Using the AWS AssumeRole capability

You manage multiple AWS accounts, and the effort to configure Splunk SOAR access to every account can feel insurmountable. You’ve diligently set up the roles you need to manage your AWS environment and want to leverage that effort in Splunk SOAR.

Use AWS temporary credentials

Amazon's Security Token Service solves this problem with temporary credentials. Temporary credentials are great, because you can give a user to access AWS resources without the overhead of creating an account, embedding credentials, or worrying about revoking or expiring them. Using the Splunk SOAR AWS Security Token Service integration, you are able to take advantage of the AssumeRole capability.

The AssumeRole functionality with the AWS Security Token Service is supported in these apps:

- AWS S3
- AWS Lambda
- AWS EC2

This is a two-step process starting with the AWS Security Token Service integration. You'll use it to AssumeRole with a specified Amazon Resource Name (ARN), then pass the credentials it returns to your AWS app, like S3 or Lambda.

1. In your playbook, start with the AWS Security Token Service integration and its assume role action.
2. Configure the role_arn parameter in AWS. This is the ID of the role you want to assume. You can type the ARN directly, or supply it from the result of a previous action.
3. The output of your assume role action becomes the credentials for the ARN. Supply that output in the AWS integration’s credentials field for the app you need.

The credentials will expire on their own, and you can create a fresh set whenever you need it.

Next steps

The content in this article comes from a previously published blog, one of the thousands of Splunk resources available to help users succeed. These additional Splunk resources might help you understand and implement these recommendations:

- Git Hub: AWS AssumeRole example playbooks