

Product ▾ Solutions ▾ Resources ▾ Open Source ▾ Enterprise ▾ Pricing

Search

Sign in

Sign up

RhinoSecurityLabs / AWS-IAM-Privilege-Escalation Public

Notifications

Fork 118

Star 896

<> Code

Issues 1

Pull requests

Actions

Security

Insights

master ▾

<> Code ▾

7 Commits

.gitignore

LICENSE

README.md

README

BSD-3-Clause license

# Table of Contents

- [AWS IAM Privilege Escalation Methods](#)
- [Prior Research, References, and Resources](#)

# AWS IAM Privilege Escalation Methods

## 1. Creating a new policy version

### How-To/Exploit Link(s)

- <https://rhinosecuritylabs.com/aws/aws-privilege-escalation-methods-mitigation/>

### Description

An attacker with the `iam:CreatePolicyVersion` permission can create a new version of an IAM policy that they have access to. This allows them to define their own custom permissions. When creating a new policy version, it needs to be set as the default version to take effect, which you would think would require the `iam:SetDefaultPolicyVersion` permission, but when creating a new policy version, it is possible to include a flag ( `--set-as-default` ) that will automatically create it as the new default version. That flag does not require the `iam:SetDefaultPolicyVersion` permission to use.

### Required Permission(s)

- `iam:CreatePolicyVersion`

### Potential Impact

This privilege escalation method could allow a user to gain full administrator access of the AWS account.

## 2. Setting the default policy version to an existing version

### How-To/Exploit Link(s)

### About

A centralized source of all AWS IAM privilege escalation methods released by Rhino Security Labs.

Readme

BSD-3-Clause license

Activity

Custom properties

896 stars

33 watching

118 forks

Report repository

### Releases

No releases published

### Packages

No packages published

- <https://rhinosecuritylabs.com/aws/aws-privilege-escalation-methods-mitigation/>

### Description

An attacker with the `iam:SetDefaultPolicyVersion` permission may be able to escalate privileges through existing policy versions that are not currently in use. If a policy that they have access to has versions that are not the default, they would be able to change the default version to any other existing version.

### Required Permission(s)

- `iam:SetDefaultPolicyVersion`

### Potential Impact

The potential impact is associated with the level of permissions that the inactive policy version has. This could range from no privilege escalation at all to gaining full administrator access to the AWS account, depending on what the inactive policy versions have access to.

## 3. Creating an EC2 instance with an existing instance profile

### How-To/Exploit Link(s)

- <https://rhinosecuritylabs.com/aws/aws-privilege-escalation-methods-mitigation/>

### Description

An attacker with the `iam:PassRole` and `ec2:RunInstances` permissions can create a new EC2 instance that they will have operating system access to and pass an existing EC2 instance profile/role to it. They can then login to the instance and request the associated AWS keys from the EC2 instance meta data, which gives them access to all the permissions that the associated instance profile/role has.

### Required Permission(s)

- `iam:PassRole`
- `ec2:RunInstances`

### Potential Impact

This attack would give an attacker access to the set of permissions that the instance profile/role has, which again could range from no privilege escalation to full administrator access of the AWS account.

## 4. Creating a new user access key

### How-To/Exploit Link(s)

- <https://rhinosecuritylabs.com/aws/aws-privilege-escalation-methods-mitigation/>

### Description

An attacker with the `iam:CreateAccessKey` permission on other users can create an access key ID and secret access key belonging to another user in the AWS environment, if they don't already have two sets associated with them (which best practice says they shouldn't).

### Required Permission(s)

- `iam:CreateAccessKey`

### Potential Impact

This method would give an attacker the same level of permissions as any user they were able to create an access key for, which could range from no privilege escalation to full administrator access to the account.

## 5. Creating a new login profile

### How-To/Exploit Link(s)

- <https://rhinosecuritylabs.com/aws/aws-privilege-escalation-methods-mitigation/>

### Description

An attacker with the `iam:CreateLoginProfile` permission on other users can create a password to use to login to the AWS console on any user that does not already have a login profile setup.

### Required Permission(s)

- `iam:CreateLoginProfile`

### Potential Impact

This method would give an attacker the same level of permissions as any user they were able to create a login profile for, which could range from no privilege escalation to full administrator access to the account.

## 6. Updating an existing login profile

### How-To/Exploit Link(s)

- <https://rhinosecuritylabs.com/aws/aws-privilege-escalation-methods-mitigation/>

### Description

An attacker with the `iam:UpdateLoginProfile` permission on other users can change the password used to login to the AWS console on any user that already has a login profile setup.

### Required Permission(s)

- `iam:UpdateLoginProfile`

### Potential Impact

This method would give an attacker the same level of permissions as any user they were able to update the login profile for, which could range from no privilege escalation to full administrator access to the account.

## 7. Attaching a policy to a user

### How-To/Exploit Link(s)

- <https://rhinosecuritylabs.com/aws/aws-privilege-escalation-methods-mitigation/>

### Description

An attacker with the `iam:AttachUserPolicy` permission can escalate privileges by attaching a policy to a user that they have access to, adding the permissions of that policy to the attacker.

### Required Permission(s)

- iam:AttachUserPolicy

### Potential Impact

An attacker would be able to use this method to attach the AdministratorAccess AWS managed policy to a user, giving them full administrator access to the AWS environment.

## 8. Attaching a policy to a group

### How-To/Exploit Link(s)

- <https://rhinosecuritylabs.com/aws/aws-privilege-escalation-methods-mitigation/>

### Description

An attacker with the iam:AttachGroupPolicy permission can escalate privileges by attaching a policy to a group that they are a part of, adding the permissions of that policy to the attacker.

### Required Permission(s)

- iam:AttachGroupPolicy

### Potential Impact

An attacker would be able to use this method to attach the AdministratorAccess AWS managed policy to a group, giving them full administrator access to the AWS environment.

## 9. Attaching a policy to a role

### How-To/Exploit Link(s)

- <https://rhinosecuritylabs.com/aws/aws-privilege-escalation-methods-mitigation/>

### Description

An attacker with the iam:AttachRolePolicy permission can escalate privileges by attaching a policy to a role that they have access to, adding the permissions of that policy to the attacker.

### Required Permission(s)

- iam:AttachRolePolicy

### Potential Impact

An attacker would be able to use this method to attach the AdministratorAccess AWS managed policy to a role, giving them full administrator access to the AWS environment.

## 10. Creating/updating an inline policy for a user

### How-To/Exploit Link(s)

- <https://rhinosecuritylabs.com/aws/aws-privilege-escalation-methods-mitigation/>

### Description

An attacker with the iam:PutUserPolicy permission can escalate privileges by creating or updating an inline policy for a user that they have access to, adding the permissions of that policy to the attacker.

Required Permission(s)

- iam:PutUserPolicy

Potential Impact

Due to the ability to specify an arbitrary policy document with this method, the attacker could specify a policy that gives permission to perform any action on any resource, ultimately escalating to full administrator privileges in the AWS environment.

11. Creating/updating an inline policy for a group

How-To/Exploit Link(s)

- <https://rhinosecuritylabs.com/aws/aws-privilege-escalation-methods-mitigation/>

Description

An attacker with the iam:PutGroupPolicy permission can escalate privileges by creating or updating an inline policy for a group that they are a part of, adding the permissions of that policy to the attacker.

Required Permission(s)

- iam:PutGroupPolicy

Potential Impact

Due to the ability to specify an arbitrary policy document with this method, the attacker could specify a policy that gives permission to perform any action on any resource, ultimately escalating to full administrator privileges in the AWS environment.

12. Creating/updating an inline policy for a role

How-To/Exploit Link(s)

- <https://rhinosecuritylabs.com/aws/aws-privilege-escalation-methods-mitigation/>

Description

An attacker with the iam:PutRolePolicy permission can escalate privileges by creating or updating an inline policy for a role that they have access to, adding the permissions of that policy to the attacker.

Required Permission(s)

- iam:PutRolePolicy

Potential Impact

Due to the ability to specify an arbitrary policy document with this method, the attacker could specify a policy that gives permission to perform any action on any resource, ultimately escalating to full administrator privileges in the AWS environment.

13. Adding a user to a group

How-To/Exploit Link(s)

- <https://rhinosecuritylabs.com/aws/aws-privilege-escalation-methods-mitigation/>

Description

An attacker with the `iam:AddUserToGroup` permission can use it to add themselves to an existing IAM Group in the AWS account.

Required Permission(s)

- `iam:AddUserToGroup`

Potential Impact

The attacker would be able to gain privileges of any existing group in the account, which could range from no privilege escalation to full administrator access to the account.

14. Updating the AssumeRolePolicyDocument of a role

How-To/Exploit Link(s)

- <https://rhinosecuritylabs.com/aws/aws-privilege-escalation-methods-mitigation/>

Description

An attacker with the `iam:UpdateAssumeRolePolicy` and `sts:AssumeRole` permissions would be able to change the assume role policy document of any existing role to allow them to assume that role.

Required Permission(s)

- `iam:UpdateAssumeRolePolicy`
- `sts:AssumeRole`

Potential Impact

This would give the attacker the privileges that are attached to any role in the account, which could range from no privilege escalation to full administrator access to the account.

15. Passing a role to a new Lambda function, then invoking it

How-To/Exploit Link(s)

- <https://rhinosecuritylabs.com/aws/aws-privilege-escalation-methods-mitigation/>

Description

A user with the `iam:PassRole` , `lambda:CreateFunction` , and `lambda:InvokeFunction` permissions can escalate privileges by passing an existing IAM role to a new Lambda function that includes code to import the relevant AWS library to their programming language of choice, then using it perform actions of their choice. The code could then be run by invoking the function through the AWS API.

Required Permission(s)

- `iam:PassRole`
- `lambda:CreateFunction`
- `lambda:InvokeFunction`

Potential Impact

This would give a user access to the privileges associated with any Lambda service role that exists in the account, which could range from no privilege escalation to full administrator access to the account.

## 16. Passing a role to a new Lambda function, then invoking it cross-account

### How-To/Exploit Link(s)

- None

### Description

A user with the `iam:PassRole` , `lambda:CreateFunction` , and `lambda:AddPermission` permissions can escalate privileges by passing an existing IAM role to a new Lambda function that includes code to import the relevant AWS library to their programming language of choice, then using it perform actions of their choice. The code could then be run by using `lambda:AddPermission` to allow cross-account invocation, then invoking it cross-account with their own attacker account.

### Required Permission(s)

- `iam:PassRole`
- `lambda:CreateFunction`
- `lambda:AddPermission`

### Potential Impact

This would give a user access to the privileges associated with any Lambda service role that exists in the account, which could range from no privilege escalation to full administrator access to the account.

## 17. Passing a role to a new Lambda function, then triggering it with DynamoDB

### How-To/Exploit Link(s)

- <https://rhinosecuritylabs.com/aws/aws-privilege-escalation-methods-mitigation/>

### Description

A user with the `iam:PassRole` , `lambda:CreateFunction` , and `lambda:CreateEventSourceMapping` (and possibly `dynamodb:PutItem` and `dynamodb:CreateTable` ) permissions, but without the `lambda:InvokeFunction` permission, can escalate privileges by passing an existing IAM role to a new Lambda function that includes code to import the relevant AWS library to their programming language of choice, then using it perform actions of their choice. They then would need to either create a DynamoDB table or use an existing one, to create an event source mapping for the Lambda function pointing to that DynamoDB table. Then they would need to either put an item into the table or wait for another method to do so that the Lambda function will be invoked.

### Required Permission(s)

- `iam:PassRole`
- `lambda:CreateFunction`
- `lambda:CreateEventSourceMapping`
- `dynamodb:PutItem` (possibly)
- `dynamodb:CreateTable` (possibly)

### Potential Impact

This would give an attacker access to the privileges associated with any Lambda service role that exists in the account, which could range from no privilege escalation to full administrator access to the account.

## 18. Updating the code of an existing Lambda function

### How-To/Exploit Link(s)

- <https://rhinosecuritylabs.com/aws/aws-privilege-escalation-methods-mitigation/>

### Description

An attacker with the `lambda:UpdateFunctionCode` permission could update the code in an existing Lambda function with an IAM role attached so that it would import the relevant AWS library in that programming language and use it to perform actions on behalf of that role. They would then need to wait for it to be invoked if they were not able to do so directly, but if it already exists, there is likely some way that it will be invoked.

### Required Permission(s)

- `lambda:UpdateFunctionCode`

### Potential Impact

This would give an attacker access to the privileges associated with the Lambda service role that is attached to that function, which could range from no privilege escalation to full administrator access to the account.

## 19. Passing a role to a Glue Development Endpoint

### How-To/Exploit Link(s)

- <https://rhinosecuritylabs.com/aws/aws-privilege-escalation-methods-mitigation/>

### Description

An attacker with the `iam:PassRole` and `glue:CreateDevEndpoint` permissions could create a new AWS Glue development endpoint and pass an existing service role to it. They then could SSH into the instance and use the AWS CLI to have access of the permissions the role has access to.

### Required Permission(s)

- `iam:PassRole`
- `glue:CreateDevEndpoint`

### Potential Impact

This would give an attacker access to the privileges associated with any Glue service role that exists in the account, which could range from no privilege escalation to full administrator access to the account.

## 20. Updating an existing Glue Dev Endpoint

### How-To/Exploit Link(s)

- <https://rhinosecuritylabs.com/aws/aws-privilege-escalation-methods-mitigation/>

### Description

An attacker with the `glue:UpdateDevEndpoint` permission would be able to update the associated SSH public key of an existing Glue development endpoint, to then SSH into it and have access to the permissions the attached role has access to.

### Required Permission(s)



- glue:UpdateDevEndpoint

### Potential Impact

This would give an attacker access to the privileges associated with the role attached to the specific Glue development endpoint, which could range from no privilege escalation to full administrator access to the account.

## 21. Passing a role to CloudFormation

### How-To/Exploit Link(s)

- <https://rhinosecuritylabs.com/aws/aws-privilege-escalation-methods-mitigation/>

### Description

An attacker with the iam:PassRole and cloudformation:CreateStack permissions would be able to escalate privileges by creating a CloudFormation template that will perform actions and create resources using the permissions of the role that was passed when creating a CloudFormation stack.

### Required Permission(s)

- iam:PassRole
- cloudformation:CreateStack

### Potential Impact

This would give an attacker access to the privileges associated with the role that was passed when creating the CloudFormation stack, which could range from no privilege escalation to full administrator access to the account.

## 22. Passing a role to Data Pipeline

### How-To/Exploit Link(s)

- <https://rhinosecuritylabs.com/aws/aws-privilege-escalation-methods-mitigation/>

### Description

An attacker with the iam:PassRole , datapipeline:CreatePipeline , and datapipeline:PutPipelineDefinition permissions would be able to escalate privileges by creating a pipeline and updating it to run an arbitrary AWS CLI command or create other resources, either once or on an interval with the permissions of the role that was passed in.

### Required Permission(s)

- iam:PassRole
- datapipeline:CreatePipeline
- datapipeline:PutPipelineDefinition

### Potential Impact

This would give the attacker access to the privileges associated with the role that was passed when creating the pipeline, which could range from no privilege escalation to full administrator access to the account.

## 23. Creating a CodeStar project from a template

### How-To/Exploit Link(s)

- <https://rhinosecuritylabs.com/aws/escalating-aws-iam-privileges-undocumented-codestar-api/>
- [https://github.com/RhinoSecurityLabs/Cloud-Security-Research/blob/master/AWS/codestar\\_createprojectfromtemplate\\_privesc/CodeStarPrivEsc.py](https://github.com/RhinoSecurityLabs/Cloud-Security-Research/blob/master/AWS/codestar_createprojectfromtemplate_privesc/CodeStarPrivEsc.py)

Description

An attacker with the `codestar:CreateProjectFromTemplate` permission can leverage an undocumented CodeStar API to escalate privileges by creating a new CodeStar project from a built-in template. This method also allows arbitrary CloudFormation resource creation under a different set of privileges.

Required Permission(s)

- `codestar:CreateProjectFromTemplate`

Potential Impact

This would give the attacker access to the privileges associated with the CodeStar project template that was chosen, along with the permissions granted to the CloudFormation role created along with the project. This results in a reasonable amount of privilege escalation, with a chance to full-administrator, depending on other resources/permissions in the environment. More information can be found in the references section.

24. Passing a role to a new CodeStar project

How-To/Exploit Link(s)

- <https://rhinosecuritylabs.com/aws/escalating-aws-iam-privileges-undocumented-codestar-api/>

Description

An attacker with the `codestar:CreateProject` and `iam:PassRole` permissions can escalate privileges by creating a new CodeStar project and passing a role to it, where the role will then be used to deploy the resources specified in the CodeStar project.

Required Permission(s)

- `codestar:CreateProject`
- `iam:PassRole`

Potential Impact

This would give the attacker the ability to escalate to a full administrator, because the default CodeStar service role has permission to escalate privileges to an administrator. If a custom CodeStar service role has been created, the impact of this privilege escalation method may vary.

25. Creating a new CodeStar project and associating a team member

How-To/Exploit Link(s)

- <https://rhinosecuritylabs.com/aws/escalating-aws-iam-privileges-undocumented-codestar-api/>

Description

An attacker with the `codestar:CreateProject` and `codestar:AssociateTeamMember` permissions can escalate privileges by creating a new CodeStar project, then associating

themselves as the Owner of the project, which will attach an IAM policy to them.

Required Permission(s)

- `codestar:CreateProject`
- `codestar:AssociateTeamMember`

Potential Impact

This would give the attacker read-only access to multiple different AWS services and full CodeStar access on the project they are now an Owner of.

26. Adding a malicious Lambda layer to an existing Lambda function

How-To/Exploit Link(s)

- <https://rhinosecuritylabs.com/aws/aws-privilege-escalation-methods-mitigation-part-2>

Description

An attacker with the `lambda:UpdateFunctionConfiguration` permission can escalate permissions by attaching a Lambda layer to an existing function to override a library that is in use by the function, where their malicious code could utilize the function's IAM role for AWS API calls.

Required Permission(s)

- `lambda:UpdateFunctionConfiguration`

Potential Impact

This would give an attacker access to the privileges associated with the Lambda service role that is attached to that function, which could range from no privilege escalation to full administrator access to the account.

27. Passing a role to a new SageMaker Jupyter notebook

How-To/Exploit Link(s)

- <https://rhinosecuritylabs.com/aws/aws-privilege-escalation-methods-mitigation-part-2>

Description

An attacker with the `sagemaker:CreateNotebookInstance` , `sagemaker:CreatePresignedNotebookInstanceUrl` , and `iam:PassRole` permissions can escalate privileges by passing a role to a new SageMaker Jupyter notebook. Then, through the Jupyter UI, they can access the credentials belonging to the notebook for further exploitation.

Required Permission(s)

- `sagemaker:CreateNotebookInstance`
- `sagemaker:CreatePresignedNotebookInstanceUrl`
- `iam:PassRole`

Potential Impact

This would give an attacker access to the privileges associated with the SageMaker service role that is attached to that Jupyter notebook, which could range from no privilege escalation to full administrator access to the account.

## 28. Gaining access to an existing SageMaker Jupyter notebook

### How-To/Exploit Link(s)

- <https://rhinosecuritylabs.com/aws/aws-privilege-escalation-methods-mitigation-part-2>

### Description

An attacker with the `sagemaker:CreatePresignedNotebookInstanceUr1` permission can escalate privileges by creating a signed URL For an existing SageMaker Jupyter notebook. Then, through the Jupyter UI, they can access the credentials belonging to the notebook for further exploitation.

### Required Permission(s)

- `sagemaker:CreatePresignedNotebookInstanceUr1`

### Potential Impact

This would give an attacker access to the privileges associated with the SageMaker service role that is attached to that Jupyter notebook, which could range from no privilege escalation to full administrator access to the account.

## Prior Research, References, and Resources

This research wouldn't be possible without the excellent work from other security researchers in the past. Listed below are a few blogs and tools that stood as the forefront into AWS IAM privilege escalation and were great resources in the aggregation of these privilege escalation methods and the discovery of new ones.

- <https://github.com/andresriancho/nimbostratus>
- <https://blog.cloudsploit.com/privilege-escalation-in-amazon-web-services-cb4827265958>

