**Creating a CloudTrail Trail and EventBridge Alert for Console Sign-Ins**

**Hands-on Lab**

We are going to create a multi-Region CloudTrail Trail with Log File Integrity and Validation enabled, as well as create an EventBridge Rule that can send a notification via Amazon SNS to alert you whenever there is a Root user console sign-in!.

**Objectives**

* Create Multi-Region CloudTrail Trail
* Create CloudTrail-specific KMS Key
* Create SNS Topic and Subscription
* Create Amazon EventBridge Rule

**Architecture**

A screenshot of a computer screen

Description automatically generated

We are going to start creating the multiregional CloudTrail called OurTrail and we are going to choose a new S3 to save our data, also we are going to leave enabled encryption and log file validation. We enter a new KMS alias called CloudTrailKey

A screenshot of a computer

Description automatically generated

In the step 2 we choose management events, and we want to select read and write event because we want to read every sing in event and we want to write the data in our log file with the cloud Eventbridge

A screenshot of a computer

Description automatically generated

We review the settings and create the cloud trail

A screenshot of a computer

Description automatically generated

We can see we create our trail and by default is multi-region

A screenshot of a computer

Description automatically generated

On the CloudTrail we can see our event history and there is and event called Console Login that stores our Loggins but the difference between our cloud trail event we are creating is that this one only can store Loggins up to 90 days by default and you can disabled it or modified

A screenshot of a computer

Description automatically generated

We are going to use the event record for test our pattern, we can see it below

A screenshot of a computer

Description automatically generated

{

"eventVersion": "1.08",

"userIdentity": {

"type": "IAMUser",

"principalId": "AIDAX66OGNACEEZGUMRF7",

"arn": "arn:aws:iam::547503892484:user/cloud\_user",

"accountId": "547503892484",

"userName": "cloud\_user"

},

"eventTime": "2024-06-16T00:12:35Z",

"eventSource": "signin.amazonaws.com",

"eventName": "ConsoleLogin",

"awsRegion": "us-east-1",

"sourceIPAddress": "24.128.137.144",

"userAgent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10\_15\_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/125.0.0.0 Safari/537.36",

"requestParameters": null,

"responseElements": {

"ConsoleLogin": "Success"

},

"additionalEventData": {

"LoginTo": "https://us-east-1.console.aws.amazon.com/console/home?hashArgs=%23&isauthcode=true&region=us-east-1&state=hashArgsFromTB\_us-east-1\_3b0d8604cf65f60d",

"MobileVersion": "No",

"MFAUsed": "No"

},

"eventID": "41d2e398-ef54-4055-a145-3bbb0571e9dc",

"readOnly": false,

"eventType": "AwsConsoleSignIn",

"managementEvent": true,

"recipientAccountId": "547503892484",

"eventCategory": "Management",

"tlsDetails": {

"tlsVersion": "TLSv1.3",

"cipherSuite": "TLS\_AES\_128\_GCM\_SHA256",

"clientProvidedHostHeader": "signin.aws.amazon.com"

}

}

The S3 that is created for store our logs we can see it in the dashboard on s3 and if we enter to our s3 we can see 2 prefixes that are going to help our performance and is going to have 2 prefixes:

1.- CloudTrail: This is going to store our log files

2.- CloudTrail-Digest: Contain the specific data for log file validation so we know our files were not modified

A screenshot of a computer

Description automatically generated

Now we going to create our SNS topic to send alerts to our security operations. The topic name is going to be ConsoleAlerts and is going to be a SQS Standard Queue, we won’t modified the default settings

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Once is created we need a subscriber so we add a new one, the protocol to alert is email and for this lab I going to use a temporally email that you can use for projects or test developments <https://temp-mail.org>

A screenshot of a computer

Description automatically generatedA screenshot of a computer

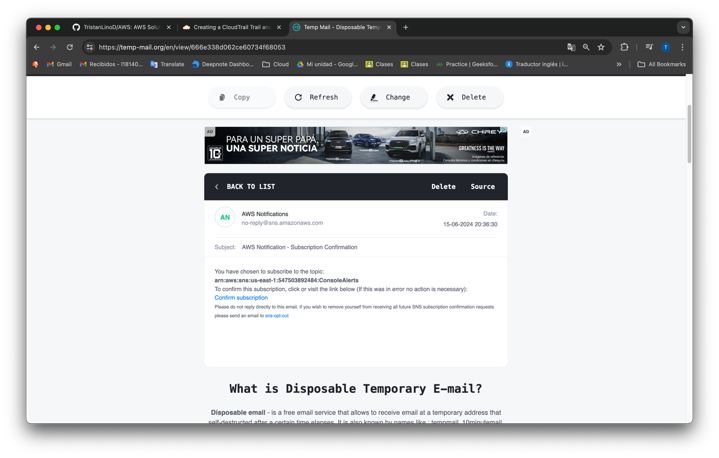
Description automatically generated

We create the subscription and left the default setting in Subscription filter policy and Redrive policy (DQL)

A screenshot of a computer

Description automatically generated

The we confirm the subscription in our temporally email account

A screenshot of a computer

Description automatically generated

And we can see now our new subscriber that is going to simulate our security team email

A screenshot of a computer

Description automatically generated

We can test the subscriber publishing a message.

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

Now the only thing we need for our architecture is create the trigger with Amazon EventBridge, we start choosing an EventBridge rule

A screenshot of a computer

Description automatically generated

The name of the trigger is ConsoleSigninAlerts we left the default setting and is important to have enable the rule of the selected event bus for start working

A screenshot of a computer

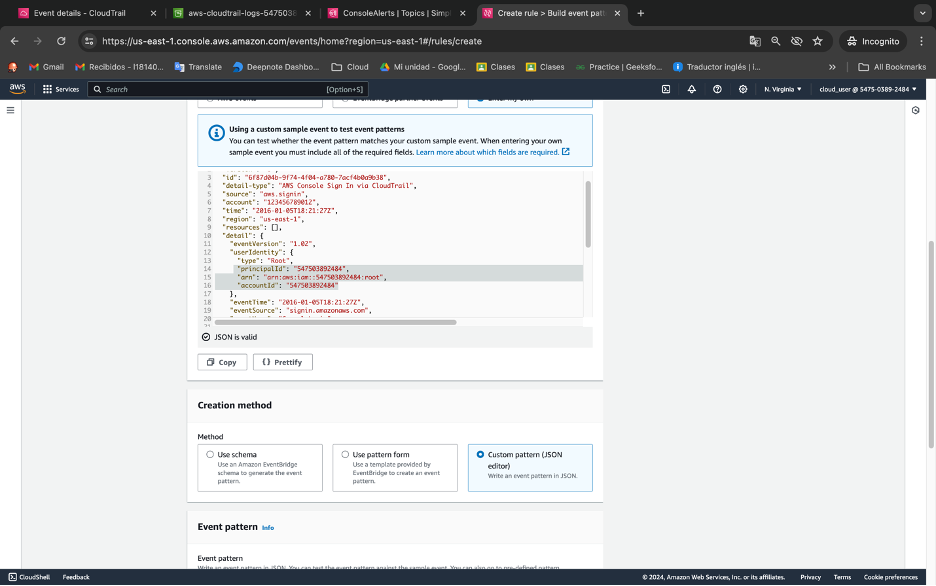
Description automatically generated

The source is an Amazon Event (Sign in) we are going to use the event record that we copy form Event History dashboard or we can select AWS events, choose AWS Console Sign In via CloudTrail and copy the event record

A screenshot of a computer

Description automatically generated

Next we paste it in Enter my own record an we need to change the id on the lines 14,15,16. We do this for test our pattern



Next we need to create our pattern to catch the event, we can use a predesign patter with Use pattern form, in this case I show you a pettern that register all sign in.

A screenshot of a computer

Description automatically generated

But we want a specific pattern that checks only IAM entities and root account Sign In, so we specify in “userIdentity” and for test we add our Account ID (yellow) that’s why we change the id in the last event record we add

{

"detail-type": ["AWS Console Sign In via CloudTrail"],

"source": ["aws.signin"],

"detail": {

"userIdentity": {

"type": ["IAMUser", "Root"],

"arn": [

"arn:aws:iam::547503892484:user/cloud\_user",

"arn:aws:iam::547503892484:root"

]

}

}

}

A screenshot of a computer

Description automatically generated

Now we can test our pattern and we can see its working.

A screenshot of a computer

Description automatically generated

And the last part is choose our SNS topic when the pattern is trigger and next we can skip to review and create Amazon EventBridge

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

We are going to test our Rule that you see below

A screenshot of a computer

Description automatically generated

For this I open a new Web Browser (safari) and Sign In with the same account. In the last picture we can see the message that receive the subscriber in this case is simulating our Security Operating team. And we this we have a log file that register every sign in and alerts to the security team

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated