Practical Worksheet 4

Version: 1.0 Date: 12/04/2018 Author: David Glance

## Learning Objectives

1. IAM, creating user accounts with profiles
2. KMS Key Management System – creating keys and using the key for symmetric encryption
3. Using AES Encryption

## Technologies Covered

Ubuntu

AWS

AWS IAM

AWS KMS

AES Encryption

Python/Boto scripts

VirtualBox

**Note**: Do this from your VirtualBox VM – if you do it from any other platform (Windows, Mac – you will need to resolve any potential issues yourself)

## Background

The aim of this lab is to write a program that will:

[1] Create a user account and profile that only has access to S3 and to a specific key in KMS

[2] Use the credentials created in step [1] for you CloudStorage application

[3] Create a key in KMS and use it to encrypt files on the client before uploading to S3 and decrypt them after downloading from S3

[4] Implement AES using python and test the difference in performance between the KMS solution and the local one.

## IAM

## [Step 1] Create an IAM user

Using the IAM console, create a policy called CloudStoragePolicy

Allow the policy to access your specific bucket

The actions for buckets you need are:

[

"s3:CreateBucket",

"s3:ListBucket",

"s3:DeleteBucket"

]

The resource:

[

"arn:aws:s3:::<bucket-name>",

"arn:aws:s3:::<bucket-name>/\*"

]

The actions for objects:

[

"s3:PutObject",

"s3:GetObject",

"s3:PutObjectVersionAcl",

"s3:PutObjectAcl",

"s3:ListMultipartUploadParts",

"s3:ListObjects"

]

The resource only needs

"arn:aws:s3:::<bucket-name>"

Create a user and attach this policy to it. Download or save the API keys. The user does not need to be able to login.

Use the API keys for your CloudStorage application and verify that it works.

Change the bucket name you use in the program to make sure that it doesn’t work since your permissions are for a specific user.

To detect errors, you will need to catch errors using exception handling.

## [Step 2] AES Encryption using KMS

Using the IAM console, go into Encryption Keys and create a key

Choose an appropriate alias for the key

Add yourself as the key administrator and choose the user you created above as the key user

In your CloudStorage application add the ability to encrypt and decrypt the files you find using the KMS Client apis of boto3.

Encrypt only operates on 4 KB of data and so if you were to use this as a means of encrypting larger files, you would have to encrypt the file in chunks and reverse the process for decryption.

## [Step 3] AES Encryption using local python library pycryptodome

Create another version of your CloudStorage program that uses the python library pycryptodome to encrypt and decrypt your files

You can use the example code for doing this from <HERE>

## Submission and Quiz

Submit the python code files you wrote and take the quiz:

## Respond to the Quiz

[1] Creating a policy that specified permissions for users, groups or roles is the only way to control access to an S3 bucket

[A] True

[B] False

[2] AES is a block cypher that operates on data blocks organised as

[A] 8 \* 8 bytes

[B] 16 \* 16 bytes

[C] 4 \* 4 bytes

[D] 256 \* 256 bytes