Case Study: AWS Multi-Account Security Audit & IAM Hardening (Anonymized)

Category: Cloud & Infrastructure Security
Duration: 3 Weeks | **Engagement Type:** AWS Security Posture Review
Tools: AWS Config, IAM Access Analyzer, ScoutSuite, GuardDuty, CloudTrail, AWS CLI

Context

A SaaS company managing multiple production, staging, and development AWS accounts wanted a **comprehensive security review** after noticing several unapproved IAM role assumptions across accounts.

With over **180 IAM roles**, **350 inline policies**, and shared access keys between environments, their growing architecture had outpaced its identity governance.

The client's primary objectives were to:

- Detect **over-permissive IAM policies** and **cross-account trust risks**.
- Identify **orphaned access keys**, **unrestricted S3 buckets**, and **potential lateral movement paths**.
- Implement **least privilege enforcement** across EC2, Lambda, and CI/CD systems.

Approach

The audit followed the **AWS Well-Architected Security Pillar** and **CIS AWS Foundations Benchmark v1.5** as the baseline.

1. **Environment Discovery & Mapping**
    - Collected IAM inventory and CloudTrail data across 4 AWS accounts using **AWS CLI + Boto3 scripts**.
    - Enumerated cross-account roles and policy attachments via **ScoutSuite**.
2. **Privilege Escalation Simulation**

    - Mapped potential `iam:PassRole` → `sts:AssumeRole` → `AdministratorAccess` chains.
    - Validated privilege inheritance and MFA enforcement gaps.
3. **Service-Level Review**
    - Audited **S3 ACLs**, **Lambda execution roles**, **KMS key policies**, and **EC2 role permissions**.
    - Checked CloudFormation stacks for inline policies granting wildcards (`"Action": "*"`, `"Resource": "*"`).
4. **Detection & Response Enablement**
    - Configured **AWS Config Aggregator** and **GuardDuty** across accounts for unified alerting.
    - Enabled **Access Analyzer** to continuously detect cross-account sharing anomalies.

Key Findings

Severity	Count	Highlight
Critical	3	IAM roles with `AdministratorAccess` trustable from external accounts
High	5	9 policies contained wildcard `"Action"` permissions
Medium	4	Unrotated access keys older than 180 days
Low	6	S3 buckets with public `GetObject` ACLs

Remediation Summary

- Implemented **least privilege policies** across core services (EC2, S3, Lambda).
- Enforced **MFA for all console users** and **rotation policy for access keys (90 days)**.
- Created **IAM policy boundaries** for staging accounts.

- Integrated **centralized CloudTrail logging** with encryption via **KMS CMK**.
- Enabled **GuardDuty + Config Aggregator** for continuous drift detection.
- Delivered **role-mapping matrix** to help DevOps teams align permissions per environment.

Outcome

- Reduced IAM exposure footprint by **~85%** across accounts
- Eliminated all privilege escalation chains
- Achieved full visibility into cross-account access patterns
- Established continuous compliance baseline using AWS Config and Access Analyzer
- Delivered documented security posture report aligned with CIS and AWS best practices

---

Executive Summary

The assessment uncovered systemic permission sprawl and cross-account trust misconfigurations that posed significant security risk.

Post-remediation, the client achieved a **principle-of-least-privilege environment** with active monitoring for deviations.

By automating audit checks with **ScoutSuite and Config Aggregators**, the organization can now sustain compliance without recurring manual reviews.

This engagement successfully bridged the gap between **identity security**, **governance**, and **operational efficiency**, enabling the team to scale securely.
