AWS IAM privilege escalation.

AWS IAM privilege escalation is when a low-level account turns itself into a powerful one, using only the permissions it already has. The paths are well known: a handful of AWS permissions, if handed to an account that should not have them, let that account give itself more power. The list of these paths was first documented in 2018, and the patterns have barely changed since.

HOW IT WORKS

01 What are the most common privilege escalation paths?

These are the paths we test on every AWS engagement. The action shown is the one that makes the escalation possible.

- `iam:PassRole` plus a service that runs code (Lambda, EC2, ECS, Glue): lets the attacker start a function or server that runs as a stronger role, then use that role freely.
- `iam:CreatePolicyVersion` on a shared policy: lets the attacker write a new version that grants more and make it the default. Everyone on that policy is now over-powered.
- `iam:AttachUserPolicy`, `iam:AttachRolePolicy`, `iam:AttachGroupPolicy`: lets the attacker attach a powerful policy like `AdministratorAccess` to themselves or a role they hold.
- `iam:PutUserPolicy`, `iam:PutRolePolicy`, `iam:PutGroupPolicy`: lets the attacker write an inline policy that grants themselves anything.
- `iam:UpdateAssumeRolePolicy` on a powerful role: lets the attacker rewrite who may assume that role, then assume it.
- `lambda:UpdateFunctionCode` on a Lambda that runs as a strong role: lets the attacker swap its code for code that steals its credentials.
- `ec2:RunInstances` with `iam:PassRole`: lets the attacker launch a server with a strong role attached and log into it.

HOW TO DEFEND

- Check for the known escalation actions. AWS IAM Access Analyzer flags identities that hold the risky actions. Reviewing those flags is an ongoing habit, not a one-off scan.
- Use permission boundaries. A permission boundary caps what an identity can do, whatever its policies say. Put one on every identity that other identities can change.
- Use Service Control Policies at the org level. Block risky actions like `iam:CreateUser` across the whole account.
- Keep sensitive identities apart. Deploy roles, IAM-admin roles, and root credentials belong in separate accounts, not all in one.
- Rotate access keys often, or drop them. Long-lived access keys are a top source of IAM compromise. Prefer assume-role and single sign-on.

SOURCES

[1] AWS IAM Access Analyzer

[2] AWS IAM Best Practices

[3] MITRE ATT&CK T1078.004 Cloud Accounts

- `glue:UpdateDevEndpoint`, `cloudformation:CreateStack`, `sagemaker:CreatePresignedNotebookInstanceUrl` and others: more obscure paths, same shape. Run code as a strong role through a service that accepts a role you can pass.

02 Why do these paths exist if least-privilege is the standard?

Three forces push against least-privilege every day.

- The default policies are broad. AWS-managed policies like `IAMFullAccess` grant more than most jobs need. Teams attach them to get going and never trim them.
- CI/CD logins collect permissions. A pipeline that deploys to many services gathers rights over time. Taking rights away is harder than adding them.
- Speed beats caution. A developer needs something done now, attaches `*:*`, plans to narrow it later, and never does.

03 How does SecureLayer7 test for IAM privilege escalation?

Every AWS engagement runs the IAM privilege-escalation matrix.

- Map the identities. Every user, role, and policy attachment in the account, including cross-account trust paths.
- Score each one. For each identity, we check which escalation actions it holds and whether a path to administrator exists.
- Prove the path. For each high-value path, we run it in a controlled way, with your permission, to show the jump from a weak identity to a strong one.
- Check the strong roles too. For high-privilege roles, we document what an attacker holding that role could actually reach.

The report includes a privilege-escalation map and a fix for each path.

Find the privilege-escalation paths before someone else does.

securelayer7.net/learn/cloud-security/iam-privesc-aws

Open online