Identity Round Robin Workshop
Using AWS IAM Roles for Delegating Access to Security Services

aws | Pop-up Loft

Agenda

- Introduction to AWS Security Services
- Access Delegation and the Principle of Least Privilege
- Lab overview
- Q & A

AWS Security Services

AWS offers a variety of services to help with your security and compliance needs.

IAM
Cognito
Secrets Manager
GuardDuty
Inspector
Macie
Organizations

Single Sign-On
Certificate Manager
CloudHSM
Directory Services
WAF / Shield
Artifact
Security Hub

Security Services – The need for delegation

When using security services, you should consider the matter of delegation – restricting who can access which features of a service as well as whether they need that access all the time. Consider these examples.

(1) Suppose there are people in the IT department that spend 10% of their time managing security.   Do they need to be signed in with security privileges 100% of the time?   **NO!**

(2) An IT Security Operator needs to be able to read CloudTrail entries. Should the Operator be able to turn off CloudTrail?   **NO!**

You should follow the Principle of Least Privilege when granting access to services.

The Principle of Least Privilege

The Principle of Least Privilege means that people should have the fewest privileges they need in order to perform their job functions.

When users need elevated or different privileges for specific tasks, they should acquire them only for the duration of the task and relinquish the privileges when they are no longer needed.   You do this with AWS IAM roles.

The Principle of Least Privilege is not meant to encumber you, but to protect you, your data, and your customers – and can protect you from yourself as well!

AWS IAM roles are the key to delegating access to AWS services.

AWS Identity and Access Management (IAM) Roles

- AWS roles are security principals you temporarily assume (or “switch to”).

- When you assume a role, your privileges are *replaced* by those of the role, similar to using *sudo* in Linux or “Run as Administrator” in Windows.

- A role has a *permissions policy* and a *trust policy*.

- A *permissions policy* describes what AWS actions (meaning API calls) the roles can take and on what resources the actions can be taken.

- A *trust policy* identifies who can assume (“switch to”) the role.

Here’s an example of an IAM role:

```
SecAdministratorRole:
    Type: AWS::IAM::Role,
    Properties: {

        AssumeRolePolicyDocument: {
            Version: 2012-10-17,
            Statement: [
                {
                    Effect: Allow,
                    Principal: AWS:123456789012:root
                    Action: sts:AssumeRole
                }
            ]
        },

        ManagedPolicyArns: [
            arn:aws:iam::aws:policy/AmazonGuardDutyFullAccess
        ],
        MaxSessionDuration: 43200
    }
}
```

Trust Policy

Allow

This principal

To assume this role

All users in the account starting from the “root” of the identity tree (in other words, all users).

Permissions Policy

AWS-managed policy

Policy types

- AWS-managed policies are provided by AWS. Think of them as reusable collections of individual IAM policies.
  For example: AmazonGuardDutyFullAccess

- Customer-managed policies are created by you. These too are reusable collections of individual policies. If you worked for ACME.COM, you could create a managed policy named AcmeFullAccess.

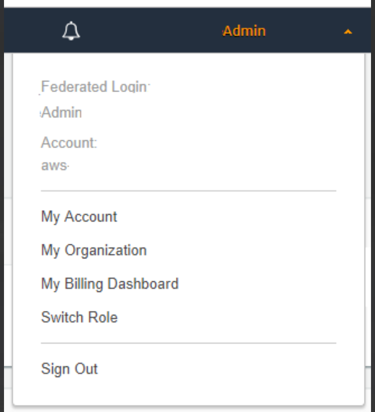
- Inline policies are added directly to a principal. We generally discourage you from using these as they are harder to audit.

There are three ways to switch your role:

1. A URL

```
https://signin.......
```

2. AWS Console →



3. The AWS CLI

```
aws sts assume-role…
```

What else can you do with AWS IAM roles?

- You can use *cross-account roles* to enable you to delegate access to your account from another account.

  For example, you can allow a user in another AWS account to use a cross account role to access the AWS Management Console in your account.

What else can you do with AWS IAM roles?

- **What if you had several accounts?**

  The answer is the same. You could create one role in a central account and let the other accounts use that role to access centralized resources.

  Roles are often an important part of a *multi-account strategy* where accounts are automatically deployed as part of an AWS Landing Zone such as that built by AWS Control Tower (currently in preview).

What else can you do with AWS IAM roles?

- You can use Amazon EC2 roles to limit what the instance can do with regard to AWS.

  If you want to launch an Amazon EC2 instance and have that instance make API calls to AWS to objects on S3, the instance will need credentials.

What else can you do with AWS IAM roles?

- Whose credentials would you use?

- If your own credentials have a lot of access privileges, you might not want the Amazon EC2 instance to use them. If that instance were somehow compromised, they could use the instance to do things on your behalf with your credentials.

- The answer: Following the Principle of Least Privilege, create an AWS IAM role with only the privileges that are truly needed. Launch the instance and attach the AWS IAM role to it.

Lab Overview

- In the lab, you will learn how to create a role and then you will practice switching roles.

- You will then test the level of access that you have by trying to access various services and seeing what you can and cannot do.

Lab Overview

- The environment creates two roles:
  - A Security Administrator role with policies that grant full access to security services.
  - A Security Operator role.
    - Initially, the policies are similar to those of the Security Administrator role.
    - You will change the policies to provide read-only access to security services.

Lab Overview

- You will learn how to switch between roles to change your *effective* access.

- Role switching can be used to enforce the Principle of *Least Privilege.* Principles should have the fewest privileges needed to perform their duties.

- Role switching in AWS is similar to *sudo* in Linux or “Run as Administrator” in Windows.

Lab Overview

- Make sure you use only the us-west-2 (Oregon) region.
- The lab has two phases:
  - In the Build Phase, you will build the environment and configure the Security Operator role., and test your work. Time permitting, you will then turn over your credentials to another team who will do the verification.
  - In the Verify Phase, again, time permitting, you will receive someone else’s credentials and then perform verification to ensure they did the lab properly.

https://awssecworkshops.com/

1. Select Workshops
2. Then Identity Round Robin
3. Then (External) Security Services Round
4. As you are doing the lab
   choose the “AWS-sponsored event” option, not “Individual”
5. If you are sharing an account, take turns on the console.
   Only deploy the stack once per account!

aws | Pop-up Loft

Everything and Anything Startups Need to Get Started on AWS

aws.amazon.com/activate