AWS S3 HRO Runbook

**Goal:**

Ensure that the Amazon S3 service is configured, managed, and operated in accordance with the security and privacy controls outlined in the National Institute of Standards and Technology (NIST) Cybersecurity Framework v1.1 and NIST Special Publication (SP) 800-53r5.

**Potential Impact of Failure:**

1. **Data Breaches**: S3 buckets that are misconfigured can be left publicly accessible, leading to unauthorized access and potential disclosure of sensitive data. This could result in financial losses, reputational damage, and potential legal liabilities.
2. **Data Loss**: Improperly configured backup and retention policies can lead to unintentional data deletion or loss.
3. **Data Corruption**: Without the proper access controls, data integrity can be compromised either accidentally or maliciously.
4. **Regulatory and Compliance Issues**: Organizations subject to regulations may face penalties, fines, or other legal actions for not adhering to required standards.
5. **Loss of Trust**: Stakeholders, including customers and partners, may lose trust in an organization that fails to secure its data adequately.
6. **Operational Disruptions**: Misconfigurations can disrupt services and business operations, leading to financial loss and increased operational costs.
7. **Financial Costs**: Remediation of issues after they arise is often more expensive than proactive measures. Organizations might have to invest in emergency response, forensic investigations, public relations campaigns, and other unexpected expenses.
8. **Legal Implications**: If sensitive data such as personal identifiable information (PII) is exposed due to misconfigurations, organizations could face lawsuits.
9. **Reputational Damage**: Negative press coverage from a data leak or breach can harm an organization's reputation, potentially leading to loss of customers or business opportunities.
10. **Auditing Failures**: Failure to align with NIST guidelines could lead to unfavorable results during security audits, which can have cascading effects, including loss of contracts or business.
11. **Increased Vulnerability**: Not following the framework can lead to vulnerabilities that cybercriminals can exploit, increasing the chances of cyberattacks.
12. **Loss of Competitive Advantage**: If business strategies, proprietary algorithms, or trade secrets stored in S3 are exposed, it might give competitors an unfair advantage.

# Identify assets to be stored in S3 Buckets

**Purpose:**

Understand the risk associated with the contained assets

**NIST CSF 1.1 Subcategories:**

**ID.AM-2** Software platforms and applications within the organization are inventoried

**NIST 800-53r5 Controls:**

**CM-8** Information System Component Inventory

**FIPS 199 Security Categorization:**

**HIGH** = (Confidentiality: HIGH, Integrity: HIGH, Availability: HIGH)

**Potential Impact of Failure:**

* Misclassifying assets stored in S3 buckets can directly lead to one or more of the failure scenarios.

**Dependencies:**

Documentation Runbook

**Output:**

Security Categorization of S3 asset

**Procedural Steps:**

1. Identify Assets:
2. **HRO Proficiency Test**: Ensure that the operator comprehends the significance and categorization criteria of assets before proceeding.
3. **HRO Reminder Prompt**: "Remember to thoroughly examine each object's metadata and cross-reference with security protocols to ensure accurate asset identification."
4. Examine the metadata of each object to discern the nature of the asset. Depending on your criteria, the specifics of this step might vary.
5. For potential confidential assets or those containing sensitive data, consider inspecting additional attributes such as object tags or even the object's content.
6. Determine FIPS 199 Security Categorization of Assets:
7. Assess the confidentiality, integrity, and availability requirements of the assets.
8. Reference the FIPS 199 standard to determine the appropriate security categorization (LOW, MODERATE, HIGH) for each requirement (Confidentiality, Integrity, Availability).
9. Combine the individual categorizations to derive the overall FIPS 199 security categorization for the asset.
10. Document the FIPS 199 security categorization for reference in subsequent processes and audits.
11. **HRO Operator Validation**: Confirm that the derived FIPS 199 security categorization for each asset is accurate and adheres to the defined criteria.
12. **HRO Watch Team Validation [required]**: Engage a designated watch team to cross-check and validate the security categorization. Any discrepancies should be flagged for further investigation.
13. Document/Inventory Assets:
    1. As assets are identified, maintain a structured inventory. This could be within your script or in an external system or database.
    2. Record pertinent details like the bucket's name, the object's key, size, last modified date, and other relevant metadata.
14. Review and Verification:
    1. After identifying and cataloging the assets, review the compiled list.
    2. Ensure that assets, especially those deemed confidential or sensitive, are correctly and comprehensively identified.
    3. **HRO Request Feedback**: Invite feedback from team members regarding the effectiveness of the asset identification and categorization process, looking for any suggestions or potential areas of improvement.

# Identify threats to the Data Stored in S3 bucket

**Purpose:**

Understand threats (vulnerabilities) to S3 buckets

**NIST CSF 1.1 Subcategories:**

**ID.RA-3** Threats, both internal and external, are identified and documented

**NIST 800-53r5 Controls:**

**RA-3** Risk Assessment

**FIPS 199 Security Categorization:**

Determined by Process 1 asset classification.

**Dependencies:**

Documentation Runbook

**Procedural Steps:**

1. Preparation:
   1. **HRO Proficiency Test**: Validate the operator's familiarity with AWS S3 configurations and threat identification process.
   2. **HRO Reminder Prompt**: "Ensure complete understanding of the specific dataset stored within the S3 bucket and its importance to the organization."
2. Inventory Asset:
3. Identify which S3 bucket contains the dataset.
4. **HRO Check Current State**: Validate that the S3 bucket's current state matches the most recent documented configuration.
5. Document bucket properties like region, creation date, owner, and permissions.
6. Analyze Bucket Policies and Permissions:
   1. **HRO Reminder Prompt**: "Permissions should adhere to the principle of least privilege."
   2. Retrieve Bucket Policy: Use the AWS API (*get\_bucket\_policy* method) to fetch the policy.
   3. Review Bucket ACLs: Retrieve and inspect the Access Control Lists (ACLs).
   4. Evaluate Cross-Account Access: Check for permissions that allow access from external AWS accounts.
7. Evaluate Data Encryption:
   1. **HRO Time sensitive**: If encryption is not found, raise an immediate alert or prompt for action.
   2. Check Encryption at Rest: Determine if server-side encryption is enabled for the S3 bucket using the AWS API (*get\_bucket\_encryption* method).
   3. Evaluate Data Transit Encryption: Ensure data is encrypted during transit.
8. Assess Logging and Monitoring:
   1. **HRO Reminder Prompt**: "Regular monitoring and logging are vital for early threat detection."
   2. Evaluate Logging: Confirm if AWS S3 access logs are enabled.
   3. Review Monitoring Setup: Ensure AWS CloudTrail is enabled for monitoring API calls.
9. Review Versioning and Backup:
   1. **HRO Time sensitive**: Ensure backup routines and versioning are timely to minimize data loss risks.
   2. Check Bucket Versioning: Determine if versioning is enabled on the bucket.
   3. Backup Assessment: Ensure a backup strategy is in place, either through S3 replication or other means.
10. Threat Modeling:
    1. **HRO Proficiency Test**: Ensure the operator understands threat modeling methodologies and techniques.
    2. Identify Likely Threat Actors: Determine potential malicious actors based on the nature of the data.
    3. Enumerate Possible Attack Vectors: List potential ways malicious actors might attempt to compromise the data.
    4. Evaluate Impact: Assess the potential impact on the organization for each identified threat.
11. Document Findings:
    1. Compile Threat Report: Document all identified threats, potential attack vectors, and their impact.
    2. Recommend Mitigations: Suggest actions to reduce the risk for each threat.
    3. **HRO Operator Validation**: Confirm that all identified threats and recommended mitigations have been comprehensively and accurately documented.
    4. **HRO Watch Team Validation**: Engage a designated watch team to review the threat report, ensuring comprehensiveness and accuracy.
12. Review and Update Periodically:
    1. Repeat this process at regular intervals to stay updated on potential risks.
    2. **HRO Request Feedback**: Encourage team members to provide feedback on the threat identification process, aiming to refine and improve the methodology based on evolving threats and organizational needs.

# Identify Authorized Users

**Purpose:**

Understand which users, groups of users or devices which require access to the stored assets

**NIST CSF 1.1 Subcategories:**

**PR.AC-1** Identities and credentials are managed for authorized devices and users

**NIST 800-53r5 Controls:**

**IA-2** Identification and Authentication (Organizational Users)

**IA-3** Device Identification and Authentication

**IA-8** Identification and Authentication (Non-Organizational Users)

**IA-9** Service Identification and Authentication

**FIPS 199 Security Categorization:**

Determined by Process 1 asset classification.

**Potential Impact of Failure:**

* + **Unauthorized Access**: If identities are not clearly defined or managed, unauthorized users or devices might gain access to sensitive data, leading to potential breaches of confidential information.
  + **Account Compromise**: Without a clear understanding and management of authorized users, accounts are more susceptible to phishing attacks, brute force, or other methods of unauthorized access.
  + **Elevation of Privileges**: Misconfiguration or lax management can allow users or devices to gain more privileges than intended, potentially giving them access to critical functions or data.

**Dependencies:**

Documentation Runbook

**Procedural Steps:**

1. Identify Applicable S3 Buckets:
   1. Before using the AWS API, you'd need to know which S3 buckets contain the dataset in question. This could be based on naming conventions, tags, or other organizational methods you have in place.
   2. **HRO Check Current State**: Verify that you're evaluating the most recent version of the S3 bucket configurations.
2. Analyze Permissions for Each Bucket:
   1. **HRO Proficiency Test**: Confirm the individual performing this task has the necessary AWS permissions and skills.
   2. **HRO Reminder Prompt**: "Thoroughly review each policy, as overlooking details could lead to unauthorized access or data exposure."
   3. For each bucket, retrieve the bucket policy using the *get\_bucket\_policy* method or *aws s3api get-bucket-policy* command.
   4. Extract identities (users, roles, and groups) from the bucket policies which have permissions to the buckets.
3. Review Bucket Access Control Lists (ACLs):
   1. **HRO Proficiency Test**: Ensure reviewer understands ACL significance.
   2. **HRO Reminder Prompt**: "Double-check each ACL entry. Misconfigurations can lead to unintended access permissions."
   3. Retrieve and inspect the Access Control Lists (ACLs) for each bucket using the *get\_bucket\_acl* method or *aws s3api get-bucket-acl* command.
   4. Identify any users or predefined groups with permissions from the ACLs.
4. Evaluate IAM Policies:
   1. **HRO Proficiency Test**: Confirm capability to interpret IAM policies.
   2. **HRO Reminder Prompt [required]**: "Ensure that IAM policies are scrutinized. Overlooked permissions can be a potential security risk."
   3. Use the *list\_policies* method or *aws iam list-policies* command to retrieve all policies.
   4. For each policy, use *get\_policy\_version* or *aws iam get-policy-version* command to get the policy details.
   5. Analyze the policies to see which ones have S3 permissions and identify associated users or roles.
5. List IAM Users, Roles, and Groups:
   1. **HRO Reminder Prompt**: "Validate that the lists are comprehensive and include all relevant IAM entities."
   2. Retrieve a list of IAM users using *list\_users* method or *aws iam list-users* command.
   3. Retrieve a list of IAM roles using *list\_roles* method or *aws iam list-role*s command.
   4. Retrieve a list of IAM groups using *list\_groups* method or *aws iam list-groups* command.
6. Cross-Reference IAM Entities with S3 Permissions:
   1. **HRO Proficiency Test**: Ensure understanding of IAM and S3 permissions.
   2. **HRO Reminder Prompt**: "Ensure that all IAM entities have the intended access to the relevant S3 buckets. Cross-check meticulously."
   3. Cross-reference the IAM users, roles, and groups from step 5 with the permissions identified in steps 2, 3, and 4 to determine which IAM entities have access to which S3 buckets.
7. Document Identified Users and Roles:
   1. Compile a report detailing which users, roles, and groups have access to each S3 bucket, including the type of access (read, write, etc.).
   2. **HRO Operator Validation**: Verify the report.
   3. **HRO Watch Team Validation [required]**: A secondary team or individual should review the findings for completeness and accuracy, ensuring that all accesses are intended and authorized.
8. Periodic Review:
   1. Regularly repeat these steps to ensure that the list of authorized users remains up-to-date, especially when there are changes in personnel or bucket configurations.
   2. **HRO Request Feedback**: Encourage feedback on the review process, aiming to refine and improve the methodology over time based on any challenges or insights encountered.

# Ensure all user access is strongly authenticated and authorized

**Purpose:**

Retain confidentiality and integrity of assets by scoping access to authorized users

**Dependencies:**

AuthN/AuthZ Runbook

## Ensure bucket policies limit access to authorized users only

**Purpose:**

Ensure that controls are in place to limit access to required uses

**NIST CSF 1.1 Subcategories:**

**PR.AC-4** Access permissions and authorizations are managed

**NIST 800-53r5 Controls:**

**AC-6** Least Privilege

**FIPS 199 Security Categorization:**

Determined by Process 1 asset classification.

**Procedural Steps:**

1. Identify Applicable S3 Buckets:
   1. Before using the AWS API, you'd need to know which S3 buckets contain the dataset in question. This could be based on naming conventions, tags, or other organizational methods you have in place.
   2. **HRO Check Current State**: Validate you are referencing the most recent and relevant S3 bucket list.
2. Retrieve Current Bucket Policy for Specific Bucket:
   1. API: *s3:GetBucketPolicy*
   2. Use this API call to retrieve the current policy for the identified bucket to inspect its existing permissions.
3. Evaluate the Retrieved Policy:
   1. **HRO Proficiency Test**: Verify the evaluator's capability to understand AWS bucket policies.
   2. **HRO Reminder Prompt**: "Check permissions meticulously against organizational norms. Adhere to the principle of least privilege."
   3. Manual or automated policy analysis to check if the permissions adhere to the principle of least privilege and only allow necessary access.
4. List IAM Users, Groups, and Roles:
   1. **HRO Reminder Prompt**: "Ensure all IAM entities are captured. Missed entities could lead to unintended access."
   2. APIs: *iam:ListUsers*, *iam:ListGroups*, and *iam:ListRoles*
   3. To set or adjust permissions correctly, understanding the potential bucket accessors (users/groups/roles) is essential.
5. Modify or Set Bucket Policy for the Specific Bucket:
   1. **HRO Preparatory Operator Validation:** Confirm understanding of necessary policy changes.
   2. **HRO Preparatory Watch Team Validation**: Ensure a secondary review of the intended policy changes.
   3. API: *s3:PutBucketPolicy*
   4. If adjustments are required based on the evaluation from step 3, use this API call to apply a new or updated policy to the identified bucket.
6. Validate Changes:
   1. Once changes are applied, attempt to access the bucket using various user roles to ensure the new permissions work as intended and that access controls are functioning correctly.
   2. **HRO Operator Validation**: Confirm the effectiveness of changes from an operator standpoint.
   3. **HRO Watch Team Validation [required]**: A mandatory secondary validation to ensure policy changes are correctly implemented.
7. Enable Bucket Policy Logging (Optional) for the Specific Bucket:
   1. API: *s3:PutBucketLogging*
   2. For additional security and auditing, enable access logging on the bucket of interest. This provides a record of all requests, helping you monitor and review access patterns.
   3. **HRO Watch Team Validation**: Ensure logs are correctly capturing the necessary information.
8. Periodic Review of Bucket Policy:
   1. Periodically re-evaluate and adjust the bucket policy as necessary to make sure it aligns with any changes in organizational or security requirements.
   2. **HRO Request Feedback**: Encourage team members to suggest improvements or report potential vulnerabilities in the policy.

## Disable ACLs by setting bucket owner enforced

**Purpose:**

ACLs do not ensure strong authentication, IAM policy is preferred control

**NIST CSF 1.1 Subcategories:**

**PR.IP-1** A baseline configuration of information technology/industrial control systems is created and maintained incorporating security principles (e.g., concept of least functionality)

**NIST 800-53r5 Controls:**

**CM-6** Configuration Settings

**FIPS 199 Security Categorization:**

Determined by Process 1 asset classification.

**Procedural Steps:**

1. Identify Applicable S3 Buckets:
   1. Before making any changes, identify which S3 buckets are related to the dataset or application in question.
   2. **HRO Check Current State**: Confirm you are working with the correct list of S3 buckets relevant to the dataset or application.
2. Retrieve Current Bucket ACL for Specific Bucket:
   1. **HRO Proficiency Test**: Verify the individual's ability to accurately retrieve and understand AWS bucket ACLs.
   2. **HRO Reminder Prompt**: "Ensure accurate retrieval of the ACL to avoid potential misconfigurations."
   3. API: *s3:GetBucketAcl*
   4. Fetch the current Access Control List for the bucket in question to understand its present state.
3. Retrieve Bucket Public Access Block Configuration:
   1. API: *s3:GetPublicAccessBlock*
   2. Check if public access is already blocked for the bucket. If not, you'd want to modify this setting as part of ensuring more secure access control.
4. Block All Public Access:
   1. **HRO Reminder Prompt:** "Always confirm the public access is blocked before proceeding."
   2. API: *s3:PutPublicAccessBlock*
   3. Disable all public access. This is a best practice and ensures that ACLs or other permissions don't inadvertently allow public access. Set the parameters (like *BlockPublicAcls* and *IgnorePublicAcls*) to *true*.
5. Remove All ACL Permissions:
   1. **HRO Reminder Prompt:** "Double-check the ACL settings to make certain only the bucket owner has access."
   2. **HRO Preparatory Operator Validation**: Operator indicates understanding of the procedure and implications of removing ACLs.
   3. **HRO Preparatory Watch Team Validation**: Secondary review to ensure no unintentional blockages occur.
   4. Since the goal is to shift towards IAM policies and not use ACLs, you might decide to reset the bucket's ACL to its default state where only the bucket owner has access.
   5. API: *s3:PutBucketAcl*
   6. Set the ACL parameter to "private", which grants full control to the owner and denies all other access.
6. Review and Adjust IAM Bucket Policies as Needed:
   1. API: *s3:GetBucketPolicy* and *s3:PutBucketPolicy*
   2. First, retrieve the existing policy (if any) using *s3:GetBucketPolicy*. Then, based on your organization's requirements, adjust or create an IAM policy using *s3:PutBucketPolicy*.
   3. **HRO Request Feedback:** Request team feedback to improve processes or identify overlooked areas.
7. Validate Changes:
   1. It's crucial to ensure that the changes haven't inadvertently blocked legitimate users or applications. Test access using different roles and verify that your IAM policies are granting the appropriate permissions while ACLs are not being used.
   2. **HRO Operator Validation**: Confirm changes from the operator's perspective.
   3. **HRO Watch Team Validation [required]**: A mandatory validation to ensure changes were appropriately implemented.

## Ensure all AWS clients use IAM roles for authN/authZ

**Purpose:**

Device identification using IAM roles provides strong authentication for services

**NIST CSF 1.1 Subcategories:**

**PR.IP-1** A baseline configuration of information technology/industrial control systems is created and maintained incorporating security principles (e.g., concept of least functionality)

**NIST 800-53r5 Controls:**

**CM-6** Configuration Settings

**FIPS 199 Security Categorization:**

Determined by Process 1 asset classification.

**Potential Impact of Failure:**

* **Insecure Access:** Without IAM roles, there's a higher chance of credentials being exposed, leading to unauthorized access.
* **Auditing Challenges:** IAM roles make it easier to trace who did what, making auditing and accountability more straightforward.
* **Increased Attack Surface:** Hardcoded or stored AWS keys (as opposed to temporary IAM role credentials) can be a security risk if exposed.

**Procedural Steps:**

1. Define IAM Roles for S3 Access:
   1. **HRO Proficiency Test**: Test operator's competency in defining and setting IAM roles correctly.
   2. **HRO Reminder Prompt**: "Ensure IAM roles strictly adhere to the principle of least privilege."
   3. Use the *CreateRole* API call to define new IAM roles.
   4. Set permissions specifically for S3 using the *PutRolePolicy* API call to attach inline policies or *AttachRolePolicy* to attach managed policies to the role.
2. Identify Applicable S3 Buckets:
   1. Before making any changes, identify which S3 buckets are related to the dataset or application in question.
   2. **HRO Check Current State**: Verify the list of S3 buckets to ensure alignment with the dataset or application.
3. Scan Bucket Policies:
   1. **HRO Reminder Prompt:** "Ensure that each policy aligns with IAM role-based access guidelines."
   2. For each bucket from step 2, use the *GetBucketPolicy* API call to check which policies are attached to them.
4. Apply IAM Role-Based Policies to Buckets:
   1. **HRO Preparatory Operator Validation**: Confirm the understanding and appropriateness of the IAM roles and policies before applying them.
   2. **HRO Preparatory Watch Team Validation**: Secondary check to validate the role-based policies.
   3. For each bucket, adjust its policies to allow access only to specific IAM roles or users, and possibly deny all other accesses. This is achieved using the *PutBucketPolicy* API call.
5. Review IAM Role Permissions:
   1. Periodically use the *GetRolePolicy* API call to review permissions of IAM roles. Ensure that roles have only the necessary permissions for S3 and not broad permissions.
   2. **HRO Operator Validation**: Confirm the permissions of the IAM roles.
   3. **HRO Watch Team Validation [required]**: Mandatory validation for IAM role permissions.
6. Rotate and Remove Old Access Keys:
   1. **HRO Reminder Prompt:** "Always cross-check access keys before deletion to prevent accidental removal of active keys."
   2. Use the *ListAccessKeys* API call to list all access keys associated with a user.
   3. Use the *DeleteAccessKey* API call to remove old or unused access keys.
   4. **HRO Watch Team Validation**: Check the removal of outdated access keys.
7. Audit and Monitor S3 Access:
   1. **HRO Reminder Prompt:** "Regularly check for any unauthorized or unusual access patterns in the logs."
   2. Utilize AWS CloudTrail logs to monitor S3 access activities.
   3. Periodically fetch and review logs with the *LookupEvents* API call to see which IAM roles or users are accessing the buckets, ensuring compliance with the expected access patterns.
   4. **HRO Watch Team Validation**: Confirm that S3 access activities align with the predefined guidelines.
8. Implement Cross-Account Access with IAM Roles (if necessary):
   1. **HRO Reminder Prompt:** "Ensure that only trusted accounts have cross-account access."
   2. If S3 buckets need to be accessed by entities in different AWS accounts, employ the *AssumeRole* API call to assume permissions in the target account, ensuring entities from other accounts are also using IAM roles.
   3. Adjust bucket policies with the *PutBucketPolicy* API call, granting permissions to the IAM role ARN from the other account.
   4. **HRO Watch Team Validation**: Validate the cross-account access setup.
9. Regularly Review and Update:
   1. Periodically revisit the procedure, especially when new buckets are created, ensuring they also follow IAM role-based access.
   2. **HRO Request Feedback**: Share any challenges or suggestions to improve the IAM role-based access process.

# Assess and document risk profile for each asset

**Purpose:**

Understand and communicate risk of unauthorized access

## Assess value of the assets and the likelihood of threat success

**Purpose:**

Understand risk profile by decomposing characteristics of risk

**NIST CSF 1.1 Subcategories:**

**ID.RA-4** Potential business impacts and likelihoods are identified

**NIST 800-53r5 Controls:**

**RA-3** Risk Assessment

**FIPS 199 Security Categorization:**

Determined by Process 1 asset classification.

**Potential Impact of Failure:**

* **Inefficient Resource Allocation**: Without proper assessment, resources might be allocated to protect less valuable assets, while more critical assets remain vulnerable.
* **Strategic Misalignment**: If the value of assets isn't clearly understood, the broader organizational or IT strategy may be misaligned with the actual business needs or risks.
* **Difficulty in Incident Response**: Without a prior understanding of the asset's value and threat likelihood, responding to security incidents can become chaotic and less effective.

**Procedural Steps:**

1. Identify Applicable S3 Buckets:
   1. Before making any changes, identify which S3 buckets are related to the dataset or application in question.
   2. **HRO Check Current State**: Validate the current state of the S3 buckets and ensure a correct list of assets.
2. Classify Assets:
   1. **HRO Proficiency Test**: Assess operator’s ability to correctly classify assets based on various criteria.
   2. **HRO Reminder Prompt**: "Remember to consider data sensitivity, business relevance, and compliance requirements during classification."
   3. Review object metadata, tags, or naming conventions to help classify the assets.
   4. You might also consider integrating with Amazon Macie, which uses machine learning to automatically discover, classify, and protect sensitive data stored in AWS.
3. Identify Bucket and Object Permissions:
   1. **HRO Reminder Prompt:** "Verify that permissions align with business needs and avoid overly permissive configurations."
   2. For each bucket, use *get-bucket-acl* and *get-bucket-policy* to determine its access controls and policies.
   3. Check object permissions with *get-object-acl*.
4. Analyze AWS CloudTrail Logs:
   1. If *CloudTrail* is enabled, review the logs to understand access patterns to the S3 buckets and objects.
   2. Analyze API call history to detect any unusual or unauthorized activities.
   3. **HRO Watch Team Validation:** Periodic examination of access patterns to identify anomalies.
5. Evaluate External Threat Intelligence:
   1. **HRO Reminder Prompt:** "Stay updated with the latest threat intelligence feeds and always correlate with current configurations."
   2. Integrate with AWS Security Hub or third-party services that provide threat intelligence feeds.
   3. Understand common threat vectors, especially those pertinent to S3, to gauge the likelihood of various threats.
6. Assess Value:
   1. Combine information from asset classification, permissions, access patterns, and external intelligence to derive a value for each asset.
   2. This may be a qualitative measure (e.g., high, medium, low) or a quantitative one.
   3. **HRO Watch Team Validation:** Ensure that the derived asset value is accurate and reflective of its significance.
7. Calculate Threat Likelihood:
   1. **HRO Reminder Prompt:** "When estimating threat likelihood, consider historical incidents, known vulnerabilities, and threat intelligence insights."
   2. Based on historical data, current configurations, and threat intelligence, estimate the likelihood of potential threats for each asset.
8. Document Findings:
   1. Use AWS services like AWS Systems Manager Parameter Store or Amazon RDS to securely document your findings. Alternatively, consider documenting findings in an internal risk management tool or database.
   2. **HRO Operator Validation**: Confirm that the findings are well-documented and comprehensible.
   3. **HRO Watch Team Validation [required]**: A mandatory second check to ensure findings are documented properly and securely.
9. Review and Update Regularly:
   1. Set up regular intervals (e.g., quarterly) to review and update your asset value assessments and threat likelihood calculations. Use AWS Lambda and CloudWatch Events to automate and schedule these reviews.
   2. **HRO Request Feedback**: Provide insights or recommendations to improve the risk assessment process.
10. Integrate with Risk Management:
11. **HRO Reminder Prompt**: "Align risk management priorities with asset values and threat likelihoods to ensure effective resource allocation."
12. Use the findings to integrate with your broader risk management processes, ensuring that high-value assets with high threat likelihoods receive appropriate security measures.

## Bucket risk profile is the highest risk profile of any asset (object) in the bucket

**Purpose:**

Ensure that high risk assets are not compromised by low risk asset colocation

**NIST CSF 1.1 Subcategories:**

**PR.AC-4** Access permissions and authorizations are managed, incorporating the principles of least privilege and separation of duties  
**PR.DS-5** Protections against data leaks are implemented

**NIST 800-53r5 Controls:**

**AC-4** Information Flow Enforcement  
**AC-6** Least Privilege

**FIPS 199 Security Categorization:**

Determined by Process 1 asset classification.

**Potential Impact of Failure:**

* **Performance and Cost:** While security is paramount, applying the highest risk profile settings and protections to all assets, including those with a lower risk profile, might introduce unnecessary performance hits and costs. Regularly reviewing and optimizing bucket configurations can help mitigate these challenges.
* **Usability and Accessibility:** Overly restrictive access might create barriers for legitimate users or services. It might be worthwhile to consider finer-grained access controls or segregation of assets, so high-risk data doesn't inadvertently limit access to low-risk data that should be more accessible.
* **Maintenance and Management Overhead:** The need to continually reassess buckets as new assets are added could introduce significant management overhead. Consider implementing tools or AWS services that can automate asset classification and the corresponding risk assessments.
* **Mitigation of Risk through Separation:** If possible, high-risk assets should be isolated in their own buckets, and strict policies should be applied to only those buckets. This strategy helps avoid the potential consequences of an overly broad policy and keeps high-risk data separate from lower-risk data.

**Procedural Steps:**

1. Inventory Assets within the Bucket:
   1. Use the *ListObjects* API call to retrieve a list of all objects within the S3 bucket.
   2. **HRO Check Current State**: Ensure an up-to-date inventory of all assets in the bucket.
2. Determine Risk Profile of Each Object:
3. **HRO Proficiency Test**: Evaluate an operator's capability to assess and tag assets based on risk correctly.
4. **HRO Reminder Prompt**: "Always refer to company guidelines or criteria when determining the risk profile of an asset."
5. This step might involve a custom process or solution since AWS does not inherently assign risk profiles to individual objects. If you have a tagging mechanism or metadata attribute indicating risk for each object, you can fetch it using the AWS SDK.
6. Evaluate the Highest Risk Profile:
7. **HRO Reminder Prompt:** "Always consider the impact, likelihood, and vulnerability attributes of each asset during this evaluation."
8. Iterate over each object's risk profile (from metadata or tags) to identify the highest risk profile among all objects in the bucket.
9. Apply Highest Risk Profile Settings to Bucket:
10. **HRO Preparatory Operator Validation**: Confirm that you are about to set the appropriate configurations and policies reflective of the highest risk profile.
11. **HRO Preparatory Watch Team Validation**: Ensure that a second team or set of eyes has checked and validated the configurations before application.
12. Depending on how you define and implement risk profiles, you will now apply the relevant configurations and policies to the S3 bucket. This typically involves setting up bucket policies, ACLs, and other AWS services integrations to enforce the highest risk profile settings.
13. Validate Bucket Policies:
14. Use the *get\_bucket\_policy* API call to retrieve the bucket's policy.
15. Analyze the policy to ensure that it aligns with the required highest risk profile.
16. **HRO Operator Validation**: Confirm that the bucket policy matches the desired risk profile settings.
17. **HRO Watch Team Validation [required]**: A mandatory secondary validation to confirm that the bucket policy is correctly configured.
18. Periodically Review and Adjust:
19. Since the bucket's content can change over time, regularly use the AWS API to review object risk profiles and adjust bucket settings as necessary.
20. **HRO Request Feedback**: After each review, provide insights and recommendations for optimization.
21. Logging and Monitoring:
22. **HRO Reminder Prompt:** "Regularly review logs to detect anomalies or potential security breaches."
23. Enable AWS CloudTrail for S3 bucket monitoring.
24. Ensure that AWS Config is set up to monitor S3 bucket settings, ensuring compliance with the highest risk profile.
25. Notifications:
26. Use S3 bucket notifications and AWS Lambda to get alerted when high-risk assets are added to the bucket. This will enable immediate review and potential reconfiguration.
27. **HRO Watch Team Validation**: Ensure that notifications are operational and the team is responsive to these alerts.
28. Documentation:
29. **HRO Reminder Prompt:** "Keep documentation clear, detailed, and easily accessible for future reference and auditing purposes."
30. Document the bucket's risk profile and relevant configurations for auditing and governance purposes.

## Ensure all objects are tagged with the appropriate risk profile setting

**Purpose:**

Track assets risk profile throughout the lifecycle of the storage container

**NIST CSF 1.1 Subcategories:**

**PR.DS-5** Protections against data leaks are implemented

**NIST 800-53r5 Controls:**

**AC-4** Information Flow Enforcement

**FIPS 199 Security Categorization:**

Determined by Process 1 asset classification.

**Potential Impact of Failure:**

* **Misclassification:** Incorrect tagging could result in higher risk data being treated as lower risk or vice versa.
* **Operational Issues:** Without consistent tagging, automations and alerts based on those tags may fail or produce false results.
* **Audit Failures:** During security audits, inconsistency in risk tagging might lead to audit failures and subsequent reputational damage.
* **Increased Cost:** Incorrect tagging can lead to data being stored on more expensive storage classes than necessary or additional costs to rectify misclassified data.

**Procedural Steps:**

1. Identify the S3 Buckets Relevant to the Dataset:
2. Use the *list\_buckets()* API call to get a list of all your S3 buckets.
3. Determine which buckets are relevant to your dataset.
4. **HRO Check Current State**: Confirm that you are working with the relevant dataset and have a clear list of associated buckets.
5. List Objects in the Bucket:
6. **HRO Proficiency Test**: Check the ability of the operator to identify and differentiate objects within the bucket correctly.
7. **HRO Reminder Prompt**: "Ensure you are considering all objects within the specified bucket for accurate risk profiling."
8. Use the *list\_objects\_v2(Bucket='your-bucket-name')* API call.
9. Retrieve a list of all objects within the chosen bucket.
10. Define the Risk Profile Tags:
11. Depending on the criteria from Process 1, craft the appropriate tags.
12. Tags might be structured like: *{ 'Key': 'RiskProfile', 'Value': 'High' }*.
13. Apply Tags to Objects:
14. **HRO Preparatory Operator Validation**: Confirm that the correct risk tags are ready for application.
15. **HRO Preparatory Watch Team Validation**: Ensure that a secondary team has verified and validated the risk tags before application.
16. For each object in the bucket, apply the risk profile tag.
17. Use the *put\_object\_tagging(Bucket='your-bucket-name', Key='your-object-key', Tagging={'TagSet': [tag]})* API call.
18. Verify Tags on Objects:
19. Ensure that tags have been applied correctly.
20. Use the *get\_object\_tagging(Bucket='your-bucket-name', Key='your-object-key')* API call for verification.
21. **HRO Operator Validation**: Ensure each object's tag aligns with its risk profile and there are no mismatches or omissions.
22. **HRO Watch Team Validation [required]**: Mandatory secondary validation to confirm correct tagging across all objects.
23. Set Up S3 Bucket Policies based on Tags (Optional):
24. **HRO Reminder Prompt:** "Ensure the bucket policies accurately reflect and enforce the desired behaviors based on the object's risk profile tags."
25. Enforce specific policies based on risk profile tags.
26. Use the *put\_bucket\_policy(Bucket='your-bucket-name', Policy='your-policy')* API call.
27. Monitor and Audit:
28. Regularly check that the tags on objects remain consistent with their risk profile.
29. Employ AWS CloudTrail and AWS Config for monitoring and auditing purposes.
30. **HRO Watch Team Validation**: Make sure periodic checks occur and deviations are promptly identified and rectified.
31. Retain Documentation:
32. Keep a record of the tagging criteria and decisions made during the classification.
33. This documentation is vital for future audits and understanding tag contexts.

# Ensure private access to S3 buckets in normal or high risk profile

**Purpose:**

Track assets risk profile throughout the lifecycle of the storage container

## Ensure S3 “Public Block Access” settings are set as follows on the bucket

**Purpose:**

Minimize attack surface for high risk assets

**NIST CSF 1.1 Subcategories:**

**PR.DS-5** Protections against data leaks are implemented

**NIST 800-53r5 Controls:**

**SC-7** Boundary Protection

**FIPS 199 Security Categorization:**

Determined by Process 1 asset classification.

**Procedural Steps:**

1. Retrieve the Current Bucket Policy Settings:
   1. Use *get\_bucket\_policy\_status* to see the current policy status.
   2. This will help determine if there are any existing policies that may grant public access.
   3. **HRO Check Current State**: Ensure that you have a clear and comprehensive understanding of the current bucket policy settings, especially with regard to public access.
2. Check Current Public Block Settings:
   1. **HRO Reminder Prompt:** "Cross-reference with known configurations to ensure accuracy in current settings retrieval."
   2. Use the *get\_public\_access\_block* API call to retrieve the bucket's current public access block settings.
3. Set or Update Public Block Access:
4. **HRO Preparatory Operator Validation**: Confirm that you understand the importance of blocking all public access, and that you have the right settings ready for implementation.
5. **HRO Preparatory Watch Team Validation**: Verify and validate the proposed block access settings before applying them to ensure no oversight.
6. Utilize the *put\_public\_access\_block* API call to set the appropriate block access settings.
7. Make sure to set all public block options to True to block all public access. This includes:
   * + *BlockPublicAcls*
     + *IgnorePublicAcls*
     + *BlockPublicPolicy*
     + *RestrictPublicBuckets*
8. Verify the Changes:
9. Repeat the *get\_public\_access\_block* call to confirm that the desired settings have been correctly applied.
10. Review any other bucket policies or IAM policies to ensure they don't inadvertently provide public access.
11. **HRO Operator Validation:** Ensure each setting accurately blocks public access and there are no mismatches or omissions.
12. **HRO Watch Team Validation [required]:** Mandatory secondary validation to confirm correct blocking across all public access settings.
13. Audit Regularly:
14. Periodically use the *list\_buckets* followed by *get\_public\_access\_block* for each bucket to ensure no public access settings have changed.
15. Consider setting up CloudWatch or AWS Config to monitor and alert on any changes to public access settings for critical buckets.
16. **HRO Request Feedback**: After a series of checks, request feedback from the team to improve the monitoring process. Ensure everyone remains vigilant about the importance of blocking all public access.

## Ensure all S3 bucket policies are free of wildcards in the identity fields

**Purpose:**

Ensure appropriate access controls

**NIST CSF 1.1 Subcategories:**

**PR.AC-7** Users, devices, and other assets are authenticated

**NIST 800-53r5 Controls:**

**IA-2** Identification and Authentication  
**IA-3** Device Identification and Authentication

**FIPS 199 Security Categorization:**

Determined by Process 1 asset classification.

**Potential Impact of Failure:**

* **Unauthorized Access**: Wildcards in identity fields could allow any AWS entity to access the S3 bucket, potentially leading to data breaches.
* **Service Disruption**: Overly permissive permissions could enable disruptive actions against the service or data.
* **Increased Risk Profile**: With wildcards in policies, the surface area of potential threats expands dramatically.
* **Compliance Issues**: Many data protection regulations and guidelines emphasize the principle of least privilege. Wildcard permissions could be in violation of such standards.

**Procedural Steps:**

1. Retrieve Current Bucket Policy:
2. Use the *get\_bucket\_policy* API call to obtain the policy attached to the S3 bucket.
3. **HRO Check Current State**: Ensure that you understand the current policy setup and that no prior deviations exist.
4. Parse and Inspect the Policy:
5. **HRO Proficiency Test**: Evaluate your understanding of AWS IAM policies and their structure.
6. **HRO Reminder Prompt**: "Wildcards (\*) in 'Principal' or 'Action' fields can lead to overly permissive access. Always adhere to the principle of least privilege."
7. Convert the policy from a JSON string to a readable format.
8. Examine the policy's statements. Look for any statements with a wildcard (\*) in the "Principal" or "Action" field.
9. Remove or Update Overly Permissive Policies:
10. If wildcards are found in inappropriate locations, modify the policy to be more specific or remove that statement entirely.
11. Apply the Updated Policy:
12. **HRO Preparatory Operator Validation**: Confirm that you are aware of the implications of changing the bucket policy and have backup or rollback options ready.
13. **HRO Preparatory Watch Team Validation**: Seek a secondary verification to ensure that the updated policy is appropriately configured without oversights.
14. Utilize the *put\_bucket\_policy* API call to replace the old policy with the revised one.
15. Verify the Changes:
16. Retrieve the bucket policy again with *get\_bucket\_policy* and ensure that the wildcard entries are removed.
17. Also, validate that the changes didn't negatively affect legitimate users or processes.
18. **HRO Operator Validation:** Reconfirm that the applied changes match the intended policy modifications.
19. **HRO Watch Team Validation [required]:** Mandatory secondary validation to guarantee that the policy is free from wildcards and no unintended permissions are granted.
20. Regular Audits:
21. **HRO Reminder Prompt:** "Maintaining security is an ongoing process. Always be vigilant of any policy changes that might introduce risks."
22. Periodically repeat the process to ensure no new wildcard permissions have been added.
23. Consider setting up AWS Config or CloudWatch alarms to monitor and alert on any changes to bucket policies that reintroduce wildcards.
24. **HRO Request Feedback**: Seek feedback after audits to continuously improve the process and ensure alignment with HRO principles.

## Ensure all S3 bucket ACLs are free of Everyone or Any authenticated AWS user

**Purpose:**

Ensure appropriate access controls

**NIST CSF 1.1 Subcategories:**

**PR.AC-1** Identities and credentials are managed for authorized devices and users

**NIST 800-53r5 Controls:**

**IA-2** Identification and Authentication

**FIPS 199 Security Categorization:**

Determined by Process 1 asset classification.

**Potential Impact of Failure:**

* **Unauthorized Access**: ACLs allowing "Everyone" or "Any authenticated AWS user" can lead to unwanted actors accessing the bucket, potentially causing data breaches.
* **Service Disruption**: Unauthorized actions against the service/data can be disruptive.
* **Increased Risk Profile**: Broad permissions raise the risk profile of the S3 bucket.
* **Compliance Issues**: Such broad permissions could violate data protection standards or regulations, leading to penalties.

**Procedural Steps:**

1. Retrieve Current Bucket ACL:
2. Use the *get\_bucket\_acl* API call to retrieve the Access Control List (ACL) for the specified S3 bucket.
3. **HRO Check Current State**: Before making any changes, ensure that you've thoroughly understood the current ACL setup, and that it's in a consistent state with what you expect.
4. Examine the ACL for Broad Permissions:
5. **HRO Proficiency Test**: Assess your familiarity with the AWS ACL structure and permissions.
6. **HRO Reminder Prompt**: "Broad permissions, especially those allowing 'Everyone' or 'Any authenticated AWS user', can significantly elevate the risk of unauthorized access. Always act with caution."
7. Inspect the returned ACL for any grants that have the "Grantee" set to "<http://acs.amazonaws.com/groups/global/AllUsers>" (which represents "Everyone") or "<http://acs.amazonaws.com/groups/global/AuthenticatedUsers>" (which represents "Any authenticated AWS user").
8. Remove Overly Permissive Grants:
9. **HRO Preparatory Operator Validation**: Ensure that you have a full understanding of the implications of removing these permissions and that you're ready to act upon it.
10. **HRO Preparatory Watch Team Validation**: Secure a secondary confirmation that these broad permissions should be eliminated, ensuring that no oversight happens.
11. If any broad permissions are found, you'll need to modify the ACL to remove these permissions.
12. Utilize the *put\_bucket\_acl* API call to update the ACL for the bucket with the overly permissive grants removed.
13. Validate the Changes:
14. After updating, retrieve the ACL again with *get\_bucket\_acl* to ensure that the changes were applied correctly and that no broad permissions remain.
15. Additionally, ensure that the changes did not inadvertently disrupt access for valid users or processes.
16. **HRO Operator Validation**: Double-check and verify that the ACL updates match the intended security adjustments.
17. **HRO Watch Team Validation [required]**: An obligatory second check to confirm that the bucket is now free from any broad permissions and is securely configured.
18. Regular Audits:
19. Periodically re-check the ACLs to ensure that no broad permissions have been reintroduced.
20. Consider setting up monitoring tools such as AWS Config to track ACL changes and to send alerts for any suspicious modifications.
21. **HRO Request Feedback**: Solicit feedback post audits to refine the process continuously and guarantee alignment with best practices and HRO principles.

## Ensure all access to the bucket is via a VPC endpoint

**Purpose:**

Minimize attack surface

**NIST CSF 1.1 Subcategories:**

**PR.AC-5** Network integrity is protected, incorporating network segregation where appropriate

**NIST 800-53r5 Controls:**

**SC-7** Boundary Protection

**FIPS 199 Security Categorization:**

**LOW** = (Confidentiality: LOW, Integrity: LOW, Availability: LOW)

### Ensure there is a VPC endpoint policy of “Access-to-specific-VPCE-only” for each authorized VPC endpoint with default “Deny”

**Purpose:**

Minimize attack surface

**NIST CSF 1.1 Subcategories:**

**PR.AC-5** Network integrity is protected, incorporating network segregation where appropriate

**NIST 800-53r5 Controls:**

**SC-7** Boundary Protection

**FIPS 199 Security Categorization:**

**LOW** = (Confidentiality: LOW, Integrity: LOW, Availability: LOW)

**Potential Impact of Failure:**

* **Inconsistent Network Policies**: Not using VPC endpoints might lead to not uniformly applying network transit policies.

**Procedural Steps:**

1. Retrieve the Bucket Policy:
2. Use *get\_bucket\_policy* API call to retrieve the current bucket policy.
3. **HRO Check Current State**: Before modifying the policy, thoroughly understand the current state and ensure that it aligns with what you anticipate.
4. Check for VPC Endpoint Restriction:
5. **HRO Proficiency Test**: Evaluate your understanding of VPC endpoint restrictions within bucket policies.
6. **HRO Reminder Prompt**: "VPC endpoint restrictions are critical to ensure controlled and secure access to AWS resources. Always be meticulous when assessing and implementing them."
7. Inspect the bucket policy for statements ensuring that the source VPC endpoint or VPC is explicitly mentioned, making sure that it's the only way to access the bucket.
8. Set Up VPC Endpoint:
9. If not already done, create a VPC endpoint for S3 using the *create\_vpc\_endpoint* API call.
10. Modify Bucket Policy:
11. **HRO Preparatory Operator Validation**: Prior to implementing changes, confirm that you are well-acquainted with the implications of adjusting the bucket policy for VPC endpoint restrictions.
12. **HRO Preparatory Watch Team Validation**: Acquire a secondary confirmation before policy alterations to ensure no detail is overlooked.
13. If the bucket policy does not restrict access to the VPC endpoint, adjust the policy to ensure that only the VPC endpoint can access it.
14. Use *put\_bucket\_policy* API call to apply the updated policy to the S3 bucket.
15. Validate the Configuration:
16. Use the *list\_vpc\_endpoints* API call to ensure the VPC endpoint for S3 is active and correctly configured.
17. Test accessing the bucket from within the VPC using the endpoint, and from outside the VPC to ensure the policy is working as expected.
18. **HRO Operator Validation**: Double-check that the endpoint restrictions are applied accurately and that there are no loopholes.
19. **HRO Watch Team Validation [required]**: A mandatory secondary check to confirm that the bucket is securely configured for the intended VPC endpoint.
20. Monitor and Audit:
21. Set up logging and monitoring, such as VPC Flow Logs and AWS CloudTrail, to monitor bucket access attempts.
22. Regularly audit the access logs to identify any unauthorized or unexpected access patterns.
23. **HRO Request Feedback**: Invite feedback after audits to constantly refine the process, ensuring it aligns with best practices and HRO principles.

## Select Client-Side Encryption

**Purpose:**

Protect confidentiality of assets

**NIST CSF 1.1 Subcategories:**

**PR.DS-1** Data-at-rest is protected

**NIST 800-53r5 Controls:**

**SC-28** Protection of Information at Rest

**FIPS 199 Security Categorization:**

Determined by Process 1 asset classification.

**Potential Impact of Failure:**

* **Data Exposure:** If data isn't encrypted client-side before being transferred, it may be exposed during transit or while at rest if the storage service's own encryption is compromised.

**Procedural Steps:**

1. Choose an Encryption SDK or Library:
2. **HRO Proficiency Test**: Assess your understanding and expertise in the selection and usage of encryption SDKs or libraries
3. **HRO Reminder Prompt**: "Selecting the correct encryption library is paramount. Ensure your choice aligns with the application's requirements and recognized security standards."
4. Select an appropriate SDK or library, like the AWS Encryption SDK, for your application's language and platform.
5. **HRO Check Current State**: Verify the present encryption methods, if any, before making modifications.
6. Generate Encryption Keys:
7. Generate or choose an appropriate encryption key.
8. Consider using AWS Key Management Service (KMS) to manage the encryption key lifecycle.
9. Encrypt Data Before Storing:
10. Use the chosen SDK or library functions to encrypt data on the client side before sending it to S3 or any other AWS service.
11. Transfer Encrypted Data:
12. Use regular AWS SDK functions, like *put\_object* for S3, to store the encrypted data.
13. Decrypt Data After Retrieval:
14. When retrieving data, use the AWS Encryption SDK or chosen library to decrypt the data after it's been downloaded.
15. Rotate Encryption Keys:
16. Periodically rotate the encryption keys used for client-side encryption.
17. If using AWS KMS, utilize its key rotation capabilities.
18. **HRO Operator Validation**: After rotating keys, confirm that the new keys function correctly without causing decryption issues or data access challenges.
19. **HRO Watch Team Validation [required**]: Key rotations can inadvertently introduce vulnerabilities. A second-layer check is imperative to ascertain that the new keys are safely deployed.
20. Monitor and Audit:
21. Set up logging, such as with AWS CloudTrail, to monitor the use of encryption and decryption operations.
22. Regularly audit the logs to ensure that encryption is consistently applied and to detect any anomalies.
23. **HRO Request Feedback**: Invite team members and stakeholders to offer feedback on the encryption process, looking for improvements or highlighting potential areas of concern.

## Ensure versioning is selected

**Purpose:**

Denial of service mitigation (ransomeware, e.g.)

**NIST CSF 1.1 Subcategories:**

**PR.IP-4** Backups of information are conducted, maintained, and tested periodically

**NIST 800-53r5 Controls:**

**SC-5** Denial of Service Protection

**CP-9** Information System Backup

**FIPS 199 Security Categorization:**

**LOW** = (Confidentiality: LOW, Integrity: LOW, Availability: LOW)

**Potential Impact of Failure:**

* **Data Loss**: Without versioning, unintentional overwrites or deletions could lead to permanent data loss.
* **Limited Recovery from Ransomware**: Ransomware typically encrypts files and demands payment for decryption. Without versioning, recovering the original versions of encrypted files would be challenging or impossible.
* **Increased Restoration Time**: In the absence of versioning, recovering an earlier state of the data would involve searching through backups, increasing the time taken for restoration.

**Procedural Steps:**

1. Enable Versioning on S3 Bucket:
2. **HRO Check Current State**: Confirm the current state of versioning on the S3 bucket to ensure no unintended changes have been made.
3. **HRO Preparatory Operator Validation**: Prior to enabling, assess the potential impact on storage and costs. Understand the implications and benefits.
4. **HRO Preparatory Watch Team Validation**: Obtain secondary approval from a monitoring team or supervisor to ensure the implications of enabling versioning are clearly understood.
5. Use the *put\_bucket\_versioning* API call with Status set to "Enabled".
6. **HRO Operator Validation**: After enabling, verify that versioning is active and properly functioning.
7. **HRO Watch Team Validation [required]**: A secondary team should confirm that versioning is successfully enabled without any issues.
8. **HRO Request Feedback**: Ask team members for feedback on the versioning process. Are there any concerns or suggestions for improvement?
9. Regularly Monitor Versioning Status:
10. Use the *get\_bucket\_versioning* API call to periodically check the versioning status of the bucket.
11. Restore Older Object Versions When Necessary:
12. Use the *get\_object* API call with the *VersionId* parameter to retrieve a specific version of an object.
13. Periodically Clean Up Old Versions:
14. While versioning helps in preserving object versions, it might lead to increased storage costs. Consider setting up lifecycle policies to transition older versions to cheaper storage classes or delete non-essential versions after a set period.
15. Ensure Monitoring and Alarms:
16. Use AWS CloudWatch to set up alarms for events related to object overwrites or deletions. This helps in taking quick actions in case of unintended changes.
17. Backup Critical Data:
18. Beyond versioning, have a strategy for backing up critical data. This could be cross-region replication to another S3 bucket or even to another AWS service.
19. Test Data Restoration:
20. Periodically test the process of restoring data from previous versions to ensure that the versioning setup works as intended.
21. **HRO Request Feedback**: Invite team members to offer feedback on the restoration process. Are there any concerns about the integrity or accessibility of restored data?

# For Buckets in the low risk profile

## If public access is required

**Purpose:**

Appropriate availability

**NIST CSF 1.1 Subcategories:**

**PR.IP-1** A baseline configuration of information technology/industrial control systems is created and maintained incorporating security principles (e.g., concept of least functionality)

**NIST 800-53r5 Controls:**

**CM-6** Configuration Settings

**FIPS 199 Security Categorization:**

**MODERATE** = (Confidentiality: MODERATE, Integrity: LOW, Availability: LOW)

**Potential Impact of Failure:**

* **Unintended Data Exposure**: Without proper controls, making a bucket or object public could lead to the accidental exposure of sensitive data.

**Procedural Steps:**

1. Review the Need for Public Access:
2. **HRO Proficiency Test**: Test the understanding of team members on the implications of public access and ensure they can differentiate between public and private access configurations.
3. **HRO Reminder Prompt**: "Remember, public access can expose data to anyone on the internet. Ensure there's a valid reason for this action."
4. Before making any data public, evaluate why it's necessary and if there are alternative solutions.
5. Set Bucket Permissions to Public:
6. **HRO Check Current State**: Verify the current state of the bucket's permissions to ensure you're not unintentionally changing permissions.
7. **HRO Preparatory Operator Validation**: Ensure that any sensitive data has been appropriately segregated or redacted before proceeding.
8. **HRO Preparatory Watch Team Validation**: Gain secondary confirmation from a separate team or supervisor about the necessity of making this bucket public.
9. Use the *put\_bucket\_acl* API call to modify the bucket's access control list (ACL) and grant public read access.
10. Review Object Permissions:
11. Even if a bucket is public, individual objects might not be. Consider using the *put\_object\_acl* API call to ensure the desired objects are publicly accessible.
12. Apply Bucket Policies:
13. Instead of using ACLs, you can use bucket policies to provide more granular access. Define a policy using the *put\_bucket\_policy* API call.
14. **HRO Operator Validation**: After defining the bucket policy, validate that the permissions are set as intended and that no unintentional access is granted.
15. **HRO Watch Team Validation [required]**: A separate team should confirm that the bucket policy is set appropriately and doesn't introduce unintended vulnerabilities.
16. Enable Logging and Monitoring:
17. Use AWS CloudWatch and S3 access logs to monitor the public bucket, ensuring you can track access patterns and detect potential malicious activity.
18. **HRO Watch Team Validation**: A separate team or automated system should continuously monitor and validate the logging results for unexpected behavior.
19. Review Periodically:
20. Periodically re-evaluate the need for public access. Use the *get\_bucket\_acl* and *get\_object\_acl* API calls to review permissions and ensure they align with the current requirements.
21. **HRO Request Feedback**: Ask team members if the public access setting still aligns with organizational needs and gather suggestions for potential changes or improvements.
22. Set Up Alerts:
23. Set up alerts to be notified of any changes to the bucket's public access settings. This can help in catching unintended changes.
24. **HRO Watch Team Validation**: Ensure a secondary team or system also gets these alerts to provide a secondary layer of oversight.

## If public access is not required

**Purpose:**

Minimize attack surface

**NIST CSF 1.1 Subcategories:**

**PR.IP-1** A baseline configuration of information technology/industrial control systems is created and maintained incorporating security principles (e.g., concept of least functionality)

**NIST 800-53r5 Controls:**

**CM-6** Configuration Settings

**FIPS 199 Security Categorization:**

**LOW** = (Confidentiality: LOW, Integrity: LOW, Availability: LOW)

**Procedural Steps:**

1. Check Current Bucket Permissions:
2. **HRO Proficiency Test**: Test team members on their ability to identify and distinguish between various bucket permission settings.
3. **HRO Reminder Prompt**: "Ensure you are inspecting the correct bucket and remember, public access should be restricted."
4. Utilize the *get\_bucket\_acl* API call to review the current access control list (ACL) of the bucket.
5. **HRO Check Current State**: Verify the initial state of bucket permissions to confirm the need for changes.
6. Remove Public Bucket Permissions:
7. If public access is detected, use the *put\_bucket\_acl* API call to modify the bucket's ACL and restrict public access.
8. Review and Modify Object Permissions:
9. Objects within the bucket might have been individually set to public access. Consider using the *get\_object\_acl* and *put\_object\_acl* API calls to review and modify permissions.
10. Update or Remove Bucket Policies Granting Public Access:
11. If a bucket policy is granting public access, update it. Use the *get\_bucket\_policy* API call to review policies and the *put\_bucket\_policy* or *delete\_bucket\_policy* to modify or remove them.
12. Enable Block Public Access Feature:
13. **HRO Preparatory Operator Validation**: Ensure that all data within the bucket is appropriately categorized and no public dependencies exist before blocking public access.
14. **HRO Preparatory Watch Team Validation**: Receive a secondary confirmation from a separate team or supervisor that the bucket should indeed have public access blocked.
15. Use the AWS Management Console or the *put\_public\_access\_block* API call to configure S3 Block Public Access settings for your account or bucket. This provides an additional layer of restriction against making data public unintentionally.
16. **HRO Operator Validation**: After setting the Block Public Access feature, validate that all data remains internally accessible and that no legitimate external requests are blocked.
17. **HRO Watch Team Validation [required]**: An independent team should confirm that the block public access settings are appropriate and no inadvertent access barriers were established.
18. Regular Monitoring:
19. Set up monitoring mechanisms using AWS CloudWatch and S3 access logs. Regularly check for changes in permissions and access patterns.
20. **HRO Request Feedback**: Engage with team members to gather insights on the monitoring process and any suggestions for improvements.
21. Alerts for Unauthorized Changes:
22. Ensure alerts are in place to notify about changes to the bucket's public access settings or unusual access patterns. This helps in identifying and rectifying any unintended changes or breaches promptly.
23. **HRO Watch Team Validation**: Ensure a separate team or system is also alerted to provide an extra layer of oversight and rapid response.

# For buckets with only immutable objects ensure S3 object lock is set

**Purpose:**

Denial of Service mitigation (e.g. ransomware)

**NIST CSF 1.1 Subcategories:**

**PR.IP-4** Backups of information are conducted, maintained, and tested periodically

**NIST 800-53r5 Controls:**

**SC-5** Denial of Service Protection

**FIPS 199 Security Categorization:**

**LOW** = (Confidentiality: LOW, Integrity: LOW, Availability: LOW)

**Potential Impact of Failure:**

* **Ransomware Attacks:** Attackers can encrypt the data and demand a ransom for its release.

**Procedural Steps:**

1. Identify the Relevant Bucket Containing the Dataset:
2. Use bucket naming conventions, tags, or metadata to pinpoint the S3 bucket containing the dataset in question.
3. **HRO Check Current State**: Verify the present state of the S3 bucket before making any alterations. Confirm that no unauthorized changes have occurred.
4. Review Current Object Lock Configuration for the Identified Bucket:
5. **HRO Proficiency Test**: Test operator’s knowledge regarding the Object Lock feature and its importance.
6. **HRO Reminder Prompt**: “Remember the significance of enabling Object Lock for data integrity and protection against unauthorized alterations.”
7. Utilize the *GetObjectLockConfiguration* API call to determine if the Object Lock is already activated for the specified bucket.
8. Enable Object Lock for the Bucket:
9. If Object Lock isn't activated, note that for existing buckets, you cannot enable Object Lock unless it's a newly created bucket. For such buckets, consider migrating the dataset to a new bucket with Object Lock activated.
10. For setting up Object Lock on a new bucket, employ the P*utObjectLockConfiguration* API call. Once activated, Object Lock can't be disabled.
11. Set the Retention Period for the Dataset:
12. **HRO Preparatory Operator Validation**: Confirm understanding of the significance of setting a retention period and ensuring data remains immutable.
13. **HRO Preparatory Watch Team Validation**: Engage the watch team to confirm the process and intent before setting the retention period.
14. Use the *PutObjectRetention* API call to establish the retention duration for the dataset, rendering it immutable for that period.
15. **HRO Operator Validation**: Verify that the retention period has been correctly set and the data is locked.
16. **HRO Watch Team Validation [required]**: Ensure that the watch team cross-checks and validates the retention settings.
17. Continuous Monitoring:
18. Monitor the bucket routinely using tools like AWS CloudWatch and S3 access logs to identify any irregularities or unauthorized modifications.
19. **HRO Watch Team Validation**: The watch team should periodically validate the monitoring process, ensuring alerts are functional and unauthorized modifications are tracked.
20. Institute Alert Mechanisms:
21. Configure alerts to be notified about any alterations to the Object Lock configurations or if attempts are made to modify the dataset in the bucket.
22. **HRO Request Feedback**: Gather feedback from the operator about the ease of configuring alerts, any potential issues faced, and suggestions for improvement.

# Ensure all communications are secured by TLS

**Purpose:**

Protect confidentiality and integrity

**NIST CSF 1.1 Subcategories:**

**PR.DS-2** Data-in-transit is protected

**NIST 800-53r5 Controls:**

**SC-8** Transmission Confidentiality and Integrity

**FIPS 199 Security Categorization:**

**MODERATE** = (Confidentiality: MODERATE, Integrity: LOW, Availability: LOW)

**Potential Impact of Failure:**

* **Data Interception**: Unencrypted data can be intercepted during transit by malicious entities leading to unauthorized access.
* **Man-in-the-Middle Attacks**: Without TLS, communications are susceptible to MITM attacks, where attackers can secretly relay or even alter the communication between two parties.

**Procedural Steps:**

1. Identify Endpoints and Services:
2. List all AWS services, applications, and endpoints that communicate data. This could include S3 buckets, EC2 instances, RDS databases, and more.
3. **HRO Check Current State**: Verify that all identified endpoints are in their expected security configurations.
4. Review Existing Communication Protocols:
5. **HRO Proficiency Test**: Ensure the operator understands the significance of using encrypted communication protocols.
6. **HRO Reminder Prompt**: "Always ensure that data communications are using TLS encryption. Unencrypted data exposes our systems to significant vulnerabilities, including interception and MITM attacks."
7. Use AWS service-specific logging and monitoring solutions, such as VPC Flow Logs and ELB access logs, to ascertain if any unencrypted traffic exists.
8. Force TLS on S3 Buckets:
9. **HRO Preparatory Operator Validation**: Confirm understanding of the steps before execution.
10. **HRO Preparatory Watch Team Validation**: Ensure a secondary review of the intended changes by another team member.
11. For S3 buckets, employ the *PutBucketPolicy* API call to enforce a bucket policy that allows only HTTPS (TLS) for communication.
12. **HRO Operator Validation**: Confirm the changes were implemented as expected.
13. **HRO Watch Team Validation [required]**: Secondary verification that the bucket policy has been set correctly.
14. Enforce TLS on Load Balancers:
15. If using Elastic Load Balancers (ELB), ensure they are configured to listen only on HTTPS and terminate SSL/TLS at the load balancer.
16. RDS & Database Encryption:
17. For RDS or other database services, leverage the appropriate API calls to modify the instance or cluster to enforce encrypted connections.
18. Continuous Monitoring & Alerts:
19. Use AWS CloudWatch or other monitoring tools to detect and alert on any non-TLS traffic in the environment.
20. Periodic Reviews:
21. Regularly review the security configurations to ensure adherence and to account for any newly introduced services or changes.
22. **HRO Request Feedback**: After the review, gather feedback on the process, any challenges faced, and areas for improvement.

# Ensure an audit log of all accesses to S3 objects is created

**Purpose:**

Detect and analyze asset behavior

**NIST CSF 1.1 Subcategories:**

**DE.CM-7** Monitoring for unauthorized personnel, connections, devices, and software is performed

**NIST 800-53r5 Controls:**

**AU-12** Audit Record Generation

**FIPS 199 Security Categorization:**

**LOW** = (Confidentiality: LOW, Integrity: LOW, Availability: LOW)

## Ensure the risk profile tag is in audit log

**Purpose:**

Detect and analyze behavior

**NIST CSF 1.1 Subcategories:**

**DE.CM-7** Monitoring for unauthorized personnel, connections, devices, and software is performed

**NIST 800-53r5 Controls:**

**AU-3** Content of Audit Records

**FIPS 199 Security Categorization:**

**MODERATE** = (Confidentiality: LOW, Integrity: MODERATE, Availability: LOW)

**Potential Impact of Failure:**

* **Lack of Visibility**: Without risk profile tags, it becomes challenging to discern the significance of an event or action in the audit logs related to an asset.
* **Delayed Incident Response**: In the absence of clear risk profile information in the logs, security incidents concerning high-risk assets might be treated with the same urgency as low-risk ones, leading to potential delays in response.
* **Inefficient Forensics**: During a post-incident analysis, the lack of a risk profile tag can make it difficult to determine the scope and impact of a security breach.

**Procedural Steps:**

1. Review Current Logging Standards:
2. Examine existing logging practices to determine if risk profile tags are currently incorporated in S3 datasets.
3. **HRO Check Current State**: Verify if current logging standards include risk profile tags. Ensure consistency across all datasets.
4. Tag S3 Objects with Risk Profile:
5. **HRO Proficiency Test**: Demonstrate your capability to add and modify risk profile tags on S3 objects.
6. **HRO Reminder Prompt**: "Remember to validate the presence of risk profile tags in the audit logs. Ensuring these tags are consistently applied helps in promptly identifying and responding to potential threats related to our assets."
7. For the S3 buckets containing the dataset under consideration, use the *PutObjectTagging* API operation to add risk profile tags to objects.
8. Periodically review and update the risk profile tags, especially if there's a change in the dataset's content or its risk categorization.
9. Enable AWS CloudTrail for S3:
10. If not already enabled, activate AWS CloudTrail to audit S3 bucket events. Ensure that CloudTrail is specifically tracking the S3 bucket containing your dataset.
11. Configure CloudTrail to Log Risk Profile Tag:
12. Adapt event selectors in CloudTrail to capture risk profile metadata tags associated with S3 objects. Ensure that these tags are appropriately reflected in the logs.
13. Monitor and Analyze Logs with Risk Profile:
14. Utilize services like Amazon CloudWatch or AWS Athena to scrutinize the CloudTrail logs, emphasizing the risk profile tags. Establish alarms or notifications for any anomalies or unauthorized actions tied to high-risk S3 assets.
15. **HRO Operator Validation**: Verify and ensure that the risk profile tags are captured accurately in the audit logs and match with the predefined risk categories.
16. **HRO Watch Team Validation [required]**: Cross-check the CloudTrail logs to confirm the presence and accuracy of risk profile tags. Document any discrepancies.
17. Periodic Review and Validation:
18. Conduct routine audits of the CloudTrail logs to verify the presence and accuracy of the risk profile tags. Make updates as required.
19. **HRO Request Feedback**: After the audit, gather insights from the team on the effectiveness of the risk profile tagging process. Use the feedback to refine and enhance the procedure.

## Ensure a time-stamp is in the audit log

**Purpose:**

Detect and analyze behavior

**NIST CSF 1.1 Subcategories:**

**DE.CM-7** Monitoring for unauthorized personnel, connections, devices, and software is performed

**NIST 800-53r5 Controls:**

**AU-3** Content of Audit Records

**FIPS 199 Security Categorization:**

**MODERATE** = (Confidentiality: LOW, Integrity: MODERATE, Availability: LOW)

**Potential Impact of Failure:**

* **Loss of Accountability**: Without accurate timestamps, it becomes difficult to determine when specific actions or events took place. This makes it challenging to hold individuals or processes accountable for their actions, as there's no clear timeline of events.
* **Impediments to Forensic Analysis**: For security incidents or data breaches, a forensic analysis is crucial to understand the scope, impact, and root cause. Without accurate timestamps, reconstructing the sequence of events becomes difficult, potentially leading to inaccurate conclusions.
* **Increased Response Time**: During a security incident, timely response is essential to limit damage. If incident responders cannot quickly understand the sequence of events due to missing or inaccurate timestamps, it can delay the response, leading to increased damage or exposure.

**Procedural Steps:**

1. Enable AWS CloudTrail:
2. Ensure that CloudTrail is enabled for your AWS account. CloudTrail captures AWS API call logs, which include timestamps for every event.
3. **HRO Check Current State**: Validate if CloudTrail is active and capturing events with accurate timestamps for the AWS account.
4. Specify S3 Bucket for Logs:
5. **HRO Proficiency Test**: Demonstrate the ability to specify and configure the S3 bucket where CloudTrail logs are stored.
6. **HRO Reminder Prompt**: "Ensure that the designated S3 bucket for CloudTrail logs is correctly configured to capture and retain all logs with precise timestamps."
7. Define the S3 bucket where you want CloudTrail to store the logs.
8. Verify Log File Integrity:
9. Enable log file validation in CloudTrail. This feature ensures that the log files haven't been tampered with, which includes the accuracy of timestamps.
10. Periodically Review CloudTrail Logs:
11. Retrieve the logs using the GetTrailStatus API call or the DescribeTrails call to ensure that the CloudTrail is operational and delivering logs.
12. Review a sample of the logs to confirm that timestamps are present.
13. Set Up CloudWatch Alarm for Log Modifications:
14. You can create CloudWatch Alarms to notify you if there's any modification in the log files. This is an added step to ensure the integrity of your logs, including the timestamps.
15. Automate Verification:
16. Develop a script or lambda function that periodically fetches a sample of the logs using the AWS SDK or AWS CLI.
17. Parse the logs to verify the presence of the "eventTime" field, which represents the timestamp. If missing, trigger an alert.
18. Maintain Secure Access:
19. Ensure that only authorized personnel have access to CloudTrail settings and the S3 bucket where logs are stored to prevent any unintentional or malicious modifications.
20. Retention Policy:
21. Specify the retention period for your CloudTrail logs in S3 to ensure you keep logs for the required time frame for audit or compliance purposes.
22. **HRO Operator Validation**: Verify that the retention policy for CloudTrail logs is set up correctly and ensures logs are kept for the necessary compliance period.
23. **HRO Watch Team Validation [required]**: Review and ensure the CloudTrail log retention settings meet compliance requirements and logs contain accurate timestamps.
24. Document and Train:
25. Document the procedures and ensure relevant personnel are trained on the importance of timestamps in audit logs and how to check and interpret them.
26. Continuous Monitoring:
27. Consider using AWS-native or third-party solutions to continuously monitor and analyze CloudTrail logs for anomalies.
28. **HRO Request Feedback**: Seek feedback from stakeholders and operators on the effectiveness of timestamp monitoring in the CloudTrail logs. Identify areas for improvement.

## Ensure the type of access and user identity is in audit log

**Purpose:**

Detect and analyze behavior

**NIST CSF 1.1 Subcategories:**

**DE.CM-7** Monitoring for unauthorized personnel, connections, devices, and software is performed

**NIST 800-53r5 Controls:**

**AU-3** Content of Audit Records

**FIPS 199 Security Categorization:**

**LOW** = (Confidentiality: LOW, Integrity: LOW, Availability: LOW)

**Potential Impact of Failure:**

* **Accountability Loss**: Without proper logging of authorized access, it becomes challenging to trace actions back to specific users or entities, leading to potential accountability issues.
* **Compromised Integrity**: Given the moderate integrity classification, failure to log authorized access can jeopardize the dataset's trustworthiness. Unauthorized changes might go unnoticed.
* **Reduced Detection Capabilities**: If authorized accesses aren't logged, detecting unusual or unauthorized behavior becomes significantly harder, potentially leading to security incidents going unnoticed.
* **Audit and Compliance Risks**: Inability to present comprehensive audit logs during regulatory or internal audits can lead to non-compliance findings, penalties, or reputational damage.

**Procedural Steps:**

1. Enable AWS CloudTrail:
2. Ensure CloudTrail is enabled for your AWS account, capturing every AWS API call, which will include logs for both authorized and unauthorized access attempts.
3. **HRO Check Current State**: Verify that CloudTrail is actively logging all AWS API calls, specifically focusing on access and identity-related events.
4. Specify S3 Bucket for Logs:
5. **HRO Proficiency Test**: Demonstrate the ability to correctly specify and configure the S3 bucket to store CloudTrail access and identity logs.
6. **HRO Reminder Prompt**: "Before proceeding, ensure the designated S3 bucket for CloudTrail logs is configured correctly to retain all pertinent access and identity events."
7. Define an S3 bucket to store the CloudTrail logs.
8. Set Up CloudWatch Alarms:
9. Create alarms for specific events or API calls that are critical to be recorded. This way, if such an event doesn't get logged, you're instantly notified.
10. Review CloudTrail Logs:
11. Periodically retrieve the logs to confirm authorized accesses are being recorded. Look for the "eventName" and "eventSource" fields in the logs.
12. Ensure Granular IAM Policies:
13. Set fine-grained permissions for resources to ensure that authorized actions are explicitly defined, making it easier to monitor and record the right actions.
14. **HRO Operator Validation**: Ensure that IAM policies are granular and capture all authorized actions. Validate the presence of necessary "eventName" and "eventSource" fields in the logs.
15. **HRO Watch Team Validation [required]**: Periodically review and validate IAM policies and configurations to ensure precise access logging and to detect any deviations.
16. Log Archival:
17. Set up a policy to regularly archive old logs, ensuring that you retain logs for the required time frame for audit or compliance purposes.
18. Regular Audit and Review:
19. Periodically review the access logs and audit configurations to ensure everything remains as expected. Check for any gaps in logging or any unauthorized changes to configurations.
20. **HRO Request Feedback**: Gather feedback from users and operators on the efficacy and comprehensiveness of the logging mechanism. Address any areas of concern or potential improvements.
21. Document and Train:
22. Maintain documentation about the logging process, and train relevant personnel on its importance and the procedures to check and interpret logs.

## Ensure authorized access is recorded in audit log

**Purpose:**

Detect and analyze behavior

**NIST CSF 1.1 Subcategories:**

**DE.CM-7** Monitoring for unauthorized personnel, connections, devices, and software is performed

**NIST 800-53r5 Controls:**

**AU-3** Content of Audit Records

**FIPS 199 Security Categorization:**

**MODERATE** = (Confidentiality: LOW, Integrity: MODERATE, Availability: LOW)

**Potential Impact of Failure:**

* **Accountability Void**: Without a clear record of authorized access, tracing actions or changes to individual users or entities becomes challenging, if not impossible. This could make investigations, troubleshooting, or establishing responsibility in incidents problematic.
* **Integrity at Stake**: The integrity classification for the dataset is moderate, implying any compromise or oversight in logging can potentially degrade the dataset's reliability or trustworthiness.
* **Compromised Detection**: A full log of authorized access forms a baseline against which anomalous behaviors can be spotted. Absence of such a baseline weakens intrusion detection systems and makes spotting unauthorized access more challenging.
* **Regulatory & Compliance Risks**: If an organization is subject to external audits or regulations that demand rigorous logging, failure to log authorized access can lead to non-compliance, potential penalties, and reputational damage.

**Procedural Steps:**

1. Enable Logging for S3 Bucket:
2. Turn on server access logging for the desired S3 bucket. This will capture requests made to the bucket.
3. **HRO Check Current State**: Verify if server access logging is already enabled on the S3 bucket. If not, determine the steps needed to activate it.
4. Define Destination for Logs:
5. **HRO Proficiency Test**: Demonstrate the ability to correctly set a destination for access logs without affecting the integrity of existing data.
6. **HRO Reminder Prompt**: "Ensure the destination S3 bucket has adequate storage and is protected from unauthorized access or modifications."
7. Specify another S3 bucket where the access logs will be stored.
8. IAM Policy Review:
9. **HRO Reminder Prompt:** "Re-evaluate the IAM policies to confirm there are no overrides or exclusions that could prevent or interrupt log capturing."
10. Ensure that the AWS Identity and Access Management (IAM) policies for the S3 bucket permit logging and have no restrictions that might prevent it.
11. Periodic Log Validation:
12. Regularly review the access logs to verify that authorized accesses are being documented. This can be done using AWS management console or AWS CLI.
13. **HRO Request Feedback:** After reviewing, note any discrepancies or anomalies that may indicate logging issues or potential security concerns.
14. Implement Alarms:
15. Use Amazon CloudWatch to create alarms for specific log events to ensure critical events are recorded and any discrepancies are promptly reported.
16. **HRO Operator Validation:** Test the CloudWatch alarms to ensure they trigger accurately for defined events and that notifications are sent to the appropriate stakeholders.
17. Retention and Backup:
18. Define policies for log retention and ensure backups of logs are made for durability.
19. **HRO Watch Team Validation [required]:** Periodically verify that log backups are intact, unmodified, and can be restored if needed. Validate the adherence to retention policies.
20. Access Control for Logs:
21. Ensure that the S3 bucket where logs are stored has strict access controls to prevent tampering.
22. **HRO Operator Validation**: Check and validate the access controls on the S3 bucket to ensure they adhere to best practices and organizational policies
23. **HRO Watch Team Validation [required]**: Run periodic access simulations to ensure unauthorized attempts to tamper with or view logs are thwarted.

## Ensure unauthorized access is recorded in audit log

**Purpose:**

Detect and analyze behavior

**NIST CSF 1.1 Subcategories:**

**DE.CM-7** Monitoring for unauthorized personnel, connections, devices, and software is performed

**NIST 800-53r5 Controls:**

**AU-3** Content of Audit Records

**FIPS 199 Security Categorization:**

**MODERATE** = (Confidentiality: LOW, Integrity: MODERATE, Availability: LOW)

**Potential Impact of Failure:**

* **Unnoticed Security Threats**: Without logs of unauthorized access, malicious attempts might go unnoticed, allowing potential attackers to repeatedly try different strategies without detection.
* **Delayed Response**: Failure to promptly detect and respond to unauthorized access attempts could give attackers more time to discover vulnerabilities.
* **Compromised Analysis**: The absence of unauthorized access records compromises the ability to conduct a thorough post-incident analysis or forensic investigation.
* **Regulatory & Compliance Risks**: Non-compliance with regulations requiring rigorous logging of unauthorized access can lead to penalties, legal repercussions, and reputational harm.

**Procedural Steps:**

1. Enable Logging for S3 Bucket:
2. Turn on server access logging for the S3 bucket to capture all requests, including unauthorized ones.
3. **HRO Check Current State**: Verify if server access logging is active and configured to capture both authorized and unauthorized requests
4. Set Up CloudTrail:
5. **HRO Proficiency Test**: Demonstrate the ability to set up and configure CloudTrail effectively, ensuring all relevant API calls are captured.
6. **HRO Reminder Prompt**: "Cross-check if CloudTrail settings cover all resources and regions pertinent to the organization."
7. AWS CloudTrail provides more granular logging of API calls, which can capture unauthorized access attempts in more detail.
8. IAM Policy Review:
9. Regularly review AWS Identity and Access Management (IAM) policies to ensure that logging permissions are set correctly.
10. **HRO Reminder Prompt:** "Check for any recent changes to IAM policies that might affect logging capabilities or create unintended access loopholes."
11. Implement Alarms:
12. Use Amazon CloudWatch to establish alarms for specific unauthorized access patterns or frequent access denied events.
13. **HRO Operator Validation:** Test the alarms for specific unauthorized patterns to ensure they trigger accurately and that stakeholders are informed.
14. Anomaly Detection:
15. Utilize AWS GuardDuty or similar services to detect and alert on unusual behavior that might indicate unauthorized access attempts.
16. **HRO Operator Validation**: Simulate an unusual access pattern and verify if GuardDuty picks it up and sends an alert.
17. **HRO Watch Team Validation [required]**: Perform periodic checks to ensure the anomaly detection system is operational and adequately calibrated.
18. Review and Analyze Logs Regularly:
19. Periodically assess the logs to identify and analyze unauthorized access attempts. Tools like Amazon Athena can assist in querying S3 logs.
20. **HRO Request Feedback:** "After analysis, seek feedback on potential areas of improvement in detection, logging, or analysis procedures."
21. Limit Permissions:
22. Refine IAM permissions to ensure that only necessary personnel have access to the logs, protecting their integrity.
23. **HRO Request Feedback**: Solicit feedback from the team about the current access levels. Ensure they align with the principle of least privilege and that no one is granted unnecessary access.

# Ensure timely detection of anomalous activity against the S3 bucket is recorded and alerts the operator

**Purpose:**

Detect unauthorized access

**NIST CSF 1.1 Subcategories:**

**RS.AN-1** Notifications from detection systems are investigated

**NIST 800-53r5 Controls:**

**IR-5** Incident Monitoring

**FIPS 199 Security Categorization:**

**MODERATE** = (Confidentiality: LOW, Integrity: MODERATE, Availability: LOW)

**Potential Impact of Failure:**

* **Undetected Threats**: A delay or failure in detecting unusual activities can allow malicious actors to exploit vulnerabilities or extract sensitive data without being noticed.
* **Delayed Incident Response**: Without timely alerts, the response to potential security incidents can be delayed, leading to potentially more significant damage or data loss.
* **Operational Disruption**: Unauthorized access or anomalous activity might disrupt operations, causing potential downtime or service degradation.
* **Regulatory & Compliance Risks**: Inability to detect and respond to incidents in real-time can lead to non-compliance with industry regulations.

**Procedural Steps:**

1. Enable AWS CloudTrail:
2. Activate CloudTrail for the S3 bucket to get detailed logging of all bucket-related API activities.
3. **HRO Check Current State**: Confirm that CloudTrail is correctly set up and capturing all necessary data related to the S3 bucket. Review any recent changes or interruptions to CloudTrail logging.
4. Activate GuardDuty:
5. **HRO Proficiency Test**: Simulate a test scenario to ensure GuardDuty is active and properly configured to detect anomalies and unauthorized behaviors.
6. **HRO Reminder Prompt**: "Before completing the GuardDuty activation, cross-check its settings against best practices and review its integration with other AWS services."
7. AWS GuardDuty provides intelligent threat detection by continuously monitoring for malicious activity. Once activated, it can identify anomalies and unauthorized behaviors.
8. Set Up CloudWatch Alarms:
9. Using Amazon CloudWatch, configure alarms for specific patterns or metrics that might indicate anomalous activity on the S3 bucket.
10. **HRO Reminder Prompt:** "Ensure that the thresholds for CloudWatch alarms are set to provide a balance between sensitivity and avoiding too many false positives."
11. Integrate with Amazon SNS:
12. Link CloudWatch alarms with Amazon Simple Notification Service (SNS) to ensure that operators receive real-time alerts when anomalies are detected.
13. **HRO Operator Validation**: Simulate an anomaly to test the SNS alert mechanism, ensuring alerts are received promptly and contain actionable information.
14. **HRO Watch Team Validation [required]**: The watch team should independently verify the alerting mechanism and response times, ensuring consistency and reliability.
15. Regularly Review Logs:
16. Conduct periodic reviews of CloudTrail and GuardDuty findings to refine detection patterns and reduce false positives.
17. **HRO Reminder Prompt:** "Periodically revisit log reviews, especially after major infrastructure changes or incident resolutions, to ensure the system remains effective."
18. Feedback Loop:
19. As incidents are investigated, continuously refine and adjust the alarm parameters based on the findings to improve detection accuracy.
20. **HRO Request Feedback**: After responding to and resolving an incident, gather feedback from the incident response team and other stakeholders to understand what went well and where improvements can be made.

# Identify and document a process to quickly halt all write access to S3 buckets

**Purpose:**

Mitigation for compromised assets

**NIST CSF 1.1 Subcategories:**

**RS.MI-2** Incidents are mitigated

**NIST 800-53r5 Controls:**

**IR-4** Incident Handling

**FIPS 199 Security Categorization:**

**MODERATE** = (Confidentiality: LOW, Integrity: MODERATE, Availability: LOW)

**Potential Impact of Failure:**

* **Unauthorized Data Modification**: If malicious actors gain access, they might change or delete data, leading to potential data loss or misinformation.
* **Operational Issues**: Continual unauthorized modifications can disrupt regular operations and services.
* **Reputational Damage**: If the breach becomes public knowledge, the organization might suffer from damaged trust and reputation.
* **Increased Incident Response Time**: Without the ability to quickly halt access, the time taken to respond to and control an incident could increase.

**Procedural Steps:**

1. Document Bucket Identifiers:
2. Have a ready list of S3 bucket names/ARNs to which the process applies.
3. **HRO Check Current State**: Regularly review the list of S3 buckets to ensure that all relevant buckets are included and that obsolete ones are removed.
4. Prepare IAM Policies:
5. **HRO Proficiency Test**: Simulate a scenario where write access needs to be quickly restricted. Validate that team members can locate and apply the predefined IAM policies efficiently.
6. **HRO Reminder Prompt**: "Double-check the IAM policy's effectiveness by testing it against write actions before it's needed in a real situation."
7. Draft IAM policies that restrict write access to the buckets while retaining read access. Store these policies in a readily accessible location.
8. Automate Policy Application via AWS Lambda:
9. Set up a Lambda function that, when triggered, applies the restrictive IAM policies to the relevant S3 buckets.
10. **HRO Reminder Prompt:** "Before deploying the Lambda function, review its permissions to ensure it only has the required permissions and no additional ones."
11. Integrate with CloudWatch and SNS:
12. Use CloudWatch Events to detect specific conditions that might necessitate triggering the Lambda function (e.g., suspicious write activities or alerts from other AWS security services).
13. Link CloudWatch with SNS for notification purposes, ensuring stakeholders are alerted when the Lambda function is activated.
14. **HRO Operator Validation**: Regularly verify that the notifications from CloudWatch through SNS are promptly delivered and contain accurate, actionable information.
15. **HRO Watch Team Validation [required]**: The watch team should independently validate the entire workflow, from CloudWatch event detection to the application of restrictive IAM policies.
16. Test Periodically:
17. Regularly test the Lambda function and policy application process to ensure it works as intended and that there are no unexpected side effects.
18. **HRO Reminder Prompt:** "After any major AWS or infrastructure changes, or at predefined intervals, revisit and test this process to ensure continued reliability."
19. Document Manual Override:
20. Prepare a documented procedure for administrators to manually apply the restrictive policies, in case automated systems fail.
21. **HRO Reminder Prompt:** "In any scenario where the automated system might fail, ensure that key personnel are well-versed with the manual override process."
22. Recovery Process:
23. Document steps to investigate the incident, ensure the threat is removed, and safely restore write access when deemed appropriate.
24. **HRO Request Feedback**: After every incident or test scenario, collect feedback from involved personnel to identify any hiccups in the process and areas for improvement.

# Ensure regular updates and verification of compliance of this runbook

**Purpose:**

Ensure control gaps can be filled quickly

**NIST CSF 1.1 Subcategories:**

**RS.IM-1** Response plans incorporate lessons learned

**NIST 800-53r5 Controls:**

**CA-2** Control Assessments

**FIPS 199 Security Categorization:**

**LOW** = (Confidentiality: LOW, Integrity: LOW, Availability: LOW)

**Potential Impact of Failure:**

* **Outdated Response Mechanisms**: If the runbook isn't updated, it might not effectively address new threats, vulnerabilities, or scenarios.
* **Missed Opportunities for Improvement**: Without periodic verification, the organization might overlook gaps in compliance, leading to potential security incidents.
* **Decreased Efficiency**: Failing to incorporate lessons learned can result in repeated mistakes and longer incident response times.
* **Reputational Risk**: An outdated or non-compliant runbook can lead to potential breaches, which might impact the organization's reputation.

**Procedural Steps:**

1. Schedule Regular Reviews:
2. **HRO Proficiency Test**: During the review, team members should be tested on their understanding and application of the runbook procedures to ensure they're not just following steps blindly but understand their significance.
3. **HRO Reminder Prompt**: "Ensure reminders are sent out a month, a week, and a day in advance to ensure key stakeholders are available and prepared for the review session."
4. Calendar bi-annual or annual runbook review sessions.
5. Establish Review Committee:
6. Include cybersecurity experts, IT personnel, and other relevant stakeholders to ensure diverse perspectives.
7. **HRO Reminder Prompt:** "Review the committee's composition yearly to ensure it reflects the current needs and expertise required for the organization."
8. Collect Feedback:
9. Gather input from team members who use the runbook in real-world scenarios. Document lessons learned from incidents, drills, or exercises.
10. Consult External Sources:
11. Stay updated on emerging threats, vulnerabilities, and best practices. Consider insights from organizations like NIST.
12. **HRO Check Current State:** Periodically validate that the runbook's procedures align with the current threat landscape and evolving best practices.
13. Update Runbook:
14. Integrate feedback and external insights to enhance the runbook.
15. **HRO Operator Validation:** Once updates are implemented, have operators validate that the changes are clear, actionable, and feasible.
16. Conduct Compliance Checks:
17. Use automated tools, checklists, or external audits to verify alignment with security standards and best practices.
18. **HRO Reminder Prompt:** "After any significant organizational change or major incident, ensure a compliance check is performed."
19. Document Changes:
20. Keep a changelog, detailing what changes were made, the reasons for such changes, and the date of modification.
21. **HRO Check Current State:** "Periodically review the changelog to ensure it provides a clear audit trail and that no significant changes have been omitted."
22. Communicate Updates:
23. Ensure that all relevant stakeholders are aware of changes to the runbook.
24. **HRO Watch Team Validation [required]:** The watch team should review communications for clarity and ensure they reach all necessary personnel.
25. Training & Drills:
26. Ensure that staff members are trained on new procedures, and conduct drills to test the efficacy of the updated runbook.
27. **HRO Proficiency Test:** Post-training, test team members on their understanding of changes to gauge the effectiveness of the training.
28. Review:
    1. **HRO Operator Validation**: Regularly gather feedback from operators post-incident to validate the runbook's effectiveness in real-world scenarios.
    2. **HRO Watch Team Validation [required]**: The watch team should periodically review the entire runbook, from procedural steps to the underlying rationale, ensuring consistency and effectiveness.