Reading 1.6: Protect the AWS Root User

What's the Big Deal About Auth?

When you're configuring access to any account, two terms come up frequently: **authentication and authorization**. Though these terms may seem basic, you need to understand them to properly configure access management on AWS. It's important to keep this mind as you progress in this course. Let's define both terms.

Understand Authentication

When you create your AWS account, you use a combination of an email address and a password to verify your identity. If the user types in the correct email and password, the system assumes the user is allowed to enter and grants them access. This is the process of *authentication*.

Authentication ensures that the user is who they say they are. Usernames and passwords are the most common types of authentication, but you may also work with other forms, such as token-based authentication or biometric data like a fingerprint. Authentication simply answers the question, "Are you who you say you are?"

Understand Authorization

Once you're inside your AWS account, you might be curious about what actions you can take. This is where *authorization* comes in. Authorization is the process of giving users permission to access AWS resources and services. Authorization determines whether the user can perform an action—whether it be to read, edit, delete, or create resources. Authorization answers the question, "What actions can you perform?"

What Is the AWS Root User?

When you first create an AWS account, you begin with a single sign-in identity that has complete access to all AWS services and resources in the account. This identity is called the AWS root user and is accessed by signing in with the email address and password that you used to create the account.

Understand the AWS Root User Credentials

The AWS root user has two sets of credentials associated with it. One set of credentials is the email address and password used to create the account. This allows you to access the AWS Management Console. The second set of credentials is called access keys, which allow you to make programmatic requests from the AWS Command Line Interface (AWS CLI) or AWS API.

Access keys consist of two parts:

- An access key ID, for example, A2lAI5EXAMPLE
- A secret access key, for example, wJalrFE/KbEKxE

Similar to a username and password combination, you need both the access key ID and secret access key to authenticate your requests via the AWS CLI or AWS API. Access keys should be managed with the same security as an email address and password.

Follow Best Practices When Working with the AWS Root User

Keep in mind that the root user has complete access to all AWS services and resources in your account, as well as your billing and personal information. Due to this, securely lock away the credentials associated with the root user and do not use the root user for everyday tasks.

To ensure the safety of the root user:

- Choose a strong password for the root user.
- Never share your root user password or access keys with anyone.
- Disable or delete the access keys associated with the root user.
- Do not use the root user for administrative tasks or everyday tasks.

When is it OK to use the AWS root user? There are some tasks where it makes sense to use the AWS root user. Check out the links at the end of this section to read about them.

Delete Your Keys to Stay Safe

If you don't already have an access key for your AWS account root user, don't create one unless you absolutely need to. If you do have an access key for your AWS account root user and want to delete the keys:

1. Go to the My Security Credentials page in the AWS Management Console and sign in with the root user's email address and password.
2. Open the Access keys section.
3. Under Actions, click **Delete**.
4. Click **Yes**.

The Case for Multi-Factor Authentication

When you create an AWS account and first log in to that account, you use single-factor authentication. Single-factor authentication is the simplest and most common form of authentication. It only requires one authentication method. In this case, you use a username and password to authenticate as the AWS root user. Other forms of single-factor authentication include a security pin or a security token.

However, sometimes a user's password is easy to guess. For example, your coworker Bob's password, IloveCats222, might be easy for someone who knows Bob personally to guess, because it's a combination of information that is easy to remember and describes certain things about Bob (1. Bob loves cats, and 2. Bob's birthday is February 22).

If a bad actor guessed or cracked Bob's password through social engineering, bots, or scripts, Bob might lose control of his account. Unfortunately, this is a common scenario that users of any website often face.

This is why using MFA has become so important in preventing unwanted account access. MFA requires two or more authentication methods to verify an identity, pulling from three different categories of information.

- Something you know, such as a username and password, or pin number
- Something you have, such as a one-time passcode from a hardware device or mobile app
- Something you are, such as fingerprint or face scanning technology

Using a combination of this information enables systems to provide a layered approach to account access. Even though the first method of authentication, Bob's password, was cracked by a malicious user, it's very unlikely that a second method of authentication, such as a fingerprint, would also be cracked.

This extra layer of security is needed when protecting your most sacred accounts, which is why it's important to enable MFA on your AWS root user.

Use MFA on AWS

If you enable MFA on your root user, you are required to present a piece of identifying information from both the something you know category and the something you have category. The first piece of identifying information the user enters is an email and password combination. The second piece of information is a temporary numeric code provided by an MFA device.

Enabling MFA adds an additional layer of security because it requires users to use a supported MFA mechanism in addition to their regular sign-in credentials. It's best practice to enable MFA on the root user.

Review Supported MFA Devices

AWS supports a variety of MFA mechanisms, such as virtual MFA devices, hardware devices, and Universal 2nd Factor (U2F) security keys. For instructions on how to set up each method, check out the Resources section.

Device	Description	Supported Devices
Virtual MFA	A software app that runs on a phone or other device that provides a one-time passcode. Keep in mind that these applications can run on unsecured mobile devices, and because of that, may not provide the same level of security as hardware or U2F devices.	Authy, Duo Mobile, LastPass Authenticator, Microsoft Authenticator, Google Authenticator

Device	Description	Supported Devices
Hardware	A hardware device, generally a key fob or display card device that generates a one-time six-digit numeric code	Key fob, display card
U2F	A hardware device that you plug into a USB port on your computer	YubiKey

Resources

- *External Site:* AWS: Enabling a Hardware MFA Device (Console)
- *External Site:* AWS: Enabling a U2F Security Key (Console)
- *External Site:* AWS: Enabling a Virtual Multi-Factor Authentication (MFA) Device (Console)
- External Site: AWS: Table of Supported MFA Devices