

AWS Access Analyzer - Hands-on Exercise

- [Introduction](#)
- [Part 1 – Remove Outside Access to a Resource](#)
 - [Goal](#)
 - [Part 1A – Login and get to Access Analyzer](#)
 - [Part 1B – Examine the Finding for "your" role](#)
 - [Part 1C – Remove access by the bad-actor Role](#)
 - [Part 1D - Rescan](#)
- [Part 2 – Archive a Finding](#)
 - [Goal](#)
 - [Part 2A – Login and get to Access Analyzer](#)
 - [Part 2B - Find the Finding for "your" S3 bucket](#)
 - [Part 2C - Archive the Finding](#)

Introduction

This hands-on exercise shows how to review IAM Access Analyzer Findings and take actions with those findings.

Part 1 – Remove Outside Access to a Resource

Goal

In this exercise, you will use Access Analyzer to find an IAM Role that allows access from an outside AWS account and remove that access.

We have prepared the cu-training AWS account with roles named **example-role-NETID**, one role for each of the training participants.

Part 1A – Login and get to Access Analyzer

1. Login to the cu-training AWS account using traditional Shibboleth login.
 - a. Use this link to initiate login: <https://signin.aws.cucloud.net/>
 - b. If you are given the option of selecting a role, select **shib-training** under the "cu-training" AWS account, and click on **Sign in**.
2. Once in the AWS Management Console, type "iam" in the search box and click on **IAM** under **Services**.
3. Click on **Access analyzer** from the left navigation section.
4. Check which AWS region your console is pointed at. You want "N. Virginia". If your console is in any other region, change it to "US East (N. Virginia) us-east-1".



Unlike other most other aspects of IAM, Access Analyzers are regional.

Part 1B – Examine the Finding for "your" role

1. Under **Active findings**, use the menu integrated into the filter to search for **Resource: example-role-NETID**
 - a. Be sure to use the pull-down menu in the search field to select **Resource**
 - b. Enter **example-role-NETID**, replacing **NETID** with your own Cornell NetId (e.g., **example-role-pea1**)
 - c. Hit "enter" on your keyboard to trigger the actual search.
2. Your search should result in one Finding that matches. Click on the Finding ID for that record to drill into the finding details.
3. Note the finding details:
 - **External principal (IAM Role):** `arn:aws:iam::2251*****:role/bad-actor`
 - **Access level:** `sts:AssumeRole`
 - **i** This indicates that our role (**example-role-NETID**) allows the **bad-actor** role from a different account to assume it. Those permissions are defined by the Trust Policy in our role.



Note that the **bad-actor** Role didn't have anything to do with our creation of the **example-role-NETID** Role in our account and trusting the **bad-actor** Role. Someone (or something) with appropriate IAM privileges to our account is the only way that **example-role-NETID** was created or configured.

Part 1C – Remove access by the bad-actor Role



Just because a Role within your AWS account trusts IAM Roles or Users from another account, doesn't mean that access is inappropriate or unnecessary. Cross-account access is perfectly fine and often necessary. For example, that is how CloudCheckr accesses our Cornell AWS accounts to do the information gathering it does to provide its services to Cornell.

For the purposes of this exercise, we'll claim that **bad-actor** no longer has need for (or never should have had) the cross-account access defined in **example-role-NETID**. To remove that access, we have two options:

1. Delete the entire **example-role-NETID** Role.
2. Change the trust policy for **example-role-NETID** so that it no longer allows the **bad-actor** Role to assume it.

In real life, the appropriate course of action depends on the situation. For the purposes of this example, we'll just delete the **example-role-NETID** Role since, in our scenario, none of our own tools or access uses **example-role-NETID**.

1. Back on the details page for your finding, click on the **Go to IAM console** button to see the role details in IAM.
2. Click on the **Trust relationships** tab and note that the role does indeed trust **arn:aws:iam::2251*****:role/bad-actor**
3. Click on the **Delete role** button, and confirm by clicking **Yes, delete**.

Part 1D - Rescan

When we change the access to a resource or delete a resource entirely, we can prompt Access Analyzer to rescan the resource to confirm that the Finding is no longer relevant (i.e., is resolved).

1. Repeat the steps in Part 1B to find the Finding about "your" **example-role-NETID** Role.
2. Drill into the finding details.
3. Click on **Rescan** to tell Access Analyzer to review the Finding and check whether the access still exists.
 - If the access remains unchanged, so will the Finding details.
 - If you have successfully deleted "your" **example-role-NETID** Role, or changed the trust policy so that it no longer trusts the **bad-actor** Role, then the status of the Finding will be changed to **Resolved**.

Part 2 – Archive a Finding

Goal

In this exercise, you will use Access Analyzer to archive a finding allowing public access to an S3 bucket. This indicates one-time review and approval for that access.

We have prepared the cu-training AWS account with S3 buckets named **my-public-web-site-NETID**, one bucket for each participant. We used these same publicly readable buckets in [AWS Config - Hands-on Exercise](#).

Part 2A – Login and get to Access Analyzer

1. If you aren't logged in to the cu-training account with role **shib-training**, follow the instructions in **Part 1A** above to login and navigate to the Access Analyzer console.


Part 2B - Find the Finding for "your" S3 bucket

1. Under **Active findings**, use the menu integrated into the filter to search for **Resource: my-public-web-site-NETID**
 - a. Be sure to use the pull-down menu in the search field to select **Resource**
 - b. Enter **my-public-web-site-NETID**, replacing **NETID** with your own Cornell NetId (e.g., **my-public-web-site-pea1**)
 - c. Hit "enter" on your keyboard to trigger the actual search.
2. Your search should result in one Finding that matches. Click on the Finding ID for that record to drill into the finding details.
3. Note the finding details:
 - **External principal: All Principals**
 - **Access level:**
 - **s3:ListBucket**
 - **s3:ListBucketMultipartUploads**
 - **s3:ListBucketVersions**
 - **i** This indicates that our bucket is publicly readable, though it is possible that not all objects within the bucket are public.

Part 2C - Archive the Finding

In this exercise scenario, we have decided that it is indeed our intention that **my-public-web-site-NETID** be publicly readable. Therefore we just need to tell Access Analyzer that this is intended access.

1. Under **Next steps** click on the **Archive** button.
 - The status of the Finding will turn to **Archived**.
2. If you wish, you can now navigate back to the main **Access analyzer** page and click on the **Archived** tab to search for your S3 bucket and confirm that the related finding is indeed archived.

 Archiving a finding is basically telling Access Analyzer that we have review the finding and that we accept the current access that is being allowed for the resource.