Policy`.
4. Ensure that a policy is listed that matches either:

```
{
    "Sid": <optional>,
    "Effect": "Deny",
    "Principal": "*",
    "Action": "s3:*",
    "Resource": "arn:aws:s3:::<bucket_name>/*",
    "Condition": {
        "Bool": {
            "aws:SecureTransport": "false"
        }
    }
}
```

or

```
{
    "Sid": "<optional>",
    "Effect": "Deny",
    "Principal": "*",
    "Action": "s3:*",
    "Resource": [
        "arn:aws:s3:::<bucket_name>",
        "arn:aws:s3:::<bucket_name>/*"
    ],
    "Condition": {
        "NumericLessThan": {
            "s3:TlsVersion": "1.2"
        }
    }
}
```

<optional> and <bucket_name> will be specific to your account, and TLS version will be site/policy specific to your organisation.

5. Repeat for all the buckets in your AWS account.

From Command Line:

1. List all of the S3 Buckets

```
aws s3 ls
```

2. Using the list of buckets, run this command on each of them:

```
aws s3api get-bucket-policy --bucket <bucket_name> | grep aws:SecureTransport
```

or

```
aws s3api get-bucket-policy --bucket <bucket_name> | grep s3:TlsVersion
```

NOTE : If an error is thrown by the CLI, it means no policy has been configured for the specified S3 bucket, and that by default it is allowing both HTTP and HTTPS requests.

3. Confirm that aws:SecureTransport is set to false (such as aws:SecureTransport:false) or that s3:TlsVersion has a site-specific value.
4. Confirm that the policy line has Effect set to Deny 'Effect:Deny'

Remediation:

From Console:

1. Login to the AWS Management Console and open the Amazon S3 console using https://console.aws.amazon.com/s3/.
2. Select the check box next to the Bucket.
3. Click on 'Permissions'.

4. Click 'Bucket Policy'.
5. Add either of the following to the existing policy, filling in the required information:

```
{
    "Sid": <optional>,
    "Effect": "Deny",
    "Principal": "*",
    "Action": "s3:*",
    "Resource": "arn:aws:s3:::<bucket_name>/*",
    "Condition": {
        "Bool": {
            "aws:SecureTransport": "false"
        }
    }
}
```

or

```
{
    "Sid": "<optional>",
    "Effect": "Deny",
    "Principal": "*",
    "Action": "s3:*",
    "Resource": [
        "arn:aws:s3:::<bucket_name>",
        "arn:aws:s3:::<bucket_name>/*"
    ],
    "Condition": {
        "NumericLessThan": {
            "s3:TlsVersion": "1.2"
        }
    }
}
```

6. Save
7. Repeat for all the buckets in your AWS account that contain sensitive data.

From Console
Using AWS Policy Generator:

1. Repeat steps 1-4 above.
2. Click on `Policy Generator` at the bottom of the Bucket Policy Editor.
3. Select Policy Type `S3 Bucket Policy`.
4. Add Statements:

- `Effect` = Deny
- `Principal` = *
- `AWS Service` = Amazon S3
- `Actions` = *
- `Amazon Resource Name` = <ARN of the S3 Bucket>

5. Generate Policy.
6. Copy the text and add it to the Bucket Policy.

From Command Line:

1. Export the bucket policy to a json file:

```
aws s3api get-bucket-policy --bucket <bucket_name> --query Policy --output
text > policy.json
```

2. Modify the policy.json file by adding either of the following:

```
{
    "Sid": <optional>,
    "Effect": "Deny",
    "Principal": "*",
    "Action": "s3:*",
    "Resource": "arn:aws:s3:::<bucket_name>/*",
    "Condition": {
        "Bool": {
            "aws:SecureTransport": "false"
        }
    }
}
```

or

```
{
    "Sid": "<optional>",
    "Effect": "Deny",
    "Principal": "*",
    "Action": "s3:*",
    "Resource": [
        "arn:aws:s3:::<bucket_name>",
        "arn:aws:s3:::<bucket_name>/*"
    ],
    "Condition": {
        "NumericLessThan": {
            "s3:TlsVersion": "1.2"
        }
    }
}
```

3. Apply this modified policy back to the S3 bucket:

```
aws s3api put-bucket-policy --bucket <bucket_name> --policy
file://policy.json
```

Default Value:

Both HTTP and HTTPS requests are allowed.

References:

1. https://aws.amazon.com/premiumsupport/knowledge-center/s3-bucket-policy-for-config-rule/
2. https://aws.amazon.com/blogs/security/how-to-use-bucket-policies-and-apply-defense-in-depth-to-help-secure-your-amazon-s3-data/
3. https://awscli.amazonaws.com/v2/documentation/api/latest/reference/s3api/get-bucket-policy.html

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.10 Encrypt Sensitive Data in Transit Encrypt sensitive data in transit. Example implementations can include: Transport Layer Security (TLS) and Open Secure Shell (OpenSSH).		●	●
v7	14.4 Encrypt All Sensitive Information in Transit Encrypt all sensitive information in transit.		●	●

2.1.2 Ensure MFA Delete is enabled on S3 buckets (Manual)

Profile Applicability:

- Level 2

Description:

Once MFA Delete is enabled on your sensitive and classified S3 bucket, it requires the user to provide two forms of authentication.

Rationale:

Adding MFA delete to an S3 bucket requires additional authentication when you change the version state of your bucket or delete an object version, adding another layer of security in the event your security credentials are compromised or unauthorized access is granted.

Impact:

Enabling MFA delete on an S3 bucket could require additional administrator oversight. Enabling MFA delete may impact other services that automate the creation and/or deletion of S3 buckets.

Audit:

Perform the steps below to confirm that MFA delete is configured on an S3 bucket:
From Console:

1. Login to the S3 console at `https://console.aws.amazon.com/s3/`.
2. Click the `check` box next to the name of the bucket you want to confirm.
3. In the window under `Properties`:

- Confirm that Versioning is `Enabled`
- Confirm that MFA Delete is `Enabled`

From Command Line:

1. Run the `get-bucket-versioning` command:

```
aws s3api get-bucket-versioning --bucket my-bucket
```

Example output:

```
<VersioningConfiguration xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
  <Status>Enabled</Status>
  <MfaDelete>Enabled</MfaDelete>
</VersioningConfiguration>
```

If the console or CLI output does not show that Versioning and MFA Delete are `enabled`, please refer to the remediation below.

Remediation:

Perform the steps below to enable MFA delete on an S3 bucket:
Note:

- You cannot enable MFA Delete using the AWS Management Console; you must use the AWS CLI or API.
- You must use your 'root' account to enable MFA Delete on S3 buckets.

From Command line:

1. Run the s3api `put-bucket-versioning` command:

```
aws s3api put-bucket-versioning --profile my-root-profile --bucket
Bucket_Name --versioning-configuration Status=Enabled,MFADelete=Enabled --mfa
“arn:aws:iam::aws_account_id:mfa/root-account-mfa-device passcode”
```

References:

1. https://docs.aws.amazon.com/AmazonS3/latest/dev/Versioning.html#MultiFactorAuthenticationDelete
2. https://docs.aws.amazon.com/AmazonS3/latest/dev/UsingMFADelete.html
3. https://aws.amazon.com/blogs/security/securing-access-to-aws-using-mfa-part-3/
4. https://docs.aws.amazon.com/IAM/latest/UserGuide/id_credentials_mfa_lost-or-broken.html

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.3 Configure Data Access Control Lists Configure data access control lists based on a user’s need to know. Apply data access control lists, also known as access permissions, to local and remote file systems, databases, and applications.	●	●	●
v8	6.5 Require MFA for Administrative Access Require MFA for all administrative access accounts, where supported, on all enterprise assets, whether managed on-site or through a third-party provider.	●	●	●
v7	14.6 Protect Information through Access Control Lists Protect all information stored on systems with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities.	●	●	●

2.1.3 Ensure all data in Amazon S3 has been discovered, classified, and secured when necessary (Manual)

Profile Applicability:

- Level 2

Description:

Amazon S3 buckets can contain sensitive data that, for security purposes, should be discovered, monitored, classified, and protected. Macie, along with other third-party tools, can automatically provide an inventory of Amazon S3 buckets.

Rationale:

Using a cloud service or third-party software to continuously monitor and automate the process of data discovery and classification for S3 buckets through machine learning and pattern matching is a strong defense in protecting that information.

Amazon Macie is a fully managed data security and privacy service that uses machine learning and pattern matching to discover and protect your sensitive data in AWS.

Impact:

There is a cost associated with using Amazon Macie, and there is typically a cost associated with third-party tools that perform similar processes and provide protection.

Audit:

Perform the following steps to determine if Macie is running:
From Console:

1. Login to the Macie console at https://console.aws.amazon.com/macie/.
2. In the left hand pane, click on `By job` under findings.
3. Confirm that you have a job set up for your S3 buckets.

When you log into the Macie console, if you are not taken to the summary page and do not have a job set up and running, then refer to the remediation procedure below.
If you are using a third-party tool to manage and protect your S3 data, you meet this recommendation.

Remediation:

Perform the steps below to enable and configure Amazon Macie:
From Console:

1. Log on to the Macie console at `https://console.aws.amazon.com/macie/`.
2. Click `Get started`.
3. Click `Enable Macie`.

Set up a repository for sensitive data discovery results:

1. In the left pane, under Settings, click `Discovery results`.
2. Make sure `Create bucket` is selected.
3. Create a bucket and enter a name for it. The name must be unique across all S3 buckets, and it must start with a lowercase letter or a number.
4. Click `Advanced`.
5. For block all public access, make sure `Yes` is selected.
6. For KMS encryption, specify the AWS KMS key that you want to use to encrypt the results. The key must be a symmetric customer master key (CMK) that is in the same region as the S3 bucket.
7. Click `Save`.

Create a job to discover sensitive data:

1. In the left pane, click `S3 buckets`. Macie displays a list of all the S3 buckets for your account.
2. Check the box for each bucket that you want Macie to analyze as part of the job.
3. Click `Create job`.
4. Click `Quick create`.
5. For the Name and Description step, enter a name and, optionally, a description of the job.
6. Click `Next`.
7. For the Review and create step, click `Submit`.

Review your findings:

1. In the left pane, click `Findings`.
2. To view the details of a specific finding, choose any field other than the check box for the finding.

If you are using a third-party tool to manage and protect your S3 data, follow the vendor documentation for implementing and configuring that tool.

References:

1. https://aws.amazon.com/macie/getting-started/
2. https://docs.aws.amazon.com/workspaces/latest/adminguide/data-protection.html
3. https://docs.aws.amazon.com/macie/latest/user/data-classification.html

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.1 Establish and Maintain a Data Management Process Establish and maintain a data management process. In the process, address data sensitivity, data owner, handling of data, data retention limits, and disposal requirements, based on sensitivity and retention standards for the enterprise. Review and update documentation annually, or when significant enterprise changes occur that could impact this Safeguard.	●	●	●
v7	5.1 Establish Secure Configurations Maintain documented, standard security configuration standards for all authorized operating systems and software.	●	●	●

2.1.4 Ensure that S3 is configured with 'Block Public Access' enabled (Automated)

Profile Applicability:

- Level 1

Description:

Amazon S3 provides `Block public access (bucket settings)` and `Block public access (account settings)` to help you manage public access to Amazon S3 resources. By default, S3 buckets and objects are created with public access disabled. However, an IAM principal with sufficient S3 permissions can enable public access at the bucket and/or object level. While enabled, `Block public access (bucket settings)` prevents an individual bucket and its contained objects from becoming publicly accessible. Similarly, `Block public access (account settings)` prevents all buckets and their contained objects from becoming publicly accessible across the entire account.

Rationale:

Amazon S3 `Block public access (bucket settings)` prevents the accidental or malicious public exposure of data contained within the respective bucket(s).

Amazon S3 `Block public access (account settings)` prevents the accidental or malicious public exposure of data contained within all buckets of the respective AWS account.

Whether to block public access to all or some buckets is an organizational decision that should be based on data sensitivity, least privilege, and use case.

Impact:

When you apply Block Public Access settings to an account, the settings apply to all AWS regions globally. The settings may not take effect in all regions immediately or simultaneously, but they will eventually propagate to all regions.

Audit:

If utilizing Block Public Access (bucket settings)
From Console:

1. Login to the AWS Management Console and open the Amazon S3 console using https://console.aws.amazon.com/s3/.
2. Select the check box next to a bucket.
3. Click on 'Edit public access settings'.
4. Ensure that the block public access settings are configured appropriately for this bucket.
5. Repeat for all the buckets in your AWS account.

From Command Line:

1. List all of the S3 buckets:

```
aws s3 ls
```

2. Find the public access settings for a specific bucket:

```
aws s3api get-public-access-block --bucket <bucket-name>
```

Output if Block Public Access is enabled:

```
{
    "PublicAccessBlockConfiguration": {
        "BlockPublicAcls": true,
        "IgnorePublicAcls": true,
        "BlockPublicPolicy": true,
        "RestrictPublicBuckets": true
    }
}
```

If the output reads `false` for the separate configuration settings, then proceed with the remediation.

If utilizing Block Public Access (account settings)

From Console:

1. Login to the AWS Management Console and open the Amazon S3 console using https://console.aws.amazon.com/s3/.
2. Choose `Block public access (account settings)`.
3. Ensure that the block public access settings are configured appropriately for your AWS account.

From Command Line:

To check the block public access settings for this account, run the following command:

```
aws s3control get-public-access-block --account-id <account-id> --region <region-name>
```

Output if Block Public Access is enabled:

```
{
    "PublicAccessBlockConfiguration": {
        "IgnorePublicAcls": true,
        "BlockPublicPolicy": true,
        "BlockPublicAcls": true,
        "RestrictPublicBuckets": true
    }
}
```

If the output reads `false` for the separate configuration settings, then proceed with the remediation.

Remediation:

If utilizing Block Public Access (bucket settings)
From Console:

1. Login to the AWS Management Console and open the Amazon S3 console using https://console.aws.amazon.com/s3/.
2. Select the check box next to a bucket.
3. Click 'Edit public access settings'.
4. Click 'Block all public access'
5. Repeat for all the buckets in your AWS account that contain sensitive data.

From Command Line:

1. List all of the S3 buckets:

```
aws s3 ls
```

2. Enable Block Public Access on a specific bucket:

```
aws s3api put-public-access-block --bucket <bucket-name> --public-access-
block-configuration
"BlockPublicAcls=true,IgnorePublicAcls=true,BlockPublicPolicy=true,RestrictPu
blicBuckets=true"
```

If utilizing Block Public Access (account settings)
From Console:
If the output reads `true` for the separate configuration settings, then Block Public Access is enabled on the account.

1. Login to the AWS Management Console and open the Amazon S3 console using https://console.aws.amazon.com/s3/.
2. Click `Block Public Access (account settings)`.
3. Click `Edit` to change the block public access settings for all the buckets in your AWS account.
4. Update the settings and click `Save`. For details about each setting, pause on the `i` icons.
5. When you're asked for confirmation, enter `confirm`. Then click `Confirm` to save your changes.

From Command Line:
To enable Block Public Access for this account, run the following command:

```
aws s3control put-public-access-block
--public-access-block-configuration BlockPublicAcls=true,
IgnorePublicAcls=true, BlockPublicPolicy=true, RestrictPublicBuckets=true
--account-id <account-id>
```

References:

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

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.3 Configure Data Access Control Lists Configure data access control lists based on a user's need to know. Apply data access control lists, also known as access permissions, to local and remote file systems, databases, and applications.	●	●	●
v7	14.6 Protect Information through Access Control Lists Protect all information stored on systems with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities.	●	●	●

2.2 Relational Database Service (RDS)

This section contains recommendations for configuring AWS Relational Database Services (RDS)

2.2.1 Ensure that encryption-at-rest is enabled for RDS instances (Automated)

Profile Applicability:

- Level 1

Description:

Amazon RDS encrypted DB instances use the industry-standard AES-256 encryption algorithm to encrypt your data on the server that hosts your Amazon RDS DB instances. After your data is encrypted, Amazon RDS handles the authentication of access and the decryption of your data transparently, with minimal impact on performance.

Rationale:

Databases are likely to hold sensitive and critical data; therefore, it is highly recommended to implement encryption to protect your data from unauthorized access or disclosure. With RDS encryption enabled, the data stored on the instance's underlying storage, the automated backups, read replicas, and snapshots are all encrypted.

Audit:

From Console:

1. Login to the AWS Management Console and open the RDS dashboard at https://console.aws.amazon.com/rds/.
2. In the navigation pane, under RDS dashboard, click `Databases`.
3. Select the RDS instance that you want to examine.
4. Click `Instance Name` to see details, then select the `Configuration` tab.
5. Under Configuration Details, in the Storage pane, search for the `Encryption Enabled` status.
6. If the current status is set to `Disabled`, encryption is not enabled for the selected RDS database instance.
7. Repeat steps 2 to 6 to verify the encryption status of other RDS instances in the same region.
8. Change the region from the top of the navigation bar, and repeat the audit steps for other regions.

From Command Line:

1. Run the `describe-db-instances` command to list all the RDS database instance names available in the selected AWS region. The output will return each database instance identifier (name):

```
aws rds describe-db-instances --region <region-name> --query
'DBInstances[*].DBInstanceIdentifier'
```

2. Run the `describe-db-instances` command again, using an RDS instance identifier returned from step 1, to determine if the selected database instance is encrypted. The output should return the encryption status `True` or `False`:

```
aws rds describe-db-instances --region <region-name> --db-instance-identifier
<db-name> --query 'DBInstances[*].StorageEncrypted'
```

3. If the StorageEncrypted parameter value is `False`, encryption is not enabled for the selected RDS database instance.
4. Repeat steps 1 to 3 to audit each RDS instance, and change the region to verify RDS instances in other regions.

Remediation:

From Console:

1. Login to the AWS Management Console and open the RDS dashboard at https://console.aws.amazon.com/rds/.
2. In the left navigation panel, click on `Databases`.
3. Select the Database instance that needs to be encrypted.
4. Click the `Actions` button placed at the top right and select `Take Snapshot`.
5. On the Take Snapshot page, enter the name of the database for which you want to take a snapshot in the `Snapshot Name` field and click on `Take Snapshot`.
6. Select the newly created snapshot, click the `Action` button placed at the top right, and select `Copy snapshot` from the Action menu.
7. On the Make Copy of DB Snapshot page, perform the following:

- In the `New DB Snapshot Identifier` field, enter a name for the new snapshot.
- Check `Copy Tags`. The new snapshot must have the same tags as the source snapshot.
- Select `Yes` from the `Enable Encryption` dropdown list to enable encryption. You can choose to use the AWS default encryption key or a custom key from the Master Key dropdown list.

8. Click `Copy Snapshot` to create an encrypted copy of the selected instance's snapshot.

9. Select the new Snapshot Encrypted Copy and click the `Action` button located at the top right. Then, select the `Restore Snapshot` option from the Action menu. This will restore the encrypted snapshot to a new database instance.
10. On the Restore DB Instance page, enter a unique name for the new database instance in the DB Instance Identifier field.
11. Review the instance configuration details and click `Restore DB Instance`.
12. As the new instance provisioning process is completed, you can update the application configuration to refer to the endpoint of the new encrypted database instance. Once the database endpoint is changed at the application level, you can remove the unencrypted instance.

From Command Line:

1. Run the `describe-db-instances` command to list the names of all RDS database instances in the selected AWS region. The command output should return database instance identifiers:

```
aws rds describe-db-instances --region <region-name> --query
'DBInstances[*].DBInstanceIdentifier'
```

2. Run the `create-db-snapshot` command to create a snapshot for a selected database instance. The command output will return the `new snapshot` with name DB Snapshot Name:

```
aws rds create-db-snapshot --region <region-name> --db-snapshot-identifier
<db-snapshot-name> --db-instance-identifier <db-name>
```

3. Now run the `list-aliases` command to list the KMS key aliases available in a specified region. The command output should return each `key alias currently available`. For our RDS encryption activation process, locate the ID of the AWS default KMS key:

```
aws kms list-aliases --region <region-name>
```

4. Run the `copy-db-snapshot` command using the default KMS key ID for the RDS instances returned earlier to create an encrypted copy of the database instance snapshot. The command output will return the `encrypted instance snapshot configuration`:

```
aws rds copy-db-snapshot --region <region-name> --source-db-snapshot-
identifier <db-snapshot-name> --target-db-snapshot-identifier <db-snapshot-
name-encrypted> --copy-tags --kms-key-id <kms-id-for-rds>
```

5. Run the `restore-db-instance-from-db-snapshot` command to restore the encrypted snapshot created in the previous step to a new database instance. If

successful, the command output should return the configuration of the new encrypted database instance:

```
aws rds restore-db-instance-from-db-snapshot --region <region-name> --db-instance-identifier <db-name-encrypted> --db-snapshot-identifier <db-snapshot-name-encrypted>
```

6. Run the `describe-db-instances` command to list all RDS database names available in the selected AWS region. The output will return the database instance identifier names. Select the encrypted database name that we just created, `db-name-encrypted`:

```
aws rds describe-db-instances --region <region-name> --query 'DBInstances[*].DBInstanceIdentifier'
```

7. Run the `describe-db-instances` command again using the RDS instance identifier returned earlier to determine if the selected database instance is encrypted. The command output should indicate that the encryption status is `True`:

```
aws rds describe-db-instances --region <region-name> --db-instance-identifier <db-name-encrypted> --query 'DBInstances[*].StorageEncrypted'
```

References:

1. https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/Overview.Encryption.html
2. https://aws.amazon.com/blogs/database/selecting-the-right-encryption-options-for-amazon-rds-and-amazon-aurora-database-engines/#:~:text=With%20RDS%2Dencrypted%20resources%2C%20data,transparent%20to%20your%20database%20engine.
3. https://aws.amazon.com/rds/features/security/

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.11 Encrypt Sensitive Data at Rest Encrypt sensitive data at rest on servers, applications, and databases containing sensitive data. Storage-layer encryption, also known as server-side encryption, meets the minimum requirement of this Safeguard. Additional encryption methods may include application-layer encryption, also known as client-side encryption, where access to the data storage device(s) does not permit access to the plain-text data.		●	●

Controls Version	Control	IG 1	IG 2	IG 3
v7	14.8 Encrypt Sensitive Information at Rest Encrypt all sensitive information at rest using a tool that requires a secondary authentication mechanism not integrated into the operating system, in order to access the information.			●

2.2.2 Ensure the Auto Minor Version Upgrade feature is enabled for RDS instances (Automated)

Profile Applicability:

- Level 1

Description:

Ensure that RDS database instances have the Auto Minor Version Upgrade flag enabled to automatically receive minor engine upgrades during the specified maintenance window. This way, RDS instances can obtain new features, bug fixes, and security patches for their database engines.

Rationale:

AWS RDS will occasionally deprecate minor engine versions and provide new ones for upgrades. When the last version number within a release is replaced, the changed version is considered minor. With the Auto Minor Version Upgrade feature enabled, version upgrades will occur automatically during the specified maintenance window, allowing your RDS instances to receive new features, bug fixes, and security patches for their database engines.

Audit:

From Console:

1. Log in to the AWS management console and navigate to the RDS dashboard at https://console.aws.amazon.com/rds/.
2. In the left navigation panel, click `Databases`.
3. Select the RDS instance that you want to examine.
4. Click on the `Maintenance and backups` panel.
5. Under the `Maintenance` section, search for the Auto Minor Version Upgrade status.

- If the current status is `Disabled`, it means that the feature is not enabled, and the minor engine upgrades released will not be applied to the selected RDS instance.

From Command Line:

1. Run the `describe-db-instances` command to list all RDS database names available in the selected AWS region:

```
aws rds describe-db-instances --region <region-name> --query
'DBInstances[*].DBInstanceIdentifier'
```

2. The command output should return each database instance identifier.
3. Run the `describe-db-instances` command again using a RDS instance identifier returned earlier to determine the Auto Minor Version Upgrade status for the selected instance:

```
aws rds describe-db-instances --region <region-name> --db-instance-identifier
<db-instance-identifier> --query 'DBInstances[*].AutoMinorVersionUpgrade'
```

4. The command output should return the current status of the feature. If the current status is set to `true`, the feature is enabled and the minor engine upgrades will be applied to the selected RDS instance.

Remediation:

From Console:

1. Log in to the AWS management console and navigate to the RDS dashboard at https://console.aws.amazon.com/rds/.
2. In the left navigation panel, click `Databases`.
3. Select the RDS instance that you want to update.
4. Click on the `Modify` button located at the top right side.
5. On the `Modify DB Instance: <instance identifier>` page, In the `Maintenance` section, select `Auto minor version upgrade` and click the `Yes` radio button.
6. At the bottom of the page, click `Continue`, and check `Apply Immediately` to apply the changes immediately, or select `Apply during the next scheduled maintenance window` to avoid any downtime.
7. Review the changes and click `Modify DB Instance`. The instance status should change from available to modifying and back to available. Once the feature is enabled, the `Auto Minor Version Upgrade` status should change to `Yes`.

From Command Line:

1. Run the `describe-db-instances` command to list all RDS database instance names available in the selected AWS region:

```
aws rds describe-db-instances --region <region-name> --query
'DBInstances[*].DBInstanceIdentifier'
```

2. The command output should return each database instance identifier.
3. Run the `modify-db-instance` command to modify the configuration of a selected RDS instance. This command will apply the changes immediately. Remove `--apply-immediately` to apply changes during the next scheduled maintenance window and avoid any downtime:

```
aws rds modify-db-instance --region <region-name> --db-instance-identifier
<db-instance-identifier> --auto-minor-version-upgrade --apply-immediately
```

4. The command output should reveal the new configuration metadata for the RDS instance, including the `AutoMinorVersionUpgrade` parameter value.
5. Run the `describe-db-instances` command to check if the Auto Minor Version Upgrade feature has been successfully enabled:

```
aws rds describe-db-instances --region <region-name> --db-instance-identifier
<db-instance-identifier> --query 'DBInstances[*].AutoMinorVersionUpgrade'
```

6. The command output should return the feature's current status set to `true`, indicating that the feature is `enabled`, and that the minor engine upgrades will be applied to the selected RDS instance.

References:

1. https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/CHAP_RDS_Managing.html
2. https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/USER_UpgradeDBInstance.Upgrading.html
3. https://aws.amazon.com/rds/faqs/

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	7.4 Perform Automated Application Patch Management Perform application updates on enterprise assets through automated patch management on a monthly, or more frequent, basis.	●	●	●
v7	3.5 Deploy Automated Software Patch Management Tools Deploy automated software update tools in order to ensure that third-party software on all systems is running the most recent security updates provided by the software vendor.	●	●	●

2.2.3 Ensure that RDS instances are not publicly accessible (Automated)

Profile Applicability:

- Level 1

Description:

Ensure and verify that the RDS database instances provisioned in your AWS account restrict unauthorized access in order to minimize security risks. To restrict access to any RDS database instance, you must disable the Publicly Accessible flag for the database and update the VPC security group associated with the instance.

Rationale:

Ensure that no public-facing RDS database instances are provisioned in your AWS account, and restrict unauthorized access in order to minimize security risks. When the RDS instance allows unrestricted access (0.0.0.0/0), anyone and anything on the Internet can establish a connection to your database, which can increase the opportunity for malicious activities such as brute force attacks, PostgreSQL injections, or DoS/DDoS attacks.

Audit:

From Console:

1. Log in to the AWS management console and navigate to the RDS dashboard at https://console.aws.amazon.com/rds/.
2. Under the navigation panel, on the RDS dashboard, click `Databases`.
3. Select the RDS instance that you want to examine.
4. Click `Instance Name` from the dashboard, under `Connectivity and Security`.
5. In the `Security` section, check if the Publicly Accessible flag status is set to `Yes`.
6. Follow the steps below to check database subnet access:

- In the `networking` section, click the subnet link under `Subnets`.
- The link will redirect you to the VPC Subnets page.
- Select the subnet listed on the page and click the `Route Table` tab from the dashboard bottom panel.
- If the route table contains any entries with the destination CIDR block set to `0.0.0.0/0` and an `Internet Gateway` attached, the selected RDS database instance was provisioned inside a public subnet; therefore, it is not running within a logically isolated environment and can be accessed from the Internet.

7. Repeat steps 3-6 to determine the configuration of other RDS database instances provisioned in the current region.
8. Change the AWS region from the navigation bar and repeat the audit process for other regions.

From Command Line:

1. Run the `describe-db-instances` command to list all available RDS database names in the selected AWS region:

```
aws rds describe-db-instances --region <region-name> --query
'DBInstances[*].DBInstanceIdentifier'
```

2. The command output should return each database instance `identifier`.
3. Run the `describe-db-instances` command again, using the `PubliclyAccessible` parameter as a query filter to reveal the status of the database instance's Publicly Accessible flag:

```
aws rds describe-db-instances --region <region-name> --db-instance-identifier
<db-instance-name> --query 'DBInstances[*].PubliclyAccessible'
```

4. Check the Publicly Accessible parameter status. If the Publicly Accessible flag is set to `Yes`, then the selected RDS database instance is publicly accessible and insecure. Follow the steps mentioned below to check database subnet access.
5. Run the `describe-db-instances` command again using the RDS database instance identifier that you want to check, along with the appropriate filtering to describe the VPC subnet(s) associated with the selected instance:

```
aws rds describe-db-instances --region <region-name> --db-instance-identifier
<db-instance-name> --query 'DBInstances[*].DBSubnetGroup.Subnets[]'
```

- The command output should list the subnets available in the selected database subnet group.

6. Run the `describe-route-tables` command using the ID of the subnet returned in the previous step to describe the routes of the VPC route table associated with the selected subnet:

```
aws ec2 describe-route-tables --region <region-name> --filters
"Name=association.subnet-id,Values=<subnet-id>" --query
'RouteTables[*].Routes[]'
```

- If the command returns the route table associated with the database instance subnet ID, check the values of the `GatewayId` and `DestinationCidrBlock` attributes returned in the output. If the route table contains any entries with the `GatewayId` value set to `igw-xxxxxxxx` and the `DestinationCidrBlock` value

set to `0.0.0.0/0`, the selected RDS database instance was provisioned within a public subnet.
- Or, if the command returns empty results, the route table is implicitly associated with the subnet; therefore, the audit process continues with the next step.

7. Run the `describe-db-instances` command again using the RDS database instance identifier that you want to check, along with the appropriate filtering to describe the VPC ID associated with the selected instance:

```
aws rds describe-db-instances --region <region-name> --db-instance-identifier
<db-instance-name> --query 'DBInstances[*].DBSubnetGroup.VpcId'
```

- The command output should show the VPC ID in the selected database subnet group.

8. Now run the `describe-route-tables` command using the ID of the VPC returned in the previous step to describe the routes of the VPC's main route table that is implicitly associated with the selected subnet:

```
aws ec2 describe-route-tables --region <region-name> --filters "Name=vpc-
id,Values=<vpc-id>" "Name=association.main,Values=true" --query
'RouteTables[*].Routes[]'
```

- The command output returns the VPC main route table implicitly associated with the database instance subnet ID. Check the values of the `GatewayId` and `DestinationCidrBlock` attributes returned in the output. If the route table contains any entries with the `GatewayId` value set to `igw-xxxxxxxx` and the `DestinationCidrBlock` value set to `0.0.0.0/0`, the selected RDS database instance was provisioned inside a public subnet; therefore, it is not running within a logically isolated environment and does not adhere to AWS security best practices.

Remediation:

From Console:

1. Log in to the AWS management console and navigate to the RDS dashboard at https://console.aws.amazon.com/rds/.
2. Under the navigation panel, on the RDS dashboard, click `Databases`.
3. Select the RDS instance that you want to update.
4. Click `Modify` from the dashboard top menu.
5. On the Modify DB Instance panel, under the `Connectivity` section, click on `Additional connectivity configuration` and update the value for `Publicly Accessible` to `Not publicly accessible` to restrict public access.
6. Follow the below steps to update subnet configurations:

- Select the `Connectivity and security` tab, and click the VPC attribute value inside the `Networking` section.
- Select the `Details` tab from the VPC dashboard's bottom panel and click the Route table configuration attribute value.
- On the Route table details page, select the Routes tab from the dashboard's bottom panel and click `Edit routes`.
- On the Edit routes page, update the Destination of Target which is set to `igw-xxxxx` and click `Save` routes.

7. On the Modify DB Instance panel, click `Continue`, and in the Scheduling of modifications section, perform one of the following actions based on your requirements:

- Select `Apply during the next scheduled maintenance window` to apply the changes automatically during the next scheduled maintenance window.
- Select `Apply immediately` to apply the changes right away. With this option, any pending modifications will be asynchronously applied as soon as possible, regardless of the maintenance window setting for this RDS database instance. Note that any changes available in the pending modifications queue are also applied. If any of the pending modifications require downtime, choosing this option can cause unexpected downtime for the application.

8. Repeat steps 3-7 for each RDS instance in the current region.
9. Change the AWS region from the navigation bar to repeat the process for other regions.

From Command Line:

1. Run the `describe-db-instances` command to list all available RDS database identifiers in the selected AWS region:

```
aws rds describe-db-instances --region <region-name> --query
'DBInstances[*].DBInstanceIdentifier'
```

2. The command output should return each database instance identifier.
3. Run the `modify-db-instance` command to modify the configuration of a selected RDS instance, disabling the `Publicly Accessible` flag for that instance. This command uses the `apply-immediately` flag. If you want to avoid any downtime, the `--no-apply-immediately` flag can be used:

```
aws rds modify-db-instance --region <region-name> --db-instance-identifier
<db-instance-name> --no-publicly-accessible --apply-immediately
```

4. The command output should reveal the `PubliclyAccessible` configuration under pending values, to be applied at the specified time.

5. Updating the Internet Gateway destination via the AWS CLI is not currently supported. To update information about the Internet Gateway, please use the AWS Console procedure.
6. Repeat steps 1-5 for each RDS instance provisioned in the current region.
7. Change the AWS region by using the --region filter to repeat the process for other regions.

References:

1. https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/UsingWithRDS.html
2. https://docs.aws.amazon.com/vpc/latest/userguide/VPC_Scenario2.html
3. https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/USER_VPC.WorkingWithRDSInstanceinaVPC.html
4. https://aws.amazon.com/rds/faqs/

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.3 Configure Data Access Control Lists Configure data access control lists based on a user's need to know. Apply data access control lists, also known as access permissions, to local and remote file systems, databases, and applications.	●	●	●
v7	14.6 Protect Information through Access Control Lists Protect all information stored on systems with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities.	●	●	●

MITRE ATT&CK Mappings:

Techniques / Sub-techniques	Tactics	Mitigations
T1530	TA0010	M1037, M1054

2.2.4 Ensure Multi-AZ deployments are used for enhanced availability in Amazon RDS (Manual)

Profile Applicability:

- Level 1

Description:

Amazon RDS offers Multi-AZ deployments that provide enhanced availability and durability for your databases, using synchronous replication to replicate data to a standby instance in a different Availability Zone (AZ). In the event of an infrastructure failure, Amazon RDS automatically fails over to the standby to minimize downtime and ensure business continuity.

Rationale:

Database availability is crucial for maintaining service uptime, particularly for applications that are critical to the business. Implementing Multi-AZ deployments with Amazon RDS ensures that your databases are protected against unplanned outages due to hardware failures, network issues, or other disruptions. This configuration enhances both the availability and durability of your database, making it a highly recommended practice for production environments.

Impact:

Multi-AZ deployments may increase costs due to the additional resources required to maintain a standby instance; however, the benefits of increased availability and reduced risk of downtime outweigh these costs for critical applications.

Audit:

From Console:

1. Login to the AWS Management Console and open the RDS dashboard at AWS RDS Console.
2. In the navigation pane, under `Databases`, select the RDS instance you want to examine.
3. Click the `Instance Name` to see details, then navigate to the `Configuration` tab.
4. Under the `Availability & Durability` section, check the `Multi-AZ` status.
    - If Multi-AZ deployment is enabled, it will display `Yes`.
    - If it is disabled, the status will display `No`.
5. Repeat steps 2-4 to verify the Multi-AZ status of other RDS instances in the same region.
6. Change the region from the top of the navigation bar and repeat the audit for other regions.

From Command Line:

1. Run the following command to list all RDS instances in the selected AWS region:

```
aws rds describe-db-instances --region <region-name> --query
'DBInstances[*].DBInstanceIdentifier'
```

2. Run the following command using the instance identifier returned earlier to check the Multi-AZ status:

```
aws rds describe-db-instances --region <region-name> --db-instance-
identifier <db-name> --query 'DBInstances[*].MultiAZ'
```

   - If the output is `True`, Multi-AZ is enabled.
   - If the output is `False`, Multi-AZ is not enabled.
3. Repeat steps 1 and 2 to audit each RDS instance, and change regions to verify in other regions.

Remediation:

From Console:

1. Login to the AWS Management Console and open the RDS dashboard at AWS RDS Console.
2. In the left navigation pane, click on `Databases`.
3. Select the database instance that needs Multi-AZ deployment to be enabled.
4. Click the `Modify` button at the top right.
5. Scroll down to the `Availability & Durability` section.
6. Under `Multi-AZ deployment`, select `Yes` to enable.
7. Review the changes and click `Continue`.
8. On the `Review` page, choose `Apply immediately` to make the change without waiting for the next maintenance window, or `Apply during the next scheduled maintenance window`.
9. Click `Modify DB Instance` to apply the changes.

From Command Line:

1. Run the following command to modify the RDS instance and enable Multi-AZ:

```
aws rds modify-db-instance --region <region-name> --db-instance-
identifier <db-name> --multi-az --apply-immediately
```

2. Confirm that the Multi-AZ deployment is enabled by running the following command:

```
aws rds describe-db-instances --region <region-name> --db-instance-
identifier <db-name> --query 'DBInstances[*].MultiAZ'
```

   - The output should return `True`, indicating that Multi-AZ is enabled.
3. Repeat the procedure for other instances as necessary.

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	12.2 Establish and Maintain a Secure Network Architecture Establish and maintain a secure network architecture. A secure network architecture must address segmentation, least privilege, and availability, at a minimum.		●	●
v7	2.10 Physically or Logically Segregate High Risk Applications Physically or logically segregated systems should be used to isolate and run software that is required for business operations but incur higher risk for the organization.			●

2.3 Elastic File System (EFS)

2.3.1 Ensure that encryption is enabled for EFS file systems (Automated)

Profile Applicability:

- Level 1

Description:

EFS data should be encrypted at rest using AWS KMS (Key Management Service).

Rationale:

Data should be encrypted at rest to reduce the risk of a data breach via direct access to the storage device.

Audit:

From Console:

1. Login to the AWS Management Console and Navigate to the Elastic File System (EFS) dashboard.
2. Select `File Systems` from the left navigation panel.
3. Each item on the list has a visible Encrypted field that displays data at rest encryption status.
4. Validate that this field reads `Encrypted` for all EFS file systems in all AWS regions.

From CLI:

1. Run the `describe-file-systems` command using custom query filters to list the identifiers of all AWS EFS file systems currently available within the selected region:

```
aws efs describe-file-systems --region <region> --output table --query
'FileSystems[*].FileSystemId'
```

2. The command output should return a table with the requested file system IDs.
3. Run the `describe-file-systems` command using the ID of the file system that you want to examine as `file-system-id` and the necessary query filters:

```
aws efs describe-file-systems --region <region> --file-system-id <file-
system-id> --query 'FileSystems[*].Encrypted'
```

4. The command output should return the file system encryption status as `true` or `false`. If the returned value is `false`, the selected AWS EFS file system is not

encrypted and if the returned value is `true`, the selected AWS EFS file system is encrypted.

Remediation:

It is important to note that EFS file system data-at-rest encryption must be turned on when creating the file system. If an EFS file system has been created without data-at-rest encryption enabled, then you must create another EFS file system with the correct configuration and transfer the data.
Steps to create an EFS file system with data encrypted at rest:
From Console:

1. Login to the AWS Management Console and Navigate to the `Elastic File System (EFS)` dashboard.
2. Select `File Systems` from the left navigation panel.
3. Click the `Create File System` button from the dashboard top menu to start the file system setup process.
4. On the `Configure file system access` configuration page, perform the following actions:

- Choose an appropriate VPC from the VPC dropdown list.
- Within the `Create mount targets` section, check the boxes for all of the Availability Zones (AZs) within the selected VPC. These will be your mount targets.
- Click `Next step` to continue.

5. Perform the following on the `Configure optional settings` page:

- Create `tags` to describe your new file system.
- Choose `performance mode` based on your requirements.
- Check the `Enable encryption` box and choose `aws/elasticfilesystem` from the `Select KMS master key` dropdown list to enable encryption for the new file system, using the default master key provided and managed by AWS KMS.
- Click `Next step` to continue.

6. Review the file system configuration details on the `review and create` page and then click `Create File System` to create your new AWS EFS file system.
7. Copy the data from the old unencrypted EFS file system onto the newly created encrypted file system.
8. Remove the unencrypted file system as soon as your data migration to the newly created encrypted file system is completed.
9. Change the AWS region from the navigation bar and repeat the entire process for the other AWS regions.

From CLI:

1. Run the `describe-file-systems` command to view the configuration information for the selected unencrypted file system identified in the Audit steps:

```
aws efs describe-file-systems --region <region> --file-system-id <file-system-id>
```

2. The command output should return the configuration information.
3. To provision a new AWS EFS file system, you need to generate a universally unique identifier (UUID) to create the token required by the `create-file-system` command. To create the required token, you can use a randomly generated UUID from "https://www.uuidgenerator.net".
4. Run the `create-file-system` command using the unique token created at the previous step:

```
aws efs create-file-system --region <region> --creation-token <uuid> --performance-mode generalPurpose --encrypted
```

5. The command output should return the new file system configuration metadata.
6. Run the `create-mount-target` command using the EFS file system ID returned from step 4 as the identifier and the ID of the Availability Zone (AZ) that will represent the mount target:

```
aws efs create-mount-target --region <region> --file-system-id <file-system-id> --subnet-id <subnet-id>
```

7. The command output should return the new mount target metadata.
8. Now you can mount your file system from an EC2 instance.
9. Copy the data from the old unencrypted EFS file system to the newly created encrypted file system.
10. Remove the unencrypted file system as soon as your data migration to the newly created encrypted file system is completed:

```
aws efs delete-file-system --region <region> --file-system-id <unencrypted-file-system-id>
```

11. Change the AWS region by updating the --region and repeat the entire process for the other AWS regions.

Default Value:

EFS file system data is encrypted at rest by default when creating a file system through the Console. However, encryption at rest is not enabled by default when creating a new file system using the AWS CLI, API, or SDKs.

References:

1. https://docs.aws.amazon.com/efs/latest/ug/encryption-at-rest.html
2. https://awscli.amazonaws.com/v2/documentation/api/latest/reference/efs/index.html#efs

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.11 Encrypt Sensitive Data at Rest Encrypt sensitive data at rest on servers, applications, and databases containing sensitive data. Storage-layer encryption, also known as server-side encryption, meets the minimum requirement of this Safeguard. Additional encryption methods may include application-layer encryption, also known as client-side encryption, where access to the data storage device(s) does not permit access to the plain-text data.		●	●
v7	14.8 Encrypt Sensitive Information at Rest Encrypt all sensitive information at rest using a tool that requires a secondary authentication mechanism not integrated into the operating system, in order to access the information.			●

3 Logging

This section contains recommendations for configuring AWS logging features.

3.1 Ensure CloudTrail is enabled in all regions (Manual)

Profile Applicability:

- Level 1

Description:

AWS CloudTrail is a web service that records AWS API calls for your account and delivers log files to you. The recorded information includes the identity of the API caller, the time of the API call, the source IP address of the API caller, the request parameters, and the response elements returned by the AWS service. CloudTrail provides a history of AWS API calls for an account, including API calls made via the Management Console, SDKs, command line tools, and higher-level AWS services (such as CloudFormation).

Rationale:

The AWS API call history produced by CloudTrail enables security analysis, resource change tracking, and compliance auditing. Additionally,

- ensuring that a multi-region trail exists will help detect unexpected activity occurring in otherwise unused regions
- ensuring that a multi-region trail exists will ensure that `Global Service Logging` is enabled for a trail by default to capture recordings of events generated on AWS global services
- for a multi-region trail, ensuring that management events are configured for all types of Read/Writes ensures the recording of management operations that are performed on all resources in an AWS account

Impact:

S3 lifecycle features can be used to manage the accumulation and management of logs over time. See the following AWS resource for more information on these features:

1. https://docs.aws.amazon.com/AmazonS3/latest/dev/object-lifecycle-mgmt.html

Audit:

Perform the following to determine if CloudTrail is enabled for all regions:
From Console:

1. Sign in to the AWS Management Console and open the CloudTrail console at https://console.aws.amazon.com/cloudtrail
2. Click on `Trails` in the left navigation pane

- You will be presented with a list of trails across all regions

3. Ensure that at least one Trail has `Yes` specified in the `Multi-region trail` column
4. Click on a trail via the link in the `Name` column
5. Ensure `Logging` is set to `ON`
6. Ensure `Multi-region trail` is set to `Yes`
7. In the section `Management Events`, ensure that `API activity` set to `ALL`

From Command Line:

1. List all trails:

```
aws cloudtrail describe-trails
```

2. Ensure `IsMultiRegionTrail` is set to `true`:

```
aws cloudtrail get-trail-status --name <trail-name>
```

3. Ensure `IsLogging` is set to `true`:

```
aws cloudtrail get-event-selectors --trail-name <trail-name>
```

4. Ensure there is at least one `fieldSelector` for a trail that equals `Management`:

- This should NOT output any results for Field: "readOnly". If either `true` or `false` is returned, one of the checkboxes (`read` or `write`) is not selected.

Example of correct output:

```
"TrailARN": "<your_trail_ARN>",
    "AdvancedEventSelectors": [
        {
            "Name": "Management events selector",
            "FieldSelectors": [
                {
                    "Field": "eventCategory",
                    "Equals": [
                        "Management"
                    ]
```

Remediation:

Perform the following to enable global (Multi-region) CloudTrail logging:
From Console:

1. Sign in to the AWS Management Console and open the IAM console at https://console.aws.amazon.com/cloudtrail.
2. Click on `Trails` in the left navigation pane.

3. Click `Get Started Now` if it is presented, then:

- Click `Add new trail`.
- Enter a trail name in the `Trail name` box.
    - A trail created in the console is a multi-region trail by default.
- Specify an S3 bucket name in the `S3 bucket` box.
- Specify the AWS KMS alias under the `Log file SSE-KMS encryption` section, or create a new key.
- Click `Next`.

4. Ensure the `Management events` check box is selected.
5. Ensure both `Read` and `Write` are checked under API activity.
6. Click `Next`.
7. Review your trail settings and click `Create trail`.

From Command Line:
Create a multi-region trail:

```
aws cloudtrail create-trail --name <trail-name> --bucket-name <s3-bucket-for-cloudtrail> --is-multi-region-trail
```

Enable multi-region on an existing trail:

```
aws cloudtrail update-trail --name <trail-name> --is-multi-region-trail
```

Note: Creating a CloudTrail trail via the CLI without providing any overriding options configures all `read` and `write Management Events` to be logged by default.

Default Value:

Not Enabled

References:

1. CCE-78913-1
2. https://docs.aws.amazon.com/awscloudtrail/latest/userguide/cloudtrail-concepts.html#cloudtrail-concepts-management-events
3. https://docs.aws.amazon.com/awscloudtrail/latest/userguide/logging-management-and-data-events-with-cloudtrail.html?icmpid=docs_cloudtrail_console#logging-management-events
4. https://docs.aws.amazon.com/awscloudtrail/latest/userguide/cloudtrail-supported-services.html#cloud-trail-supported-services-data-events

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	8.5 Collect Detailed Audit Logs Configure detailed audit logging for enterprise assets containing sensitive data. Include event source, date, username, timestamp, source addresses, destination addresses, and other useful elements that could assist in a forensic investigation.		●	●
v7	6.2 Activate audit logging Ensure that local logging has been enabled on all systems and networking devices.	●	●	●

3.2 Ensure CloudTrail log file validation is enabled (Automated)

Profile Applicability:

- Level 2

Description:

CloudTrail log file validation creates a digitally signed digest file containing a hash of each log that CloudTrail writes to S3. These digest files can be used to determine whether a log file was changed, deleted, or remained unchanged after CloudTrail delivered the log. It is recommended that file validation be enabled for all CloudTrails.

Rationale:

Enabling log file validation will provide additional integrity checks for CloudTrail logs.

Audit:

Perform the following on each trail to determine if log file validation is enabled:
From Console:

1. Sign in to the AWS Management Console and open the IAM console at https://console.aws.amazon.com/cloudtrail.
2. Click on `Trails` in the left navigation pane.
3. For every trail:

- Click on a trail via the link in the `Name` column.
- Under the `General details` section, ensure `Log file validation` is set to `Enabled`.

From Command Line:
List all trails:

```
aws cloudtrail describe-trails
```

Ensure `LogFileValidationEnabled` is set to `true` for each trail.

Remediation:

Perform the following to enable log file validation on a given trail:
From Console:

1. Sign in to the AWS Management Console and open the IAM console at https://console.aws.amazon.com/cloudtrail.
2. Click on `Trails` in the left navigation pane.
3. Click on the target trail.
4. Within the `General details` section, click `edit`.
5. Under `Advanced settings`, check the `enable` box under `Log file validation`.

6. Click Save changes.

From Command Line:
Enable log file validation on a trail:

```
aws cloudtrail update-trail --name <trail_name> --enable-log-file-validation
```

Note that periodic validation of logs using these digests can be carried out by running the following command:

```
aws cloudtrail validate-logs --trail-arn <trail_arn> --start-time
<start_time> --end-time <end_time>
```

Default Value:

Not Enabled

References:

1. https://docs.aws.amazon.com/awscloudtrail/latest/userguide/cloudtrail-log-file-validation-enabling.html
2. CCE-78914-9

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	8.11 Conduct Audit Log Reviews Conduct reviews of audit logs to detect anomalies or abnormal events that could indicate a potential threat. Conduct reviews on a weekly, or more frequent, basis.		●	●
v7	6.1 Utilize Three Synchronized Time Sources Use at least three synchronized time sources from which all servers and network devices retrieve time information on a regular basis so that timestamps in logs are consistent.		●	●

3.3 Ensure AWS Config is enabled in all regions (Automated)

Profile Applicability:

- Level 2

Description:

AWS Config is a web service that performs configuration management of supported AWS resources within your account and delivers log files to you. The recorded information includes the configuration items (AWS resources), relationships between configuration items (AWS resources), and any configuration changes between resources. It is recommended that AWS Config be enabled in all regions.

Rationale:

The AWS configuration item history captured by AWS Config enables security analysis, resource change tracking, and compliance auditing.

Impact:

Enabling AWS Config in all regions provides comprehensive visibility into resource configurations, enhancing security and compliance monitoring. However, this may incur additional costs and require proper configuration management.

Audit:

Process to evaluate AWS Config configuration per region:
From Console:

1. Sign in to the AWS Management Console and open the AWS Config console at https://console.aws.amazon.com/config/.
2. On the top right of the console select the target region.
3. If a Config Recorder is enabled in this region, you should navigate to the Settings page from the navigation menu on the left-hand side. If a Config Recorder is not yet enabled in this region, proceed to the remediation steps.
4. Ensure "Record all resources supported in this region" is checked.
5. Ensure "Include global resources (e.g., AWS IAM resources)" is checked, unless it is enabled in another region (this is only required in one region).
6. Ensure the correct S3 bucket has been defined.
7. Ensure the correct SNS topic has been defined.
8. Repeat steps 2 to 7 for each region.

From Command Line:

1. Run this command to show all AWS Config Recorders and their properties:

```
aws configservice describe-configuration-recorders
```

2. Evaluate the output to ensure that all recorders have a `recordingGroup` object which includes `"allSupported": true`. Additionally, ensure that at least one recorder has `"includeGlobalResourceTypes": true`.

Note: There is one more parameter, "ResourceTypes," in the recordingGroup object. We don't need to check it, as whenever we set "allSupported" to true, AWS enforces the resource types to be empty ("ResourceTypes": []).
Sample output:

```
{
    "ConfigurationRecorders": [
        {
            "recordingGroup": {
                "allSupported": true,
                "resourceTypes": [],
                "includeGlobalResourceTypes": true
            },
            "roleARN": "arn:aws:iam::<AWS_Account_ID>:role/service-
role/<config-role-name>",
            "name": "default"
        }
    ]
}
```

3. Run this command to show the status for all AWS Config Recorders:

```
aws configservice describe-configuration-recorder-status
```

4. In the output, find recorders with `name` key matching the recorders that were evaluated in step 2. Ensure that they include `"recording": true` and `"lastStatus": "SUCCESS"`.

Remediation:

To implement AWS Config configuration:
From Console:

1. Select the region you want to focus on in the top right of the console.
2. Click `Services`.
3. Click `Config`.
4. If a Config Recorder is enabled in this region, navigate to the Settings page from the navigation menu on the left-hand side. If a Config Recorder is not yet enabled in this region, select "Get Started".
5. Select "Record all resources supported in this region".
6. Choose to include global resources (IAM resources).
7. Specify an S3 bucket in the same account or in another managed AWS account.
8. Create an SNS Topic from the same AWS account or another managed AWS account.

From Command Line:

1. Ensure there is an appropriate S3 bucket, SNS topic, and IAM role per the AWS Config Service prerequisites.
2. Run this command to create a new configuration recorder:

```
aws configservice put-configuration-recorder --configuration-recorder
name=<config-recorder-name>,roleARN=arn:aws:iam::<account-id>:role/<iam-role>
--recording-group allSupported=true,includeGlobalResourceTypes=true
```

3. Create a delivery channel configuration file locally which specifies the channel attributes, populated from the prerequisites set up previously:

```
{
  "name": "<delivery-channel-name>",
  "s3BucketName": "<bucket-name>",
  "snsTopicARN": "arn:aws:sns:<region>:<account-id>:<sns-topic>",
  "configSnapshotDeliveryProperties": {
    "deliveryFrequency": "Twelve_Hours"
  }
}
```

4. Run this command to create a new delivery channel, referencing the json configuration file made in the previous step:

```
aws configservice put-delivery-channel --delivery-channel file://<delivery-
channel-file>.json
```

5. Start the configuration recorder by running the following command:

```
aws configservice start-configuration-recorder --configuration-recorder-name
<config-recorder-name>
```

References:

1. CCE-78917-2
2. https://awscli.amazonaws.com/v2/documentation/api/latest/reference/configservice/describe-configuration-recorder-status.html
3. https://awscli.amazonaws.com/v2/documentation/api/latest/reference/configservice/describe-configuration-recorders.html
4. https://docs.aws.amazon.com/config/latest/developerguide/gs-cli-prereq.html

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	1.1 Establish and Maintain Detailed Enterprise Asset Inventory Establish and maintain an accurate, detailed, and up-to-date inventory of all enterprise assets with the potential to store or process data, to include: end-user devices (including portable and mobile), network devices, non-computing/IoT devices, and servers. Ensure the inventory records the network address (if static), hardware address, machine name, enterprise asset owner, department for each asset, and whether the asset has been approved to connect to the network. For mobile end-user devices, MDM type tools can support this process, where appropriate. This inventory includes assets connected to the infrastructure physically, virtually, remotely, and those within cloud environments. Additionally, it includes assets that are regularly connected to the enterprise's network infrastructure, even if they are not under control of the enterprise. Review and update the inventory of all enterprise assets bi-annually, or more frequently.	●	●	●
v7	1.4 Maintain Detailed Asset Inventory Maintain an accurate and up-to-date inventory of all technology assets with the potential to store or process information. This inventory shall include all hardware assets, whether connected to the organization's network or not.	●	●	●

3.4 Ensure that server access logging is enabled on the CloudTrail S3 bucket (Manual)

Profile Applicability:

- Level 1

Description:

Server 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 server access logging be enabled on the CloudTrail S3 bucket.

Rationale:

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

Audit:

Perform the following ensure that the CloudTrail S3 bucket has access logging is enabled:
From Console:

1. Go to the Amazon CloudTrail console at https://console.aws.amazon.com/cloudtrail/home.
2. In the API activity history pane on the left, click `Trails`.
3. In the Trails pane, note the bucket names in the S3 bucket column.
4. Sign in to the AWS Management Console and open the S3 console at https://console.aws.amazon.com/s3.
5. Under `All Buckets` click on a target S3 bucket.
6. Click on `Properties` in the top right of the console.
7. Under `Bucket: <bucket-name>`, click `Logging`.
8. Ensure `Enabled` is checked.

From Command Line:

1. Get the name of the S3 bucket that CloudTrail is logging to:

```
aws cloudtrail describe-trails --query 'trailList[*].S3BucketName'
```

2. Ensure logging is enabled on the bucket:

```
aws s3api get-bucket-logging --bucket <s3-bucket-for-cloudtrail>
```

Ensure the command does not return an empty output.
Sample output for a bucket with logging enabled:

```
{
    "LoggingEnabled": {
        "TargetPrefix": "<log-file-prefix>",
        "TargetBucket": "<logging-bucket>"
    }
}
```

Remediation:

Perform the following to enable server access logging:
From Console:

1. Sign in to the AWS Management Console and open the S3 console at https://console.aws.amazon.com/s3.
2. Under `All Buckets` click on the target S3 bucket.
3. Click on `Properties` in the top right of the console.
4. Under `Bucket: <bucket-name>`, click `Logging`.
5. Configure bucket logging:
    - Check the `Enabled` box.
    - Select a Target Bucket from the list.
    - Enter a Target Prefix.
6. Click `Save`.

From Command Line:

1. Get the name of the S3 bucket that CloudTrail is logging to:

```
aws cloudtrail describe-trails --region <region-name> --query
trailList[*].S3BucketName
```

2. Copy and add the target bucket name at `<bucket-name>`, the prefix for the log file at `<log-file-prefix>`, and optionally add an email address in the following template, then save it as `<file-name>.json`:

```
{
        "LoggingEnabled": {
                "TargetBucket": "<bucket-name>",
                "TargetPrefix": "<log-file-prefix>",
                "TargetGrants": [
                        {
                        "Grantee": {
                                "Type": "AmazonCustomerByEmail",
                                "EmailAddress": "<email-address>"
                                },
                        "Permission": "FULL_CONTROL"
                        }
                ]
        }
}
```

3. Run the `put-bucket-logging` command with bucket name and `<file-name>.json` as input; for more information, refer to put-bucket-logging:

```
aws s3api put-bucket-logging --bucket <bucket-name> --bucket-logging-status
file://<file-name>.json
```

Default Value:

Logging is disabled.

References:

1. CCE-78918-0
2. https://docs.aws.amazon.com/AmazonS3/latest/dev/ServerLogs.html
3. https://docs.aws.amazon.com/AmazonS3/latest/userguide/enable-server-access-logging.html

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.14 Log Sensitive Data Access Log sensitive data access, including modification and disposal.			●
v8	8.2 Collect Audit Logs Collect audit logs. Ensure that logging, per the enterprise's audit log management process, has been enabled across enterprise assets.	●	●	●
v7	6.2 Activate audit logging Ensure that local logging has been enabled on all systems and networking devices.	●	●	●

Controls Version	Control	IG 1	IG 2	IG 3
v7	14.9 Enforce Detail Logging for Access or Changes to Sensitive Data Enforce detailed audit logging for access to sensitive data or changes to sensitive data (utilizing tools such as File Integrity Monitoring or Security Information and Event Monitoring).			●

3.5 Ensure CloudTrail logs are encrypted at rest using KMS CMKs (Automated)

Profile Applicability:

- Level 2

Description:

AWS CloudTrail is a web service that records AWS API calls for an account and makes those logs available to users and resources in accordance with IAM policies. AWS Key Management Service (KMS) is a managed service that helps create and control the encryption keys used to encrypt account data, and uses Hardware Security Modules (HSMs) to protect the security of encryption keys. CloudTrail logs can be configured to leverage server side encryption (SSE) and KMS customer-created master keys (CMK) to further protect CloudTrail logs. It is recommended that CloudTrail be configured to use SSE-KMS.

Rationale:

Configuring CloudTrail to use SSE-KMS provides additional confidentiality controls on log data, as a given user must have S3 read permission on the corresponding log bucket and must be granted decrypt permission by the CMK policy.

Impact:

Customer-created keys incur an additional cost. See https://aws.amazon.com/kms/pricing/ for more information.

Audit:

Perform the following to determine if CloudTrail is configured to use SSE-KMS:
From Console:

1. Sign in to the AWS Management Console and open the CloudTrail console at https://console.aws.amazon.com/cloudtrail.
2. In the left navigation pane, choose `Trails`.
3. Select a trail.
4. In the `General details` section, select `Edit` to edit the trail configuration.
5. Ensure the box at `Log file SSE-KMS encryption` is checked and that a valid `AWS KMS alias` of a KMS key is entered in the respective text box.

From Command Line:

1. Run the following command:

```
aws cloudtrail describe-trails
```

2. For each trail listed, SSE-KMS is enabled if the trail has a `KmsKeyId` property defined.

Remediation:

Perform the following to configure CloudTrail to use SSE-KMS:
From Console:

1. Sign in to the AWS Management Console and open the CloudTrail console at https://console.aws.amazon.com/cloudtrail.
2. In the left navigation pane, choose `Trails`.
3. Click on a trail.
4. Under the `S3` section, click the edit button (pencil icon).
5. Click `Advanced`.
6. Select an existing CMK from the `KMS key Id` drop-down menu.

- **Note:** Ensure the CMK is located in the same region as the S3 bucket.
- **Note:** You will need to apply a KMS key policy on the selected CMK in order for CloudTrail, as a service, to encrypt and decrypt log files using the CMK provided. View the AWS documentation for editing the selected CMK Key policy.

7. Click `Save`.
8. You will see a notification message stating that you need to have decryption permissions on the specified KMS key to decrypt log files.
9. Click `Yes`.

From Command Line:
Run the following command to specify a KMS key ID to use with a trail:

```
aws cloudtrail update-trail --name <trail-name>  --kms-id <cloudtrail-kms-key>
```

Run the following command to attach a key policy to a specified KMS key:

```
aws kms put-key-policy --key-id <cloudtrail-kms-key> --policy <cloudtrail-kms-key-policy>
```

References:

1. https://docs.aws.amazon.com/awscloudtrail/latest/userguide/encrypting-cloudtrail-log-files-with-aws-kms.html
2. https://docs.aws.amazon.com/kms/latest/developerguide/create-keys.html
3. CCE-78919-8
4. https://awscli.amazonaws.com/v2/documentation/api/latest/reference/cloudtrail/update-trail.html
5. https://awscli.amazonaws.com/v2/documentation/api/latest/reference/kms/put-key-policy.html

Additional Information:

Three statements that need to be added to the CMK policy:

1. Enable CloudTrail to describe CMK properties:

```
<pre class="programlisting" style="font-style: normal;">{
  "Sid": "Allow CloudTrail access",
  "Effect": "Allow",
  "Principal": {
    "Service": "cloudtrail.amazonaws.com"
  },
  "Action": "kms:DescribeKey",
  "Resource": "*"
}
```

2. Granting encrypt permissions:

```
<pre class="programlisting" style="font-style: normal;">{
  "Sid": "Allow CloudTrail to encrypt logs",
  "Effect": "Allow",
  "Principal": {
    "Service": "cloudtrail.amazonaws.com"
  },
  "Action": "kms:GenerateDataKey*",
  "Resource": "*",
  "Condition": {
    "StringLike": {
      "kms:EncryptionContext:aws:cloudtrail:arn": [
        "arn:aws:cloudtrail:*:aws-account-id:trail/*"
      ]
    }
  }
}
```

3. Granting decrypt permissions:

```
<pre class="programlisting" style="font-style: normal;">{
  "Sid": "Enable CloudTrail log decrypt permissions",
  "Effect": "Allow",
  "Principal": {
    "AWS": "arn:aws:iam::aws-account-id:user/username"
  },
  "Action": "kms:Decrypt",
  "Resource": "*",
  "Condition": {
    "Null": {
      "kms:EncryptionContext:aws:cloudtrail:arn": "false"
    }
  }
}
```

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.11 Encrypt Sensitive Data at Rest Encrypt sensitive data at rest on servers, applications, and databases containing sensitive data. Storage-layer encryption, also known as server-side encryption, meets the minimum requirement of this Safeguard. Additional encryption methods may include application-layer encryption, also known as client-side encryption, where access to the data storage device(s) does not permit access to the plain-text data.		●	●
v8	8.1 Establish and Maintain an Audit Log Management Process Establish and maintain an audit log management process that defines the enterprise's logging requirements. At a minimum, address the collection, review, and retention of audit logs for enterprise assets. Review and update documentation annually, or when significant enterprise changes occur that could impact this Safeguard.	●	●	●
v7	14.8 Encrypt Sensitive Information at Rest Encrypt all sensitive information at rest using a tool that requires a secondary authentication mechanism not integrated into the operating system, in order to access the information.			●

3.6 Ensure rotation for customer-created symmetric CMKs is enabled (Automated)

Profile Applicability:

- Level 2

Description:

AWS Key Management Service (KMS) allows customers to rotate the backing key, which is key material stored within the KMS that is tied to the key ID of the customer-created customer master key (CMK). The backing key is used to perform cryptographic operations such as encryption and decryption. Automated key rotation currently retains all prior backing keys so that decryption of encrypted data can occur transparently. It is recommended that CMK key rotation be enabled for symmetric keys. Key rotation cannot be enabled for any asymmetric CMK.

Rationale:

Rotating encryption keys helps reduce the potential impact of a compromised key, as data encrypted with a new key cannot be accessed with a previous key that may have been exposed. Keys should be rotated every year or upon an event that could result in the compromise of that key.

Impact:

Creation, management, and storage of CMKs may require additional time from an administrator.

Audit:

From Console:

1. Sign in to the AWS Management Console and open the KMS console at: https://console.aws.amazon.com/kms.
2. In the left navigation pane, click `Customer-managed keys`.
3. Select a customer-managed CMK where `Key spec = SYMMETRIC_DEFAULT`.
4. Select the `Key rotation` tab.
5. Ensure the `Automatically rotate this KMS key every year` box is checked.
6. Repeat steps 3–5 for all customer-managed CMKs where `Key spec = SYMMETRIC_DEFAULT`.

From Command Line:

1. Run the following command to get a list of all keys and their associated `KeyIds`:

```
aws kms list-keys
```

2. For each key, note the KeyId and run the following command:

```
describe-key --key-id <kms-key-id>
```

3. If the response contains `"KeySpec = SYMMETRIC_DEFAULT"`, run the following command:

```
aws kms get-key-rotation-status --key-id <kms-key-id>
```

4. Ensure `KeyRotationEnabled` is set to `true`.
5. Repeat steps 2–4 for all remaining CMKs.

Remediation:

From Console:

1. Sign in to the AWS Management Console and open the KMS console at: https://console.aws.amazon.com/kms.
2. In the left navigation pane, click `Customer-managed keys`.
3. Select a key with `Key spec = SYMMETRIC_DEFAULT` that does not have automatic rotation enabled.
4. Select the `Key rotation` tab.
5. Check the `Automatically rotate this KMS key every year` box.
6. Click `Save`.
7. Repeat steps 3–6 for all customer-managed CMKs that do not have automatic rotation enabled.

From Command Line:

1. Run the following command to enable key rotation:

```
aws kms enable-key-rotation --key-id <kms-key-id>
```

References:

1. https://aws.amazon.com/kms/pricing/
2. https://csrc.nist.gov/publications/detail/sp/800-57-part-1/rev-5/final
3. CCE-78920-6

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.11 Encrypt Sensitive Data at Rest Encrypt sensitive data at rest on servers, applications, and databases containing sensitive data. Storage-layer encryption, also known as server-side encryption, meets the minimum requirement of this Safeguard. Additional encryption methods may include application-layer encryption, also known as client-side encryption, where access to the data storage device(s) does not permit access to the plain-text data.		●	●
v7	14.8 Encrypt Sensitive Information at Rest Encrypt all sensitive information at rest using a tool that requires a secondary authentication mechanism not integrated into the operating system, in order to access the information.			●

3.7 Ensure VPC flow logging is enabled in all VPCs (Automated)

Profile Applicability:

- Level 2

Description:

VPC Flow Logs is a feature that enables you to capture information about the IP traffic going to and from network interfaces in your VPC. After you've created a flow log, you can view and retrieve its data in Amazon CloudWatch Logs. It is recommended that VPC Flow Logs be enabled for packet "Rejects" for VPCs.

Rationale:

VPC Flow Logs provide visibility into network traffic that traverses the VPC and can be used to detect anomalous traffic or gain insights during security workflows.

Impact:

By default, CloudWatch Logs will store logs indefinitely unless a specific retention period is defined for the log group. When choosing the number of days to retain, keep in mind that the average time it takes for an organization to realize they have been breached is 210 days (at the time of this writing). Since additional time is required to research a breach, a minimum retention policy of 365 days allows for detection and investigation. You may also wish to archive the logs to a cheaper storage service rather than simply deleting them. See the following AWS resource to manage CloudWatch Logs retention periods:

1. https://docs.aws.amazon.com/AmazonCloudWatch/latest/DeveloperGuide/SettingLogRetention.html

Audit:

Perform the following to determine if VPC Flow logs are enabled:
From Console:

1. Sign into the management console.
2. Select `Services`, then select `VPC`.
3. In the left navigation pane, select `Your VPCs`.
4. Select a VPC.
5. In the right pane, select the `Flow Logs` tab.
6. Ensure a Log Flow exists that has `Active` in the `Status` column.

From Command Line:

1. Run the `describe-vpcs` command (OSX/Linux/UNIX) to list the VPC networks available in the current AWS region:

```
aws ec2 describe-vpcs --region <region> --query Vpcs[].VpcId
```

2. The command output returns the `VpcId` of VPCs available in the selected region.
3. Run the `describe-flow-logs` command (OSX/Linux/UNIX) using the VPC ID to determine if the selected virtual network has the Flow Logs feature enabled:

```
aws ec2 describe-flow-logs --filter "Name=resource-id,Values=<vpc-id>"
```

- If there are no Flow Logs created for the selected VPC, the command output will return an empty list `[]`.

4. Repeat step 3 for other VPCs in the same region.
5. Change the region by updating `--region`, and repeat steps 1-4 for each region.

Remediation:

Perform the following to enable VPC Flow Logs:
From Console:

1. Sign into the management console.
2. Select `Services`, then select `VPC`.
3. In the left navigation pane, select `Your VPCs`.
4. Select a VPC.
5. In the right pane, select the `Flow Logs` tab.
6. If no Flow Log exists, click `Create Flow Log`.
7. For Filter, select `Reject`.
8. Enter a `Role` and `Destination Log Group`.
9. Click `Create Log Flow`.
10. Click on `CloudWatch Logs Group`.

Note: Setting the filter to "Reject" will dramatically reduce the accumulation of logging data for this recommendation and provide sufficient information for the purposes of breach detection, research, and remediation. However, during periods of least privilege security group engineering, setting the filter to "All" can be very helpful in discovering existing traffic flows required for the proper operation of an already running environment.
From Command Line:

1. Create a policy document, name it `role_policy_document.json`, and paste the following content:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "test",
            "Effect": "Allow",
            "Principal": {
                "Service": "ec2.amazonaws.com"
            },
            "Action": "sts:AssumeRole"
        }
    ]
}
```

2. Create another policy document, name it `iam_policy.json`, and paste the following content:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action":[
                "logs:CreateLogGroup",
                "logs:CreateLogStream",
                "logs:DescribeLogGroups",
                "logs:DescribeLogStreams",
                "logs:PutLogEvents",
                "logs:GetLogEvents",
                "logs:FilterLogEvents"
            ],
            "Resource": "*"
        }
    ]
}
```

3. Run the following command to create an IAM role:

```
aws iam create-role --role-name <aws-support-iam-role> --assume-role-policy-
document file://<file-path>role_policy_document.json
```

4. Run the following command to create an IAM policy:

```
aws iam create-policy --policy-name <iam-policy-name> --policy-document
file://<file-path>iam-policy.json
```

5. Run the `attach-group-policy` command, using the IAM policy ARN returned from the previous step to attach the policy to the IAM role:

```
aws iam attach-group-policy --policy-arn arn:aws:iam::<aws-account-
id>:policy/<iam-policy-name> --group-name <group-name>
```

- If the command succeeds, no output is returned.

6. Run the `describe-vpcs` command to get a list of VPCs in the selected region:

```
aws ec2 describe-vpcs --region <region>
```

- The command output should return a list of VPCs in the selected region.

7. Run the `create-flow-logs` command to create a flow log for a VPC:

```
aws ec2 create-flow-logs --resource-type VPC --resource-ids <vpc-id> --
traffic-type REJECT --log-group-name <log-group-name> --deliver-logs-
permission-arn <iam-role-arn>
```

8. Repeat step 7 for other VPCs in the selected region.
9. Change the region by updating --region, and repeat the remediation procedure for each region.

References:

1. CCE-79202-8
2. https://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/flow-logs.html

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	8.2 Collect Audit Logs Collect audit logs. Ensure that logging, per the enterprise's audit log management process, has been enabled across enterprise assets.	●	●	●
v8	13.6 Collect Network Traffic Flow Logs Collect network traffic flow logs and/or network traffic to review and alert upon from network devices.		●	●
v7	6.2 Activate audit logging Ensure that local logging has been enabled on all systems and networking devices.	●	●	●

Controls Version	Control	IG 1	IG 2	IG 3
v7	12.5 Configure Monitoring Systems to Record Network Packets Configure monitoring systems to record network packets passing through the boundary at each of the organization's network boundaries.		●	●

3.8 Ensure that object-level logging for write events is enabled for S3 buckets (Automated)

Profile Applicability:

- Level 2

Description:

S3 object-level API operations, such as GetObject, DeleteObject, and PutObject, are referred to as data events. By default, CloudTrail trails do not log data events, so it is recommended to enable object-level logging for S3 buckets.

Rationale:

Enabling object-level logging will help you meet data compliance requirements within your organization, perform comprehensive security analyses, monitor specific patterns of user behavior in your AWS account, or take immediate actions on any object-level API activity within your S3 buckets using Amazon CloudWatch Events.

Impact:

Enabling logging for these object-level events may significantly increase the number of events logged and may incur additional costs.

Audit:

From Console:

1. Login to the AWS Management Console and navigate to the CloudTrail dashboard at `https://console.aws.amazon.com/cloudtrail/`.
2. In the left panel, click `Trails`, and then click the name of the trail that you want to examine.
3. Review `General details`.
4. Confirm that `Multi-region trail` is set to `Yes`.
5. Scroll down to `Data events` and confirm the configuration:

- If `advanced event selectors` is being used, it should read:

```
Data Events: S3
Log selector template
Log all events
```

- If `basic event selectors` is being used, it should read:

```
Data events: S3
Bucket Name: All current and future S3 buckets
Write: Enabled
```

6. Repeat steps 2-5 to verify that each trail has multi-region enabled and is configured to log data events. If a trail does not have multi-region enabled and data event logging configured, refer to the remediation steps.

From Command Line:

1. Run the `list-trails` command to list all trails:

```
aws cloudtrail list-trails
```

2. The command output will be a list of trails:

```
"TrailARN": "arn:aws:cloudtrail:<region>:<account#>:trail/<trail-name>",
"Name": "<trail-name>",
"HomeRegion": "<region>"
```

3. Run the `get-trail` command to determine whether a trail is a multi-region trail:

```
aws cloudtrail get-trail --name <trail-name> --region <region-name>
```

4. The command output should include: `"IsMultiRegionTrail": true`.
5. Run the `get-event-selectors` command, using the `Name` of the trail and the `region` returned in step 2, to determine if data event logging is configured:

```
aws cloudtrail get-event-selectors --region <home-region> --trail-name
<trail-name> --query EventSelectors[*].DataResources[]
```

6. The command output should be an array that includes the S3 bucket defined for data event logging:

```
"Type": "AWS::S3::Object",
        "Values": [
            "arn:aws:s3"
```

7. If the `get-event-selectors` command returns an empty array, data events are not included in the trail's logging configuration; therefore, object-level API operations performed on S3 buckets within your AWS account are not being recorded.
8. Repeat steps 1-7 to verify that each trail has multi-region enabled and is configured to log data events. If a trail does not have multi-region enabled and data event logging configured, refer to the remediation steps.

Remediation:

From Console:

1. Login to the AWS Management Console and navigate to the S3 dashboard at `https://console.aws.amazon.com/s3/`.
2. In the left navigation panel, click `buckets`, and then click the name of the S3 bucket you want to examine.
3. Click the `Properties` tab to see the bucket configuration in detail.
4. In the `AWS CloudTrail data events` section, select the trail name for recording activity. You can choose an existing trail or create a new one by clicking the `Configure in CloudTrail` button or navigating to the CloudTrail console.
5. Once the trail is selected, select the `Data Events` check box.
6. Select `S3` from the `Data event type` drop-down.
7. Select `Log all events` from the `Log selector template` drop-down.
8. Repeat steps 2-7 to enable object-level logging of write events for other S3 buckets.

From Command Line:

1. To enable `object-level` data events logging for S3 buckets within your AWS account, run the `put-event-selectors` command using the name of the trail that you want to reconfigure as identifier:

```
aws cloudtrail put-event-selectors --region <region-name> --trail-name
<trail-name> --event-selectors '[{ "ReadWriteType": "WriteOnly",
"IncludeManagementEvents":true, "DataResources": [{ "Type":
"AWS::S3::Object", "Values": ["arn:aws:s3:::<s3-bucket-name>/"] }] }]'
```

2. The command output will be `object-level` event trail configuration.
3. If you want to enable it for all buckets at once, change the Values parameter to `["arn:aws:s3"]` in the previous command.
4. Repeat step 1 for each s3 bucket to update `object-level` logging of write events.
5. Change the AWS region by updating the `--region` command parameter, and perform the process for the other regions.

References:

1. https://docs.aws.amazon.com/AmazonS3/latest/user-guide/enable-cloudtrail-events.html

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	8.5 Collect Detailed Audit Logs Configure detailed audit logging for enterprise assets containing sensitive data. Include event source, date, username, timestamp, source addresses, destination addresses, and other useful elements that could assist in a forensic investigation.		●	●
v7	6.3 Enable Detailed Logging Enable system logging to include detailed information such as an event source, date, user, timestamp, source addresses, destination addresses, and other useful elements.		●	●

3.9 Ensure that object-level logging for read events is enabled for S3 buckets (Automated)

Profile Applicability:

- Level 2

Description:

S3 object-level API operations, such as GetObject, DeleteObject, and PutObject, are referred to as data events. By default, CloudTrail trails do not log data events, so it is recommended to enable object-level logging for S3 buckets.

Rationale:

Enabling object-level logging will help you meet data compliance requirements within your organization, perform comprehensive security analyses, monitor specific patterns of user behavior in your AWS account, or take immediate actions on any object-level API activity within your S3 buckets using Amazon CloudWatch Events.

Impact:

Enabling logging for these object-level events may significantly increase the number of events logged and may incur additional costs.

Audit:

From Console:

1. Login to the AWS Management Console and navigate to the CloudTrail dashboard at `https://console.aws.amazon.com/cloudtrail/`.
2. In the left panel, click `Trails`, and then click the name of the trail that you want to examine.
3. Review `General details`.
4. Confirm that `Multi-region trail` is set to `Yes`
5. Scroll down to `Data events`
6. Scroll down to `Data events` and confirm the configuration:

- If `advanced event selectors` is being used, it should read:

```
Data Events: S3
Log selector template
Log all events
```

- If `basic event selectors` is being used, it should read:

```
Data events: S3
Bucket Name: All current and future S3 buckets
Read: Enabled
```

6. Repeat steps 2-5 to verify that each trail has multi-region enabled and is configured to log data events. If a trail does not have multi-region enabled and data event logging configured, refer to the remediation steps.

From Command Line:

1. Run the `describe-trails` command to list all trail names:

```
aws cloudtrail describe-trails --region <region-name> --output table --query
trailList[*].Name
```

2. The command output will be table of the trail names.
3. Run the `get-event-selectors` command using the name of a trail returned at the previous step and custom query filters to determine if data event logging is configured:

```
aws cloudtrail get-event-selectors --region <region-name> --trail-name
<trail-name> --query EventSelectors[*].DataResources[]
```

4. The command output should be an array that includes the S3 bucket defined for data event logging.
5. If the `get-event-selectors` command returns an empty array, data events are not included in the trail's logging configuration; therefore, object-level API operations performed on S3 buckets within your AWS account are not being recorded.
6. Repeat steps 1-5 to verify the configuration of each trail.
7. Change the AWS region by updating the `--region` command parameter, and perform the audit process for other regions.

Remediation:

From Console:

1. Login to the AWS Management Console and navigate to S3 dashboard at `https://console.aws.amazon.com/s3/`.
2. In the left navigation panel, click `buckets` and then click the name of the S3 bucket that you want to examine.
3. Click the `Properties` tab to see the bucket configuration in detail.
4. In the `AWS Cloud Trail data events` section, select the trail name for recording activity. You can choose an existing trail or create a new one by clicking the `Configure in CloudTrail` button or navigating to the CloudTrail console.
5. Once the trail is selected, select the `Data Events` check box.

6. Select `S3` from the `Data event type` drop-down.
7. Select `Log all events` from the `Log selector template` drop-down.
8. Repeat steps 2-7 to enable object-level logging of read events for other S3 buckets.

From Command Line:

1. To enable `object-level` data events logging for S3 buckets within your AWS account, run the `put-event-selectors` command using the name of the trail that you want to reconfigure as identifier:

```
aws cloudtrail put-event-selectors --region <region-name> --trail-name
<trail-name> --event-selectors '[{ "ReadWriteType": "ReadOnly",
"IncludeManagementEvents":true, "DataResources": [{ "Type":
"AWS::S3::Object", "Values": ["arn:aws:s3:::<s3-bucket-name>/"] }] }]'
```

2. The command output will be `object-level` event trail configuration.
3. If you want to enable it for all buckets at once, change the Values parameter to `["arn:aws:s3"]` in the previous command.
4. Repeat step 1 for each s3 bucket to update `object-level` logging of read events.
5. Change the AWS region by updating the `--region` command parameter, and perform the process for the other regions.

References:

1. https://docs.aws.amazon.com/AmazonS3/latest/user-guide/enable-cloudtrail-events.html

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	8.5 Collect Detailed Audit Logs Configure detailed audit logging for enterprise assets containing sensitive data. Include event source, date, username, timestamp, source addresses, destination addresses, and other useful elements that could assist in a forensic investigation.		●	●
v7	6.3 Enable Detailed Logging Enable system logging to include detailed information such as an event source, date, user, timestamp, source addresses, destination addresses, and other useful elements.		●	●

4 Monitoring

This section contains recommendations for configuring AWS to assist with monitoring and responding to account activities.

Metric filter-related recommendations in this section are dependent on the `Ensure CloudTrail is enabled in all regions` and `Ensure CloudTrail trails are integrated with CloudWatch Logs` recommendations in the "Logging" section.

4.1 Ensure unauthorized API calls are monitored (Manual)

Profile Applicability:

- Level 2

Description:

Real-time monitoring of API calls can be achieved by directing CloudTrail Logs to CloudWatch Logs or an external Security Information and Event Management (SIEM) environment, and establishing corresponding metric filters and alarms.

It is recommended that a metric filter and alarm be established for unauthorized API calls.

Rationale:

CloudWatch is an AWS native service that allows you to observe and monitor resources and applications. CloudTrail logs can also be sent to an external Security Information and Event Management (SIEM) environment for monitoring and alerting.

Monitoring unauthorized API calls will help reduce the time it takes to detect malicious activity and can alert you to potential security incidents.

Impact:

This alert may be triggered by normal read-only console activities that attempt to opportunistically gather optional information but gracefully fail if they lack the necessary permissions.

If an excessive number of alerts are generated, then an organization may wish to consider adding read access to the limited IAM user permissions solely to reduce the number of alerts.

In some cases, doing this may allow users to actually view some areas of the system; any additional access granted should be reviewed for alignment with the original limited IAM user intent.

Audit:

If you are using CloudTrail trails and CloudWatch, perform the following to ensure that there is at least one active multi-region CloudTrail trail with the prescribed metric filters and alarms configured:

1. Identify the log group name that is configured for use with the active multi-region CloudTrail trail:

- List all CloudTrail trails: `aws cloudtrail describe-trails`
- Identify multi-region CloudTrail trails: `Trails with "IsMultiRegionTrail" set to true`

- Note the value associated with "Name":<trail-name>
- Note the <trail-log-group-name> within the value associated with "CloudWatchLogsLogGroupArn"
    - Example: arn:aws:logs:<region>:<account-id>:log-group:<trail-log-group-name>:*
- Ensure the identified multi-region CloudTrail trail is active:
    - aws cloudtrail get-trail-status --name <trail-name>
        - Ensure IsLogging is set to TRUE
- Ensure the identified multi-region CloudTrail trail captures all management events:
    - aws cloudtrail get-event-selectors --trail-name <trail-name>
        - Ensure there is at least one event selector for a trail with IncludeManagementEvents set to true and ReadWriteType set to All

2. Get a list of all associated metric filters for the <trail-log-group-name> captured in step 1:

```
aws logs describe-metric-filters --log-group-name <trail-log-group-name>
```

3. Ensure the output from the above command contains the following:

```
"filterPattern": "{ ($.errorCode ="*UnauthorizedOperation") ||
($.errorCode ="AccessDenied*") &&
($.sourceIPAddress!="delivery.logs.amazonaws.com") &&
($.eventName!="HeadBucket") }",
```

4. Note the <unauthorized-api-calls-metric> value associated with the filterPattern from step 3.
5. Get a list of CloudWatch alarms, and filter on the <unauthorized-api-calls-metric> captured in step 4:

```
aws cloudwatch describe-alarms --query "MetricAlarms[?MetricName ==
<unauthorized-api-calls-metric>]"
```

6. Note the AlarmActions value; this will provide the SNS topic ARN value.
7. Ensure there is at least one active subscriber to the SNS topic:

```
aws sns list-subscriptions-by-topic --topic-arn <sns-topic-arn>
```

- At least one subscription should have "SubscriptionArn" with a valid AWS ARN.
    - Example of valid "SubscriptionArn": arn:aws:sns:<region>:<account-id>:<sns-topic-name>:<subscription-id>

Remediation:

If you are using CloudTrail trails and CloudWatch, perform the following steps to set up the metric filter, alarm, SNS topic, and subscription:

1. Create a metric filter based on the provided filter pattern that checks for unauthorized API calls and uses the `<trail-log-group-name>` taken from audit step 1:

```
aws logs put-metric-filter --log-group-name <trail-log-group-name> --
filter-name <unauthorized-api-calls-metric> --metric-transformations
metricName=unauthorized_api_calls_metric,metricNamespace=CISBenchmark,m
etricValue=1 --filter-pattern "{ ($.errorCode
="*UnauthorizedOperation") || ($.errorCode ="AccessDenied*") &&
($.sourceIPAddress!="delivery.logs.amazonaws.com") &&
($.eventName!="HeadBucket") }"
```

   Note: You can choose your own `metricName` and `metricNamespace` strings. Using the same `metricNamespace` for all Foundations Benchmark metrics will group them together.

2. Create an SNS topic that the alarm will notify:

```
aws sns create-topic --name <sns-topic-name>
```

   Note: You can execute this command once and then reuse the same topic for all monitoring alarms.
   Note: Capture the `TopicArn` that is displayed when creating the SNS topic in step 2.

3. Create an SNS subscription for the topic created in step 2:

```
aws sns subscribe --topic-arn <sns-topic-arn> --protocol <sns-protocol>
--notification-endpoint <sns-subscription-endpoints>
```

   Note: You can execute this command once and then reuse the same subscription for all monitoring alarms.

4. Create an alarm that is associated with the CloudWatch Logs metric filter created in step 1 and the SNS topic created in step 2:

```
aws cloudwatch put-metric-alarm --alarm-name
"unauthorized_api_calls_alarm"  --metric-name
"unauthorized_api_calls_metric"  --statistic Sum --period 300 --
threshold 1 --comparison-operator GreaterThanOrEqualToThreshold --
evaluation-periods 1 --namespace "CISBenchmark" --alarm-actions <sns-
topic-arn>
```

References:

1. https://aws.amazon.com/sns/
2. CCE-79186-3
3. https://docs.aws.amazon.com/awscloudtrail/latest/userguide/receive-cloudtrail-log-files-from-multiple-regions.html
4. https://docs.aws.amazon.com/awscloudtrail/latest/userguide/cloudwatch-alarms-for-cloudtrail.html
5. https://docs.aws.amazon.com/sns/latest/dg/SubscribeTopic.html

Additional Information:

Configuring a log metric filter and alarm on a multi-region (global) CloudTrail trail:

- ensures that activities from all regions (both used and unused) are monitored
- ensures that activities on all supported global services are monitored
- ensures that all management events across all regions are monitored

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	8.5 Collect Detailed Audit Logs Configure detailed audit logging for enterprise assets containing sensitive data. Include event source, date, username, timestamp, source addresses, destination addresses, and other useful elements that could assist in a forensic investigation.		●	●
v8	8.11 Conduct Audit Log Reviews Conduct reviews of audit logs to detect anomalies or abnormal events that could indicate a potential threat. Conduct reviews on a weekly, or more frequent, basis.		●	●
v7	6.3 Enable Detailed Logging Enable system logging to include detailed information such as an event source, date, user, timestamp, source addresses, destination addresses, and other useful elements.		●	●
v7	6.5 Central Log Management Ensure that appropriate logs are being aggregated to a central log management system for analysis and review.		●	●
v7	6.7 Regularly Review Logs On a regular basis, review logs to identify anomalies or abnormal events.		●	●

4.2 Ensure management console sign-in without MFA is monitored (Manual)

Profile Applicability:

- Level 1

Description:

Real-time monitoring of API calls can be achieved by directing CloudTrail Logs to CloudWatch Logs or an external Security Information and Event Management (SIEM) environment, and establishing corresponding metric filters and alarms.

It is recommended that a metric filter and alarm be established for console logins that are not protected by multi-factor authentication (MFA).

Rationale:

CloudWatch is an AWS native service that allows you to observe and monitor resources and applications. CloudTrail logs can also be sent to an external Security Information and Event Management (SIEM) environment for monitoring and alerting.

Monitoring for single-factor console logins will increase visibility into accounts that are not protected by MFA. These type of accounts are more susceptible to compromise and unauthorized access.

Audit:

If you are using CloudTrail trails and CloudWatch, perform the following to ensure that there is at least one active multi-region CloudTrail trail with the prescribed metric filters and alarms configured:

1. Identify the log group name that is configured for use with the active multi-region CloudTrail trail:

- List all CloudTrail trails:

```
aws cloudtrail describe-trails
```

- Identify multi-region CloudTrail trails: `Trails with "IsMultiRegionTrail" set to true`
- Note the value associated with "Name":`<trail-name>`
- Note the `<trail-log-group-name>` within the value associated with "CloudWatchLogsLogGroupArn"
    - Example: `arn:aws:logs:<region>:<account-id>:log-group:<trail-log-group-name>:*`
- Ensure the identified multi-region CloudTrail trail is active:

```
aws cloudtrail get-trail-status --name <trail-name>
```

    - ensure `IsLogging` is set to `TRUE`
- Ensure the identified multi-region CloudTrail trail captures all management events:

```
aws cloudtrail get-event-selectors --trail-name <trail-name>
```

    - Ensure there is at least one `event selector` for a trail with `IncludeManagementEvents` set to `true` and `ReadWriteType` set to `All`

2. Get a list of all associated metric filters for the `<trail-log-group-name>` captured in step 1:

```
aws logs describe-metric-filters --log-group-name <trail-log-group-name>
```

3. Ensure the output from the above command contains the following:

```
"filterPattern": "{ ($.eventName = "ConsoleLogin") && ($.additionalEventData.MFAUsed != "Yes") }"
```

   Or, to reduce false positives in case Single Sign-On (SSO) is used in the organization:

```
"filterPattern": "{ ($.eventName = "ConsoleLogin") && ($.additionalEventData.MFAUsed != "Yes") && ($.userIdentity.type = "IAMUser") && ($.responseElements.ConsoleLogin = "Success") }"
```

4. Note the `<no-mfa-console-signin-metric>` value associated with the `filterPattern` from step 3.
5. Get a list of CloudWatch alarms, and filter on the `<no-mfa-console-signin-metric>` captured in step 4.

```
aws cloudwatch describe-alarms --query 'MetricAlarms[?MetricName== <no-mfa-console-signin-metric>]'
```

6. Note the `AlarmActions` value; this will provide the SNS topic ARN value.
7. Ensure there is at least one active subscriber to the SNS topic:

```
aws sns list-subscriptions-by-topic --topic-arn <sns-topic-arn>
```

- At least one subscription should have "SubscriptionArn" with a valid AWS ARN.
    - Example of valid "SubscriptionArn": `arn:aws:sns:<region>:<account-id>:<sns-topic-name>:<subscription-id>`

Remediation:

If you are using CloudTrail trails and CloudWatch, perform the following steps to set up the metric filter, alarm, SNS topic, and subscription:

1. Create a metric filter based on the provided filter pattern that checks for AWS Management Console sign-ins without MFA and uses the `<trail-log-group-name>` taken from audit step 1.

```
aws logs put-metric-filter --log-group-name <trail-log-group-name> --
filter-name `<no-mfa-console-signin-metric>` --metric-transformations
metricName= `<no-mfa-console-signin-
metric>`,metricNamespace='CISBenchmark',metricValue=1 --filter-pattern
'{ ($.eventName = "ConsoleLogin") && ($.additionalEventData.MFAUsed !=
"Yes") }'
```

   Or, to reduce false positives in case Single Sign-On (SSO) is used in the organization:

```
aws logs put-metric-filter --log-group-name <trail-log-group-name> --
filter-name `<no-mfa-console-signin-metric>` --metric-transformations
metricName= `<no-mfa-console-signin-
metric>`,metricNamespace='CISBenchmark',metricValue=1 --filter-pattern
'{ ($.eventName = "ConsoleLogin") && ($.additionalEventData.MFAUsed !=
"Yes") && ($.userIdentity.type = "IAMUser") &&
($.responseElements.ConsoleLogin = "Success") }'
```

   Note: You can choose your own `metricName` and `metricNamespace` strings. Using the same `metricNamespace` for all Foundations Benchmark metrics will group them together.

2. Create an SNS topic that the alarm will notify:

```
aws sns create-topic --name <sns-topic-name>
```

   Note: You can execute this command once and then reuse the same topic for all monitoring alarms.
   Note: Capture the `TopicArn` that is displayed when creating the SNS topic in step 2.

3. Create an SNS subscription for the topic created in step 2:

```
aws sns subscribe --topic-arn <sns-topic-arn> --protocol <sns-protocol>
--notification-endpoint <sns-subscription-endpoints>
```

   Note: You can execute this command once and then reuse the same subscription for all monitoring alarms.

4. Create an alarm that is associated with the CloudWatch Logs metric filter created in step 1 and the SNS topic created in step 2:

```
aws cloudwatch put-metric-alarm --alarm-name `<no-mfa-console-signin-
alarm>` --metric-name `<no-mfa-console-signin-metric>` --statistic Sum
--period 300 --threshold 1 --comparison-operator
GreaterThanOrEqualToThreshold --evaluation-periods 1 --namespace
'CISBenchmark' --alarm-actions <sns-topic-arn>
```

References:

1. https://docs.aws.amazon.com/AmazonCloudWatch/latest/DeveloperGuide/viewing_metrics_with_cloudwatch.html
2. CCE-79187-1
3. https://docs.aws.amazon.com/awscloudtrail/latest/userguide/receive-cloudtrail-log-files-from-multiple-regions.html
4. https://docs.aws.amazon.com/awscloudtrail/latest/userguide/cloudwatch-alarms-for-cloudtrail.html
5. https://docs.aws.amazon.com/sns/latest/dg/SubscribeTopic.html

Additional Information:

Configuring a log metric filter and alarm on a multi-region (global) CloudTrail trail:

- ensures that activities from all regions (both used and unused) are monitored
- ensures that activities on all supported global services are monitored
- ensures that all management events across all regions are monitored

Filter pattern set to `{ ($.eventName = "ConsoleLogin") && ($.additionalEventData.MFAUsed != "Yes") && ($.userIdentity.type = "IAMUser") && ($.responseElements.ConsoleLogin = "Success")}`:

- reduces false alarms raised when a user logs in via SSO

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	8.5 Collect Detailed Audit Logs Configure detailed audit logging for enterprise assets containing sensitive data. Include event source, date, username, timestamp, source addresses, destination addresses, and other useful elements that could assist in a forensic investigation.		●	●
v8	8.11 Conduct Audit Log Reviews Conduct reviews of audit logs to detect anomalies or abnormal events that could indicate a potential threat. Conduct reviews on a weekly, or more frequent, basis.		●	●

Controls Version	Control	IG 1	IG 2	IG 3
v7	6.3 Enable Detailed Logging Enable system logging to include detailed information such as an event source, date, user, timestamp, source addresses, destination addresses, and other useful elements.		●	●

4.3 Ensure usage of the 'root' account is monitored (Manual)

Profile Applicability:

- Level 1

Description:

Real-time monitoring of API calls can be achieved by directing CloudTrail Logs to CloudWatch Logs or an external Security Information and Event Management (SIEM) environment, and establishing corresponding metric filters and alarms.

It is recommended that a metric filter and alarm be established for 'root' login attempts to detect unauthorized use or attempts to use the root account.

Rationale:

CloudWatch is an AWS native service that allows you to observe and monitor resources and applications. CloudTrail logs can also be sent to an external Security Information and Event Management (SIEM) environment for monitoring and alerting.

Monitoring 'root' account logins will provide visibility into the use of a fully privileged account and the opportunity to reduce its usage.

Audit:

If you are using CloudTrail trails and CloudWatch, perform the following to ensure that there is at least one active multi-region CloudTrail trail with the prescribed metric filters and alarms configured:

1. Identify the log group name that is configured for use with the active multi-region CloudTrail trail:

- List all CloudTrail trails:

```
aws cloudtrail describe-trails
```

- Identify multi-region CloudTrail trails: `Trails with "IsMultiRegionTrail" set to true`
- Note the value associated with "Name":`<trail-name>`
- Note the `<trail-log-group-name>` within the value associated with "CloudWatchLogsLogGroupArn"
    - Example: `arn:aws:logs:<region>:<account-id>:log-group:<trail-log-group-name>:*`
- Ensure the identified multi-region CloudTrail trail is active:

```
aws cloudtrail get-trail-status --name <trail-name>
```

    - Ensure `IsLogging` is set to `TRUE`

- Ensure the identified multi-region CloudTrail trail captures all management events:

```
aws cloudtrail get-event-selectors --trail-name <trail-name>
```

    - Ensure there is at least one `event selector` for a trail with `IncludeManagementEvents` set to `true` and `ReadWriteType` set to `All`

2. Get a list of all associated metric filters for the `<trail-log-group-name>` captured in step 1:

```
aws logs describe-metric-filters --log-group-name <trail-log-group-
name>
```

3. Ensure the output from the above command contains the following:

```
"filterPattern": "{ $.userIdentity.type = "Root" &&
$.userIdentity.invokedBy NOT EXISTS && $.eventType != "AwsServiceEvent"
}"
```

4. Note the `<root-usage-metric>` value associated with the `filterPattern` from step 3.
5. Get a list of CloudWatch alarms, and filter on the `<root-usage-metric>` captured in step 4:

```
aws cloudwatch describe-alarms --query
'MetricAlarms[?MetricName==<root-usage-metric>]'
```

6. Note the `AlarmActions` value; this will provide the SNS topic ARN value.
7. Ensure there is at least one active subscriber to the SNS topic:

```
aws sns list-subscriptions-by-topic --topic-arn <sns-topic-arn>
```

- At least one subscription should have "SubscriptionArn" with a valid AWS ARN.
    - Example of valid "SubscriptionArn": `arn:aws:sns:<region>:<account-id>:<sns-topic-name>:<subscription-id>`

Remediation:

If you are using CloudTrail trails and CloudWatch, perform the following steps to set up the metric filter, alarm, SNS topic, and subscription:

1. Create a metric filter based on the provided filter pattern that checks for 'root' account usage and uses the `<trail-log-group-name>` taken from audit step 1:

```
aws logs put-metric-filter --log-group-name `<trail-log-group-name>` --
filter-name `<root-usage-metric>` --metric-transformations metricName=
`<root-usage-metric>` ,metricNamespace='CISBenchmark',metricValue=1 --
filter-pattern '{ $.userIdentity.type = "Root" &&
$.userIdentity.invokedBy NOT EXISTS && $.eventType != "AwsServiceEvent"
}'
```

Note: You can choose your own `metricName` and `metricNamespace` strings. Using the same `metricNamespace` for all Foundations Benchmark metrics will group them together.

2. Create an SNS topic that the alarm will notify:

```
aws sns create-topic --name <sns-topic-name>
```

Note: You can execute this command once and then reuse the same topic for all monitoring alarms.
Note: Capture the `TopicArn` that is displayed when creating the SNS topic in step 2.

3. Create an SNS subscription for the topic created in step 2:

```
aws sns subscribe --topic-arn <sns-topic-arn> --protocol <sns-protocol>
--notification-endpoint <sns-subscription-endpoints>
```

Note: You can execute this command once and then reuse the same subscription for all monitoring alarms.

4. Create an alarm that is associated with the CloudWatch Logs metric filter created in step 1 and the SNS topic created in step 2:

```
aws cloudwatch put-metric-alarm --alarm-name `<root-usage-alarm>` --
metric-name `<root-usage-metric>` --statistic Sum --period 300 --
threshold 1 --comparison-operator GreaterThanOrEqualToThreshold --
evaluation-periods 1 --namespace 'CISBenchmark' --alarm-actions
<sns_topic_arn>
```

References:

1. CCE-79188-9
2. https://docs.aws.amazon.com/awscloudtrail/latest/userguide/receive-cloudtrail-log-files-from-multiple-regions.html
3. https://docs.aws.amazon.com/awscloudtrail/latest/userguide/cloudwatch-alarms-for-cloudtrail.html
4. https://docs.aws.amazon.com/sns/latest/dg/SubscribeTopic.html

Additional Information:

Configuring a log metric filter and alarm on a multi-region (global) CloudTrail trail:

- ensures that activities from all regions (both used and unused) are monitored
- ensures that activities on all supported global services are monitored
- ensures that all management events across all regions are monitored

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	8.5 Collect Detailed Audit Logs Configure detailed audit logging for enterprise assets containing sensitive data. Include event source, date, username, timestamp, source addresses, destination addresses, and other useful elements that could assist in a forensic investigation.		●	●
v8	8.11 Conduct Audit Log Reviews Conduct reviews of audit logs to detect anomalies or abnormal events that could indicate a potential threat. Conduct reviews on a weekly, or more frequent, basis.		●	●
v7	4.9 Log and Alert on Unsuccessful Administrative Account Login Configure systems to issue a log entry and alert on unsuccessful logins to an administrative account.		●	●
v7	6.3 Enable Detailed Logging Enable system logging to include detailed information such as an event source, date, user, timestamp, source addresses, destination addresses, and other useful elements.		●	●

4.4 Ensure IAM policy changes are monitored (Manual)

Profile Applicability:

- Level 1

Description:

Real-time monitoring of API calls can be achieved by directing CloudTrail Logs to CloudWatch Logs or an external Security Information and Event Management (SIEM) environment, and establishing corresponding metric filters and alarms.

It is recommended that a metric filter and alarm be established for changes made to Identity and Access Management (IAM) policies.

Rationale:

CloudWatch is an AWS native service that allows you to observe and monitor resources and applications. CloudTrail logs can also be sent to an external Security Information and Event Management (SIEM) environment for monitoring and alerting.

Monitoring changes to IAM policies will help ensure authentication and authorization controls remain intact.

Impact:

Monitoring these changes may result in a number of "false positives," especially in larger environments. This alert may require more tuning than others to eliminate some of those erroneous notifications.

Audit:

If you are using CloudTrail trails and CloudWatch, perform the following to ensure that there is at least one active multi-region CloudTrail trail with the prescribed metric filters and alarms configured:

1. Identify the log group name that is configured for use with the active multi-region CloudTrail trail:

- List all CloudTrails:

```
aws cloudtrail describe-trails
```

- Identify multi-region CloudTrail trails: `Trails with "IsMultiRegionTrail" set to true`
- Note the value associated with "Name":`<trail-name>`
- Note the `<trail-log-group-name>` within the value associated with "CloudWatchLogsLogGroupArn"

    - Example: `arn:aws:logs:<region>:<account-id>:log-group:<trail-log-group-name>:*`
- Ensure the identified multi-region CloudTrail trail is active:

```
aws cloudtrail get-trail-status --name <trail-name>
```

    - Ensure `IsLogging` is set to `TRUE`
- Ensure the identified multi-region CloudTrail trail captures all management events:

```
aws cloudtrail get-event-selectors --trail-name <trail-name>
```

    - Ensure there is at least one `event selector` for a trail with `IncludeManagementEvents` set to `true` and `ReadWriteType` set to `All`

2. Get a list of all associated metric filters for the `<trail-log-group-name>` captured in step 1:

```
aws logs describe-metric-filters --log-group-name <trail-log-group-name>
```

3. Ensure the output from the above command contains the following:

```
"filterPattern":
"{($.eventName=DeleteGroupPolicy)||($.eventName=DeleteRolePolicy)||($.eventName=DeleteUserPolicy)||($.eventName=PutGroupPolicy)||($.eventName=PutRolePolicy)||($.eventName=PutUserPolicy)||($.eventName=CreatePolicy)||($.eventName=DeletePolicy)||($.eventName=CreatePolicyVersion)||($.eventName=DeletePolicyVersion)||($.eventName=AttachRolePolicy)||($.eventName=DetachRolePolicy)||($.eventName=AttachUserPolicy)||($.eventName=DetachUserPolicy)||($.eventName=AttachGroupPolicy)||($.eventName=DetachGroupPolicy)}"
```

4. Note the `<iam-changes-metric>` value associated with the `filterPattern` from step 3.
5. Get a list of CloudWatch alarms, and filter on the `<iam-change-metric>` captured in step 4:

```
aws cloudwatch describe-alarms --query 'MetricAlarms[?MetricName==<iam-changes-metric>]'
```

6. Note the `AlarmActions` value; this will provide the SNS topic ARN value.
7. Ensure there is at least one active subscriber to the SNS topic:

```
aws sns list-subscriptions-by-topic --topic-arn <sns-topic-arn>
```

- At least one subscription should have "SubscriptionArn" with a valid AWS ARN.

- Example of valid "SubscriptionArn": `arn:aws:sns:<region>:<account-id>:<sns-topic-name>:<subscription-id>`

Remediation:

If you are using CloudTrail trails and CloudWatch, perform the following steps to set up the metric filter, alarm, SNS topic, and subscription:

1. Create a metric filter based on the provided filter pattern that checks for IAM policy changes and the `<trail-log-group-name>` taken from audit step 1:

```
aws logs put-metric-filter --log-group-name `<trail-log-group-name>` --
filter-name `<iam-changes-metric>` --metric-transformations metricName=
`<iam-changes-metric>`,metricNamespace='CISBenchmark',metricValue=1 --
filter-pattern
'{($.eventName=DeleteGroupPolicy)||($.eventName=DeleteRolePolicy)||($.e
ventName=DeleteUserPolicy)||($.eventName=PutGroupPolicy)||($.eventName=
PutRolePolicy)||($.eventName=PutUserPolicy)||($.eventName=CreatePolicy)
||($.eventName=DeletePolicy)||($.eventName=CreatePolicyVersion)||($.eve
ntName=DeletePolicyVersion)||($.eventName=AttachRolePolicy)||($.eventNa
me=DetachRolePolicy)||($.eventName=AttachUserPolicy)||($.eventName=Deta
chUserPolicy)||($.eventName=AttachGroupPolicy)||($.eventName=DetachGrou
pPolicy)}'
```

   Note: You can choose your own `metricName` and `metricNamespace` strings. Using the same `metricNamespace` for all Foundations Benchmark metrics will group them together.

2. Create an SNS topic that the alarm will notify:

```
aws sns create-topic --name <sns-topic-name>
```

   Note: You can execute this command once and then reuse the same topic for all monitoring alarms.
   Note: Capture the `TopicArn` that is displayed when creating the SNS topic in step 2.

3. Create an SNS subscription for the topic created in step 2:

```
aws sns subscribe --topic-arn <sns-topic-arn> --protocol <sns-protocol>
--notification-endpoint <sns-subscription-endpoints>
```

   Note: You can execute this command once and then reuse the same subscription for all monitoring alarms.

4. Create an alarm that is associated with the CloudWatch Logs metric filter created in step 1 and the SNS topic created in step 2:

```
aws cloudwatch put-metric-alarm --alarm-name `<iam-changes-alarm>` --
metric-name `<iam-changes-metric>` --statistic Sum --period 300 --
threshold 1 --comparison-operator GreaterThanOrEqualToThreshold --
evaluation-periods 1 --namespace 'CISBenchmark' --alarm-actions <sns-
topic-arn>
```

References:

1. CCE-79189-7
2. https://docs.aws.amazon.com/awscloudtrail/latest/userguide/receive-cloudtrail-log-files-from-multiple-regions.html
3. https://docs.aws.amazon.com/awscloudtrail/latest/userguide/cloudwatch-alarms-for-cloudtrail.html
4. https://docs.aws.amazon.com/sns/latest/dg/SubscribeTopic.html

Additional Information:

Configuring a log metric filter and alarm on a multi-region (global) CloudTrail trail:

- ensures that activities from all regions (both used and unused) are monitored
- ensures that activities on all supported global services are monitored
- ensures that all management events across all regions are monitored

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	8.5 Collect Detailed Audit Logs Configure detailed audit logging for enterprise assets containing sensitive data. Include event source, date, username, timestamp, source addresses, destination addresses, and other useful elements that could assist in a forensic investigation.		●	●
v8	8.11 Conduct Audit Log Reviews Conduct reviews of audit logs to detect anomalies or abnormal events that could indicate a potential threat. Conduct reviews on a weekly, or more frequent, basis.		●	●
v7	6.3 Enable Detailed Logging Enable system logging to include detailed information such as an event source, date, user, timestamp, source addresses, destination addresses, and other useful elements.		●	●

4.5 Ensure CloudTrail configuration changes are monitored (Manual)

Profile Applicability:

- Level 1

Description:

Real-time monitoring of API calls can be achieved by directing CloudTrail Logs to CloudWatch Logs or an external Security Information and Event Management (SIEM) environment, and establishing corresponding metric filters and alarms.

It is recommended that a metric filter and alarm be used to detect changes to CloudTrail's configurations.

Rationale:

CloudWatch is an AWS native service that allows you to observe and monitor resources and applications. CloudTrail logs can also be sent to an external Security Information and Event Management (SIEM) environment for monitoring and alerting.

Monitoring changes to CloudTrail's configuration will help ensure sustained visibility into the activities performed in the AWS account.

Impact:

Ensuring that changes to CloudTrail configurations are monitored enhances security by maintaining the integrity of logging mechanisms. Automated monitoring can provide real-time alerts; however, it may require additional setup and resources to configure and manage these alerts effectively. These steps can be performed manually within a company's existing SIEM platform in cases where CloudTrail logs are monitored outside of the AWS monitoring tools in CloudWatch.

Audit:

If you are using CloudTrail trails and CloudWatch, perform the following to ensure that there is at least one active multi-region CloudTrail trail with the prescribed metric filters and alarms configured:

1. Identify the log group name that is configured for use with the active multi-region CloudTrail trail:

- List all CloudTrail trails: `aws cloudtrail describe-trails`
- Identify multi-region CloudTrail trails: `Trails with "IsMultiRegionTrail" set to true`
- Note the value associated with "Name":`<trail-name>`
- Note the `<trail-log-group-name>` within the value associated with "CloudWatchLogsLogGroupArn"

    - Example: arn:aws:logs:<region>:<account-id>:log-group:<trail-log-group-name>:*
- Ensure the identified multi-region CloudTrail trail is active:
    - aws cloudtrail get-trail-status --name <trail-name>
        - Ensure IsLogging is set to TRUE
- Ensure the identified multi-region CloudTrail trail captures all management events:
    - aws cloudtrail get-event-selectors --trail-name <trail-name>
        - Ensure there is at least one event selector for a trail with IncludeManagementEvents set to true and ReadWriteType set to All

2. Get a list of all associated metric filters for the <trail-log-group-name> captured in step 1:

```
aws logs describe-metric-filters --log-group-name <trail-log-group-name>
```

3. Ensure the output from the above command contains the following:

```
"filterPattern": "{ ($.eventName = CreateTrail) || ($.eventName = UpdateTrail) || ($.eventName = DeleteTrail) || ($.eventName = StartLogging) || ($.eventName = StopLogging) }"
```

4. Note the <cloudtrail-cfg-changes-metric> value associated with the filterPattern from step 3.
5. Get a list of CloudWatch alarms, and filter on the <cloudtrail-cfg-changes-metric> captured in step 4:

```
aws cloudwatch describe-alarms --query 'MetricAlarms[?MetricName==<cloudtrail-cfg-changes-metric>]'
```

6. Note the AlarmActions value; this will provide the SNS topic ARN value.
7. Ensure there is at least one active subscriber to the SNS topic:

```
aws sns list-subscriptions-by-topic --topic-arn <sns-topic-arn>
```

- At least one subscription should have "SubscriptionArn" with a valid AWS ARN.
    - Example of valid "SubscriptionArn": arn:aws:sns:<region>:<account-id>:<sns-topic-name>:<subscription-id>

Remediation:

If you are using CloudTrail trails and CloudWatch, perform the following steps to set up the metric filter, alarm, SNS topic, and subscription:

1. Create a metric filter based on the provided filter pattern that checks for CloudTrail configuration changes and the `<trail-log-group-name>` taken from audit step 1:

```
aws logs put-metric-filter --log-group-name <trail-log-group-name> --
filter-name <cloudtrail-cfg-changes-metric> --metric-transformations
metricName=<cloudtrail-cfg-changes-
metric>,metricNamespace='CISBenchmark',metricValue=1 --filter-pattern
'{ ($.eventName = CreateTrail) || ($.eventName = UpdateTrail) ||
($.eventName = DeleteTrail) || ($.eventName = StartLogging) ||
($.eventName = StopLogging) }'
```

   Note: You can choose your own `metricName` and `metricNamespace` strings. Using the same `metricNamespace` for all Foundations Benchmark metrics will group them together.

2. Create an SNS topic that the alarm will notify:

```
aws sns create-topic --name <sns-topic-name>
```

   Note: You can execute this command once and then reuse the same topic for all monitoring alarms.
   Note: Capture the `TopicArn` that is displayed when creating the SNS topic in step 2.

3. Create an SNS subscription for the topic created in step 2:

```
aws sns subscribe --topic-arn <sns-topic-arn> --protocol <sns-protocol>
--notification-endpoint <sns-subscription-endpoints>
```

   Note: You can execute this command once and then reuse the same subscription for all monitoring alarms.

4. Create an alarm that is associated with the CloudWatch Logs metric filter created in step 1 and the SNS topic created in step 2:

```
aws cloudwatch put-metric-alarm --alarm-name <cloudtrail-cfg-changes-
alarm> --metric-name <cloudtrail-cfg-changes-metric> --statistic Sum --
period 300 --threshold 1 --comparison-operator
GreaterThanOrEqualToThreshold --evaluation-periods 1 --namespace
'CISBenchmark' --alarm-actions <sns-topic-arn>
```

References:

1. CCE-79190-5
2. https://docs.aws.amazon.com/awscloudtrail/latest/userguide/receive-cloudtrail-log-files-from-multiple-regions.html

3. https://docs.aws.amazon.com/awscloudtrail/latest/userguide/cloudwatch-alarms-for-cloudtrail.html
4. https://docs.aws.amazon.com/sns/latest/dg/SubscribeTopic.html

Additional Information:

Configuring a log metric filter and alarm on a multi-region (global) CloudTrail trail:

- ensures that activities from all regions (both used and unused) are monitored
- ensures that activities on all supported global services are monitored
- ensures that all management events across all regions are monitored

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	8.5 Collect Detailed Audit Logs Configure detailed audit logging for enterprise assets containing sensitive data. Include event source, date, username, timestamp, source addresses, destination addresses, and other useful elements that could assist in a forensic investigation.		●	●
v8	8.11 Conduct Audit Log Reviews Conduct reviews of audit logs to detect anomalies or abnormal events that could indicate a potential threat. Conduct reviews on a weekly, or more frequent, basis.		●	●
v7	6.3 Enable Detailed Logging Enable system logging to include detailed information such as an event source, date, user, timestamp, source addresses, destination addresses, and other useful elements.		●	●

4.6 Ensure AWS Management Console authentication failures are monitored (Manual)

Profile Applicability:

- Level 2

Description:

Real-time monitoring of API calls can be achieved by directing CloudTrail Logs to CloudWatch Logs or an external Security Information and Event Management (SIEM) environment, and establishing corresponding metric filters and alarms.

It is recommended that a metric filter and alarm be established for failed console authentication attempts.

Rationale:

CloudWatch is an AWS native service that allows you to observe and monitor resources and applications. CloudTrail logs can also be sent to an external Security Information and Event Management (SIEM) environment for monitoring and alerting.

Monitoring failed console logins may decrease the lead time to detect an attempt to brute-force a credential, which may provide an indicator, such as the source IP address, that can be used in other event correlations.

Impact:

Monitoring for these failures may generate a large number of alerts, especially in larger environments.

Audit:

If you are using CloudTrail trails and CloudWatch, perform the following to ensure that there is at least one active multi-region CloudTrail trail with the prescribed metric filters and alarms configured:

1. Identify the log group name that is configured for use with the active multi-region CloudTrail trail:

- List all CloudTrail trails: `aws cloudtrail describe-trails`
- Identify multi-region CloudTrail trails: `Trails with "IsMultiRegionTrail" set to true`
- Note the value associated with "Name":`<trail-name>`
- Note the `<trail-log-group-name>` within the value associated with "CloudWatchLogsLogGroupArn"
    - Example: `arn:aws:logs:<region>:<account-id>:log-group:<trail-log-group-name>:*`
- Ensure the identified multi-region CloudTrail trail is active:

    - `aws cloudtrail get-trail-status --name <trail-name>`
        - Ensure `IsLogging` is set to `TRUE`
- Ensure the identified multi-region CloudTrail trail captures all management events:
    - `aws cloudtrail get-event-selectors --trail-name <trail-name>`
        - Ensure there is at least one `event selector` for a trail with `IncludeManagementEvents` set to `true` and `ReadWriteType` set to `All`

2. Get a list of all associated metric filters for the `<trail-log-group-name>` captured in step 1:

```
aws logs describe-metric-filters --log-group-name <trail-log-group-
name>
```

3. Ensure the output from the above command contains the following:

```
"filterPattern": "{ ($.eventName = ConsoleLogin) && ($.errorMessage =
"Failed authentication") }"
```

4. Note the `<console-signin-failure-metric>` value associated with the `filterPattern` from step 3.
5. Get a list of CloudWatch alarms, and filter on the `<console-signin-failure-metric>` captured in step 4:

```
aws cloudwatch describe-alarms --query
'MetricAlarms[?MetricName==<console-signin-failure-metric>]'
```

6. Note the `AlarmActions` value; this will provide the SNS topic ARN value.
7. Ensure there is at least one active subscriber to the SNS topic:

```
aws sns list-subscriptions-by-topic --topic-arn <sns-topic-arn>
```

- At least one subscription should have "SubscriptionArn" with a valid AWS ARN.
    - Example of valid "SubscriptionArn": `arn:aws:sns:<region>:<account-id>:<sns-topic-name>:<subscription-id>`

Remediation:

If you are using CloudTrail trails and CloudWatch, perform the following steps to set up the metric filter, alarm, SNS topic, and subscription:

1. Create a metric filter based on the provided filter pattern that checks for AWS management Console login failures and uses the `<trail-log-group-name>` taken from audit step 1:

```
aws logs put-metric-filter --log-group-name <trail-log-group-name> --
filter-name <console-signin-failure-metric> --metric-transformations
metricName=<console-signin-failure-
metric>,metricNamespace='CISBenchmark',metricValue=1 --filter-pattern
'{ ($.eventName = ConsoleLogin) && ($.errorMessage = "Failed
authentication") }'
```

Note: You can choose your own `metricName` and `metricNamespace` strings. Using the same `metricNamespace` for all Foundations Benchmark metrics will group them together.

2. Create an SNS topic that the alarm will notify:

```
aws sns create-topic --name <sns-topic-name>
```

Note: You can execute this command once and then reuse the same topic for all monitoring alarms.
Note: Capture the `TopicArn` that is displayed when creating the SNS topic in step 2.

3. Create an SNS subscription for the topic created in step 2:

```
aws sns subscribe --topic-arn <sns-topic-arn> --protocol <sns-protocol>
--notification-endpoint <sns-subscription-endpoints>
```

Note: You can execute this command once and then reuse the same subscription for all monitoring alarms.

4. Create an alarm that is associated with the CloudWatch Logs metric filter created in step 1 and the SNS topic created in step 2:

```
aws cloudwatch put-metric-alarm --alarm-name <console-signin-failure-
alarm> --metric-name <console-signin-failure-metric> --statistic Sum --
period 300 --threshold 1 --comparison-operator
GreaterThanOrEqualToThreshold --evaluation-periods 1 --namespace
'CISBenchmark' --alarm-actions <sns-topic-arn>
```

References:

1. CCE-79191-3
2. https://docs.aws.amazon.com/awscloudtrail/latest/userguide/receive-cloudtrail-log-files-from-multiple-regions.html
3. https://docs.aws.amazon.com/awscloudtrail/latest/userguide/cloudwatch-alarms-for-cloudtrail.html
4. https://docs.aws.amazon.com/sns/latest/dg/SubscribeTopic.html

Additional Information:

Configuring a log metric filter and alarm on a multi-region (global) CloudTrail trail:

- ensures that activities from all regions (both used and unused) are monitored
- ensures that activities on all supported global services are monitored
- ensures that all management events across all regions are monitored

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	8.5 Collect Detailed Audit Logs Configure detailed audit logging for enterprise assets containing sensitive data. Include event source, date, username, timestamp, source addresses, destination addresses, and other useful elements that could assist in a forensic investigation.		●	●
v8	8.11 Conduct Audit Log Reviews Conduct reviews of audit logs to detect anomalies or abnormal events that could indicate a potential threat. Conduct reviews on a weekly, or more frequent, basis.		●	●
v7	4.9 Log and Alert on Unsuccessful Administrative Account Login Configure systems to issue a log entry and alert on unsuccessful logins to an administrative account.		●	●
v7	6.3 Enable Detailed Logging Enable system logging to include detailed information such as an event source, date, user, timestamp, source addresses, destination addresses, and other useful elements.		●	●

4.7 Ensure disabling or scheduled deletion of customer created CMKs is monitored (Manual)

Profile Applicability:

- Level 2

Description:

Real-time monitoring of API calls can be achieved by directing CloudTrail Logs to CloudWatch Logs or an external Security Information and Event Management (SIEM) environment, and establishing corresponding metric filters and alarms.

It is recommended that a metric filter and alarm be established for customer-created CMKs that have changed state to disabled or are scheduled for deletion.

Rationale:

CloudWatch is an AWS native service that allows you to observe and monitor resources and applications. CloudTrail logs can also be sent to an external Security Information and Event Management (SIEM) environment for monitoring and alerting.

Data encrypted with disabled or deleted keys will no longer be accessible. Changes in the state of a CMK should be monitored to ensure that the change is intentional.

Impact:

Creation, storage, and management of CMK may require additional labor compared to the use of AWS-managed keys.

Audit:

If you are using CloudTrail trails and CloudWatch, perform the following to ensure that there is at least one active multi-region CloudTrail trail with the prescribed metric filters and alarms configured:

1. Identify the log group name that is configured for use with the active multi-region CloudTrail trail:

- List all CloudTrail trails: `aws cloudtrail describe-trails`
- Identify multi-region CloudTrail trails: `Trails with "IsMultiRegionTrail" set to true`
- Note the value associated with "Name":`<trail-name>`
- Note the `<trail-log-group-name>` within the value associated with "CloudWatchLogsLogGroupArn"
    - Example: `arn:aws:logs:<region>:<account-id>:log-group:<trail-log-group-name>:*`
- Ensure the identified multi-region CloudTrail trail is active:
    - `aws cloudtrail get-trail-status --name <trail-name>`

        - Ensure `IsLogging` is set to `TRUE`
- Ensure the identified multi-region CloudTrail trail captures all management events:
    - `aws cloudtrail get-event-selectors --trail-name <trail-name>`
        - Ensure there is at least one `event selector` for a trail with `IncludeManagementEvents` set to `true` and `ReadWriteType` set to `All`

2. Get a list of all associated metric filters for the `<trail-log-group-name>` captured in step 1:

```
aws logs describe-metric-filters --log-group-name <trail-log-group-
name>
```

3. Ensure the output from the above command contains the following:

```
"filterPattern": "{($.eventSource = kms.amazonaws.com) &&
($.eventName=DisableKey)||($.eventName=ScheduleKeyDeletion)) }"
```

4. Note the `<disable-or-delete-cmk-changes-metric>` value associated with the `filterPattern` from step 3.
5. Get a list of CloudWatch alarms, and filter on the `<disable-or-delete-cmk-changes-metric>` captured in step 4:

```
aws cloudwatch describe-alarms --query
'MetricAlarms[?MetricName==<disable-or-delete-cmk-changes-metric>]'
```

6. Note the `AlarmActions` value; this will provide the SNS topic ARN value.
7. Ensure there is at least one active subscriber to the SNS topic:

```
aws sns list-subscriptions-by-topic --topic-arn <sns-topic-arn>
```

- At least one subscription should have "SubscriptionArn" with a valid AWS ARN.
    - Example of valid "SubscriptionArn": `arn:aws:sns:<region>:<account-id>:<sns-topic-name>:<subscription-id>`

Remediation:

If you are using CloudTrail trails and CloudWatch, perform the following steps to set up the metric filter, alarm, SNS topic, and subscription:

1. Create a metric filter based on the provided filter pattern that checks for CMKs that have been disabled or scheduled for deletion and uses the `<trail-log-group-name>` taken from audit step 1:

```
aws logs put-metric-filter --log-group-name <trail-log-group-name> --
filter-name <disable-or-delete-cmk-changes-metric> --metric-
transformations metricName=<disable-or-delete-cmk-changes-
metric>,metricNamespace='CISBenchmark',metricValue=1 --filter-pattern
'{($.eventSource = kms.amazonaws.com) &&
(($.eventName=DisableKey)||($.eventName=ScheduleKeyDeletion)) }'
```

Note: You can choose your own `metricName` and `metricNamespace` strings. Using the same `metricNamespace` for all Foundations Benchmark metrics will group them together.

2. Create an SNS topic that the alarm will notify:

```
aws sns create-topic --name <sns-topic-name>
```

Note: You can execute this command once and then reuse the same topic for all monitoring alarms.
Note: Capture the `TopicArn` that is displayed when creating the SNS topic in step 2.

3. Create an SNS subscription for the topic created in step 2:

```
aws sns subscribe --topic-arn <sns-topic-arn> --protocol <sns-protocol>
--notification-endpoint <sns-subscription-endpoints>
```

Note: You can execute this command once and then reuse the same subscription for all monitoring alarms.

4. Create an alarm that is associated with the CloudWatch Logs metric filter created in step 1 and the SNS topic created in step 2:

```
aws cloudwatch put-metric-alarm --alarm-name <disable-or-delete-cmk-
changes-alarm> --metric-name <disable-or-delete-cmk-changes-metric> --
statistic Sum --period 300 --threshold 1 --comparison-operator
GreaterThanOrEqualToThreshold --evaluation-periods 1 --namespace
'CISBenchmark' --alarm-actions <sns-topic-arn>
```

References:

1. CCE-79192-1
2. https://docs.aws.amazon.com/awscloudtrail/latest/userguide/cloudwatch-alarms-for-cloudtrail.html
3. https://docs.aws.amazon.com/awscloudtrail/latest/userguide/receive-cloudtrail-log-files-from-multiple-regions.html
4. https://docs.aws.amazon.com/sns/latest/dg/SubscribeTopic.html

Additional Information:

Configuring a log metric filter and alarm on a multi-region (global) CloudTrail trail:

- ensures that activities from all regions (both used and unused) are monitored
- ensures that activities on all supported global services are monitored
- ensures that all management events across all regions are monitored

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	8.5 Collect Detailed Audit Logs Configure detailed audit logging for enterprise assets containing sensitive data. Include event source, date, username, timestamp, source addresses, destination addresses, and other useful elements that could assist in a forensic investigation.		●	●
v8	8.11 Conduct Audit Log Reviews Conduct reviews of audit logs to detect anomalies or abnormal events that could indicate a potential threat. Conduct reviews on a weekly, or more frequent, basis.		●	●
v7	6.3 Enable Detailed Logging Enable system logging to include detailed information such as an event source, date, user, timestamp, source addresses, destination addresses, and other useful elements.		●	●

4.8 Ensure S3 bucket policy changes are monitored (Manual)

Profile Applicability:

- Level 1

Description:

Real-time monitoring of API calls can be achieved by directing CloudTrail Logs to CloudWatch Logs or an external Security Information and Event Management (SIEM) environment, and establishing corresponding metric filters and alarms.

It is recommended that a metric filter and alarm be established for changes to S3 bucket policies.

Rationale:

CloudWatch is an AWS native service that allows you to observe and monitor resources and applications. CloudTrail logs can also be sent to an external Security Information and Event Management (SIEM) environment for monitoring and alerting.

Monitoring changes to S3 bucket policies may reduce the time it takes to detect and correct permissive policies on sensitive S3 buckets.

Audit:

If you are using CloudTrail trails and CloudWatch, perform the following to ensure that there is at least one active multi-region CloudTrail trail with the prescribed metric filters and alarms configured:

1. Identify the log group name that is configured for use with the active multi-region CloudTrail trail:

- List all CloudTrail trails: `aws cloudtrail describe-trails`
- Identify multi-region CloudTrail trails: `Trails with "IsMultiRegionTrail" set to true`
- Note the value associated with "Name":`<trail-name>`
- Note the `<trail-log-group-name>` within the value associated with "CloudWatchLogsLogGroupArn"
  - Example: `arn:aws:logs:<region>:<account-id>:log-group:<trail-log-group-name>:*`
- Ensure the identified multi-region CloudTrail trail is active:
  - `aws cloudtrail get-trail-status --name <trail-name>`
    - Ensure `IsLogging` is set to `TRUE`
- Ensure the identified multi-region CloudTrail trail captures all management events:
  - `aws cloudtrail get-event-selectors --trail-name <trail-name>`

        - Ensure there is at least one `event selector` for a trail with `IncludeManagementEvents` set to `true` and `ReadWriteType` set to `All`

2. Get a list of all associated metric filters for the `<trail-log-group-name>` captured in step 1:

```
aws logs describe-metric-filters --log-group-name <trail-log-group-
name>
```

3. Ensure the output from the above command contains the following:

```
"filterPattern": "{ ($.eventSource = s3.amazonaws.com) && (($.eventName
= PutBucketAcl) || ($.eventName = PutBucketPolicy) || ($.eventName =
PutBucketCors) || ($.eventName = PutBucketLifecycle) || ($.eventName =
PutBucketReplication) || ($.eventName = DeleteBucketPolicy) ||
($.eventName = DeleteBucketCors) || ($.eventName =
DeleteBucketLifecycle) || ($.eventName = DeleteBucketReplication)) }"
```

4. Note the `<s3-bucket-policy-changes-metric>` value associated with the `filterPattern` from step 3.
5. Get a list of CloudWatch alarms, and filter on the `<s3-bucket-policy-changes-metric>` captured in step 4:

```
aws cloudwatch describe-alarms --query 'MetricAlarms[?MetricName==<s3-
bucket-policy-changes-metric>]'
```

6. Note the `AlarmActions` value; this will provide the SNS topic ARN value.
7. Ensure there is at least one active subscriber to the SNS topic:

```
aws sns list-subscriptions-by-topic --topic-arn <sns-topic-arn>
```

- At least one subscription should have "SubscriptionArn" with a valid AWS ARN.
    - Example of valid "SubscriptionArn": `arn:aws:sns:<region>:<account-id>:<sns-topic-name>:<subscription-id>`

Remediation:

If you are using CloudTrail trails and CloudWatch, perform the following steps to set up the metric filter, alarm, SNS topic, and subscription:

1. Create a metric filter based on the provided filter pattern that checks for changes to S3 bucket policies and uses the `<trail-log-group-name>` taken from audit step 1:

```
aws logs put-metric-filter --log-group-name <trail-log-group-name> --
filter-name <s3-bucket-policy-changes-metric> --metric-transformations
metricName=<s3-bucket-policy-changes-
metric>,metricNamespace='CISBenchmark',metricValue=1 --filter-pattern
'{ ($.eventSource = s3.amazonaws.com) && (($.eventName = PutBucketAcl)
|| ($.eventName = PutBucketPolicy) || ($.eventName = PutBucketCors) ||
($.eventName = PutBucketLifecycle) || ($.eventName =
PutBucketReplication) || ($.eventName = DeleteBucketPolicy) ||
($.eventName = DeleteBucketCors) || ($.eventName =
DeleteBucketLifecycle) || ($.eventName = DeleteBucketReplication)) }'
```

Note: You can choose your own `metricName` and `metricNamespace` strings. Using the same `metricNamespace` for all Foundations Benchmark metrics will group them together.

2. Create an SNS topic that the alarm will notify:

```
aws sns create-topic --name <sns-topic-name>
```

Note: You can execute this command once and then reuse the same topic for all monitoring alarms.
Note: Capture the `TopicArn` that is displayed when creating the SNS topic in step 2.

3. Create an SNS subscription for the topic created in step 2:

```
aws sns subscribe --topic-arn <sns-topic-arn> --protocol <sns-protocol>
--notification-endpoint <sns-subscription-endpoints>
```

Note: You can execute this command once and then reuse the same subscription for all monitoring alarms.

4. Create an alarm that is associated with the CloudWatch Logs metric filter created in step 1 and the SNS topic created in step 2:

```
aws cloudwatch put-metric-alarm --alarm-name <s3-bucket-policy-changes-
alarm> --metric-name <s3-bucket-policy-changes-metric> --statistic Sum
--period 300 --threshold 1 --comparison-operator
GreaterThanOrEqualToThreshold --evaluation-periods 1 --namespace
'CISBenchmark' --alarm-actions <sns-topic-arn>
```

References:

1. CCE-79193-9
2. https://docs.aws.amazon.com/awscloudtrail/latest/userguide/cloudwatch-alarms-for-cloudtrail.html
3. https://docs.aws.amazon.com/awscloudtrail/latest/userguide/receive-cloudtrail-log-files-from-multiple-regions.html
4. https://docs.aws.amazon.com/sns/latest/dg/SubscribeTopic.html

Additional Information:

Configuring a log metric filter and alarm on a multi-region (global) CloudTrail trail:

- ensures that activities from all regions (both used and unused) are monitored
- ensures that activities on all supported global services are monitored
- ensures that all management events across all regions are monitored

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	8.5 Collect Detailed Audit Logs Configure detailed audit logging for enterprise assets containing sensitive data. Include event source, date, username, timestamp, source addresses, destination addresses, and other useful elements that could assist in a forensic investigation.		●	●
v8	8.11 Conduct Audit Log Reviews Conduct reviews of audit logs to detect anomalies or abnormal events that could indicate a potential threat. Conduct reviews on a weekly, or more frequent, basis.		●	●
v7	6.2 Activate audit logging Ensure that local logging has been enabled on all systems and networking devices.	●	●	●
v7	6.3 Enable Detailed Logging Enable system logging to include detailed information such as an event source, date, user, timestamp, source addresses, destination addresses, and other useful elements.		●	●

4.9 Ensure AWS Config configuration changes are monitored (Manual)

Profile Applicability:

- Level 2

Description:

Real-time monitoring of API calls can be achieved by directing CloudTrail Logs to CloudWatch Logs or an external Security Information and Event Management (SIEM) environment, and establishing corresponding metric filters and alarms.

It is recommended that a metric filter and alarm be established for detecting changes to AWS Config's configurations.

Rationale:

CloudWatch is an AWS native service that allows you to observe and monitor resources and applications. CloudTrail logs can also be sent to an external Security Information and Event Management (SIEM) environment for monitoring and alerting.

Monitoring changes to the AWS Config configuration will help ensure sustained visibility of the configuration items within the AWS account.

Audit:

If you are using CloudTrail trails and CloudWatch, perform the following to ensure that there is at least one active multi-region CloudTrail trail with the prescribed metric filters and alarms configured:

1. Identify the log group name that is configured for use with the active multi-region CloudTrail trail:

- List all CloudTrail trails: `aws cloudtrail describe-trails`
- Identify multi-region CloudTrail trails: `Trails with "IsMultiRegionTrail" set to true`
- Note the value associated with "Name":`<trail-name>`
- Note the `<trail-log-group-name>` within the value associated with "CloudWatchLogsLogGroupArn"
    - Example: `arn:aws:logs:<region>:<account-id>:log-group:<trail-log-group-name>:*`
- Ensure the identified multi-region CloudTrail trail is active:
    - `aws cloudtrail get-trail-status --name <trail-name>`
        - Ensure `IsLogging` is set to `TRUE`
- Ensure the identified multi-region CloudTrail trail captures all management events:
    - `aws cloudtrail get-event-selectors --trail-name <trail-name>`

        - Ensure there is at least one `event selector` for a trail with `IncludeManagementEvents` set to `true` and `ReadWriteType` set to `All`

2. Get a list of all associated metric filters for the `<trail-log-group-name>` captured in step 1:

```
aws logs describe-metric-filters --log-group-name <trail-log-group-
name>
```

3. Ensure the output from the above command contains the following:

```
"filterPattern": "{ ($.eventSource = config.amazonaws.com) &&
(($.eventName=StopConfigurationRecorder)||($.eventName=DeleteDeliveryCh
annel)||($.eventName=PutDeliveryChannel)||($.eventName=PutConfiguration
Recorder)) }"
```

4. Note the `<aws-config-changes-metric>` value associated with the `filterPattern` from step 3.
5. Get a list of CloudWatch alarms, and filter on the `<aws-config-changes-metric>` captured in step 4:

```
aws cloudwatch describe-alarms --query 'MetricAlarms[?MetricName==<aws-
config-changes-metric>]'
```

6. Note the `AlarmActions` value; this will provide the SNS topic ARN value.
7. Ensure there is at least one active subscriber to the SNS topic:

```
aws sns list-subscriptions-by-topic --topic-arn <sns-topic-arn>
```

- At least one subscription should have "SubscriptionArn" with a valid AWS ARN.
    - Example of valid "SubscriptionArn": `arn:aws:sns:<region>:<account-id>:<sns-topic-name>:<subscription-id>`

Remediation:

If you are using CloudTrail trails and CloudWatch, perform the following steps to set up the metric filter, alarm, SNS topic, and subscription:

1. Create a metric filter based on the provided filter pattern that checks for AWS Configuration changes and uses the `<trail-log-group-name>` taken from audit step 1:

```
aws logs put-metric-filter --log-group-name <trail-log-group-name> --
filter-name <aws-config-changes-metric> --metric-transformations
metricName=<aws-config-changes-
metric>,metricNamespace='CISBenchmark',metricValue=1 --filter-pattern
'{ ($.eventSource = config.amazonaws.com) &&
(($.eventName=StopConfigurationRecorder)||($.eventName=DeleteDeliveryCh
annel)||($.eventName=PutDeliveryChannel)||($.eventName=PutConfiguration
Recorder)) }'
```

Note: You can choose your own `metricName` and `metricNamespace` strings. Using the same `metricNamespace` for all Foundations Benchmark metrics will group them together.

2. Create an SNS topic that the alarm will notify:

```
aws sns create-topic --name <sns-topic-name>
```

Note: You can execute this command once and then reuse the same topic for all monitoring alarms.
Note: Capture the `TopicArn` that is displayed when creating the SNS topic in step 2.

3. Create an SNS subscription for the topic created in step 2:

```
aws sns subscribe --topic-arn <sns-topic-arn> --protocol <sns-protocol>
--notification-endpoint <sns-subscription-endpoints>
```

Note: You can execute this command once and then reuse the same subscription for all monitoring alarms.

4. Create an alarm that is associated with the CloudWatch Logs metric filter created in step 1 and the SNS topic created in step 2:

```
aws cloudwatch put-metric-alarm --alarm-name <aws-config-changes-alarm>
--metric-name <aws-config-changes-metric> --statistic Sum --period 300
--threshold 1 --comparison-operator GreaterThanOrEqualToThreshold --
evaluation-periods 1 --namespace 'CISBenchmark' --alarm-actions <sns-
topic-arn>
```

References:

1. CCE-79194-7
2. https://docs.aws.amazon.com/awscloudtrail/latest/userguide/cloudwatch-alarms-for-cloudtrail.html
3. https://docs.aws.amazon.com/awscloudtrail/latest/userguide/receive-cloudtrail-log-files-from-multiple-regions.html
4. https://docs.aws.amazon.com/sns/latest/dg/SubscribeTopic.html

Additional Information:

Configuring a log metric filter and alarm on a multi-region (global) CloudTrail trail:

- ensures that activities from all regions (both used and unused) are monitored
- ensures that activities on all supported global services are monitored
- ensures that all management events across all regions are monitored

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	8.5 Collect Detailed Audit Logs Configure detailed audit logging for enterprise assets containing sensitive data. Include event source, date, username, timestamp, source addresses, destination addresses, and other useful elements that could assist in a forensic investigation.		●	●
v8	8.11 Conduct Audit Log Reviews Conduct reviews of audit logs to detect anomalies or abnormal events that could indicate a potential threat. Conduct reviews on a weekly, or more frequent, basis.		●	●
v7	6.3 Enable Detailed Logging Enable system logging to include detailed information such as an event source, date, user, timestamp, source addresses, destination addresses, and other useful elements.		●	●
v7	11.2 Document Traffic Configuration Rules All configuration rules that allow traffic to flow through network devices should be documented in a configuration management system with a specific business reason for each rule, a specific individual's name responsible for that business need, and an expected duration of the need.		●	●

4.10 Ensure security group changes are monitored (Manual)

Profile Applicability:

- Level 2

Description:

Real-time monitoring of API calls can be achieved by directing CloudTrail Logs to CloudWatch Logs or an external Security Information and Event Management (SIEM) environment, and establishing corresponding metric filters and alarms.

Security groups are stateful packet filters that control ingress and egress traffic within a VPC.

It is recommended that a metric filter and alarm be established to detect changes to security groups.

Rationale:

CloudWatch is an AWS native service that allows you to observe and monitor resources and applications. CloudTrail logs can also be sent to an external Security Information and Event Management (SIEM) environment for monitoring and alerting.

Monitoring changes to security groups will help ensure that resources and services are not unintentionally exposed.

Impact:

This may require additional 'tuning' to eliminate false positives and filter out expected activity so that anomalies are easier to detect.

Audit:

If you are using CloudTrail trails and CloudWatch, perform the following to ensure that there is at least one active multi-region CloudTrail trail with the prescribed metric filters and alarms configured:

1. Identify the log group name that is configured for use with the active multi-region CloudTrail trail:

- List all CloudTrail trails: `aws cloudtrail describe-trails`
- Identify multi-region CloudTrail trails: `Trails with "IsMultiRegionTrail" set to true`
- Note the value associated with "Name":`<trail-name>`
- Note the `<trail-log-group-name>` within the value associated with "CloudWatchLogsLogGroupArn"
    - Example: `arn:aws:logs:<region>:<account-id>:log-group:<trail-log-group-name>:*`
- Ensure the identified multi-region CloudTrail trail is active:

    - `aws cloudtrail get-trail-status --name <trail-name>`
        - Ensure `IsLogging` is set to `TRUE`
- Ensure the identified multi-region CloudTrail trail captures all management events:
    - `aws cloudtrail get-event-selectors --trail-name <trail-name>`
        - Ensure there is at least one `event selector` for a trail with `IncludeManagementEvents` set to `true` and `ReadWriteType` set to `All`

2. Get a list of all associated metric filters for the `<trail-log-group-name>` captured in step 1:

```
aws logs describe-metric-filters --log-group-name <trail-log-group-
name>
```

3. Ensure the output from the above command contains the following:

```
"filterPattern": "{ ($.eventName = AuthorizeSecurityGroupIngress) ||
($.eventName = AuthorizeSecurityGroupEgress) || ($.eventName =
RevokeSecurityGroupIngress) || ($.eventName =
RevokeSecurityGroupEgress) || ($.eventName = CreateSecurityGroup) ||
($.eventName = DeleteSecurityGroup) || ($.eventName =
ModifySecurityGroupRules) }"
```

4. Note the `<security-group-changes-metric>` value associated with the `filterPattern` from step 3.
5. Get a list of CloudWatch alarms, and filter on the `<security-group-changes-metric>` captured in step 4:

```
aws cloudwatch describe-alarms --query
"MetricAlarms[?MetricName==<security-group-changes-metric>]"
```

6. Note the `AlarmActions` value; this will provide the SNS topic ARN value.
7. Ensure there is at least one active subscriber to the SNS topic:

```
aws sns list-subscriptions-by-topic --topic-arn <sns-topic-arn>
```

- At least one subscription should have "SubscriptionArn" with a valid AWS ARN.
    - Example of valid "SubscriptionArn": `arn:aws:sns:<region>:<account-id>:<sns-topic-name>:<subscription-id>`

Remediation:

If you are using CloudTrail trails and CloudWatch, perform the following steps to set up the metric filter, alarm, SNS topic, and subscription:

1. Create a metric filter based on the provided filter pattern that checks for security groups changes and uses the `<trail-log-group-name>` taken from audit step 1:

```
aws logs put-metric-filter --log-group-name <trail-log-group-name> --
filter-name <security-group-changes-metric> --metric-transformations
metricName=<security-group-changes-
metric>,metricNamespace="CISBenchmark",metricValue=1 --filter-pattern
"{ ($.eventName = AuthorizeSecurityGroupIngress) || ($.eventName =
AuthorizeSecurityGroupEgress) || ($.eventName =
RevokeSecurityGroupIngress) || ($.eventName =
RevokeSecurityGroupEgress) || ($.eventName = CreateSecurityGroup) ||
($.eventName = DeleteSecurityGroup) || ($.eventName =
ModifySecurityGroupRules) }"
```

   Note: You can choose your own `metricName` and `metricNamespace` strings. Using the same `metricNamespace` for all Foundations Benchmark metrics will group them together.

2. Create an SNS topic that the alarm will notify:

```
aws sns create-topic --name <sns-topic-name>
```

   Note: You can execute this command once and then reuse the same topic for all monitoring alarms.
   Note: Capture the `TopicArn` that is displayed when creating the SNS topic in step 2.

3. Create an SNS subscription for the topic created in step 2:

```
aws sns subscribe --topic-arn <sns-topic-arn> --protocol <sns-protocol>
--notification-endpoint <sns-subscription-endpoints>
```

   Note: You can execute this command once and then reuse the same subscription for all monitoring alarms.

4. Create an alarm that is associated with the CloudWatch Logs metric filter created in step 1 and the SNS topic created in step 2:

```
aws cloudwatch put-metric-alarm --alarm-name <security-group-changes-
alarm> --metric-name <security-group-changes-metric> --statistic Sum --
period 300 --threshold 1 --comparison-operator
GreaterThanOrEqualToThreshold --evaluation-periods 1 --namespace
"CISBenchmark" --alarm-actions <sns-topic-arn>
```

References:

1. CCE-79195-4

2. https://docs.aws.amazon.com/awscloudtrail/latest/userguide/receive-cloudtrail-log-files-from-multiple-regions.html
3. https://docs.aws.amazon.com/awscloudtrail/latest/userguide/cloudwatch-alarms-for-cloudtrail.html
4. https://docs.aws.amazon.com/sns/latest/dg/SubscribeTopic.html
5. https://docs.aws.amazon.com/AWSEC2/latest/APIReference/API_ModifySecurityGroupRules.html

Additional Information:

Configuring a log metric filter and alarm on a multi-region (global) CloudTrail trail:

- ensures that activities from all regions (both used and unused) are monitored
- ensures that activities on all supported global services are monitored
- ensures that all management events across all regions are monitored

AWS has recently introduced a new API, ModifySecurityGroupRules, which modifies the rules of a security group.

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.3 Configure Data Access Control Lists Configure data access control lists based on a user's need to know. Apply data access control lists, also known as access permissions, to local and remote file systems, databases, and applications.	●	●	●
v8	8.5 Collect Detailed Audit Logs Configure detailed audit logging for enterprise assets containing sensitive data. Include event source, date, username, timestamp, source addresses, destination addresses, and other useful elements that could assist in a forensic investigation.		●	●
v8	8.11 Conduct Audit Log Reviews Conduct reviews of audit logs to detect anomalies or abnormal events that could indicate a potential threat. Conduct reviews on a weekly, or more frequent, basis.		●	●
v7	6.2 Activate audit logging Ensure that local logging has been enabled on all systems and networking devices.	●	●	●
v7	6.3 Enable Detailed Logging Enable system logging to include detailed information such as an event source, date, user, timestamp, source addresses, destination addresses, and other useful elements.		●	●

Controls Version	Control	IG 1	IG 2	IG 3
v7	14.6 Protect Information through Access Control Lists Protect all information stored on systems with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities.	●	●	●

4.11 Ensure Network Access Control List (NACL) changes are monitored (Manual)

Profile Applicability:

- Level 2

Description:

Real-time monitoring of API calls can be achieved by directing CloudTrail Logs to CloudWatch Logs or an external Security Information and Event Management (SIEM) environment, and establishing corresponding metric filters and alarms.

NACLs are used as a stateless packet filter to control ingress and egress traffic for subnets within a VPC. It is recommended that a metric filter and alarm be established for any changes made to NACLs.

Rationale:

CloudWatch is an AWS native service that allows you to observe and monitor resources and applications. CloudTrail logs can also be sent to an external Security Information and Event Management (SIEM) environment for monitoring and alerting.

Monitoring changes to NACLs will help ensure that AWS resources and services are not unintentionally exposed.

Audit:

If you are using CloudTrail trails and CloudWatch, perform the following to ensure that there is at least one active multi-region CloudTrail trail with the prescribed metric filters and alarms configured:

1. Identify the log group name that is configured for use with the active multi-region CloudTrail trail:

- List all CloudTrail trails: `aws cloudtrail describe-trails`
- Identify multi-region CloudTrail trails: `Trails with "IsMultiRegionTrail" set to true`
- Note the value associated with "Name":`<trail-name>`
- Note the `<trail-log-group-name>` within the value associated with "CloudWatchLogsLogGroupArn"
    - Example: `arn:aws:logs:<region>:<account-id>:log-group:<trail-log-group-name>:*`
- Ensure the identified multi-region CloudTrail trail is active:
    - `aws cloudtrail get-trail-status --name <trail-name>`
        - Ensure `IsLogging` is set to `TRUE`
- Ensure the identified multi-region CloudTrail trail captures all management events:

- `aws cloudtrail get-event-selectors --trail-name <trail-name>`
  - Ensure there is at least one `event selector` for a trail with `IncludeManagementEvents` set to `true` and `ReadWriteType` set to `All`

2. Get a list of all associated metric filters for the `<trail-log-group-name>` captured in step 1:

```
aws logs describe-metric-filters --log-group-name <trail-log-group-
name>
```

3. Ensure the output from the above command contains the following:

```
"filterPattern": "{ ($.eventName = CreateNetworkAcl) || ($.eventName =
CreateNetworkAclEntry) || ($.eventName = DeleteNetworkAcl) ||
($.eventName = DeleteNetworkAclEntry) || ($.eventName =
ReplaceNetworkAclEntry) || ($.eventName = ReplaceNetworkAclAssociation)
}"
```

4. Note the `<nacl-changes-metric>` value associated with the `filterPattern` from step 3.
5. Get a list of CloudWatch alarms, and filter on the `<nacl-changes-metric>` captured in step 4:

```
aws cloudwatch describe-alarms --query
'MetricAlarms[?MetricName==<nacl-changes-metric>]'
```

6. Note the `AlarmActions` value; this will provide the SNS topic ARN value.
7. Ensure there is at least one active subscriber to the SNS topic:

```
aws sns list-subscriptions-by-topic --topic-arn <sns-topic-arn>
```

- At least one subscription should have "SubscriptionArn" with a valid AWS ARN.
  - Example of valid "SubscriptionArn": `arn:aws:sns:<region>:<account-id>:<sns-topic-name>:<subscription-id>`

Remediation:

If you are using CloudTrail trails and CloudWatch, perform the following steps to set up the metric filter, alarm, SNS topic, and subscription:

1. Create a metric filter based on the provided filter pattern that checks for NACL changes and uses the `<trail-log-group-name>` taken from audit step 1:

```
aws logs put-metric-filter --log-group-name <trail-log-group-name> --
filter-name <nacl-changes-metric> --metric-transformations
metricName=<nacl-changes-
metric>,metricNamespace='CISBenchmark',metricValue=1 --filter-pattern
'{ ($.eventName = CreateNetworkAcl) || ($.eventName =
CreateNetworkAclEntry) || ($.eventName = DeleteNetworkAcl) ||
($.eventName = DeleteNetworkAclEntry) || ($.eventName =
ReplaceNetworkAclEntry) || ($.eventName = ReplaceNetworkAclAssociation)
}'
```

Note: You can choose your own `metricName` and `metricNamespace` strings. Using the same `metricNamespace` for all Foundations Benchmark metrics will group them together.

2. Create an SNS topic that the alarm will notify:

```
aws sns create-topic --name <sns-topic-name>
```

Note: You can execute this command once and then reuse the same topic for all monitoring alarms.
Note: Capture the `TopicArn` that is displayed when creating the SNS topic in step 2.

3. Create an SNS subscription for the topic created in step 2:

```
aws sns subscribe --topic-arn <sns-topic-arn> --protocol <sns-protocol>
--notification-endpoint <sns-subscription-endpoints>
```

Note: You can execute this command once and then reuse the same subscription for all monitoring alarms.

4. Create an alarm that is associated with the CloudWatch Logs metric filter created in step 1 and the SNS topic created in step 2:

```
aws cloudwatch put-metric-alarm --alarm-name <nacl-changes-alarm> --
metric-name <nacl-changes-metric> --statistic Sum --period 300 --
threshold 1 --comparison-operator GreaterThanOrEqualToThreshold --
evaluation-periods 1 --namespace 'CISBenchmark' --alarm-actions <sns-
topic-arn>
```

References:

1. CCE-79196-2
2. https://docs.aws.amazon.com/awscloudtrail/latest/userguide/receive-cloudtrail-log-files-from-multiple-regions.html
3. https://docs.aws.amazon.com/awscloudtrail/latest/userguide/cloudwatch-alarms-for-cloudtrail.html
4. https://docs.aws.amazon.com/sns/latest/dg/SubscribeTopic.html

Additional Information:

Configuring a log metric filter and alarm on a multi-region (global) CloudTrail trail:

- ensures that activities from all regions (both used and unused) are monitored
- ensures that activities on all supported global services are monitored
- ensures that all management events across all regions are monitored

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	8.5 Collect Detailed Audit Logs Configure detailed audit logging for enterprise assets containing sensitive data. Include event source, date, username, timestamp, source addresses, destination addresses, and other useful elements that could assist in a forensic investigation.		●	●
v8	8.11 Conduct Audit Log Reviews Conduct reviews of audit logs to detect anomalies or abnormal events that could indicate a potential threat. Conduct reviews on a weekly, or more frequent, basis.		●	●
v7	6.3 Enable Detailed Logging Enable system logging to include detailed information such as an event source, date, user, timestamp, source addresses, destination addresses, and other useful elements.		●	●
v7	11.3 Use Automated Tools to Verify Standard Device Configurations and Detect Changes Compare all network device configuration against approved security configurations defined for each network device in use and alert when any deviations are discovered.		●	●

4.12 Ensure changes to network gateways are monitored (Manual)

Profile Applicability:

- Level 1

Description:

Real-time monitoring of API calls can be achieved by directing CloudTrail Logs to CloudWatch Logs or an external Security Information and Event Management (SIEM) environment, and establishing corresponding metric filters and alarms.

Network gateways are required to send and receive traffic to a destination outside of a VPC. It is recommended that a metric filter and alarm be established for changes to network gateways.

Rationale:

CloudWatch is an AWS native service that allows you to observe and monitor resources and applications. CloudTrail logs can also be sent to an external Security Information and Event Management (SIEM) environment for monitoring and alerting.

Monitoring changes to network gateways will help ensure that all ingress/egress traffic traverses the VPC border via a controlled path.

Impact:

Monitoring changes to network gateways helps detect unauthorized modifications that could compromise network security. Implementing automated monitoring and alerts can improve incident response times, but it may require additional configuration and maintenance efforts.

Audit:

If you are using CloudTrail trails and CloudWatch, perform the following to ensure that there is at least one active multi-region CloudTrail trail with the prescribed metric filters and alarms configured:

1. Identify the log group name that is configured for use with the active multi-region CloudTrail trail:

- List all CloudTrail trails: `aws cloudtrail describe-trails`
- Identify multi-region CloudTrail trails: `Trails with "IsMultiRegionTrail" set to true`
- Note the value associated with "Name":`<trail-name>`
- Note the `<trail-log-group-name>` within the value associated with "CloudWatchLogsLogGroupArn"

    - Example: `arn:aws:logs:<region>:<account-id>:log-group:<trail-log-group-name>:*`
- Ensure the identified multi-region CloudTrail trail is active:
    - `aws cloudtrail get-trail-status --name <trail-name>`
        - Ensure `IsLogging` is set to `TRUE`
- Ensure the identified multi-region CloudTrail trail captures all management events:
    - `aws cloudtrail get-event-selectors --trail-name <trail-name>`
        - Ensure there is at least one `event selector` for a trail with `IncludeManagementEvents` set to `true` and `ReadWriteType` set to `All`

2. Get a list of all associated metric filters for the `<trail-log-group-name>` captured in step 1:

```
aws logs describe-metric-filters --log-group-name <trail-log-group-name>
```

3. Ensure the output from the above command contains the following:

```
"filterPattern": "{ ($.eventName = CreateCustomerGateway) || ($.eventName = DeleteCustomerGateway) || ($.eventName = AttachInternetGateway) || ($.eventName = CreateInternetGateway) || ($.eventName = DeleteInternetGateway) || ($.eventName = DetachInternetGateway) }"
```

4. Note the `<network-gw-changes-metric>` value associated with the `filterPattern` from step 3.
5. Get a list of CloudWatch alarms, and filter on the `<network-gw-changes-metric>` captured in step 4:

```
aws cloudwatch describe-alarms --query 'MetricAlarms[?MetricName==<network-gw-changes-metric>]'
```

6. Note the `AlarmActions` value; this will provide the SNS topic ARN value.
7. Ensure there is at least one active subscriber to the SNS topic:

```
aws sns list-subscriptions-by-topic --topic-arn <sns-topic-arn>
```

- At least one subscription should have "SubscriptionArn" with a valid AWS ARN.
    - Example of valid "SubscriptionArn": `arn:aws:sns:<region>:<account-id>:<sns-topic-name>:<subscription-id>`

8. Ensure automated monitoring is enabled:

```
aws cloudwatch put-metric-alarm --alarm-name NetworkGatewayChanges --
metric-name GatewayChanges --namespace AWS/EC2 --statistic Sum --period
300 --threshold 1 --comparison-operator GreaterThanOrEqualToThreshold -
-evaluation-periods 1 --alarm-actions <sns-topic-arn>
```

Remediation:

If you are using CloudTrail trails and CloudWatch, perform the following steps to set up the metric filter, alarm, SNS topic, and subscription:

1. Create a metric filter based on the provided filter pattern that checks for network gateways changes and uses the `<trail-log-group-name>` taken from audit step 1:

```
aws logs put-metric-filter --log-group-name <trail-log-group-name> --
filter-name <network-gw-changes-metric> --metric-transformations
metricName=<network-gw-changes-
metric>,metricNamespace='CISBenchmark',metricValue=1 --filter-pattern
'{ ($.eventName = CreateCustomerGateway) || ($.eventName =
DeleteCustomerGateway) || ($.eventName = AttachInternetGateway) ||
($.eventName = CreateInternetGateway) || ($.eventName =
DeleteInternetGateway) || ($.eventName = DetachInternetGateway) }'
```

   Note: You can choose your own `metricName` and `metricNamespace` strings. Using the same `metricNamespace` for all Foundations Benchmark metrics will group them together.

2. Create an SNS topic that the alarm will notify:

```
aws sns create-topic --name <sns-topic-name>
```

   Note: You can execute this command once and then reuse the same topic for all monitoring alarms.
   Note: Capture the `TopicArn` that is displayed when creating the SNS topic in step 2.

3. Create an SNS subscription for the topic created in step 2:

```
aws sns subscribe --topic-arn <sns-topic-arn> --protocol <sns-protocol>
--notification-endpoint <sns-subscription-endpoints>
```

   Note: You can execute this command once and then reuse the same subscription for all monitoring alarms.

4. Create an alarm that is associated with the CloudWatch Logs metric filter created in step 1 and the SNS topic created in step 2:

```
aws cloudwatch put-metric-alarm --alarm-name <network-gw-changes-alarm>
--metric-name <network-gw-changes-metric> --statistic Sum --period 300
--threshold 1 --comparison-operator GreaterThanOrEqualToThreshold --
evaluation-periods 1 --namespace 'CISBenchmark' --alarm-actions <sns-
topic-arn>
```

5. Implement logging and alerting mechanisms:

```
aws sns create-topic --name NetworkGatewayChangesAlerts
aws sns subscribe --topic-arn <sns-topic-arn> --protocol email --
notification-endpoint <email-address>
aws cloudwatch put-metric-alarm --alarm-name NetworkGatewayChangesAlarm
--metric-name GatewayChanges --namespace AWS/EC2 --statistic Sum --
period 300 --threshold 1 --comparison-operator
GreaterThanOrEqualToThreshold --evaluation-periods 1 --alarm-actions
<sns-topic-arn>
```

References:

1. CCE-79197-0
2. https://docs.aws.amazon.com/awscloudtrail/latest/userguide/receive-cloudtrail-log-files-from-multiple-regions.html
3. https://docs.aws.amazon.com/awscloudtrail/latest/userguide/cloudwatch-alarms-for-cloudtrail.html
4. https://docs.aws.amazon.com/sns/latest/dg/SubscribeTopic.html

Additional Information:

Configuring a log metric filter and alarm on a multi-region (global) CloudTrail trail:

- ensures that activities from all regions (both used and unused) are monitored
- ensures that activities on all supported global services are monitored
- ensures that all management events across all regions are monitored

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	8.5 Collect Detailed Audit Logs Configure detailed audit logging for enterprise assets containing sensitive data. Include event source, date, username, timestamp, source addresses, destination addresses, and other useful elements that could assist in a forensic investigation.		●	●
v8	8.11 Conduct Audit Log Reviews Conduct reviews of audit logs to detect anomalies or abnormal events that could indicate a potential threat. Conduct reviews on a weekly, or more frequent, basis.		●	●

Controls Version	Control	IG 1	IG 2	IG 3
v7	6.2 Activate audit logging Ensure that local logging has been enabled on all systems and networking devices.	●	●	●
v7	6.3 Enable Detailed Logging Enable system logging to include detailed information such as an event source, date, user, timestamp, source addresses, destination addresses, and other useful elements.		●	●

4.13 Ensure route table changes are monitored (Manual)

Profile Applicability:

- Level 1

Description:

Real-time monitoring of API calls can be achieved by directing CloudTrail Logs to CloudWatch Logs or an external Security Information and Event Management (SIEM) environment, and establishing corresponding metric filters and alarms.

Routing tables are used to route network traffic between subnets and to network gateways.

It is recommended that a metric filter and alarm be established for changes to route tables.

Rationale:

CloudWatch is an AWS native service that allows you to observe and monitor resources and applications. CloudTrail logs can also be sent to an external Security Information and Event Management (SIEM) environment for monitoring and alerting.

Monitoring changes to route tables will help ensure that all VPC traffic flows through the expected path and prevent any accidental or intentional modifications that may lead to uncontrolled network traffic. An alarm should be triggered every time an AWS API call is performed to create, replace, delete, or disassociate a route table.

Audit:

If you are using CloudTrail trails and CloudWatch, perform the following to ensure that there is at least one active multi-region CloudTrail trail with the prescribed metric filters and alarms configured:

1. Identify the log group name that is configured for use with the active multi-region CloudTrail trail:

- List all CloudTrail trails: `aws cloudtrail describe-trails`
- Identify multi-region CloudTrail trails: `Trails with "IsMultiRegionTrail" set to true`
- Note the value associated with "Name":`<trail-name>`
- Note the `<trail-log-group-name>` within the value associated with "CloudWatchLogsLogGroupArn"
    - Example: `arn:aws:logs:<region>:<account-id>:log-group:<trail-log-group-name>:*`
- Ensure the identified multi-region CloudTrail trail is active:
    - `aws cloudtrail get-trail-status --name <trail-name>`
        - Ensure `IsLogging` is set to `TRUE`

- Ensure the identified multi-region CloudTrail trail captures all management events:
    - `aws cloudtrail get-event-selectors --trail-name <trail-name>`
        - Ensure there is at least one `event selector` for a trail with `IncludeManagementEvents` set to `true` and `ReadWriteType` set to `All`

2. Get a list of all associated metric filters for the `<trail-log-group-name>` captured in step 1:

```
aws logs describe-metric-filters --log-group-name <trail-log-group-
name>
```

3. Ensure the output from the above command contains the following:

```
"filterPattern": "{($.eventSource = ec2.amazonaws.com) && ($.eventName
= CreateRoute) || ($.eventName = CreateRouteTable) || ($.eventName =
ReplaceRoute) || ($.eventName = ReplaceRouteTableAssociation) ||
($.eventName = DeleteRouteTable) || ($.eventName = DeleteRoute) ||
($.eventName = DisassociateRouteTable) }"
```

4. Note the `<route-table-changes-metric>` value associated with the `filterPattern` from step 3.
5. Get a list of CloudWatch alarms, and filter on the `<route-table-changes-metric>` captured in step 4:

```
aws cloudwatch describe-alarms --query
'MetricAlarms[?MetricName==<route-table-changes-metric>]'
```

6. Note the `AlarmActions` value; this will provide the SNS topic ARN value.
7. Ensure there is at least one active subscriber to the SNS topic:

```
aws sns list-subscriptions-by-topic --topic-arn <sns-topic-arn>
```

- At least one subscription should have "SubscriptionArn" with a valid AWS ARN.
    - Example of valid "SubscriptionArn": `arn:aws:sns:<region>:<account-id>:<sns-topic-name>:<subscription-id>`

Remediation:

If you are using CloudTrail trails and CloudWatch, perform the following steps to set up the metric filter, alarm, SNS topic, and subscription:

1. Create a metric filter based on the provided filter pattern that checks for route table changes and uses the `<trail-log-group-name>` taken from audit step 1:

```
aws logs put-metric-filter --log-group-name <trail-log-group-name> --
filter-pattern '{ ($.eventName = CreateRoute) || ($.eventName =
CreateRouteTable) || ($.eventName = ReplaceRoute) || ($.eventName =
ReplaceRouteTableAssociation) || ($.eventName = DeleteRouteTable) ||
($.eventName = DeleteRoute) || ($.eventName = DisassociateRouteTable)
}'
```

Note: You can choose your own `metricName` and `metricNamespace` strings. Using the same `metricNamespace` for all Foundations Benchmark metrics will group them together.

2. Create an SNS topic that the alarm will notify:

```
aws sns create-topic --name <sns-topic-name>
```

Note: You can execute this command once and then reuse the same topic for all monitoring alarms.
Note: Capture the `TopicArn` that is displayed when creating the SNS topic in step 2.

3. Create an SNS subscription for the topic created in step 2:

```
aws sns subscribe --topic-arn <sns-topic-arn> --protocol <sns-protocol>
--notification-endpoint <sns-subscription-endpoints>
```

Note: You can execute this command once and then reuse the same subscription for all monitoring alarms.

4. Create an alarm that is associated with the CloudWatch Logs metric filter created in step 1 and the SNS topic created in step 2:

```
aws cloudwatch put-metric-alarm --alarm-name <route-table-changes-
alarm> --metric-name <route-table-changes-metric> --statistic Sum --
period 300 --threshold 1 --comparison-operator
GreaterThanOrEqualToThreshold --evaluation-periods 1 --namespace
'CISBenchmark' --alarm-actions <sns-topic-arn>
```

References:

1. CCE-79198-8
2. https://docs.aws.amazon.com/awscloudtrail/latest/userguide/receive-cloudtrail-log-files-from-multiple-regions.html
3. https://docs.aws.amazon.com/awscloudtrail/latest/userguide/cloudwatch-alarms-for-cloudtrail.html
4. https://docs.aws.amazon.com/sns/latest/dg/SubscribeTopic.html

Additional Information:

Configuring a log metric filter and alarm on a multi-region (global) CloudTrail trail:

- ensures that activities from all regions (both used and unused) are monitored
- ensures that activities on all supported global services are monitored
- ensures that all management events across all regions are monitored

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	8.5 Collect Detailed Audit Logs Configure detailed audit logging for enterprise assets containing sensitive data. Include event source, date, username, timestamp, source addresses, destination addresses, and other useful elements that could assist in a forensic investigation.		●	●
v8	8.11 Conduct Audit Log Reviews Conduct reviews of audit logs to detect anomalies or abnormal events that could indicate a potential threat. Conduct reviews on a weekly, or more frequent, basis.		●	●
v7	6.2 Activate audit logging Ensure that local logging has been enabled on all systems and networking devices.	●	●	●
v7	6.3 Enable Detailed Logging Enable system logging to include detailed information such as an event source, date, user, timestamp, source addresses, destination addresses, and other useful elements.		●	●

4.14 Ensure VPC changes are monitored (Manual)

Profile Applicability:

- Level 1

Description:

Real-time monitoring of API calls can be achieved by directing CloudTrail Logs to CloudWatch Logs or an external Security Information and Event Management (SIEM) environment, and establishing corresponding metric filters and alarms.

It is possible to have more than one VPC within an account; additionally, it is also possible to create a peer connection between two VPCs, enabling network traffic to route between them.

It is recommended that a metric filter and alarm be established for changes made to VPCs.

Rationale:

CloudWatch is an AWS native service that allows you to observe and monitor resources and applications. CloudTrail logs can also be sent to an external Security Information and Event Management (SIEM) environment for monitoring and alerting.

VPCs in AWS are logically isolated virtual networks that can be used to launch AWS resources. Monitoring changes to VPC configurations will help ensure that VPC traffic flow is not negatively impacted. Changes to VPCs can affect network accessibility from the public internet and additionally impact VPC traffic flow to and from the resources launched in the VPC.

Audit:

If you are using CloudTrail trails and CloudWatch, perform the following to ensure that there is at least one active multi-region CloudTrail trail with the prescribed metric filters and alarms configured:

1. Identify the log group name that is configured for use with the active multi-region CloudTrail trail:

- List all CloudTrail trails: `aws cloudtrail describe-trails`
- Identify multi-region CloudTrail trails: `Trails with "IsMultiRegionTrail" set to true`
- Note the value associated with "Name":`<trail-name>`
- Note the `<trail-log-group-name>` within the value associated with "CloudWatchLogsLogGroupArn"
    - Example: `arn:aws:logs:<region>:<account-id>:log-group:<trail-log-group-name>:*`
- Ensure the identified multi-region CloudTrail trail is active:

    - `aws cloudtrail get-trail-status --name <trail-name>`
        - Ensure `IsLogging` is set to `TRUE`
- Ensure the identified multi-region CloudTrail trail captures all management events:
    - `aws cloudtrail get-event-selectors --trail-name <trail-name>`
        - Ensure there is at least one `event selector` for a trail with `IncludeManagementEvents` set to `true` and `ReadWriteType` set to `All`

2. Get a list of all associated metric filters for the `<trail-log-group-name>` captured in step 1:

```
aws logs describe-metric-filters --log-group-name <trail-log-group-
name>
```

3. Ensure the output from the above command contains the following:

```
"filterPattern": "{ ($.eventName = CreateVpc) || ($.eventName =
DeleteVpc) || ($.eventName = ModifyVpcAttribute) || ($.eventName =
AcceptVpcPeeringConnection) || ($.eventName =
CreateVpcPeeringConnection) || ($.eventName =
DeleteVpcPeeringConnection) || ($.eventName =
RejectVpcPeeringConnection) || ($.eventName = AttachClassicLinkVpc) ||
($.eventName = DetachClassicLinkVpc) || ($.eventName =
DisableVpcClassicLink) || ($.eventName = EnableVpcClassicLink) }"
```

4. Note the `<vpc-changes-metric>` value associated with the `filterPattern` from step 3.
5. Get a list of CloudWatch alarms, and filter on the `<vpc-changes-metric>` captured in step 4:

```
aws cloudwatch describe-alarms --query 'MetricAlarms[?MetricName==<vpc-
changes-metric>]'
```

6. Note the `AlarmActions` value; this will provide the SNS topic ARN value.
7. Ensure there is at least one active subscriber to the SNS topic:

```
aws sns list-subscriptions-by-topic --topic-arn <sns-topic-arn>
```

- At least one subscription should have "SubscriptionArn" with a valid AWS ARN.
    - Example of valid "SubscriptionArn": `arn:aws:sns:<region>:<account-id>:<sns-topic-name>:<subscription-id>`

Remediation:

If you are using CloudTrail trails and CloudWatch, perform the following steps to set up the metric filter, alarm, SNS topic, and subscription:

1. Create a metric filter based on the provided filter pattern that checks for VPC changes and uses the `<trail-log-group-name>` taken from audit step 1:

```
aws logs put-metric-filter --log-group-name <trail-log-group-name> --
filter-name <vpc-changes-metric> --metric-transformations
metricName=<vpc-changes-
metric>,metricNamespace='CISBenchmark',metricValue=1 --filter-pattern
'{ ($.eventName = CreateVpc) || ($.eventName = DeleteVpc) ||
($.eventName = ModifyVpcAttribute) || ($.eventName =
AcceptVpcPeeringConnection) || ($.eventName =
CreateVpcPeeringConnection) || ($.eventName =
DeleteVpcPeeringConnection) || ($.eventName =
RejectVpcPeeringConnection) || ($.eventName = AttachClassicLinkVpc) ||
($.eventName = DetachClassicLinkVpc) || ($.eventName =
DisableVpcClassicLink) || ($.eventName = EnableVpcClassicLink) }'
```

   Note: You can choose your own `metricName` and `metricNamespace` strings. Using the same `metricNamespace` for all Foundations Benchmark metrics will group them together.

2. Create an SNS topic that the alarm will notify:

```
aws sns create-topic --name <sns-topic-name>
```

   Note: You can execute this command once and then reuse the same topic for all monitoring alarms.
   Note: Capture the `TopicArn` that is displayed when creating the SNS topic in step 2.

3. Create an SNS subscription for the topic created in step 2:

```
aws sns subscribe --topic-arn <sns-topic-arn> --protocol <sns-protocol>
--notification-endpoint <sns-subscription-endpoints>
```

   Note: You can execute this command once and then reuse the same subscription for all monitoring alarms.

4. Create an alarm that is associated with the CloudWatch Logs metric filter created in step 1 and the SNS topic created in step 2:

```
aws cloudwatch put-metric-alarm --alarm-name <vpc-changes-alarm> --
metric-name <vpc-changes-metric> --statistic Sum --period 300 --
threshold 1 --comparison-operator GreaterThanOrEqualToThreshold --
evaluation-periods 1 --namespace 'CISBenchmark' --alarm-actions <sns-
topic-arn>
```

References:

1. CCE-79199-6
2. https://docs.aws.amazon.com/awscloudtrail/latest/userguide/receive-cloudtrail-log-files-from-multiple-regions.html
3. https://docs.aws.amazon.com/awscloudtrail/latest/userguide/cloudwatch-alarms-for-cloudtrail.html
4. https://docs.aws.amazon.com/sns/latest/dg/SubscribeTopic.html

Additional Information:

Configuring a log metric filter and alarm on a multi-region (global) CloudTrail trail:

- ensures that activities from all regions (both used and unused) are monitored
- ensures that activities on all supported global services are monitored
- ensures that all management events across all regions are monitored

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	8.5 Collect Detailed Audit Logs Configure detailed audit logging for enterprise assets containing sensitive data. Include event source, date, username, timestamp, source addresses, destination addresses, and other useful elements that could assist in a forensic investigation.		●	●
v8	8.11 Conduct Audit Log Reviews Conduct reviews of audit logs to detect anomalies or abnormal events that could indicate a potential threat. Conduct reviews on a weekly, or more frequent, basis.		●	●
v7	5.5 Implement Automated Configuration Monitoring Systems Utilize a Security Content Automation Protocol (SCAP) compliant configuration monitoring system to verify all security configuration elements, catalog approved exceptions, and alert when unauthorized changes occur.		●	●
v7	6.3 Enable Detailed Logging Enable system logging to include detailed information such as an event source, date, user, timestamp, source addresses, destination addresses, and other useful elements.		●	●

4.15 Ensure AWS Organizations changes are monitored (Manual)

Profile Applicability:

- Level 1

Description:

Real-time monitoring of API calls can be achieved by directing CloudTrail Logs to CloudWatch Logs or an external Security Information and Event Management (SIEM) environment, and establishing corresponding metric filters and alarms.

It is recommended that a metric filter and alarm be established for changes made to AWS Organizations in the master AWS account.

Rationale:

CloudWatch is an AWS native service that allows you to observe and monitor resources and applications. CloudTrail logs can also be sent to an external Security Information and Event Management (SIEM) environment for monitoring and alerting.

Monitoring AWS Organizations changes can help you prevent unwanted, accidental, or intentional modifications that may lead to unauthorized access or other security breaches. This monitoring technique helps ensure that any unexpected changes made within your AWS Organizations can be investigated and that any unwanted changes can be rolled back.

Audit:

If you are using CloudTrail trails and CloudWatch, perform the following to ensure that there is at least one active multi-region CloudTrail trail with the prescribed metric filters and alarms configured:

1. Identify the log group name that is configured for use with the active multi-region CloudTrail trail:

- List all CloudTrail trails: `aws cloudtrail describe-trails`
- Identify multi-region CloudTrail trails: `Trails with "IsMultiRegionTrail" set to true`
- Note the value associated with "Name":`<trail-name>`
- Note the `<trail-log-group-name>` within the value associated with "CloudWatchLogsLogGroupArn"
    - Example: `arn:aws:logs:<region>:<account-id>:log-group:<trail-log-group-name>:*`
- Ensure the identified multi-region CloudTrail trail is active:
    - `aws cloudtrail get-trail-status --name <trail-name>`
        - Ensure `IsLogging` is set to `TRUE`
- Ensure the identified multi-region CloudTrail trail captures all management events:

- `aws cloudtrail get-event-selectors --trail-name <trail-name>`
  - Ensure there is at least one `event selector` for a trail with `IncludeManagementEvents` set to `true` and `ReadWriteType` set to `All`

2. Get a list of all associated metric filters for the `<trail-log-group-name>` captured in step 1:

```
aws logs describe-metric-filters --log-group-name <trail-log-group-
name>
```

3. Ensure the output from the above command contains the following:

```
"filterPattern": "{ ($.eventSource = organizations.amazonaws.com) &&
(($.eventName = "AcceptHandshake")  || ($.eventName = "AttachPolicy")
|| ($.eventName = "CreateAccount") || ($.eventName =
"CreateOrganizationalUnit") || ($.eventName = "CreatePolicy") ||
($.eventName = "DeclineHandshake") || ($.eventName =
"DeleteOrganization") || ($.eventName = "DeleteOrganizationalUnit") ||
($.eventName = "DeletePolicy") ||  ($.eventName = "DetachPolicy") ||
($.eventName = "DisablePolicyType") || ($.eventName =
"EnablePolicyType") || ($.eventName = "InviteAccountToOrganization") ||
($.eventName = "LeaveOrganization") || ($.eventName = "MoveAccount") ||
($.eventName = "RemoveAccountFromOrganization") || ($.eventName =
"UpdatePolicy") || ($.eventName = "UpdateOrganizationalUnit")) }"
```

4. Note the `<organizations-changes>` value associated with the `filterPattern` from step 3.
5. Get a list of CloudWatch alarms, and filter on the `<organizations-changes>` captured in step 4:

```
aws cloudwatch describe-alarms --query
'MetricAlarms[?MetricName==<organizations-changes>]'
```

6. Note the `AlarmActions` value; this will provide the SNS topic ARN value.
7. Ensure there is at least one active subscriber to the SNS topic:

```
aws sns list-subscriptions-by-topic --topic-arn <sns-topic-arn>
```

- At least one subscription should have "SubscriptionArn" with a valid AWS ARN.
  - Example of valid "SubscriptionArn": `arn:aws:sns:<region>:<account-id>:<sns-topic-name>:<subscription-id>`

Remediation:

If you are using CloudTrail trails and CloudWatch, perform the following steps to set up the metric filter, alarm, SNS topic, and subscription:

1. Create a metric filter based on the provided filter pattern that checks for AWS Organizations changes and uses the `<trail-log-group-name>` taken from audit step 1:

```
aws logs put-metric-filter --log-group-name <trail-log-group-name> --
filter-name <organizations-changes> --metric-transformations
metricName=<organizations-
changes>,metricNamespace='CISBenchmark',metricValue=1 --filter-pattern
'{ ($.eventSource = organizations.amazonaws.com) && (($.eventName =
"AcceptHandshake")  || ($.eventName = "AttachPolicy")  || ($.eventName
= "CreateAccount") || ($.eventName = "CreateOrganizationalUnit") ||
($.eventName = "CreatePolicy") || ($.eventName = "DeclineHandshake") ||
($.eventName = "DeleteOrganization") || ($.eventName =
"DeleteOrganizationalUnit") || ($.eventName = "DeletePolicy") ||
($.eventName = "DetachPolicy") || ($.eventName = "DisablePolicyType")
|| ($.eventName = "EnablePolicyType") || ($.eventName =
"InviteAccountToOrganization") || ($.eventName = "LeaveOrganization")
|| ($.eventName = "MoveAccount") || ($.eventName =
"RemoveAccountFromOrganization") || ($.eventName = "UpdatePolicy") ||
($.eventName = "UpdateOrganizationalUnit")) }'
```

Note: You can choose your own `metricName` and `metricNamespace` strings. Using the same `metricNamespace` for all Foundations Benchmark metrics will group them together.

2. Create an SNS topic that the alarm will notify:

```
aws sns create-topic --name <sns-topic-name>
```

Note: You can execute this command once and then reuse the same topic for all monitoring alarms.
Note: Capture the `TopicArn` that is displayed when creating the SNS topic in step 2.

3. Create an SNS subscription for the topic created in step 2:

```
aws sns subscribe --topic-arn <sns-topic-arn> --protocol <sns-protocol>
--notification-endpoint <sns-subscription-endpoints>
```

Note: You can execute this command once and then reuse the same subscription for all monitoring alarms.

4. Create an alarm that is associated with the CloudWatch Logs metric filter created in step 1 and the SNS topic created in step 2:

```
aws cloudwatch put-metric-alarm --alarm-name <organizations-changes> --
metric-name <organizations-changes> --statistic Sum --period 300 --
threshold 1 --comparison-operator GreaterThanOrEqualToThreshold --
evaluation-periods 1 --namespace 'CISBenchmark' --alarm-actions <sns-
topic-arn>
```

References:

1. https://docs.aws.amazon.com/awscloudtrail/latest/userguide/cloudwatch-alarms-for-cloudtrail.html
2. https://docs.aws.amazon.com/organizations/latest/userguide/orgs_security_incident-response.html

Additional Information:

Configuring a log metric filter and alarm on a multi-region (global) CloudTrail trail:

- ensures that activities from all regions (both used and unused) are monitored
- ensures that activities on all supported global services are monitored
- ensures that all management events across all regions are monitored

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	8.5 Collect Detailed Audit Logs Configure detailed audit logging for enterprise assets containing sensitive data. Include event source, date, username, timestamp, source addresses, destination addresses, and other useful elements that could assist in a forensic investigation.		●	●
v8	8.11 Conduct Audit Log Reviews Conduct reviews of audit logs to detect anomalies or abnormal events that could indicate a potential threat. Conduct reviews on a weekly, or more frequent, basis.		●	●
v7	6.2 Activate audit logging Ensure that local logging has been enabled on all systems and networking devices.	●	●	●
v7	6.3 Enable Detailed Logging Enable system logging to include detailed information such as an event source, date, user, timestamp, source addresses, destination addresses, and other useful elements.		●	●

Controls Version	Control	IG 1	IG 2	IG 3
v7	14.6 Protect Information through Access Control Lists Protect all information stored on systems with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities.	●	●	●

4.16 Ensure AWS Security Hub is enabled (Automated)

Profile Applicability:

- Level 2

Description:

Security Hub collects security data from various AWS accounts, services, and supported third-party partner products, helping you analyze your security trends and identify the highest-priority security issues. When you enable Security Hub, it begins to consume, aggregate, organize, and prioritize findings from the AWS services that you have enabled, such as Amazon GuardDuty, Amazon Inspector, and Amazon Macie. You can also enable integrations with AWS partner security products.

Rationale:

AWS Security Hub provides you with a comprehensive view of your security state in AWS and helps you check your environment against security industry standards and best practices, enabling you to quickly assess the security posture across your AWS accounts.

Impact:

It is recommended that AWS Security Hub be enabled in all regions. AWS Security Hub requires that AWS Config be enabled.

Audit:

Follow this process to evaluate AWS Security Hub configuration per region:
From Console:

1. Sign in to the AWS Management Console and open the AWS Security Hub console at https://console.aws.amazon.com/securityhub/.
2. On the top right of the console, select the target Region.
3. If the Security Hub > Summary page is displayed, then Security Hub is set up for the selected region.
4. If presented with "Setup Security Hub" or "Get Started With Security Hub," refer to the remediation steps.
5. Repeat steps 2 to 4 for each region.

From Command Line:
Run the following command to verify the Security Hub status:

```
aws securityhub describe-hub
```

This will list the Security Hub status by region. Check for a 'SubscribedAt' value.
Example output:

```
{
    "HubArn": "<security-hub-arn>",
    "SubscribedAt": "2022-08-19T17:06:42.398Z",
    "AutoEnableControls": true
}
```

An error will be returned if Security Hub is not enabled.
Example error:

```
An error occurred (InvalidAccessException) when calling the DescribeHub
operation: Account <Account ID> is not subscribed to AWS Security Hub
```

Remediation:

To grant the permissions required to enable Security Hub, attach the Security Hub managed policy `AWSSecurityHubFullAccess` to an IAM user, group, or role.
Enabling Security Hub:
From Console:

1. Use the credentials of the IAM identity to sign in to the Security Hub console.
2. When you open the Security Hub console for the first time, choose `Go to Security Hub`.
3. The `Security standards` section on the welcome page lists supported security standards. Check the box for a standard to enable it.
4. Choose `Enable Security Hub`.

From Command Line:

1. Run the `enable-security-hub` command, including `--enable-default-standards` to enable the default standards:

```
aws securityhub enable-security-hub --enable-default-standards
```

2. To enable Security Hub without the default standards, include `--no-enable-default-standards`:

```
aws securityhub enable-security-hub --no-enable-default-standards
```

References:

1. https://docs.aws.amazon.com/securityhub/latest/userguide/securityhub-get-started.html
2. https://docs.aws.amazon.com/securityhub/latest/userguide/securityhub-enable.html#securityhub-enable-api
3. https://awscli.amazonaws.com/v2/documentation/api/latest/reference/securityhub/enable-security-hub.html

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	7.1 Establish and Maintain a Vulnerability Management Process Establish and maintain a documented vulnerability management process for enterprise assets. Review and update documentation annually, or when significant enterprise changes occur that could impact this Safeguard.	●	●	●
v7	11.3 Use Automated Tools to Verify Standard Device Configurations and Detect Changes Compare all network device configuration against approved security configurations defined for each network device in use and alert when any deviations are discovered.		●	●

5 Networking

This section contains recommendations for AWS networking configuration.

5.1 Elastic Compute Cloud (EC2)

This section contains recommendations for configuring AWS Elastic Compute Cloud (EC2)

5.1.1 Ensure EBS volume encryption is enabled in all regions (Automated)

Profile Applicability:

- Level 1

Description:

Elastic Compute Cloud (EC2) supports encryption at rest when using the Elastic Block Store (EBS) service. While disabled by default, forcing encryption at EBS volume creation is supported.

Rationale:

Encrypting data at rest reduces the likelihood of unintentional exposure and can nullify the impact of disclosure if the encryption remains unbroken.

Impact:

Losing access to or removing the KMS key used by the EBS volumes will result in the inability to access the volumes.

Audit:

From Console:

1. Login to the AWS Management Console and open the Amazon EC2 console using https://console.aws.amazon.com/ec2/.
2. Under `Settings`, click `EBS encryption`.
3. Verify `Always encrypt new EBS volumes` displays `Enabled`.
4. Repeat for each region in use.

Note: EBS volume encryption is configured per region.
From Command Line:

1. Run the following command:

```
aws --region <region> ec2 get-ebs-encryption-by-default
```

2. Verify that `"EbsEncryptionByDefault": true` is displayed.
3. Repeat for each region in use.

Note: EBS volume encryption is configured per region.

Remediation:

From Console:

1. Login to the AWS Management Console and open the Amazon EC2 console using https://console.aws.amazon.com/ec2/.
2. Under `Account attributes`, click `EBS encryption`.
3. Click `Manage`.
4. Check the `Enable` box.
5. Click `Update EBS encryption`.
6. Repeat for each region in which EBS volume encryption is not enabled by default.

Note: EBS volume encryption is configured per region.
From Command Line:

1. Run the following command:

```
aws --region <region> ec2 enable-ebs-encryption-by-default
```

2. Verify that `"EbsEncryptionByDefault": true` is displayed.
3. Repeat for each region in which EBS volume encryption is not enabled by default.

Note: EBS volume encryption is configured per region.

References:

1. https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EBSEncryption.html
2. https://aws.amazon.com/blogs/aws/new-opt-in-to-default-encryption-for-new-ebs-volumes/

Additional Information:

Default EBS volume encryption only applies to newly created EBS volumes; existing EBS volumes are **not** converted automatically.

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.11 Encrypt Sensitive Data at Rest Encrypt sensitive data at rest on servers, applications, and databases containing sensitive data. Storage-layer encryption, also known as server-side encryption, meets the minimum requirement of this Safeguard. Additional encryption methods may include application-layer encryption, also known as client-side encryption, where access to the data storage device(s) does not permit access to the plain-text data.		●	●
v7	14.8 Encrypt Sensitive Information at Rest Encrypt all sensitive information at rest using a tool that requires a secondary authentication mechanism not integrated into the operating system, in order to access the information.			●

5.1.2 Ensure CIFS access is restricted to trusted networks to prevent unauthorized access (Automated)

Profile Applicability:

- Level 1

Description:

Common Internet File System (CIFS) is a network file-sharing protocol that allows systems to share files over a network. However, unrestricted CIFS access can expose your data to unauthorized users, leading to potential security risks. It is important to restrict CIFS access to only trusted networks and users to prevent unauthorized access and data breaches.

Rationale:

Allowing unrestricted CIFS access can lead to significant security vulnerabilities, as it may allow unauthorized users to access sensitive files and data. By restricting CIFS access to known and trusted networks, you can minimize the risk of unauthorized access and protect sensitive data from exposure to potential attackers. Implementing proper network access controls and permissions is essential for maintaining the security and integrity of your file-sharing systems.

Impact:

Restricting CIFS access may require additional configuration and management effort. However, the benefits of enhanced security and reduced risk of unauthorized access to sensitive data far outweigh the potential challenges.

Audit:

From Console:

1. Login to the AWS Management Console.
2. Navigate to the EC2 Dashboard and select the Security Groups section under `Network & Security`.
3. Identify the security groups associated with instances or resources that may be using CIFS.
4. Review the inbound rules of each security group to check for rules that allow unrestricted access on port 445 (the port used by CIFS).
    - Specifically, look for inbound rules that allow access from `0.0.0.0/0` or `::/0` on port 445.
5. Document any instances where unrestricted access is allowed and verify whether it is necessary for the specific use case.

From Command Line:

1. Run the following command to list all security groups and identify those associated with CIFS:

```
aws ec2 describe-security-groups --region <region-name> --query
'SecurityGroups[*].GroupId'
```

2. Check for any inbound rules that allow unrestricted access on port 445 using the following command:

```
aws ec2 describe-security-groups --region <region-name> --group-ids
<security-group-id> --query
'SecurityGroups[*].IpPermissions[?ToPort==`445`].{CIDR:IpRanges[*].Cidr
Ip,Port:ToPort}'
```

    - Look for `0.0.0.0/0` or `::/0` in the output, which indicates unrestricted access.
3. Repeat the audit for other regions and security groups as necessary.

Remediation:

From Console:

1. Login to the AWS Management Console.
2. Navigate to the EC2 Dashboard and select the Security Groups section under `Network & Security`.
3. Identify the security group that allows unrestricted ingress on port 445.
4. Select the security group and click the `Edit Inbound Rules` button.
5. Locate the rule allowing unrestricted access on port 445 (typically listed as `0.0.0.0/0` or `::/0`).
6. Modify the rule to restrict access to specific IP ranges or trusted networks only.
7. Save the changes to the security group.

From Command Line:

1. Run the following command to remove or modify the unrestricted rule for CIFS access:

```
aws ec2 revoke-security-group-ingress --region <region-name> --group-id
<security-group-id> --protocol tcp --port 445 --cidr 0.0.0.0/0
```

    - Optionally, run the `authorise-security-group-ingress` command to create a new rule, specifying a trusted CIDR range instead of `0.0.0.0/0`.
2. Confirm the changes by describing the security group again and ensuring the unrestricted access rule has been removed or appropriately restricted:

```
aws ec2 describe-security-groups --region <region-name> --group-ids
<security-group-id> --query
'SecurityGroups[*].IpPermissions[?ToPort==`445`].{CIDR:IpRanges[*].Cidr
Ip,Port:ToPort}'
```

3. Repeat the remediation for other security groups and regions as necessary.

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	4.5 Implement and Manage a Firewall on End-User Devices Implement and manage a host-based firewall or port-filtering tool on end-user devices, with a default-deny rule that drops all traffic except those services and ports that are explicitly allowed.	●	●	●
v7	9.4 Apply Host-based Firewalls or Port Filtering Apply host-based firewalls or port filtering tools on end systems, with a default-deny rule that drops all traffic except those services and ports that are explicitly allowed.	●	●	●

5.2 Ensure no Network ACLs allow ingress from 0.0.0.0/0 to remote server administration ports (Automated)

Profile Applicability:

- Level 1

Description:

The Network Access Control List (NACL) function provides stateless filtering of ingress and egress network traffic to AWS resources. It is recommended that no NACL allows unrestricted ingress access to remote server administration ports, such as SSH on port `22` and RDP on port `3389`, using either the TCP (6), UDP (17), or ALL (-1) protocols.

Rationale:

Public access to remote server administration ports, such as 22 (when used for SSH, not SFTP) and 3389, increases the attack surface of resources and unnecessarily raises the risk of resource compromise.

Audit:

From Console:
Perform the following steps to determine if the account is configured as prescribed:

1. Login to the AWS VPC Console at https://console.aws.amazon.com/vpc/home.
2. In the left pane, click `Network ACLs`.
3. For each network ACL, perform the following:
    - Select the network ACL.
    - Click the `Inbound Rules` tab.
    - Ensure that no rule exists which has a port range that includes port `22` or `3389`, uses the protocols TCP (6), UDP (17), or ALL (-1), or other remote server administration ports for your environment, has a `Source` of `0.0.0.0/0`, and shows `ALLOW`.

Note: A port value of `ALL` or a port range such as `0-3389` includes port `22`, `3389`, and potentially other remote server administration ports.

Remediation:

From Console:
Perform the following steps to remediate a network ACL:

1. Login to the AWS VPC Console at https://console.aws.amazon.com/vpc/home.
2. In the left pane, click `Network ACLs`.
3. For each network ACL that needs remediation, perform the following:
    - Select the network ACL.
    - Click the `Inbound Rules` tab.

- Click `Edit inbound rules`.
- Either A) update the Source field to a range other than 0.0.0.0/0, or B) click `Delete` to remove the offending inbound rule.
- Click `Save`.

References:

1. https://docs.aws.amazon.com/vpc/latest/userguide/vpc-network-acls.html
2. https://docs.aws.amazon.com/vpc/latest/userguide/VPC_Security.html#VPC_Security_Comparison

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v7	9.2 Ensure Only Approved Ports, Protocols and Services Are Running Ensure that only network ports, protocols, and services listening on a system with validated business needs, are running on each system.		●	●
v7	12.4 Deny Communication over Unauthorized Ports Deny communication over unauthorized TCP or UDP ports or application traffic to ensure that only authorized protocols are allowed to cross the network boundary in or out of the network at each of the organization's network boundaries.	●	●	●

5.3 Ensure no security groups allow ingress from 0.0.0.0/0 to remote server administration ports (Automated)

Profile Applicability:

- Level 1

Description:

Security groups provide stateful filtering of ingress and egress network traffic to AWS resources. It is recommended that no security group allows unrestricted ingress access to remote server administration ports, such as SSH on port `22` and RDP on port `3389`, using either the TCP (6), UDP (17), or ALL (-1) protocols.

Rationale:

Public access to remote server administration ports, such as 22 (when used for SSH, not SFTP) and 3389, increases the attack surface of resources and unnecessarily raises the risk of resource compromise.

Impact:

When updating an existing environment, ensure that administrators have access to remote server administration ports through another mechanism before removing access by deleting the 0.0.0.0/0 inbound rule.

Audit:

Perform the following to determine if the account is configured as prescribed:

1. Login to the AWS VPC Console at https://console.aws.amazon.com/vpc/home.
2. In the left pane, click `Security Groups`.
3. For each security group, perform the following:
    - Select the security group.
    - Click the `Inbound Rules` tab.
    - Ensure that no rule exists which has a port range including port `22` or `3389`, uses the protocols TCP (6), UDP (17), or ALL (-1), or other remote server administration ports for your environment, and has a `Source` of `0.0.0.0/0`.

Note: A port value of `ALL` or a port range such as `0-3389` includes port `22`, `3389`, and potentially other remote server administration ports.

Remediation:

Perform the following to implement the prescribed state:

1. Login to the AWS VPC Console at https://console.aws.amazon.com/vpc/home.

2. In the left pane, click `Security Groups`.
3. For each security group, perform the following:
    - Select the security group.
    - Click the `Inbound Rules` tab.
    - Click the `Edit inbound rules` button.
    - Identify the rules to be edited or removed.
    - Either A) update the Source field to a range other than 0.0.0.0/0, or B) click `Delete` to remove the offending inbound rule.
    - Click `Save rules`.

References:

1. https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-security-groups.html#deleting-security-group-rule

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	4.5 Implement and Manage a Firewall on End-User Devices Implement and manage a host-based firewall or port-filtering tool on end-user devices, with a default-deny rule that drops all traffic except those services and ports that are explicitly allowed.	●	●	●
v7	9.4 Apply Host-based Firewalls or Port Filtering Apply host-based firewalls or port filtering tools on end systems, with a default-deny rule that drops all traffic except those services and ports that are explicitly allowed.	●	●	●

5.4 Ensure no security groups allow ingress from ::/0 to remote server administration ports (Automated)

Profile Applicability:

- Level 1

Description:

Security groups provide stateful filtering of ingress and egress network traffic to AWS resources. It is recommended that no security group allows unrestricted ingress access to remote server administration ports, such as SSH on port `22` and RDP on port `3389`.

Rationale:

Public access to remote server administration ports, such as 22 (when used for SSH, not SFTP) and 3389, increases attack surface of resources and unnecessarily raises the risk of resource compromise.

Impact:

When updating an existing environment, ensure that administrators have access to remote server administration ports through another mechanism before removing access by deleting the ::/0 inbound rule.

Audit:

Perform the following to determine if the account is configured as prescribed:

1. Login to the AWS VPC Console at https://console.aws.amazon.com/vpc/home.
2. In the left pane, click `Security Groups`.
3. For each security group, perform the following:
    - Select the security group.
    - Click the `Inbound Rules` tab.
    - Ensure that no rule exists which has a port range including port `22`, `3389`, or other remote server administration ports for your environment, and has a `Source` of `::/0`.

Note: A port value of `ALL` or a port range such as `0-3389` includes port `22`, `3389`, and potentially other remote server administration ports.

Remediation:

Perform the following to implement the prescribed state:

1. Login to the AWS VPC Console at https://console.aws.amazon.com/vpc/home.
2. In the left pane, click `Security Groups`.
3. For each security group, perform the following:

- Select the security group.
- Click the `Inbound Rules` tab.
- Click the `Edit inbound rules` button.
- Identify the rules to be edited or removed.
- Either A) update the Source field to a range other than ::/0, or B) Click `Delete` to remove the offending inbound rule.
- Click `Save rules`.

References:

1. https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-security-groups.html#deleting-security-group-rule

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	4.5 Implement and Manage a Firewall on End-User Devices Implement and manage a host-based firewall or port-filtering tool on end-user devices, with a default-deny rule that drops all traffic except those services and ports that are explicitly allowed.	●	●	●
v7	9.4 Apply Host-based Firewalls or Port Filtering Apply host-based firewalls or port filtering tools on end systems, with a default-deny rule that drops all traffic except those services and ports that are explicitly allowed.	●	●	●

5.5 Ensure the default security group of every VPC restricts all traffic (Automated)

Profile Applicability:

- Level 2

Description:

A VPC comes with a default security group whose initial settings deny all inbound traffic, allow all outbound traffic, and allow all traffic between instances assigned to the security group. If a security group is not specified when an instance is launched, it is automatically assigned to this default security group. Security groups provide stateful filtering of ingress/egress network traffic to AWS resources. It is recommended that the default security group restrict all traffic, both inbound and outbound.

The default VPC in every region should have its default security group updated to comply with the following:

- No inbound rules.
- No outbound rules.

Any newly created VPCs will automatically contain a default security group that will need remediation to comply with this recommendation.

Note: When implementing this recommendation, VPC flow logging is invaluable in determining the least privilege port access required by systems to work properly, as it can log all packet acceptances and rejections occurring under the current security groups. This dramatically reduces the primary barrier to least privilege engineering by discovering the minimum ports required by systems in the environment. Even if the VPC flow logging recommendation in this benchmark is not adopted as a permanent security measure, it should be used during any period of discovery and engineering for least privileged security groups.

Rationale:

Configuring all VPC default security groups to restrict all traffic will encourage the development of least privilege security groups and promote the mindful placement of AWS resources into security groups, which will, in turn, reduce the exposure of those resources.

Impact:

Implementing this recommendation in an existing VPC that contains operating resources requires extremely careful migration planning, as the default security groups are likely enabling many ports that are unknown. Enabling VPC flow logging (for accepted connections) in an existing environment that is known to be breach-free will reveal the current pattern of ports being used for each instance to communicate successfully. The migration process should include:

- Analyzing VPC flow logs to understand current traffic patterns.
- Creating least privilege security groups based on the analyzed data.
- Testing the new security group rules in a staging environment before applying them to production.

Audit:

Perform the following to determine if the account is configured as prescribed:
Security Group State

1. Login to the AWS VPC Console at https://console.aws.amazon.com/vpc/home.
2. Repeat the following steps for all VPCs, including the default VPC in each AWS region:
3. In the left pane, click `Security Groups`.
4. For each default security group, perform the following:
    - Select the `default` security group.
    - Click the `Inbound Rules` tab and ensure no rules exist.
    - Click the `Outbound Rules` tab and ensure no rules exist.

Security Group Members

1. Login to the AWS VPC Console at https://console.aws.amazon.com/vpc/home.
2. Repeat the following steps for all default groups in all VPCs, including the default VPC in each AWS region:
3. In the left pane, click `Security Groups`.
4. Copy the ID of the default security group.
5. Change to the EC2 Management Console at https://console.aws.amazon.com/ec2/v2/home.
6. In the filter column type `Security Group ID : <security-group-id-from-step-4>`.

Remediation:

Perform the following to implement the prescribed state:
Security Group Members

1. Identify AWS resources that exist within the default security group.
2. Create a set of least-privilege security groups for those resources.

3. Place the resources in those security groups, removing the resources noted in step 1 from the default security group.

Security Group State

1. Login to the AWS VPC Console at https://console.aws.amazon.com/vpc/home.
2. Repeat the following steps for all VPCs, including the default VPC in each AWS region:
3. In the left pane, click `Security Groups`.
4. For each default security group, perform the following:
    - Select the `default` security group.
    - Click the `Inbound Rules` tab.
    - Remove any inbound rules.
    - Click the `Outbound Rules` tab.
    - Remove any Outbound rules.

Recommended
IAM groups allow you to edit the "name" field. After remediating default group rules for all VPCs in all regions, edit this field to add text similar to "DO NOT USE. DO NOT ADD RULES."

References:

1. CCE-79201-0
2. https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-network-security.html
3. https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-security-groups.html#default-security-group

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	3.3 Configure Data Access Control Lists Configure data access control lists based on a user's need to know. Apply data access control lists, also known as access permissions, to local and remote file systems, databases, and applications.	●	●	●
v8	4.5 Implement and Manage a Firewall on End-User Devices Implement and manage a host-based firewall or port-filtering tool on end-user devices, with a default-deny rule that drops all traffic except those services and ports that are explicitly allowed.	●	●	●

Controls Version	Control	IG 1	IG 2	IG 3
v7	9.4 Apply Host-based Firewalls or Port Filtering Apply host-based firewalls or port filtering tools on end systems, with a default-deny rule that drops all traffic except those services and ports that are explicitly allowed.	●	●	●
v7	14.6 Protect Information through Access Control Lists Protect all information stored on systems with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities.	●	●	●

5.6 Ensure routing tables for VPC peering are "least access" (Manual)

Profile Applicability:

- Level 2

Description:

Once a VPC peering connection is established, routing tables must be updated to enable any connections between the peered VPCs. These routes can be as specific as desired, even allowing for the peering of a VPC to only a single host on the other side of the connection.

Rationale:

Being highly selective in peering routing tables is a very effective way to minimize the impact of a breach, as resources outside of these routes are inaccessible to the peered VPC.

Audit:

Review the routing tables of peered VPCs to determine whether they route all subnets of each VPC and whether this is necessary to accomplish the intended purposes of peering the VPCs.
From Command Line:

1. List all the route tables from a VPC and check if the "GatewayId" is pointing to a `<peering-connection-id>` (e.g., pcx-1a2b3c4d) and if the "DestinationCidrBlock" is as specific as desired:

```
aws ec2 describe-route-tables --filter "Name=vpc-id,Values=<vpc-id>" --query
"RouteTables[*].{RouteTableId:RouteTableId, VpcId:VpcId, Routes:Routes,
AssociatedSubnets:Associations[*].SubnetId}"
```

Remediation:

Remove and add route table entries to ensure that the least number of subnets or hosts required to accomplish the purpose of peering are routable.
From Command Line:

1. For each `<route-table-id>` that contains routes that are non-compliant with your routing policy (granting more access than desired), delete the non-compliant route:

```
aws ec2 delete-route --route-table-id <route-table-id> --destination-cidr-
block <non-compliant-destination-cidr>
```

2. Create a new compliant route:

```
aws ec2 create-route --route-table-id <route-table-id> --destination-cidr-
block <compliant-destination-cidr> --vpc-peering-connection-id <peering-
connection-id>
```

References:

1. https://docs.aws.amazon.com/AmazonVPC/latest/PeeringGuide/peering-configurations-partial-access.html
2. https://awscli.amazonaws.com/v2/documentation/api/latest/reference/ec2/create-vpc-peering-connection.html

Additional Information:

If an organization has an AWS Transit Gateway implemented in its VPC architecture, it should look to apply the recommendation above for a "least access" routing architecture at the AWS Transit Gateway level, in combination with what must be implemented at the standard VPC route table. More specifically, to route traffic between two or more VPCs via a Transit Gateway, VPCs must have an attachment to a Transit Gateway route table as well as a route. Therefore, to avoid routing traffic between VPCs, an attachment to the Transit Gateway route table should only be added where there is an intention to route traffic between the VPCs. As Transit Gateways are capable of hosting multiple route tables, it is possible to group VPCs by attaching them to a common route table.

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	13.4 Perform Traffic Filtering Between Network Segments Perform traffic filtering between network segments, where appropriate.		●	●
v7	9.2 Ensure Only Approved Ports, Protocols and Services Are Running Ensure that only network ports, protocols, and services listening on a system with validated business needs, are running on each system.		●	●

5.7 Ensure that the EC2 Metadata Service only allows IMDSv2 (Automated)

Profile Applicability:

- Level 1

Description:

When enabling the Metadata Service on AWS EC2 instances, users have the option of using either Instance Metadata Service Version 1 (IMDSv1; a request/response method) or Instance Metadata Service Version 2 (IMDSv2; a session-oriented method).

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, such as host name, events, and security groups.

When enabling the Metadata Service on AWS EC2 instances, users have the option of using either Instance Metadata Service Version 1 (IMDSv1; a request/response method) or Instance Metadata Service Version 2 (IMDSv2; a session-oriented method). With IMDSv2, every request is now protected by session authentication. A session begins and ends a series of requests that software running on an EC2 instance uses to access the locally stored EC2 instance metadata and credentials.

Allowing Version 1 of the service may open EC2 instances to Server-Side Request Forgery (SSRF) attacks, so Amazon recommends utilizing Version 2 for better instance security.

Audit:

From Console:

1. Sign in to the AWS Management Console and navigate to the EC2 dashboard at https://console.aws.amazon.com/ec2/.
2. In the left navigation panel, under the `INSTANCES` section, choose `Instances`.
3. Select the EC2 instance that you want to examine.
4. Check the `IMDSv2` status, and ensure that it is set to `Required`.

From Command Line:

1. Run the `describe-instances` command using appropriate filters to list the IDs of all existing EC2 instances currently available in the selected region:

```
aws ec2 describe-instances --region <region-name> --output table --
query "Reservations[*].Instances[*].InstanceId"
```

2. The command output should return a table with the requested instance IDs.
3. Run the `describe-instances` command using the instance ID returned in the previous step and apply custom filtering to determine whether the selected instance is using IMDSv2:

```
aws ec2 describe-instances --region <region-name> --instance-ids
<instance-id> --query "Reservations[*].Instances[*].MetadataOptions" --
output table
```

4. Ensure that for all EC2 instances, `HttpTokens` is set to `required` and `State` is set to `applied`.
5. Repeat steps 3 and 4 to verify the other EC2 instances provisioned within the current region.
6. Repeat steps 1–5 to perform the audit process for other AWS regions.

Remediation:

From Console:

1. Sign in to the AWS Management Console and navigate to the EC2 dashboard at https://console.aws.amazon.com/ec2/.
2. In the left navigation panel, under the `INSTANCES` section, choose `Instances`.
3. Select the EC2 instance that you want to examine.
4. Choose `Actions > Instance Settings > Modify instance metadata options`.
5. Set `Instance metadata service` to `Enable`.
6. Set `IMDSv2` to `Required`.
7. Repeat steps 1-6 to perform the remediation process for other EC2 instances in all applicable AWS region(s).

From Command Line:

1. Run the `describe-instances` command, applying the appropriate filters to list the IDs of all existing EC2 instances currently available in the selected region:

```
aws ec2 describe-instances --region <region-name> --output table --
query "Reservations[*].Instances[*].InstanceId"
```

2. The command output should return a table with the requested instance IDs.
3. Run the `modify-instance-metadata-options` command with an instance ID obtained from the previous step to update the Instance Metadata Version:

```
aws ec2 modify-instance-metadata-options --instance-id <instance-id> --
http-tokens required --region <region-name>
```

4. Repeat steps 1-3 to perform the remediation process for other EC2 instances in the same AWS region.
5. Change the region by updating `--region` and repeat the process for other regions.

References:

1. https://aws.amazon.com/blogs/security/defense-in-depth-open-firewalls-reverse-proxies-ssrf-vulnerabilities-ec2-instance-metadata-service/
2. https://docs.aws.amazon.com/cli/latest/reference/ec2/describe-instances.html

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	4.8 Uninstall or Disable Unnecessary Services on Enterprise Assets and Software Uninstall or disable unnecessary services on enterprise assets and software, such as an unused file sharing service, web application module, or service function.		●	●
v7	5.2 Maintain Secure Images Maintain secure images or templates for all systems in the enterprise based on the organization's approved configuration standards. Any new system deployment or existing system that becomes compromised should be imaged using one of those images or templates.		●	●

Appendix: Summary Table

CIS Benchmark Recommendation		Set Correctly	
		Yes	No
1	**Identity and Access Management**		
1.1	Maintain current contact details (Manual)	☐	☐
1.2	Ensure security contact information is registered (Manual)	☐	☐
1.3	Ensure no 'root' user account access key exists (Automated)	☐	☐
1.4	Ensure MFA is enabled for the 'root' user account (Automated)	☐	☐
1.5	Ensure hardware MFA is enabled for the 'root' user account (Manual)	☐	☐
1.6	Eliminate use of the 'root' user for administrative and daily tasks (Manual)	☐	☐
1.7	Ensure IAM password policy requires minimum length of 14 or greater (Automated)	☐	☐
1.8	Ensure IAM password policy prevents password reuse (Automated)	☐	☐
1.9	Ensure multi-factor authentication (MFA) is enabled for all IAM users that have a console password (Automated)	☐	☐
1.10	Do not create access keys during initial setup for IAM users with a console password (Manual)	☐	☐
1.11	Ensure credentials unused for 45 days or more are disabled (Automated)	☐	☐
1.12	Ensure there is only one active access key for any single IAM user (Automated)	☐	☐

CIS Benchmark Recommendation		Set Correctly	
		Yes	No
1.13	Ensure access keys are rotated every 90 days or less (Automated)	☐	☐
1.14	Ensure IAM users receive permissions only through groups (Automated)	☐	☐
1.15	Ensure IAM policies that allow full "*:*" administrative privileges are not attached (Automated)	☐	☐
1.16	Ensure a support role has been created to manage incidents with AWS Support (Automated)	☐	☐
1.17	Ensure IAM instance roles are used for AWS resource access from instances (Automated)	☐	☐
1.18	Ensure that all expired SSL/TLS certificates stored in AWS IAM are removed (Automated)	☐	☐
1.19	Ensure that IAM External Access Analyzer is enabled for all regions (Automated)	☐	☐
1.20	Ensure IAM users are managed centrally via identity federation or AWS Organizations for multi-account environments (Manual)	☐	☐
1.21	Ensure access to AWSCloudShellFullAccess is restricted (Manual)	☐	☐
2	**Storage**		
2.1	**Simple Storage Service (S3)**		
2.1.1	Ensure S3 Bucket Policy is set to deny HTTP requests (Automated)	☐	☐
2.1.2	Ensure MFA Delete is enabled on S3 buckets (Manual)	☐	☐
2.1.3	Ensure all data in Amazon S3 has been discovered, classified, and secured when necessary (Manual)	☐	☐
2.1.4	Ensure that S3 is configured with 'Block Public Access' enabled (Automated)	☐	☐

CIS Benchmark Recommendation		Set Correctly	
		Yes	No
2.2	**Relational Database Service (RDS)**		
2.2.1	Ensure that encryption-at-rest is enabled for RDS instances (Automated)	☐	☐
2.2.2	Ensure the Auto Minor Version Upgrade feature is enabled for RDS instances (Automated)	☐	☐
2.2.3	Ensure that RDS instances are not publicly accessible (Automated)	☐	☐
2.2.4	Ensure Multi-AZ deployments are used for enhanced availability in Amazon RDS (Manual)	☐	☐
2.3	**Elastic File System (EFS)**		
2.3.1	Ensure that encryption is enabled for EFS file systems (Automated)	☐	☐
3	**Logging**		
3.1	Ensure CloudTrail is enabled in all regions (Manual)	☐	☐
3.2	Ensure CloudTrail log file validation is enabled (Automated)	☐	☐
3.3	Ensure AWS Config is enabled in all regions (Automated)	☐	☐
3.4	Ensure that server access logging is enabled on the CloudTrail S3 bucket (Manual)	☐	☐
3.5	Ensure CloudTrail logs are encrypted at rest using KMS CMKs (Automated)	☐	☐
3.6	Ensure rotation for customer-created symmetric CMKs is enabled (Automated)	☐	☐
3.7	Ensure VPC flow logging is enabled in all VPCs (Automated)	☐	☐

CIS Benchmark Recommendation		Set Correctly	
		Yes	No
3.8	Ensure that object-level logging for write events is enabled for S3 buckets (Automated)	☐	☐
3.9	Ensure that object-level logging for read events is enabled for S3 buckets (Automated)	☐	☐
4	**Monitoring**		
4.1	Ensure unauthorized API calls are monitored (Manual)	☐	☐
4.2	Ensure management console sign-in without MFA is monitored (Manual)	☐	☐
4.3	Ensure usage of the 'root' account is monitored (Manual)	☐	☐
4.4	Ensure IAM policy changes are monitored (Manual)	☐	☐
4.5	Ensure CloudTrail configuration changes are monitored (Manual)	☐	☐
4.6	Ensure AWS Management Console authentication failures are monitored (Manual)	☐	☐
4.7	Ensure disabling or scheduled deletion of customer created CMKs is monitored (Manual)	☐	☐
4.8	Ensure S3 bucket policy changes are monitored (Manual)	☐	☐
4.9	Ensure AWS Config configuration changes are monitored (Manual)	☐	☐
4.10	Ensure security group changes are monitored (Manual)	☐	☐
4.11	Ensure Network Access Control List (NACL) changes are monitored (Manual)	☐	☐
4.12	Ensure changes to network gateways are monitored (Manual)	☐	☐
4.13	Ensure route table changes are monitored (Manual)	☐	☐

CIS Benchmark Recommendation		Set Correctly	
		Yes	No
4.14	Ensure VPC changes are monitored (Manual)	☐	☐
4.15	Ensure AWS Organizations changes are monitored (Manual)	☐	☐
4.16	Ensure AWS Security Hub is enabled (Automated)	☐	☐
5	**Networking**		
5.1	**Elastic Compute Cloud (EC2)**		
5.1.1	Ensure EBS volume encryption is enabled in all regions (Automated)	☐	☐
5.1.2	Ensure CIFS access is restricted to trusted networks to prevent unauthorized access (Automated)	☐	☐
5.2	Ensure no Network ACLs allow ingress from 0.0.0.0/0 to remote server administration ports (Automated)	☐	☐
5.3	Ensure no security groups allow ingress from 0.0.0.0/0 to remote server administration ports (Automated)	☐	☐
5.4	Ensure no security groups allow ingress from ::/0 to remote server administration ports (Automated)	☐	☐
5.5	Ensure the default security group of every VPC restricts all traffic (Automated)	☐	☐
5.6	Ensure routing tables for VPC peering are "least access" (Manual)	☐	☐
5.7	Ensure that the EC2 Metadata Service only allows IMDSv2 (Automated)	☐	☐

Appendix: CIS Controls v7 IG 1 Mapped Recommendations

Recommendation		Set Correctly	
		Yes	No
1.1	Maintain current contact details	☐	☐
1.3	Ensure no 'root' user account access key exists	☐	☐
1.6	Eliminate use of the 'root' user for administrative and daily tasks	☐	☐
1.11	Ensure credentials unused for 45 days or more are disabled	☐	☐
1.18	Ensure that all expired SSL/TLS certificates stored in AWS IAM are removed	☐	☐
1.19	Ensure that IAM External Access Analyzer is enabled for all regions	☐	☐
2.1.2	Ensure MFA Delete is enabled on S3 buckets	☐	☐
2.1.3	Ensure all data in Amazon S3 has been discovered, classified, and secured when necessary	☐	☐
2.1.4	Ensure that S3 is configured with 'Block Public Access' enabled	☐	☐
2.2.2	Ensure the Auto Minor Version Upgrade feature is enabled for RDS instances	☐	☐
2.2.3	Ensure that RDS instances are not publicly accessible	☐	☐
3.1	Ensure CloudTrail is enabled in all regions	☐	☐
3.3	Ensure AWS Config is enabled in all regions	☐	☐
3.4	Ensure that server access logging is enabled on the CloudTrail S3 bucket	☐	☐
3.7	Ensure VPC flow logging is enabled in all VPCs	☐	☐
4.8	Ensure S3 bucket policy changes are monitored	☐	☐
4.9	Ensure AWS Config configuration changes are monitored	☐	☐
4.10	Ensure security group changes are monitored	☐	☐
4.12	Ensure changes to network gateways are monitored	☐	☐
4.13	Ensure route table changes are monitored	☐	☐
4.15	Ensure AWS Organizations changes are monitored	☐	☐

Recommendation		Set Correctly	
		Yes	No
5.1.2	Ensure CIFS access is restricted to trusted networks to prevent unauthorized access	☐	☐
5.2	Ensure no Network ACLs allow ingress from 0.0.0.0/0 to remote server administration ports	☐	☐
5.3	Ensure no security groups allow ingress from 0.0.0.0/0 to remote server administration ports	☐	☐
5.4	Ensure no security groups allow ingress from ::/0 to remote server administration ports	☐	☐
5.5	Ensure the default security group of every VPC restricts all traffic	☐	☐

Appendix: CIS Controls v7 IG 2 Mapped Recommendations

Recommendation		Set Correctly	
		Yes	No
1.1	Maintain current contact details	☐	☐
1.2	Ensure security contact information is registered	☐	☐
1.3	Ensure no 'root' user account access key exists	☐	☐
1.4	Ensure MFA is enabled for the 'root' user account	☐	☐
1.5	Ensure hardware MFA is enabled for the 'root' user account	☐	☐
1.6	Eliminate use of the 'root' user for administrative and daily tasks	☐	☐
1.7	Ensure IAM password policy requires minimum length of 14 or greater	☐	☐
1.8	Ensure IAM password policy prevents password reuse	☐	☐
1.9	Ensure multi-factor authentication (MFA) is enabled for all IAM users that have a console password	☐	☐
1.10	Do not create access keys during initial setup for IAM users with a console password	☐	☐
1.11	Ensure credentials unused for 45 days or more are disabled	☐	☐
1.12	Ensure there is only one active access key for any single IAM user	☐	☐
1.13	Ensure access keys are rotated every 90 days or less	☐	☐
1.14	Ensure IAM users receive permissions only through groups	☐	☐
1.15	Ensure IAM policies that allow full "*:*" administrative privileges are not attached	☐	☐
1.16	Ensure a support role has been created to manage incidents with AWS Support	☐	☐
1.17	Ensure IAM instance roles are used for AWS resource access from instances	☐	☐
1.18	Ensure that all expired SSL/TLS certificates stored in AWS IAM are removed	☐	☐

Recommendation		Set Correctly	
		Yes	No
1.19	Ensure that IAM External Access Analyzer is enabled for all regions	☐	☐
1.20	Ensure IAM users are managed centrally via identity federation or AWS Organizations for multi-account environments	☐	☐
1.21	Ensure access to AWSCloudShellFullAccess is restricted	☐	☐
2.1.1	Ensure S3 Bucket Policy is set to deny HTTP requests	☐	☐
2.1.2	Ensure MFA Delete is enabled on S3 buckets	☐	☐
2.1.3	Ensure all data in Amazon S3 has been discovered, classified, and secured when necessary	☐	☐
2.1.4	Ensure that S3 is configured with 'Block Public Access' enabled	☐	☐
2.2.2	Ensure the Auto Minor Version Upgrade feature is enabled for RDS instances	☐	☐
2.2.3	Ensure that RDS instances are not publicly accessible	☐	☐
3.1	Ensure CloudTrail is enabled in all regions	☐	☐
3.2	Ensure CloudTrail log file validation is enabled	☐	☐
3.3	Ensure AWS Config is enabled in all regions	☐	☐
3.4	Ensure that server access logging is enabled on the CloudTrail S3 bucket	☐	☐
3.7	Ensure VPC flow logging is enabled in all VPCs	☐	☐
3.8	Ensure that object-level logging for write events is enabled for S3 buckets	☐	☐
3.9	Ensure that object-level logging for read events is enabled for S3 buckets	☐	☐
4.1	Ensure unauthorized API calls are monitored	☐	☐
4.2	Ensure management console sign-in without MFA is monitored	☐	☐
4.3	Ensure usage of the 'root' account is monitored	☐	☐
4.4	Ensure IAM policy changes are monitored	☐	☐
4.5	Ensure CloudTrail configuration changes are monitored	☐	☐
4.6	Ensure AWS Management Console authentication failures are monitored	☐	☐

Recommendation		Set Correctly	
		Yes	No
4.7	Ensure disabling or scheduled deletion of customer created CMKs is monitored	☐	☐
4.8	Ensure S3 bucket policy changes are monitored	☐	☐
4.9	Ensure AWS Config configuration changes are monitored	☐	☐
4.10	Ensure security group changes are monitored	☐	☐
4.11	Ensure Network Access Control List (NACL) changes are monitored	☐	☐
4.12	Ensure changes to network gateways are monitored	☐	☐
4.13	Ensure route table changes are monitored	☐	☐
4.14	Ensure VPC changes are monitored	☐	☐
4.15	Ensure AWS Organizations changes are monitored	☐	☐
4.16	Ensure AWS Security Hub is enabled	☐	☐
5.1.2	Ensure CIFS access is restricted to trusted networks to prevent unauthorized access	☐	☐
5.2	Ensure no Network ACLs allow ingress from 0.0.0.0/0 to remote server administration ports	☐	☐
5.3	Ensure no security groups allow ingress from 0.0.0.0/0 to remote server administration ports	☐	☐
5.4	Ensure no security groups allow ingress from ::/0 to remote server administration ports	☐	☐
5.5	Ensure the default security group of every VPC restricts all traffic	☐	☐
5.6	Ensure routing tables for VPC peering are "least access"	☐	☐
5.7	Ensure that the EC2 Metadata Service only allows IMDSv2	☐	☐

Appendix: CIS Controls v7 IG 3 Mapped Recommendations

Recommendation		Set Correctly	
		Yes	No
1.1	Maintain current contact details	☐	☐
1.2	Ensure security contact information is registered	☐	☐
1.3	Ensure no 'root' user account access key exists	☐	☐
1.4	Ensure MFA is enabled for the 'root' user account	☐	☐
1.5	Ensure hardware MFA is enabled for the 'root' user account	☐	☐
1.6	Eliminate use of the 'root' user for administrative and daily tasks	☐	☐
1.7	Ensure IAM password policy requires minimum length of 14 or greater	☐	☐
1.8	Ensure IAM password policy prevents password reuse	☐	☐
1.9	Ensure multi-factor authentication (MFA) is enabled for all IAM users that have a console password	☐	☐
1.10	Do not create access keys during initial setup for IAM users with a console password	☐	☐
1.11	Ensure credentials unused for 45 days or more are disabled	☐	☐
1.12	Ensure there is only one active access key for any single IAM user	☐	☐
1.13	Ensure access keys are rotated every 90 days or less	☐	☐
1.14	Ensure IAM users receive permissions only through groups	☐	☐
1.15	Ensure IAM policies that allow full "*:*" administrative privileges are not attached	☐	☐
1.16	Ensure a support role has been created to manage incidents with AWS Support	☐	☐
1.17	Ensure IAM instance roles are used for AWS resource access from instances	☐	☐
1.18	Ensure that all expired SSL/TLS certificates stored in AWS IAM are removed	☐	☐

Recommendation		Set Correctly	
		Yes	No
1.19	Ensure that IAM External Access Analyzer is enabled for all regions	☐	☐
1.20	Ensure IAM users are managed centrally via identity federation or AWS Organizations for multi-account environments	☐	☐
1.21	Ensure access to AWSCloudShellFullAccess is restricted	☐	☐
2.1.1	Ensure S3 Bucket Policy is set to deny HTTP requests	☐	☐
2.1.2	Ensure MFA Delete is enabled on S3 buckets	☐	☐
2.1.3	Ensure all data in Amazon S3 has been discovered, classified, and secured when necessary	☐	☐
2.1.4	Ensure that S3 is configured with 'Block Public Access' enabled	☐	☐
2.2.1	Ensure that encryption-at-rest is enabled for RDS instances	☐	☐
2.2.2	Ensure the Auto Minor Version Upgrade feature is enabled for RDS instances	☐	☐
2.2.3	Ensure that RDS instances are not publicly accessible	☐	☐
2.2.4	Ensure Multi-AZ deployments are used for enhanced availability in Amazon RDS	☐	☐
2.3.1	Ensure that encryption is enabled for EFS file systems	☐	☐
3.1	Ensure CloudTrail is enabled in all regions	☐	☐
3.2	Ensure CloudTrail log file validation is enabled	☐	☐
3.3	Ensure AWS Config is enabled in all regions	☐	☐
3.4	Ensure that server access logging is enabled on the CloudTrail S3 bucket	☐	☐
3.5	Ensure CloudTrail logs are encrypted at rest using KMS CMKs	☐	☐
3.6	Ensure rotation for customer-created symmetric CMKs is enabled	☐	☐
3.7	Ensure VPC flow logging is enabled in all VPCs	☐	☐
3.8	Ensure that object-level logging for write events is enabled for S3 buckets	☐	☐
3.9	Ensure that object-level logging for read events is enabled for S3 buckets	☐	☐
4.1	Ensure unauthorized API calls are monitored	☐	☐

Recommendation		Set Correctly	
		Yes	No
4.2	Ensure management console sign-in without MFA is monitored	☐	☐
4.3	Ensure usage of the 'root' account is monitored	☐	☐
4.4	Ensure IAM policy changes are monitored	☐	☐
4.5	Ensure CloudTrail configuration changes are monitored	☐	☐
4.6	Ensure AWS Management Console authentication failures are monitored	☐	☐
4.7	Ensure disabling or scheduled deletion of customer created CMKs is monitored	☐	☐
4.8	Ensure S3 bucket policy changes are monitored	☐	☐
4.9	Ensure AWS Config configuration changes are monitored	☐	☐
4.10	Ensure security group changes are monitored	☐	☐
4.11	Ensure Network Access Control List (NACL) changes are monitored	☐	☐
4.12	Ensure changes to network gateways are monitored	☐	☐
4.13	Ensure route table changes are monitored	☐	☐
4.14	Ensure VPC changes are monitored	☐	☐
4.15	Ensure AWS Organizations changes are monitored	☐	☐
4.16	Ensure AWS Security Hub is enabled	☐	☐
5.1.1	Ensure EBS volume encryption is enabled in all regions	☐	☐
5.1.2	Ensure CIFS access is restricted to trusted networks to prevent unauthorized access	☐	☐
5.2	Ensure no Network ACLs allow ingress from 0.0.0.0/0 to remote server administration ports	☐	☐
5.3	Ensure no security groups allow ingress from 0.0.0.0/0 to remote server administration ports	☐	☐
5.4	Ensure no security groups allow ingress from ::/0 to remote server administration ports	☐	☐
5.5	Ensure the default security group of every VPC restricts all traffic	☐	☐
5.6	Ensure routing tables for VPC peering are "least access"	☐	☐
5.7	Ensure that the EC2 Metadata Service only allows IMDSv2	☐	☐

Appendix: CIS Controls v7 Unmapped Recommendations

Recommendation		Set Correctly	
		Yes	No
	No unmapped recommendations to CIS Controls v7	☐	☐

Appendix: CIS Controls v8 IG 1 Mapped Recommendations

Recommendation		Set Correctly	
		Yes	No
1.1	Maintain current contact details	☐	☐
1.2	Ensure security contact information is registered	☐	☐
1.3	Ensure no 'root' user account access key exists	☐	☐
1.4	Ensure MFA is enabled for the 'root' user account	☐	☐
1.5	Ensure hardware MFA is enabled for the 'root' user account	☐	☐
1.6	Eliminate use of the 'root' user for administrative and daily tasks	☐	☐
1.7	Ensure IAM password policy requires minimum length of 14 or greater	☐	☐
1.8	Ensure IAM password policy prevents password reuse	☐	☐
1.9	Ensure multi-factor authentication (MFA) is enabled for all IAM users that have a console password	☐	☐
1.10	Do not create access keys during initial setup for IAM users with a console password	☐	☐
1.11	Ensure credentials unused for 45 days or more are disabled	☐	☐
1.12	Ensure there is only one active access key for any single IAM user	☐	☐
1.13	Ensure access keys are rotated every 90 days or less	☐	☐
1.15	Ensure IAM policies that allow full "*:*" administrative privileges are not attached	☐	☐
1.16	Ensure a support role has been created to manage incidents with AWS Support	☐	☐
1.18	Ensure that all expired SSL/TLS certificates stored in AWS IAM are removed	☐	☐
1.19	Ensure that IAM External Access Analyzer is enabled for all regions	☐	☐
1.21	Ensure access to AWSCloudShellFullAccess is restricted	☐	☐
2.1.2	Ensure MFA Delete is enabled on S3 buckets	☐	☐

Recommendation		Set Correctly	
		Yes	No
2.1.3	Ensure all data in Amazon S3 has been discovered, classified, and secured when necessary	☐	☐
2.1.4	Ensure that S3 is configured with 'Block Public Access' enabled	☐	☐
2.2.2	Ensure the Auto Minor Version Upgrade feature is enabled for RDS instances	☐	☐
2.2.3	Ensure that RDS instances are not publicly accessible	☐	☐
3.3	Ensure AWS Config is enabled in all regions	☐	☐
3.4	Ensure that server access logging is enabled on the CloudTrail S3 bucket	☐	☐
3.5	Ensure CloudTrail logs are encrypted at rest using KMS CMKs	☐	☐
3.7	Ensure VPC flow logging is enabled in all VPCs	☐	☐
4.10	Ensure security group changes are monitored	☐	☐
4.16	Ensure AWS Security Hub is enabled	☐	☐
5.1.2	Ensure CIFS access is restricted to trusted networks to prevent unauthorized access	☐	☐
5.3	Ensure no security groups allow ingress from 0.0.0.0/0 to remote server administration ports	☐	☐
5.4	Ensure no security groups allow ingress from ::/0 to remote server administration ports	☐	☐
5.5	Ensure the default security group of every VPC restricts all traffic	☐	☐

Appendix: CIS Controls v8 IG 2 Mapped Recommendations

Recommendation		Set Correctly	
		Yes	No
1.1	Maintain current contact details	☐	☐
1.2	Ensure security contact information is registered	☐	☐
1.3	Ensure no 'root' user account access key exists	☐	☐
1.4	Ensure MFA is enabled for the 'root' user account	☐	☐
1.5	Ensure hardware MFA is enabled for the 'root' user account	☐	☐
1.6	Eliminate use of the 'root' user for administrative and daily tasks	☐	☐
1.7	Ensure IAM password policy requires minimum length of 14 or greater	☐	☐
1.8	Ensure IAM password policy prevents password reuse	☐	☐
1.9	Ensure multi-factor authentication (MFA) is enabled for all IAM users that have a console password	☐	☐
1.10	Do not create access keys during initial setup for IAM users with a console password	☐	☐
1.11	Ensure credentials unused for 45 days or more are disabled	☐	☐
1.12	Ensure there is only one active access key for any single IAM user	☐	☐
1.13	Ensure access keys are rotated every 90 days or less	☐	☐
1.15	Ensure IAM policies that allow full "*:*" administrative privileges are not attached	☐	☐
1.16	Ensure a support role has been created to manage incidents with AWS Support	☐	☐
1.18	Ensure that all expired SSL/TLS certificates stored in AWS IAM are removed	☐	☐
1.19	Ensure that IAM External Access Analyzer is enabled for all regions	☐	☐

Recommendation		Set Correctly	
		Yes	No
1.20	Ensure IAM users are managed centrally via identity federation or AWS Organizations for multi-account environments	☐	☐
1.21	Ensure access to AWSCloudShellFullAccess is restricted	☐	☐
2.1.1	Ensure S3 Bucket Policy is set to deny HTTP requests	☐	☐
2.1.2	Ensure MFA Delete is enabled on S3 buckets	☐	☐
2.1.3	Ensure all data in Amazon S3 has been discovered, classified, and secured when necessary	☐	☐
2.1.4	Ensure that S3 is configured with 'Block Public Access' enabled	☐	☐
2.2.1	Ensure that encryption-at-rest is enabled for RDS instances	☐	☐
2.2.2	Ensure the Auto Minor Version Upgrade feature is enabled for RDS instances	☐	☐
2.2.3	Ensure that RDS instances are not publicly accessible	☐	☐
2.2.4	Ensure Multi-AZ deployments are used for enhanced availability in Amazon RDS	☐	☐
2.3.1	Ensure that encryption is enabled for EFS file systems	☐	☐
3.1	Ensure CloudTrail is enabled in all regions	☐	☐
3.2	Ensure CloudTrail log file validation is enabled	☐	☐
3.3	Ensure AWS Config is enabled in all regions	☐	☐
3.4	Ensure that server access logging is enabled on the CloudTrail S3 bucket	☐	☐
3.5	Ensure CloudTrail logs are encrypted at rest using KMS CMKs	☐	☐
3.6	Ensure rotation for customer-created symmetric CMKs is enabled	☐	☐
3.7	Ensure VPC flow logging is enabled in all VPCs	☐	☐
3.8	Ensure that object-level logging for write events is enabled for S3 buckets	☐	☐
3.9	Ensure that object-level logging for read events is enabled for S3 buckets	☐	☐
4.1	Ensure unauthorized API calls are monitored	☐	☐
4.2	Ensure management console sign-in without MFA is monitored	☐	☐

Recommendation		Set Correctly	
		Yes	No
4.3	Ensure usage of the 'root' account is monitored	☐	☐
4.4	Ensure IAM policy changes are monitored	☐	☐
4.5	Ensure CloudTrail configuration changes are monitored	☐	☐
4.6	Ensure AWS Management Console authentication failures are monitored	☐	☐
4.7	Ensure disabling or scheduled deletion of customer created CMKs is monitored	☐	☐
4.8	Ensure S3 bucket policy changes are monitored	☐	☐
4.9	Ensure AWS Config configuration changes are monitored	☐	☐
4.10	Ensure security group changes are monitored	☐	☐
4.11	Ensure Network Access Control List (NACL) changes are monitored	☐	☐
4.12	Ensure changes to network gateways are monitored	☐	☐
4.13	Ensure route table changes are monitored	☐	☐
4.14	Ensure VPC changes are monitored	☐	☐
4.15	Ensure AWS Organizations changes are monitored	☐	☐
4.16	Ensure AWS Security Hub is enabled	☐	☐
5.1.1	Ensure EBS volume encryption is enabled in all regions	☐	☐
5.1.2	Ensure CIFS access is restricted to trusted networks to prevent unauthorized access	☐	☐
5.3	Ensure no security groups allow ingress from 0.0.0.0/0 to remote server administration ports	☐	☐
5.4	Ensure no security groups allow ingress from ::/0 to remote server administration ports	☐	☐
5.5	Ensure the default security group of every VPC restricts all traffic	☐	☐
5.6	Ensure routing tables for VPC peering are "least access"	☐	☐
5.7	Ensure that the EC2 Metadata Service only allows IMDSv2	☐	☐

Appendix: CIS Controls v8 IG 3 Mapped Recommendations

Recommendation		Set Correctly	
		Yes	No
1.1	Maintain current contact details	☐	☐
1.2	Ensure security contact information is registered	☐	☐
1.3	Ensure no 'root' user account access key exists	☐	☐
1.4	Ensure MFA is enabled for the 'root' user account	☐	☐
1.5	Ensure hardware MFA is enabled for the 'root' user account	☐	☐
1.6	Eliminate use of the 'root' user for administrative and daily tasks	☐	☐
1.7	Ensure IAM password policy requires minimum length of 14 or greater	☐	☐
1.8	Ensure IAM password policy prevents password reuse	☐	☐
1.9	Ensure multi-factor authentication (MFA) is enabled for all IAM users that have a console password	☐	☐
1.10	Do not create access keys during initial setup for IAM users with a console password	☐	☐
1.11	Ensure credentials unused for 45 days or more are disabled	☐	☐
1.12	Ensure there is only one active access key for any single IAM user	☐	☐
1.13	Ensure access keys are rotated every 90 days or less	☐	☐
1.14	Ensure IAM users receive permissions only through groups	☐	☐
1.15	Ensure IAM policies that allow full "*:*" administrative privileges are not attached	☐	☐
1.16	Ensure a support role has been created to manage incidents with AWS Support	☐	☐
1.17	Ensure IAM instance roles are used for AWS resource access from instances	☐	☐
1.18	Ensure that all expired SSL/TLS certificates stored in AWS IAM are removed	☐	☐

Recommendation		Set Correctly	
		Yes	No
1.19	Ensure that IAM External Access Analyzer is enabled for all regions	☐	☐
1.20	Ensure IAM users are managed centrally via identity federation or AWS Organizations for multi-account environments	☐	☐
1.21	Ensure access to AWSCloudShellFullAccess is restricted	☐	☐
2.1.1	Ensure S3 Bucket Policy is set to deny HTTP requests	☐	☐
2.1.2	Ensure MFA Delete is enabled on S3 buckets	☐	☐
2.1.3	Ensure all data in Amazon S3 has been discovered, classified, and secured when necessary	☐	☐
2.1.4	Ensure that S3 is configured with 'Block Public Access' enabled	☐	☐
2.2.1	Ensure that encryption-at-rest is enabled for RDS instances	☐	☐
2.2.2	Ensure the Auto Minor Version Upgrade feature is enabled for RDS instances	☐	☐
2.2.3	Ensure that RDS instances are not publicly accessible	☐	☐
2.2.4	Ensure Multi-AZ deployments are used for enhanced availability in Amazon RDS	☐	☐
2.3.1	Ensure that encryption is enabled for EFS file systems	☐	☐
3.1	Ensure CloudTrail is enabled in all regions	☐	☐
3.2	Ensure CloudTrail log file validation is enabled	☐	☐
3.3	Ensure AWS Config is enabled in all regions	☐	☐
3.4	Ensure that server access logging is enabled on the CloudTrail S3 bucket	☐	☐
3.5	Ensure CloudTrail logs are encrypted at rest using KMS CMKs	☐	☐
3.6	Ensure rotation for customer-created symmetric CMKs is enabled	☐	☐
3.7	Ensure VPC flow logging is enabled in all VPCs	☐	☐
3.8	Ensure that object-level logging for write events is enabled for S3 buckets	☐	☐
3.9	Ensure that object-level logging for read events is enabled for S3 buckets	☐	☐
4.1	Ensure unauthorized API calls are monitored	☐	☐

Recommendation		Set Correctly	
		Yes	No
4.2	Ensure management console sign-in without MFA is monitored	☐	☐
4.3	Ensure usage of the 'root' account is monitored	☐	☐
4.4	Ensure IAM policy changes are monitored	☐	☐
4.5	Ensure CloudTrail configuration changes are monitored	☐	☐
4.6	Ensure AWS Management Console authentication failures are monitored	☐	☐
4.7	Ensure disabling or scheduled deletion of customer created CMKs is monitored	☐	☐
4.8	Ensure S3 bucket policy changes are monitored	☐	☐
4.9	Ensure AWS Config configuration changes are monitored	☐	☐
4.10	Ensure security group changes are monitored	☐	☐
4.11	Ensure Network Access Control List (NACL) changes are monitored	☐	☐
4.12	Ensure changes to network gateways are monitored	☐	☐
4.13	Ensure route table changes are monitored	☐	☐
4.14	Ensure VPC changes are monitored	☐	☐
4.15	Ensure AWS Organizations changes are monitored	☐	☐
4.16	Ensure AWS Security Hub is enabled	☐	☐
5.1.1	Ensure EBS volume encryption is enabled in all regions	☐	☐
5.1.2	Ensure CIFS access is restricted to trusted networks to prevent unauthorized access	☐	☐
5.3	Ensure no security groups allow ingress from 0.0.0.0/0 to remote server administration ports	☐	☐
5.4	Ensure no security groups allow ingress from ::/0 to remote server administration ports	☐	☐
5.5	Ensure the default security group of every VPC restricts all traffic	☐	☐
5.6	Ensure routing tables for VPC peering are "least access"	☐	☐
5.7	Ensure that the EC2 Metadata Service only allows IMDSv2	☐	☐

Appendix: CIS Controls v8 Unmapped Recommendations

Recommendation		Set Correctly	
		Yes	No
5.2	Ensure no Network ACLs allow ingress from 0.0.0.0/0 to remote server administration ports	☐	☐

Appendix: Change History

Date	Version	Changes for this version
Mar 31, 2025	5.0.0	UPDATE – Ensure CloudTrail is enabled in all regions - Cloudtrail get-trail-status command fails (Ticket 23894)
Mar 31, 2025	5.0.0	UPDATE – Ensure CIFS access is restricted to trusted networks to prevent unauthorized access - Ports(s) should likely be clarified in CLI (Ticket 23924)
Mar 31, 2025	5.0.0	UPDATE – Ensure that server access logging is enabled on the CloudTrail S3 bucket - Cloudtrail s3api get-bucket-logging command fails (Ticket 23895)
Jan 30, 2024	4.0.0	UPDATE - Ensure CloudTrail is enabled in all regions - Audit and remediation update for console and CLI steps (Ticket 20200)
Feb 27, 2024	4.0.0	UPDATE - Ensure no Network ACLs allow ingress from 0.0.0.0/0 to remote server administration ports - Protocol typo in document content. (Ticket 21042)
Aug 13, 2024	4.0.0	UPDATE - Ensure access to AWSCloudShellFullAccess is restricted - Change Assessment Status to Automated (Ticket 20722)
Aug 13, 2024	4.0.0	ADD - Ensure that encryption in transit using SSL/TLS is enabled for RDS Instances (Ticket 20820)
Sep 16, 2024	4.0.0	The filter on the audit procedure does not match the filter in the Remediation Procedure (Ticket 21738)
Sep 26, 2024	4.0.0	ADD - Ensure CIFS access is restricted to trusted networks to prevent unauthorized access (Ticket 16568)
Sep 26, 2024	4.0.0	UPDATE - Block public access (bucket settings) - Title and content mismatch (Ticket 21203)
Sep 26, 2024	4.0.0	UPDATE - Ensure security questions are registered in the AWS account - update V8 Controls Mapping (Ticket 22624)
Sep 26, 2024	4.0.0	UPDATE - Ensure no 'root' user account access key exists - CIS Control v8 and Remediation Procedure (Ticket 22625)

Date	Version	Changes for this version
Sep 26, 2024	4.0.0	UPDATE - Ensure IAM password policy prevents password reuse - update MITRE mappings (Ticket 22634)