AWS Setup and Security Alerting Project

Date: March 28, 2025

Prepared by: Agbai Joseph Okwara

Project Overview

This project involved setting up an AWS Free Tier account and implementing security alerting mechanisms using AWS CloudTrail, SNS, and EventBridge. The primary goal was to monitor and receive notifications whenever the AWS root user performed a **GetCallerIdentity** API call, which can indicate unauthorized access attempts or security breaches.

Objectives

- Set up an **AWS Free Tier** account for cloud-based experiments.
- Configure **CloudTrail** to log AWS API activity.
- Use **Amazon SNS** (Simple Notification Service) to send email alerts.
- Set up **Amazon EventBridge** to trigger alerts on root user actions.
- Test the setup using AWS CLI to confirm functionality.

Technologies Used

- **AWS CloudTrail**: Tracks API calls and logs events.
- **Amazon SNS**: Sends notifications to subscribed users.
- **Amazon EventBridge**: Creates event-driven triggers.
- **Kali linux CLI**: Used to test the security setup.

Project Steps

1. **AWS Free Tier Account Setup**
   - Created a Free Tier AWS account on aws.amazon.com.
   - Configured billing information and identity verification.
   - Selected "Basic Support – Free" for cost efficiency.
2. **Configured AWS CloudTrail**
   - Accessed **CloudTrail** in the AWS Management Console.
   - Created a new trail named MyCloudTrail.
   - Enabled **Management Event Logging** for all read/write events.
   - Configured logs to be stored in an **S3 bucket** (my-cloudtrail-logs-uniqueid).
   - Enabled **multi-region logging** for full coverage.
3. **Set Up SNS for Email Alerts**
   - Created an **SNS topic** named APICallAlert.
   - Subscribed an email address to receive notifications.
   - Confirmed the email subscription via the AWS verification link.
   - Linked **CloudTrail** to SNS to trigger notifications when specific events occur.
4. **Configuring EventBridge for Root API Call Alerts**
   - Created an **EventBridge Rule** named GetCallerIdentityAlert.
   - Defined a **custom event pattern** to detect GetCallerIdentity API calls from the root user:

```
{
 "source": ["aws.sts"],
 "detail-type": ["AWS API Call via CloudTrail"],
 "detail": {
  "eventSource": ["sts.amazonaws.com"],
  "eventName": ["GetCallerIdentity"],
  "userIdentity": {
   "type": ["Root"]
  }
 }
}
```

   - Configured **SNS** as the target for this rule to send notifications.
   - Allowed EventBridge to create an IAM role for permissions.

5. Testing the Setup via kali linux CLI

- Installed and configured **kali linux** with root credentials.
- Ran the following command to simulate an API call:

  aws sts get-caller-identity

- Verified that the API call was logged in **CloudTrail**.
- Checked the SNS email inbox for the alert notification.

Results & Key Takeaways

- Successfully implemented **real-time monitoring** for root user API calls.
- Demonstrated how to configure **AWS security alerting** using CloudTrail, SNS, and EventBridge.
- Gained hands-on experience with AWS **event-driven automation**.
- Ensured that **email notifications** were triggered on unauthorized root account activity.

This project showcases my ability to implement AWS security measures and monitor cloud infrastructure effectively.