1. **AWS-security-assessment-solution/CloudFormation-Templates/Self ServiceSec.yml**

AWSTemplateFormatVersion: 2010-09-09

Description: Cloudformation templates for <https://github.com/awslabs/aws-security-assessment-solution>

Parameters:

TemplateS3Bucket:

Description: The name of the bucket you created to upload the project files.

Type: String

Default: YourBucketNameHere

RansomwareChecks:

Description: "Enable Ransomware checks"

Type: String

Default: 'false'

AllowedValues:

- 'true'

- 'false'

CommonSecurityMistakesChecks:

Description: "Check for common security mistakes as per https://www.youtube.com/watch?v=tmuClE3nWlk "

Type: String

Default: 'false'

AllowedValues:

- 'true'

- 'false'

Conditions:

# This will enable the optional ransomware specific modules

EnableRansomwareChecks: !Equals

- !Ref RansomwareChecks

- 'true'

# This will enable the optional common security mistake checks specific modules

EnableCommonSecurityMistakesChecks: !Equals

- !Ref CommonSecurityMistakesChecks

- 'true'

Resources:

# Basic VPC required for the Security Review. This will create a VPC with 2 subnets in a single AZ. It will also create a NATGateway and a few VPC Endpoints.

rVPCStack:

Type: AWS::CloudFormation::Stack

Properties:

TemplateURL: !Join

- ''

- - 'https://'

- !Ref TemplateS3Bucket

- '.s3.'

- !Ref "AWS::Region"

- '.amazonaws.com/SelfServiceSecVPC.yml'

# This will create a single Role for EC2 Instance, leverage a few AWS Managed Policies and a few custom policies so Prowler and ScoutSuite can run.

rIAMStack:

Type: AWS::CloudFormation::Stack

Properties:

TemplateURL: !Join

- ''

- - 'https://'

- !Ref TemplateS3Bucket

- '.s3.'

- !Ref "AWS::Region"

- '.amazonaws.com/SelfServiceSecIAM.yml'

Parameters:

SelfServiceSecS3Bucket: !GetAtt

- rS3Stack

- Outputs.SelfServiceSecS3Bucket

#This will create a Bucket where the output of Prowler and ScoutSuite will be delivered

rS3Stack:

Type: AWS::CloudFormation::Stack

Properties:

TemplateURL: !Join

- ''

- - 'https://'

- !Ref TemplateS3Bucket

- '.s3.'

- !Ref "AWS::Region"

- '.amazonaws.com/SelfServiceSecS3.yml'

#This will create a single Instance using Amazon Linux 2, and install/deploy Prowler and ScoutSuite

rEC2Stack:

Type: AWS::CloudFormation::Stack

Properties:

TemplateURL: !Join

- ''

- - 'https://'

- !Ref TemplateS3Bucket

- '.s3.'

- !Ref "AWS::Region"

- '.amazonaws.com/SelfServiceSecEC2.yml'

Parameters:

SelfServiceSecVPCID: !GetAtt

- rVPCStack

- Outputs.SelfServiceSecVPCID

SubnetAID: !GetAtt

- rVPCStack

- Outputs.SubnetAID

InstanceRoleName: !GetAtt

- rIAMStack

- Outputs.InstanceRoleName

SelfServiceSecS3Bucket: !GetAtt

- rS3Stack

- Outputs.SelfServiceSecS3Bucket

#This will enable the optional ransomeware checks

rRansomwareChecks:

Condition: EnableRansomwareChecks

Type: AWS::CloudFormation::Stack

Properties:

TemplateURL: !Join

- ''

- - 'https://'

- !Ref TemplateS3Bucket

- '.s3.'

- !Ref "AWS::Region"

- '.amazonaws.com/SelfServiceSecRansomware.yml'

Parameters:

InstanceRoleARN: !GetAtt

- rIAMStack

- Outputs.InstanceRoleARN

TemplatesS3Bucket: !Ref TemplateS3Bucket

SelfServiceSecS3Bucket: !GetAtt

- rS3Stack

- Outputs.SelfServiceSecS3Bucket

#This will enable the optional checks for common security mistakes

rCommonSecurityMistakesChecks:

Condition: EnableCommonSecurityMistakesChecks

Type: AWS::CloudFormation::Stack

Properties:

TemplateURL: !Join

- ''

- - 'https://'

- !Ref TemplateS3Bucket

- '.s3.'

- !Ref "AWS::Region"

- '.amazonaws.com/SelfServiceSecCommonSecurityMistakes.yml'

Parameters:

InstanceRoleARN: !GetAtt

- rIAMStack

- Outputs.InstanceRoleARN

TemplatesS3Bucket: !Ref TemplateS3Bucket

SelfServiceSecS3Bucket: !GetAtt

- rS3Stack

- Outputs.SelfServiceSecS3Bucket

Outputs:

ReportBucketName:

Description: The name of the newly generated S3 Bucket with reports from the tool

Value: !GetAtt rS3Stack.Outputs.SelfServiceSecS3Bucket

1. **CloudFormation-Templates/SelfServiceSecEC2.yml**

AWSTemplateFormatVersion: '2010-09-09'

Description: AWS CloudFormation template to launch an Instance for self service Security Review

Parameters:

SelfServiceSecVPCID:

Type: String

SubnetAID:

Type: String

SelfServiceSecS3Bucket:

Type: String

InstanceRoleName:

Type: String

LatestAmiId:

Type: 'AWS::SSM::Parameter::Value<AWS::EC2::Image::Id>'

Default: '/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86\_64-gp2'

Resources:

rInstanceSG:

Type: AWS::EC2::SecurityGroup

Properties:

GroupDescription: "SelfServiceSec Security Instance SG"

VpcId: !Ref SelfServiceSecVPCID

Tags:

- Key: Name

Value: "SelfServiceSecSecurityGroup"

rEC2Instance:

Type: AWS::EC2::Instance

Properties:

InstanceType: m5a.large

Tags:

- Key: Name

Value: "SelfServiceSecSecurityInstance"

SecurityGroupIds:

- !GetAtt "rInstanceSG.GroupId"

ImageId: !Ref 'LatestAmiId'

SubnetId: !Ref SubnetAID

IamInstanceProfile: !Ref InstanceRoleName

# This will update the Instance, pull Prowler, run and then write to the output to the S3 Bucket

UserData:

Fn::Base64: !Sub |

#!/bin/bash

yum -y update

yum groupinstall "Development Tools" -y

sudo yum -y install gcc openssl-devel bzip2-devel libffi-devel

wget https://www.python.org/ftp/python/3.9.16/Python-3.9.16.tgz

tar zxf Python-3.9.16.tgz

cd Python-3.9.16/

./configure --enable-optimizations

sudo make altinstall

export PATH=$PATH:/usr/local/bin

pip3.9 install prowler

cd

aws configure set default.region ${AWS::Region}

prowler aws | tee prowler-output.txt

aws s3 cp prowler-output.txt s3://${SelfServiceSecS3Bucket}

aws s3 cp output s3://${SelfServiceSecS3Bucket} --recursive

/sbin/init 0

1. **CloudFormation-Templates/SelfServiceSecS3.yml**

AWSTemplateFormatVersion: 2010-09-09

Description: Cloudformation templates to create a new Bucket for self service Security Reviews

Resources:

# Create a random name S3 bucket for the outputs to be placed

rCentralizedBucket:

DeletionPolicy: Delete

Type: 'AWS::S3::Bucket'

Properties:

BucketEncryption:

ServerSideEncryptionConfiguration:

- ServerSideEncryptionByDefault:

SSEAlgorithm: AES256

Outputs:

SelfServiceSecS3Bucket:

Description: 'Bucket Name'

Value: !Ref rCentralizedBucket

1. **CloudFormation-Templates/SelfServiceSecRansomware.yml**

AWSTemplateFormatVersion: 2010-09-09

Description: Cloudformation templates to create ransomware checks in the account. https://github.com/awslabs/aws-security-assessment-solution/

Parameters:

InstanceRoleARN:

Type: String

SelfServiceSecS3Bucket:

Type: String

TemplatesS3Bucket:

Type: String

Conditions:

CreateGovCloudResources: !Equals [ !Ref "AWS::Partition", aws ]

# Currently two features are not supported in GovCloud so we omit these two checks below with this partition check.

Resources:

libs:

Type: AWS::Lambda::LayerVersion

Properties:

LayerName: boto3

Description: Dependencies for the ransomware modules

Content:

S3Bucket: !Ref TemplatesS3Bucket

S3Key: "modules/RansomwareDetection/boto3\_layer.zip"

CompatibleRuntimes:

- python3.9

- python3.8

- python3.7

rLambdaCheckForLogging:

Type: "AWS::Lambda::Function"

Properties:

Code:

S3Bucket: !Ref TemplatesS3Bucket

S3Key: "modules/RansomwareDetection/check\_logging.py.zip"

FunctionName: "LambdaCheckForLogging"

Handler: "lambda\_function.lambda\_handler"

MemorySize: 256

Layers:

- !Ref libs

Role: !Ref InstanceRoleARN

Runtime: "python3.8"

Timeout: 900

DeadLetterConfig: !Ref "AWS::NoValue"

TracingConfig:

Mode: "PassThrough"

Environment:

Variables:

reportbucket: !Ref SelfServiceSecS3Bucket

xLambdaCheckForLogging:

Type: AWS::CloudFormation::CustomResource

Properties:

ServiceToken: !GetAtt rLambdaCheckForLogging.Arn

rLambdaCheckForRoute53DNSFirewall:

Type: "AWS::Lambda::Function"

Properties:

Code:

S3Bucket: !Ref TemplatesS3Bucket

S3Key: "modules/RansomwareDetection/check\_route53\_dns\_firewall.py.zip"

FunctionName: "LambdaCheckForRoute53DNSFirewall"

Handler: "lambda\_function.lambda\_handler"

MemorySize: 128

Layers:

- !Ref libs

Role: !Ref InstanceRoleARN

Runtime: "python3.8"

Timeout: 600

DeadLetterConfig: !Ref "AWS::NoValue"

TracingConfig:

Mode: "PassThrough"

Environment:

Variables:

reportbucket: !Ref SelfServiceSecS3Bucket

xLambdaCheckForRoute53DNSFirewall:

Type: AWS::CloudFormation::CustomResource

Properties:

ServiceToken: !GetAtt rLambdaCheckForRoute53DNSFirewall.Arn

rLambdaCheckForUnusedAccessKeys:

Type: "AWS::Lambda::Function"

Properties:

Code:

S3Bucket: !Ref TemplatesS3Bucket

S3Key: "modules/RansomwareDetection/check\_unused\_access\_keys.py.zip"

FunctionName: "LambdaCheckForUnusedAccessKeys"

Handler: "lambda\_function.lambda\_handler"

MemorySize: 128

Role: !Ref InstanceRoleARN

Runtime: "python3.8"

Timeout: 600

DeadLetterConfig: !Ref "AWS::NoValue"

TracingConfig:

Mode: "PassThrough"

Environment:

Variables:

reportbucket: !Ref SelfServiceSecS3Bucket

xLambdaCheckForUnusedAccessKeys:

Type: AWS::CloudFormation::CustomResource

Properties:

ServiceToken: !GetAtt rLambdaCheckForUnusedAccessKeys.Arn

#

rLambdaCheckForDNSSEC:

Type: "AWS::Lambda::Function"

Properties:

Code:

S3Bucket: !Ref TemplatesS3Bucket

S3Key: "modules/RansomwareDetection/check\_dnssec.py.zip"

FunctionName: "LambdaCheckForDNSSEC"

Handler: "lambda\_function.lambda\_handler"

MemorySize: 128

Layers:

- !Ref libs

Role: !Ref InstanceRoleARN

Runtime: "python3.8"

Timeout: 600

DeadLetterConfig: !Ref "AWS::NoValue"

TracingConfig:

Mode: "PassThrough"

Environment:

Variables:

reportbucket: !Ref SelfServiceSecS3Bucket

xLambdaCheckForDNSSEC:

Type: AWS::CloudFormation::CustomResource

Properties:

ServiceToken: !GetAtt rLambdaCheckForDNSSEC.Arn

#

rLambdaCheckFor2k8:

Type: "AWS::Lambda::Function"

Properties:

Code:

S3Bucket: !Ref TemplatesS3Bucket

S3Key: "modules/RansomwareDetection/check\_for\_outdated\_os.py.zip"

FunctionName: "LambdaCheckFor2k8"

Handler: "lambda\_function.lambda\_handler"

MemorySize: 128

Role: !Ref InstanceRoleARN

Runtime: "python3.8"

Timeout: 90

DeadLetterConfig: !Ref "AWS::NoValue"

TracingConfig:

Mode: "PassThrough"

Environment:

Variables:

reportbucket: !Ref SelfServiceSecS3Bucket

xLambdaCheckFor2k8:

Type: AWS::CloudFormation::CustomResource

Properties:

ServiceToken: !GetAtt rLambdaCheckFor2k8.Arn

rLambdaCheckForBackups:

Type: "AWS::Lambda::Function"

Properties:

Code:

S3Bucket: !Ref TemplatesS3Bucket

S3Key: "modules/RansomwareDetection/check\_backup.py.zip"

FunctionName: "LambdaCheckForBackups"

Handler: "lambda\_function.lambda\_handler"

MemorySize: 128

Role: !Ref InstanceRoleARN

Runtime: "python3.8"

Timeout: 90

DeadLetterConfig: !Ref "AWS::NoValue"

TracingConfig:

Mode: "PassThrough"

Environment:

Variables:

reportbucket: !Ref SelfServiceSecS3Bucket

xLambdaCheckForBackups:

Type: AWS::CloudFormation::CustomResource

Properties:

ServiceToken: !GetAtt rLambdaCheckForBackups.Arn

rLambdaCheckForGuardDuty:

Type: "AWS::Lambda::Function"

Properties:

Code:

S3Bucket: !Ref TemplatesS3Bucket

S3Key: "modules/RansomwareDetection/check\_guardduty.py.zip"

FunctionName: "LambdaCheckForGuardDuty"

Handler: "lambda\_function.lambda\_handler"

MemorySize: 128

Role: !Ref InstanceRoleARN

Runtime: "python3.8"

Timeout: 90

DeadLetterConfig: !Ref "AWS::NoValue"

TracingConfig:

Mode: "PassThrough"

Environment:

Variables:

reportbucket: !Ref SelfServiceSecS3Bucket

xLambdaCheckForGuardDuty:

Type: AWS::CloudFormation::CustomResource

Properties:

ServiceToken: !GetAtt rLambdaCheckForGuardDuty.Arn

rLambdaCheckForEBSSnapshot:

Type: "AWS::Lambda::Function"

Properties:

Code:

S3Bucket: !Ref TemplatesS3Bucket

S3Key: "modules/RansomwareDetection/check\_ebs\_snapshots.py.zip"

FunctionName: "LambdaCheckForEBSSnapShots"

Handler: "lambda\_function.lambda\_handler"

MemorySize: 128

Role: !Ref InstanceRoleARN

Runtime: "python3.8"

Timeout: 90

DeadLetterConfig: !Ref "AWS::NoValue"

TracingConfig:

Mode: "PassThrough"

Environment:

Variables:

reportbucket: !Ref SelfServiceSecS3Bucket

xLambdaCheckForEBSSnapshot:

Type: AWS::CloudFormation::CustomResource

Properties:

ServiceToken: !GetAtt rLambdaCheckForEBSSnapshot.Arn

rLambdaCheckForS3StaleAccess:

Type: "AWS::Lambda::Function"

Condition: CreateGovCloudResources

Properties:

Code:

S3Bucket: !Ref TemplatesS3Bucket

S3Key: "modules/RansomwareDetection/check\_s3\_stale\_access.py.zip"

FunctionName: "LambdaCheckForS3StaleAccess"

Handler: "lambda\_function.lambda\_handler"

MemorySize: 128

Role: !Ref InstanceRoleARN

Runtime: "python3.8"

Timeout: 180

DeadLetterConfig: !Ref "AWS::NoValue"

TracingConfig:

Mode: "PassThrough"

Environment:

Variables:

reportbucket: !Ref SelfServiceSecS3Bucket

xLambdaCheckForS3StaleAccess:

Type: AWS::CloudFormation::CustomResource

Condition: CreateGovCloudResources

Properties:

ServiceToken: !GetAtt rLambdaCheckForS3StaleAccess.Arn

rLambdaCheckForSSMUse:

Type: "AWS::Lambda::Function"

Condition: CreateGovCloudResources

Properties:

Code:

S3Bucket: !Ref TemplatesS3Bucket

S3Key: "modules/RansomwareDetection/check\_ssm\_used.py.zip"

FunctionName: "LambdaCheckForSSMUse"

Handler: "lambda\_function.lambda\_handler"

MemorySize: 128

Role: !Ref InstanceRoleARN

Runtime: "python3.8"

Timeout: 90

DeadLetterConfig: !Ref "AWS::NoValue"

TracingConfig:

Mode: "PassThrough"

Environment:

Variables:

reportbucket: !Ref SelfServiceSecS3Bucket

xLambdaCheckForSSMUse:

Type: AWS::CloudFormation::CustomResource

Condition: CreateGovCloudResources

Properties:

ServiceToken: !GetAtt rLambdaCheckForSSMUse.Arn