IRON CLOUD
SECURITY

Script Details

Sierra Maldonado

Email Alert

The Lambda function is triggered by CloudTrail events. Specifically, it is triggered when a user attempts to log in to the AWS Management Console ('ConsoleLogin' event).

The function first extracts the instance ID from the CloudTrail event.

It checks if the event is related to a failed authentication attempt. It does this by verifying that the event name is 'ConsoleLogin' and the error message is 'Failed authentication'.

The function then creates a DynamoDB client to interact with the DynamoDB table that stores the failed attempt counters.

It checks if the instance already has a counter in the DynamoDB table by retrieving the item based on the instance ID.

If the instance does not have a counter (i.e., 'Item' is not present in the response), it means this is the first failed attempt for the instance. In this case, the function initializes the counter to 1 by putting a new item in the DynamoDB table.

If the instance already has a counter, it retrieves the current number of attempts and increments it by 1.

Next, the function checks if the number of failed attempts has reached or exceeded the threshold of 3.

If the number of failed attempts is equal to or greater than 3, the function creates an SNS client to interact with the Amazon SNS service.

It then publishes a message to the SNS topic specified by the ARN provided. The message contains information about the detected brute force attack, including the instance ID and the number of failed attempts.

Finally, the function updates the counter in the DynamoDB table to reflect the new number of attempts.

File Organizer

CloudTrail Log Analysis:

The Lambda function is triggered by an S3 event when a CloudTrail log file is uploaded to an S3 bucket.

It retrieves the CloudTrail log file from S3, processes it, and extracts relevant information from each log entry.

It performs threat detection logic based on the extracted information.

Two examples of threat detection logic are provided:

If the CloudTrail event is a PutObject API call and the user agent contains the word "malicious," it prints a potential threat detection message.

If the CloudTrail event is a StartInstances API call and the source IP address is '10.0.0.1', it prints a potential threat detection message.

The script is intended to be a starting point for implementing custom threat detection logic based on CloudTrail logs.

GuardDuty Finding Creation:

The script includes a function called create_guardduty_finding that uses the AWS SDK for Python (Boto3) to create a GuardDuty finding.

The create_guardduty_finding function takes parameters such as severity, description, and resource ARN to create a GuardDuty finding.

In the provided code, an example GuardDuty finding is created with the severity set to "High," finding type set to "UnusualBehaviors:EC2/SSHBruteForce," and a description provided.

The function uses the GuardDuty client from Boto3 to make the API call to create the finding.

The GuardDuty finding creation functionality is included as an example, but it is not directly used in the Lambda function. You would need to modify the Lambda function to incorporate this functionality if desired.

Attack Script:

Mode 1: Iterator

Asks for a file path to a dictionary (wordlist) file.

Reads the file line by line and prints each line (word) with a delay of 1 second between each word.

Mode 2: Password Check

Asks for a file path to a wordlist file.

Asks the user to enter a new password.

Checks if the new password is present in the wordlist file. If it is, it prints "Password is weak!" indicating that the password is not strong.

Mode 3: Brute Force SSH

Asks for a file path to a wordlist file.

Asks for an IP address to connect to and a username.

Attempts to establish an SSH connection to the provided IP address using each password from the wordlist file until a successful authentication occurs or the wordlist is exhausted.

Mode 4: Zipfiles

Asks for the path to a zip file and a password file.

Reads the passwords from the password file into a list.

Attempts to extract the files from the zip file using each password from the list until a successful extraction occurs or all passwords are tried.

Exit

Allows the user to exit the script.