Dynamic Privilege Validation in AWS Environments via CloudTrail-Athena Analysis[*]

Soo-Min Nam[1], Sun-Woo Kwon[1] , Ye-Eun Shin[1] and Seongmin Kim[2†]

1 Undergraduate Student, Department of Convergence Security Engineering, Sungshin Women's University

2 Professor, Department of Convergence Security Engineering, Sungshin Women's University

{20231076, 20231061, 20220124, sm.kim}@sungshin.ac.kr

Abstract

This study empirically analyzed the relationship between permissions allowed by policyand those actually usedin order to address the issue of over-privileged accessin AWS environments. To achieve this, six representative scenarios were designed, and CloudTrail–Athena log analysiswas employed to validate privilege usage patterns. The analysis results demonstrated the necessity of reducing excessive privileges.This research proposes a Dynamic Privilege Validation Frameworkthat goes beyond static policy review and introduces a Permission Minimization Baselinefor IAM and Cognito environments, derived through real-world AWS experiments.

keywords - CloudTrail, Athena, IAM and Cognito, PoLP, Allowed-Only, Allowed-Used

1 Introduction

In cloud environments, over-privileged access remains a major security concern. In Amazon Web Services (AWS), administrators often assign overly broad IAM (Identity and Access Management) permissions for operational convenience, increasing the risk of potential breaches. To enforce the Principle of Least Privilege (PoLP) in practice, it is essential to analyze and validate actual permission usage patterns. This study empirically examines the gap between permissions granted by policies and those actually used across diverse authorization structures in AWS. We designed six representative experimental scenarios to quantify the risks of over-privileged configurations and demonstrate their security impact. Based on this analysis, we propose a *minimization baseline* applicable to both IAM and Cognito, contributing to the practical enhancement of cloud operational security.

2 Scenario Description

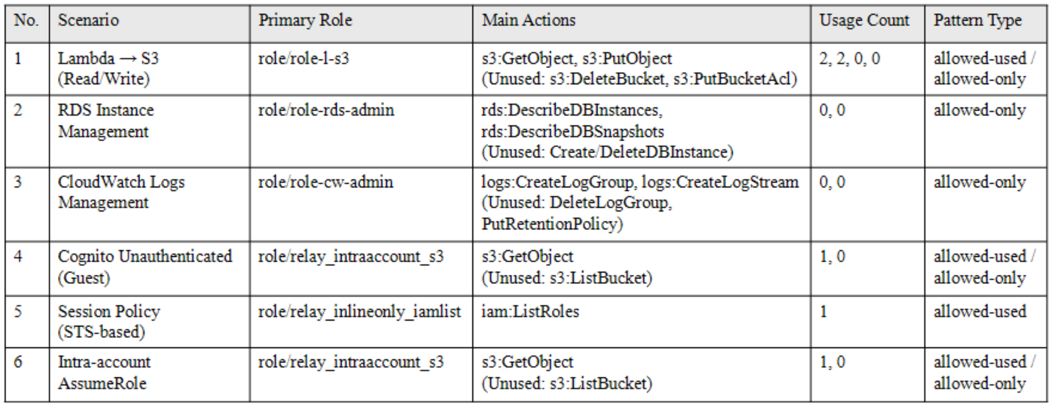
We designed six representative scenarios to examine privilege usage behaviors across different credential models, including IAM User, Cognito Federated Identity, and STS Session [1]. The scenarios were constructed around major AWS services, such as S3 bucket, Relational Database Service (RDS), and CloudWatch, to emulate real-world cases of over-privileged access and delegated authorization commonly observed in operational environments. All experiments were conducted

[*] Proceedings of the 9th International Conference on Mobile Internet Security (MobiSec'25), Article No. P-13, December 16-18, 2025, Sapporo, Japan.

[†] Corresponding author

using both the AWS Management Console and Command Line Interface (CLI), encompassing the entire process of IAM role creation, policy assignment, Lambda deployment, and STS session issuance in an actual AWS environment [2].

Scenarios 1–3 focused on detecting excessive permissions within individual services (Over-privilege Group), whereas Scenarios 4–6 validated delegated permissions in temporary credential environments (Delegation Group). Table 1 summarizes the scenarios.



No.	Scenario	Primary Role	Main Actions	Usage Count	Pattern Type
1	Lambda → S3 (Read/Write)	role/role-l-s3	s3:GetObject, s3:PutObject (Unused: s3:DeleteBucket, s3:PutBucketAcl)	2, 2, 0, 0	allowed-used / allowed-only
2	RDS Instance Management	role/role-rds-admin	rds:DescribeDBInstances, rds:DescribeDBSnapshots (Unused: Create/DeleteDBInstance)	0, 0	allowed-only
3	CloudWatch Logs Management	role/role-cw-admin	logs:CreateLogGroup, logs:CreateLogStream (Unused: DeleteLogGroup, PutRetentionPolicy)	0, 0	allowed-only
4	Cognito Unauthenticated (Guest)	role/relay_intraaccount_s3	s3:GetObject (Unused: s3:ListBucket)	1, 0	allowed-used / allowed-only
5	Session Policy (STS-based)	role/relay_inlineonly_iamlist	iam:ListRoles	1	allowed-used
6	Intra-account AssumeRole	role/relay_intraaccount_s3	s3:GetObject (Unused: s3:ListBucket)	1, 0	allowed-used / allowed-only

Table.1 Six realistic usage scenarios and credential models in AWS environment

We then empirically evaluated inter-service privilege propagation and potential privilege escalation risks. For this analysis, CloudTrail logs were collected for each scenario and examined using Amazon Athena, a serverless query service that enables SQL-based analysis of CloudTrail logs stored in S3 without additional preprocessing. This approach allowed for rapid and consistent examination of privilege usage patterns [3]. Using Athena, we compared the permissions granted in IAM policies with those actually invoked in execution logs and identified two primary patterns: *allowed-only* (permissions granted but not used) and *allowed-used* (permissions granted and used). The *allowed-only* pattern represents permissions that were defined in the policy but never appeared in CloudTrail logs, whereas *allowed-used* corresponds to permissions that were both defined and executed. Notably, *allowed-only* permissions can serve as key candidates for privilege minimization in enforcing the principle of least privilege.

Based on the Athena query analysis, Scenarios 1–3 (Lambda, RDS, and CloudWatch) exhibited a significant number of *allowed-only* patterns. Accordingly, permissions with no invocation records (*allowed-only*) within the 30-day observation window defined for this experiment were defined as candidates for permission minimization. API actions recorded with uses = 0 would either be removed from the operational policy or refined into more granular, resource-level permissions to enforce a stricter application of the PoLP. This study provides practical insights into applying the PoLP beyond conventional static reviews by incorporating dynamic, log-based validation, and proposes concrete, data-driven criteria for privilege minimization in real-world cloud environments.

References

[1] Amazon Web Services,“IAM Best Practices – AWS Identity and Access Management User Guide,” available at:https://docs.aws.amazon.com/IAM/latest/UserGuide/best-practices.html, accessed October 13, 2025.

[2] Choi, Y.-J., Park, H.-J., Kim, J.-Y., Kim, T.-Y., Shin, Y.-J., & Cha, W.-J. (2025). Template-Based Database Design and Validation for Dynamic Least Privilege Policy Generation in AWS.Journal of The Korea Institute of Information Security & Cryptology, 35(3), 493–504.

[3] Amazon Web Services,"What is Amazon Athena? – Amazon Athena User Guide," available at: https://docs.aws.amazon.com/athena/latest/ug/what-is.html, accessed October 13, 2025.