Security Monitoring & Alerting System Using AWS

Overview

This project demonstrates the implementation of a basic security monitoring and alerting system on AWS. The goal is to track and alert on access to sensitive secrets stored in AWS Secrets Manager using CloudTrail, CloudWatch Logs, Metrics, Alarms, and SNS notifications.

Components Used

- AWS Secrets Manager
- AWS CloudTrail
- Amazon CloudWatch
- Amazon SNS
- AWS CLI

Step-by-Step Implementation

Step 1: Create a Secret in AWS Secrets Manager

Details:

- **Secret name:** secrete
- **Encryption key:** aws/secretsmanager
- **Secret description:** A secrete created for security monitoring system.

Secret ARN:

arn:aws:secretsmanager:ap-south-1:157466815761:secret:secrete-jeJdZg

---

Step 2: Create a CloudTrail Trail

Trail Configuration:

- **Trail name:** secrete-manager
- **Multi-region trail:** Yes
- **Apply trail to my organization:** Not enabled

- **Trail log location:** secrete-manager-d/AWSLogs/157466815761
- **Log file SSE-KMS encryption:** Not enabled
- **Log file validation:** Enabled
- **SNS notification delivery:** Disabled

Step 2.1: Enable Log Event Types

- **Management events:** Enabled
  - **API activity:** All
  - **Exclude AWS KMS events:** Yes
  - **Exclude Amazon RDS Data API events:** Yes

Step 3: Validate Logging

Used the following AWS CLI command to simulate secret access:

```
aws secretsmanager get-secret-value --secret-id "secrete" --region ap-south-1
```

Checked **CloudTrail > Event History**, confirmed the GetSecretValue events were recorded:

Event Name	Event Time (UTC+05:30)	User Name	Event Source	Resource Type	Resource Name
GetSecretValue	April 23, 2025, 11:54:14	devops	secretsmanager.amazonaws.com	AWS::SecretsManager::Secret	secrete
GetSecretValue	April 23, 2025, 11:46:44	devops	secretsmanager.amazonaws.com	AWS::SecretsManager::Secret	arn:aws:...

Step 4: Enable CloudWatch Logs

- Enabled CloudWatch logs in the trail settings
- Verified the creation of the log group in **CloudWatch > Log Groups**

Step 5: Create Metric Filter

Filter Pattern:

"GetSecretValue"

Metric Details:

- **Metric namespace:** securitymetric
- **Metric name:** secret accessed
- **Metric value:** 1
- **Default value:** 0

This filter captures every time the secret value is accessed.

Step 6: Create CloudWatch Alarm

Alarm Configuration:

- **Metric:** secret accessed from securitymetric namespace
- **Threshold type:** Static
- **Whenever metric is >= 1** (indicating at least one access)

Step 7: Create SNS Notification

- Created a new SNS Topic
- Subscribed an email endpoint to receive alerts
- Confirmed subscription via email

Step 8: Test and Verify Alert

- Accessed the secret in Secrets Manager **two more times**
- **Received SNS email notifications** each time the secret was accessed

---

Outcome

The setup successfully monitors access to sensitive secrets in Secrets Manager and sends real-time alerts using SNS when access occurs.

Improvements/Future Scope

- Enable logging to centralized S3 bucket with tighter access controls
- Configure alerts on additional API operations (e.g., PutSecretValue, DeleteSecret)
- Integrate with a SIEM system for correlation and deeper analysis

---

AWS CLI Reference

```
aws secretsmanager get-secret-value --secret-id "secrete" --region ap-south-1
```

Credits

Implemented by: Darshan BR
Date: April 2025

All the project screenshots can be seen in the screenshots folder.