Test Project Day four

Cloud Computing

Submitted by Taze Miller US

Contents

This Test Project consists of the following documentation/files:

- WS2019_TP53_Main_document
- WSC2019_TP53_DayOne_actual
- WSC2019_TP53_DayTwo_actual
- WSC2019_TP53_DayThree_actual
- **WSC2019_TP53_DayFour_actual**

Description of project and tasks

Today's Test Project is three hours.
The goal this test project is further test specific cloud computing skills through a series of unique modules.
There will be three tasks for you to complete in any order that you choose.

Tasks

1. Log into the AWS Provided testing platform using the same credentials you've used previously
2. Read the documentation thoroughly (Outlined below)
3. Continue until the test project day has completed (Six hours)

Technical Details

AWS Jam – Find the rogue script

One of your EC2 web servers has been compromised by an insider! A rogue script is invoking many AWS APIs. Find the instance and isolate it using automation, so you can investigate later. You will use services like Amazon EC2, AWS CloudTrail, AWS Athena and AWS Lambda.

Summary

Our e-commerce web site has gone viral! Who would have thought that selling organic jeans would have so much demand! But something is wrong.

A CloudWatch Alarm is warning about unauthorized activity. One of the e-commerce web servers seems to be compromised. It is invoking many AWS APIs. We're worried that someone on the development team has leaked credentials with authority to deploy scripts on our web servers. You need to find the instance running a rogue script and isolate it using automation. We assume forensics will be performed later.

Instance isolation for forensics involves a few steps:

1. Disconnect the web server from the application load balancer.
2. Allow inbound traffic ONLY from the forensics EC2 instance.
3. Block all outbound traffic originating from the compromised instance.

Objectives

1. Identify the compromised instance. Thankfully, we have a CloudWatch Alarm Unauthorized Activity Attempt monitoring unauthorized access attempts.
2. You need to isolate the compromised instance using automation. You should modify and extend the InstanceIsolater Lambda function that is provided.
3. You will need to SSH into the compromised instance. Remember you can only reach it from the forensics EC2 instance that has been provided. None of the web server EC2 instances SSH access from anywhere else.
4. Identify and inspect the rogue script to find the challenge answer.

Inventory

The following resources have been provisioned for you. Make sure you review them before you start hacking. Pay special attention to security groups.

- 2 web server instances
- 1 Application Load Balancer
- 1 forensics EC2 instance
- Some security groups
- 1 CloudWatch Alarm named Unauthorized Activity Attempt
- 1 Lambda function named InstanceIsolater
- 1 AWS CloudTrail trail delivering log files to a S3 bucket.

AWS Jam – Strengthen Your API Defense!!

Your API has been attacked from the Internet because the API Key seems to been leaking out. Your CTO, Steve, has already announced that the new secure API will be released by the end of today.

Your colleague, Akira, sent you an email late last night and he is away on a business trip with Steve. You have to implement the new secure API instead of him.

1. Situation

- Your current API has been using leaked key and Akira had been working to implement new secure API to protect resources from these attacks. But he did not complet it.
  You need to complete it!!!

2. Mission and Requirements

- You need to change configuration for the current API and implement new secure API that Akira had been working on it. The new secure API had just created, so please complete the configuration of Amazon API Gateway.
- Current API is only allowed to access from application via NAT Gateway on Current VPC. Please configure access control using API Gateway
- Secure API is only allowed to access from Secure VPC. Please configure "Private API" to access API without the Internet access. API Gateway supports creating "Private API".
- Secure API must access backend application without the internet access. API Gateway support it is using "VPC Link" that is one of API Gateway function.
- Please don't use the "Use proxy integration" setting when you configure secure API (If you enable it, the backend application will not work well.)

- Akira had generated 2 new API keys, but he had not configured them. Please configure these in the right way. The current API MUST support both keys(Current and New) and the new Secure API MUST support only the new API key.
- Please set a new API Key or new API URL to Parameter stores on AWS Systems manager after adding new API keys or deploying new secure API.
- NOTE: please set the value correctly. please remove additional "spaces" or "line break"

3. Additional Information

- You don't have any permission to configure API yet. Akira created an IAM Role, named starts from "JAMUserRole-") for you. You need to switch role at first.
- NOTE: make sure that the region matches the challenge's region.
- CloudFormation output section is useful. If you get lost, take a look at CloudFormation console.
- Please check "Status Check URL". It gives your current status. It also gives answer if you solve the issue.
- You don't have to change any resources in "Current VPC".
- Please select "prod" stage if you deploy API.
- You have a permission to execute API teting. Please ignore the error message from API Gataway if you test it.
- Please ignore error message when you click New Secure API's GET Link. If you get error message when you saved API Configuration, please check your configuration is saved or not.

4. Important

- Akira has no knowledge and experience on developing APIs, so new Secure API is developed only with the first few steps.
- If you have no idea, please open clue. Clue #1 tell you outline of solution. Clue #2 tells you how to block external access for current API. Clue #3 tells you how to create Secure API.

5. Reference materials

- API Key and Usage Plan
- API Gateway Resource Policy
- API Gateway VPC Integration
- Private API
- CloudWatch Logs Insights

AWS Jam – Identify and mitigate configuration drift

When you run a large fleet of EC2 instances, it's important to know the versions of software you run. And when instances run vulnerable software, you need to identify them and remediate them. In this challenge you will use services such as Amazon EC2, AWS Config and EC2 Systems Manager.

Summary

You are an enterprise running a large fleet of EC2 instances and have a well-defined software release process. The team that manages deployments and operations is struggling to get an accurate list of the applications and versions deployed on each EC2 instance within the fleet. It appears that one of the EC2 instances is running an older version of the application, which you know has numerous security vulnerabilities. You need to find the instance and upgrade the application software using automation.

Your Mission

You have two web server EC2 instances behind a load balancer that run your web site.

The web site has several software dependencies: PHP runtime, Apache and Redis. One of these instances has an older version of this software which has known vulnerabilities and was not approved by your AppSec team. None of the web server EC2 instances allow direct SSH access. You need to

1. Build an inventory of software running on each of these EC2 instances
2. Upgrade the application software version to ensure it meets your AppSec team's security policy.
3. Ensure that the AWS Config rule named **EC2-managedinstance-applications-required** reports the Compliance status as **Compliant**.
4. Invoke the Lambda function named **RevealSecretMessage** to check your work. If you have done it correctly, this function will reveal the challenge's answer.

Inventory

The following resources have been provisioned for you. Make sure you review them before you start hacking!

- Two web server instances
- An Application Load Balancer
- A Lambda function named **RevealSecretMessage**
- A AWS Config rule named **EC2-managedinstance-applications-required**

5. Please refer to the **WS2019_TP53_Main_document** document for more details.