AWS Account Audit (Sample)

May 2021

v1.0.0

Prepared for SafeCo, a high-profile SMB

Auditors
Stephen Kuenzli, k9 Security

Table of Contents

Executive Summary

This report identifies issues, risks, and recommendations during the IAM audit of the SafeCo production AWS account (`123456789012`) for technical management and staff.

The AWS Identity and Access Management (IAM) service manages access to all resources and capabilities within an AWS cloud environment. AWS customers configure IAM specifically for the workloads and operational processes of each account, and there's a lot that can go wrong, despite best efforts.

SafeCo engaged k9 Security's Professional Services team to audit the environment and identify the greatest risks and make recommendations to remediate those risks. This audit is not meant to be exhaustive, rather it focuses on the most important issues to address in the first round of improvement.

Key Issues & Recommendations

These are the three most important issues we found with the AWS IAM configuration of the SafeCo AWS account.

Excess IAM administrators
There are **8** IAM principals with IAM administration capabilities that are unnecessary for operations and application delivery (details).

Recommendation: Confirm those capabilities are not needed and decommission access to IAM APIs.

Excess privileges to critical data sources

There are **4** applications whose critical data is accessible by **+100** IAM principals who do not need it (details). Those data resources include five S3 buckets and one RDS database, containing both PCI and customer PII data.

Recommendations:

1. Implement least privilege S3 bucket policies using k9's infrastructure code libraries for all `Confidential` or `Restricted` data sources.
2. Confirm RDS administration capabilities are not needed and decommission access.

External access to account

There are 3 privileged IAM identities accessible via another AWS account (details). Two of those identities are no longer needed, and the other's accessibility should be narrowed.

Recommendations:

1. Decommission the two unneeded identities.
2. Create a fine-grained trust policy on `safeco-prod-cicd` allowing only the intended role in the Identity account to assume it. Verify the trusted Identity account implements appropriate access controls.

Next Steps

k9 Security recommends that you address the excess administrators and external access findings immediately. Remediate excess privileges to critical data next.

We also recommend that you operationalize this analysis by executing an audit of your AWS IAM security posture monthly using the k9 Security access reports and katas.

k9 Security Professional Services can assist with these efforts by supporting analysis, training, or leading improvement. We are available to review reports for your production and security critical accounts on a quarterly basis.

Introduction

This report summarizes security issues, characterizes risks, and recommends improvements identified in the IAM audit of the SafeCo production AWS account (`123456789012`) for technical management and engineering staff.

The AWS Identity and Access Management (IAM) service manages access to all resources and capabilities within an AWS cloud environment. AWS customers configure IAM specifically for the workloads and operational processes of each account. AWS IAM is very powerful and very flexible, a combination that frequently results in misconfiguration.

SafeCo engaged k9 Security's Professional Services team to audit the environment and identify the greatest risks and make recommendations to remediate those risks.

This report answers these critical questions:

1. Who can administer IAM and thus has full control of the account and its data?
2. Who can access this AWS account from another AWS account or publically? (a cross-account pivot) What access do they have in this account?
3. What excess privileges to critical data sources do IAM identities have?
4. Who can decrypt data in this account?
5. Which identities are unused and should be analyzed for decommissioning?

We explain how to integrate these answers into your information security, risk management, and product development processes.

Each section:

- Provides context about why the question and control is important
- Summarizes the issues found and illustrate with specific examples
- Assesses the risk for fine-tuning by SafeCo
- Recommends remediations
- Refers to materials that will help you operationalize this control within SafeCo

Reduce excess IAM Administrators

Reducing excess IAM administration capabilities is the first AWS access audit and improvement step. This is because any IAM user or role who can administer the AWS Identity and Access Management (IAM) service can give themselves privileges to do anything in the account if they don't already have it:

- create or destroy compute & data resources
- read, write, delete data
- run applications and scripts

It is common to find 5-15 excess administrators in AWS accounts that have been used for more than 5 years.

Findings

k9 Security found **17** IAM users or roles with the ability to administer IAM configurations within the account. k9 classifies an IAM user or role as an administrator if they can:

- create or delete IAM users, roles, and groups
- create, modify, attach, detach IAM policies

You can identify the IAM administrators yourself by executing k9 Security Kata 1.

See Appendix - IAM Administrators for a complete list.

Discussion
Automated delivery processes, operations teams, and security teams usually need an IAM principal with IAM administration capabilities. This capability often varies by environment. For example, you may allow application development teams to administer IAM in a development environment, but not production.

*AWS accounts managed with automation for infrastructure and applications typically need fewer than **10** IAM identities with IAM administration capabilities.*

Further, we found that 6 of the administrators had not been used recently:

1. Role-1
2. Role-2
3. Role-3
4. User-A, has two active, but unused, API keys

5. User-B, no keys
6. User-C, has one inactive API key

Risks

The risks of abuse or accident with any of the **17** IAM identities with administrator privileges cannot be overstated. The risks include loss of SafeCo data in the account, extortion for control of the account, and abuse of compute resources to mine cryptocurrency.

We estimate the expected loss range from one such event to be:

- likely minimum (5th-percentile) $250k USD
- likely maximum (95th-percentile) $8M USD

The annual likelihood for this class of event is difficult to estimate, but likely between 1% to 10% per year for a high-profile SMB like SafeCo.

These estimates were calibrated using brief discussions with Customer’s engineering team and publicly available industry data.

Recommendations

k9 Security recommends SafeCo *immediately*:

1. Review the unused IAM principals with administrator access to verify they are not needed and remove the excess privileges by updating or detaching policies
    a. If they are needed for a “break glass” recovery process or another activity that is infrequent, then that process should be executed regularly so that you know it works.
2. Review the IAM administrators that are in use and determine if the underlying activity can be used with a less privileged role or a session boundary policy applied to it in common usage.

Reduce external access to account

AWS customers frequently provision access to resources such as S3 buckets or IAM roles to external partners and other AWS accounts in the customer's organization.

AWS calls granting access to an identity outside of a given AWS account as granting external access because the account is AWS' fundamental security partition.

External access may be granted to:

- All AWS accounts (effectively public)
- A named AWS account or set of accounts
- IAM entities within another AWS account; these may be identified specifically by name, with wildcards, or other IAM policy conditions

External access is both frequently necessary and accidentally misconfigured.

You can identify these excess privilege problems yourself by executing the k9 Security Kata 4 and focusing on IAM roles that are accessible by a principal outside of the account.

Findings

Importantly, this analysis found no IAM roles that were effectively public, which is a common accident with serious results.

Several IAM roles with external access require remediation. 3 privileged IAM roles may be assumed solely at the discretion of SafeCo's Identity account, `234567890123`

A single (intentionally) public bucket was identified: `safeco-dt-status-page`

Discussion

IAM Roles
3 privileged IAM roles may be assumed solely at the discretion of the customer's Identity account, `234567890123`:

- `safeco-prod-cicd`
- `safeco-prod-admin`
- `temp-incident-2019-03`

An ‘Identity account’ is a common pattern to centralize access management to IAM principals used by internal services.

After speaking with the SafeCo engineering team, we concluded:

`safeco-prod-cicd` should remain accessible via the Identity account, but should use an IAM Trust policy that only permits the Identity account’s `safeco-cicd` role to assume it. This prevents an accidentally over permissioned principal in the Identity account from being able to assume `safeco-prod-cicd` in prod.

`safeco-prod-admin` should *not* be accessible via the Identity account any more. That access is now granted by SafeCo’s SSO, but the Identity account path was not decommissioned.

`temp-incident-2019-03` was provisioned to respond to an incident. The role is no longer used and should be decommissioned.

S3 Buckets
We did identify one world-readable bucket, `safeco-dt-status-page`. Public read access was granted intentionally to serve a downtime status website directly from S3.

There is no material risk to this bucket’s data. However, the recommended way to serve a static website from S3 is to use a CloudFront distribution.

When you front an S3 bucket with CloudFront, you only need to allow the AWS CloudFront service role to read the bucket instead of everyone.

This allows you to enable S3’s ‘Block public access’ feature within the account to protect all the other buckets from accidental misconfiguration.

Risks

The risks of these findings are included in the ‘Excess IAM administrators’ risk estimate.

Recommendations

First, implement the recommendations for improving access via the Identity account.

Second, serve the downtime status website via CloudFront, then enable S3’s block public access feature.

Reduce excess privileges to critical data sources

Reducing excess privileges to administer, change, and read capabilities critical data sources is the next AWS access improvement step. Data is a precious organizational asset. IAM principals with excess privileges to administer, change, or read data resources creates a latent risk of accident or abuse (c.f. Appendix - Excess Privilege Abuse).

k9 Security worked with the Customer team to identify critical data resources within the account. Those data resources are now tagged with two critical pieces of information per the Customer's Tagging Standard:

- Confidentiality: identifying that expected confidentiality of the data, 'critical' data is either Confidential or Restricted
- `Application`: identifying the application the resource belongs to

With application and confidentiality metadata in place, we reviewed the k9 access audit report to identify critical data sources that are accessible by more than the application principals to which the data belongs.

You can identify these excess privilege problems yourself by executing the k9 Security Kata 4.

Excess privileges to data are a very common problem in AWS because (details):

- Writing least privilege security policies manually is difficult and time consuming
- Using AWS Managed Policies grants access capabilities to all resources within the account
- AWS policy language and evaluation model is very powerful, flexible, and complex

The following excess privilege findings are unsurprising for a well-used account.

Findings

We identified **68** unique identities with excess privileges to **four** Customer applications with critical data.

Discussion

Customer Profile

The Customer Profile application has 3 S3 buckets and one RDS database whose data is accessible by IAM principals that do not need access.

+45 IAM identities with excess capabilities to read, write, and delete data in these buckets:

1. safeco-prod-customer-profile
2. safeco-prod-customer-profile-bak

Tokenization

There are **24** IAM identities with administration capabilities to the `safeco-pci-tokens` RDS database cluster, which contains PCI data. Those excess privileges could be used to:

- modify or delete the DB cluster and its associated data
- Export a snapshot the data to an attacker controlled account

Cart

The Cart application stores copies of customer orders in safeco-customer-orders. There are **45** IAM identities with excess privileges to read and write data in the `safeco-customer-orders` bucket.

Backups

63 IAM identities have excess privileges to read database and other backup data in the safeco-prod-backups S3 bucket.

Risks

An abuse or accident with any of the **68** IAM identities identified with excessive access could result in breach or destruction of critical SafeCo data.

We estimate the expected loss range from one such event to be:

- likely minimum (5th-percentile) $100k USD
- likely maximum (95th-percentile) $6M USD

The annual likelihood of an application breach event for a high-profile SMB is estimated to be 5% to 25% per year for a high-profile SMB. When that application uses an IAM identity with excess privileges to critical data, the critical data is put at risk unnecessarily.

These estimates were calibrated using brief discussions with Customer's engineering team and publicly available industry data.

Recommendations

k9 Security recommends SafeCo:

3. Implement least privilege S3 bucket policies using k9's infrastructure code libraries for all `Confidential` or `Restricted` data sources.

4. Confirm excess RDS capabilities are not needed and decommission that access to RDS APIs.

Reduce excess privileges to encryption keys

Reducing excess privileges to administer encryption keys or read encrypted data helps you achieve compliance and actual information security in AWS.

AWS Key Management Service (KMS) is a fully managed encryption API and key vault. KMS exposes encryption operations like encrypt, decrypt, sign, and verify as APIs for use by your applications and by other AWS services. When you encrypt data in AWS data services like S3 or SQS, KMS is the service that safely performs those operations on your behalf. KMS integrates with many AWS services, currently 66.

Most engineers' first encounter with KMS is to satisfy a requirement to encrypt data at rest, but you can and often should take it much further.

You can control access to data in your account by encrypting data in a particular data domain with a dedicated KMS Customer Managed CMK (key) in your account (details). Then manage which encrypt and decrypt operations an IAM identity can invoke using a resource policy applied to that domain-specific key.

By contrast, you cannot manage access to the 'default' AWS Managed CMKs used by AWS services within your account. AWS Managed Keys may be used by any IAM identity in the account to encrypt or decrypt data and do not create a meaningful access control boundary.

Consequently, 'who can access encrypted data in our account?' breaks down into a few questions:

1. Which keys are used to encrypt critical data sources?
2. Who has access to Customer Managed keys encrypting critical data?
3. Who has access to AWS Managed keys encrypting critical data?

We answered these questions by determining which encryption keys each application with critical data used, and then analyzed access to each of those keys.

Findings

We identified **65** unique identities with excess privileges to the four Customer applications with critical data analyzed previously.

Discussion

Customer Profile
The Customer Profile application encrypts data stored in its 3 S3 buckets and one RDS database.

Data in the S3 buckets is encrypted with the 'default' AWS Managed Key for S3 in the us-east-1 region. The RDS database cluster also uses the default AWS Managed Key for RDS.

This means the Customer Profile data in S3 and RDS can be decrypted by any of the **65** IAM identities that have access to the underlying S3 objects and RDS snapshots.

Tokenization
The Tokenization application encrypts its data with a domain-specific key, alias `safeco-prod-tokens`. Access to this key is well-controlled.

There are **8** IAM identities with the capability to administer or read data capabilities to the `safeco-prod-tokens` RDS database cluster data (volume and snapshots). Analysis identified only a single principal with excess access to read the encrypted data. The other 7 principals' access is appropriate.

Cart
The Cart application encrypts order objects in S3 with a key used by several applications in the ecommerce domain, key id `abcd-1234`. There are **17** IAM identities with excess privileges to read order data encrypted with this key. Most of these identities are used by ecommerce applications and teams. Four of the identities are from outside the ecommerce domain and have no clear purpose for decrypting ecommerce or order data.

Backups
Backups are a special case. By SafeCo policy, backups are encrypted with the key of the source data. Backup processes have access to encrypt & decrypt as appropriate. We verified the backup principals could encrypt and decrypt with the keys used by the Customer Profile, Tokenization, and Cart applications.

Risks

The risk of excess access to encryption keys is similar to that of excess privileges to critical data sources.

An abuse or accident with any of the **65** IAM identities identified with excessive access to encryption keys could result in exfiltration and exposure of critical SafeCo data.

Importantly:

- access to the Tokenization key and the data it protects is well-controlled
- Customer Profile data is most vulnerable

We estimate the expected loss range from one such event to be:

- likely minimum (5th-percentile) $50k USD
- likely maximum (95th-percentile) $2M USD

The annual likelihood of an application breach event for a high-profile SMB is estimated to be 5% to 25% per year for a high-profile SMB.

These estimates were calibrated using brief discussions with Customer's engineering team and publicly available industry data.

Recommendations

Secure data in AWS with KMS, describes how you can partition data domains with KMS and control access to that data with KMS resource policies.

Controlling access to data using KMS encryption keys is particularly useful when:

1. Encryption is required by regulation or policy
2. Data services that do not support resource policies directly, particularly RDS
3. A shared data container is in use, such as with Backups

k9 Security recommends:
First, remove unneeded access to the Ecommerce key, key id `abcd-1234`.

Second, review the Customer Profile application's goals for encryption. Determine which data sources, if any, should be migrated to a domain-specific CMK. Migrating the RDS database to a dedicated CMK should be relatively easy. Migrating to a dedicated CMK for S3 objects in its 3 buckets may be possible with bucket-level configuration or may involve application changes and re-encrypting objects in the bucket. k9 Security is happy to advise.

Decommission Unused Principals

Unused principals expand the attack surface of an AWS account needlessly. These principals provide no utility to the organization and an attacker that gains access may abuse them.

It is common to find tens or even hundreds of unused IAM principals in AWS accounts that have been used for more than 5 years.

You can identify unused principals yourself by executing k9 Security Kata 2.

Findings

This AWS account has 192 IAM principals total, including both users and roles. A significant portion of these principals are inactive:

	Users	Roles	**Total**
Inactive 90 days	8	35	**43**
Inactive 365 days	6	32	**38**

There are **43** IAM principals that have not been used in the past 90 days. **38** of those principals have not been used in the past 365 days.

To be considered an active, the principal must be:
- IAM user used via the AWS console or an AWS API Access Key
- IAM role used to invoke an API action

Discussion
Security best practice and many information security regulations require disabling or decommissioning principals that are inactive for more than 90 days.

As identified in reduce excess IAM Administrators and reduce external access to account, some of these principals have privileged access to the AWS account or data.

Cleaning up unused IAM principals is an essential activity of AWS account hygiene, as is cleaning up unused Cloud deployments generally.

Some principals have no or infrequent usage by design: 'break glass' administrators, certain types monitoring, or trial of a new AWS service. Many unused principals will require some investigation prior to removal. Record the owner and what application uses the principal as tags for principals that need to stick around.

Risks

An attacker with a foothold in the account with a principal that can assume a role can escalate their privileges by assuming most of the 35 unused roles. This enables an abuse of those privileges.

Recommendations

You can decommission IAM principals in three steps

1. Disable access to the principal
2. Verify the principal is unused
3. Delete the principal

Unused IAM Users are generally the greater concern because they are designed to be used 'externally' after authenticating with a credential.

Use the steps below to decommission a principal.

Users

1. Deactivate all credentials for unused users by deactivating all AWS API keys and resetting the user's password.
2. Check with the principal's owner to verify the user is no longer needed
3. Delete the IAM user

Roles

Unused IAM Roles.

1. Apply a Trust Policy to the unused role that denies everyone the ability to assume the role (`sts:Assume*`)
2. Check with the principal's owner to verify the user is no longer needed
3. Delete the IAM role

Appendix - IAM Administrators

Principal Name	Principal Arn	Principal Type	Principal Last Used
ci	arn:aws:iam::123456789012:user/ci	IAMUser	2021-04-30 22:53:00
training	arn:aws:iam::123456789012:user/training	IAMUser	2019-10-07 21:21:41
AccountAdminAccessRole-Sandbox	arn:aws:iam::123456789012:role/AccountAdminAccessRole-Sandbox	IAMRole	2021-02-15 18:19:43
AWS-CodePipeline-Service	arn:aws:iam::123456789012:role/AWS-CodePipeline-Service	IAMRole	
k9-dev-appeng	arn:aws:iam::123456789012:role/k9-dev-appeng	IAMRole	2021-05-03 22:13:06
MLSecWorkshopSageMakerRole	arn:aws:iam::123456789012:role/MLSecWorkshopSageMakerRole	IAMRole	

Appendix - Excess privilege abuse

The Excess Privileges to Critical Data Sources category identifies the most important link in the exploit path used in the 2019q3 CapitalOne breach where an attacker exploited a web application firewall and then used that process' excess privileges to exfiltrate critical, unrelated credit application data.