

[illegible]

| | |
|-------------|---|
| Name | API Gateway Enumeration |
| URL | https://attackdefense.com/challengedetails?cid=2275 |
| Type | AWS Cloud Security : API Gateway |

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.

Solution:

Console Based Enumeration

Step 1: Click on the lab link to get AWS access credentials.

Access Credentials to your AWS lab Account

| | |
|--------------------------|---|
| Login URL | https://367163872066.signin.aws.amazon.com/console |
| Region | US East (N. Virginia) us-east-1 |
| Username | student |
| Password | Ad63Q9Q0vNQUEGui |
| Access Key ID | AKIAVK7FMCNBB2Z4VRPY |
| Secret Access Key | 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 |

Step 2: Sign in to AWS console.

Sign in as IAM user

Account ID (12 digits) or account alias

367163872066

IAM user name

student

Password

●●●●●●●●●●●●●●●●

Sign in

[Sign in using root user email](#)

[Forgot password?](#)

Build, train, and deploy ML models quickly

Get ML models into production faster with less effort and lower cost with Amazon SageMaker

[Learn more »](#)

aws machine learning

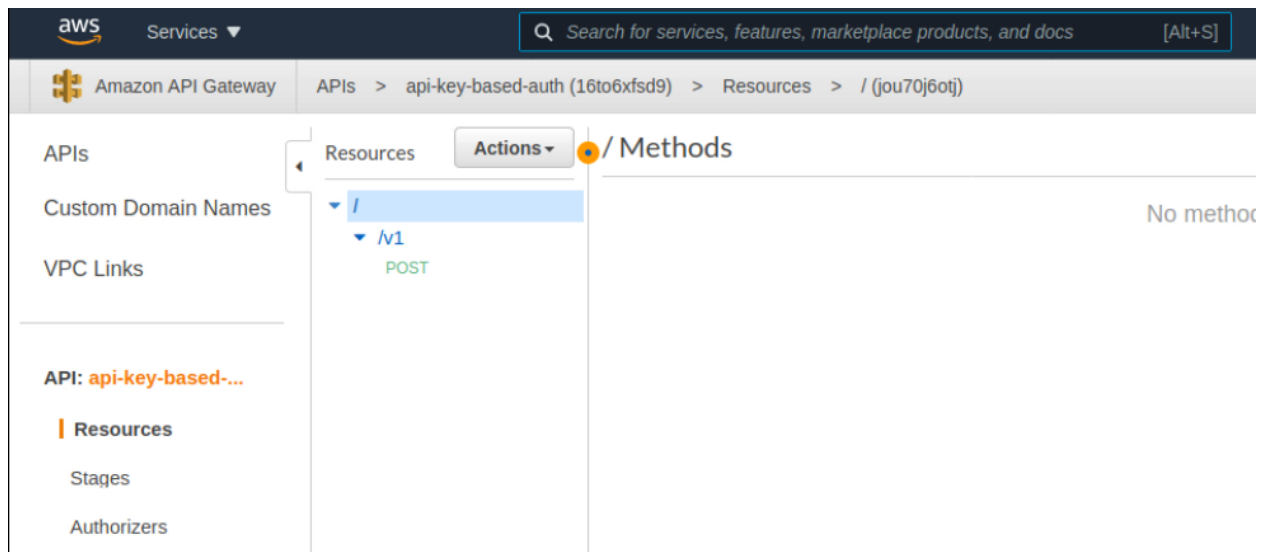


Step 3: Navigate to the API gateway dashboard.

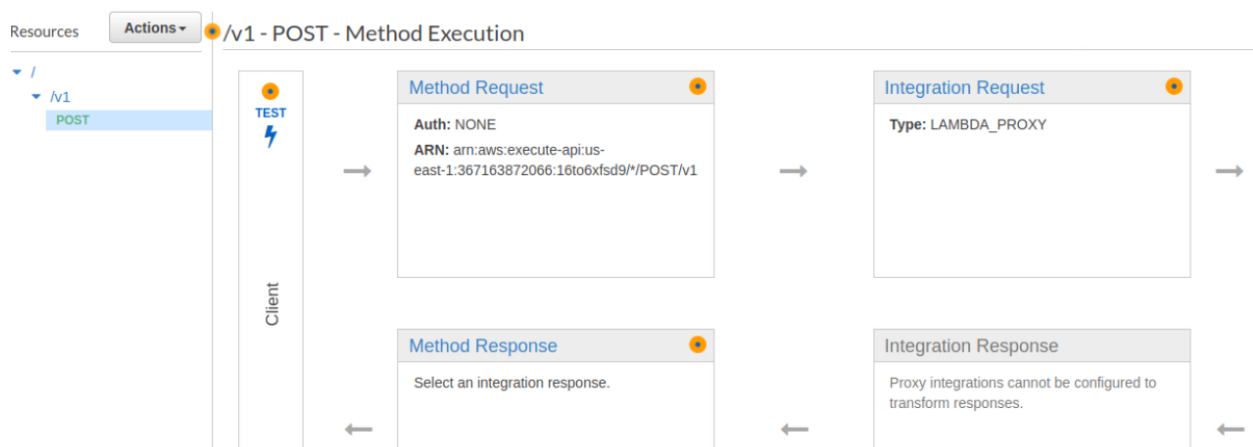
The screenshot shows the AWS API Gateway console. The left sidebar has a search bar and a list of links: "APIs", "Custom domain names", and "VPC links". The main content area is titled "APIs (3)" and contains a search bar with the text "Find APIs". Below the search bar is a table with the following columns: "Name", "Description", "ID", and "Protocol". The table lists three APIs:

| Name | Description | ID | Protocol |
|------------------------------------|-------------|------------|----------|
| api-key-based-auth | | 16to6xfsd9 | REST |
| demo | | byn61y7sl7 | REST |
| iam-based-auth | | fypgf9z86l | REST |

Step 4: Click on API name to open API details.



Step 5: Click on Resources and enumerate Method Request, Integration request and proxy.




← Method Execution /v1 - POST - Method Request

Provide information about this method's authorization settings and the parameters it can receive.

Settings

Authorization NONE  

Request Validator NONE  

API Key Required true 

▶ URL Query String Parameters

▶ HTTP Request Headers

▶ Request Body

▶ SDK Settings

← Method Execution /v1 - POST - Integration Request

Provide information about the target backend that this method will call and whether the incoming request data should be modified.

Integration type ☒ Lambda Function ⓘ
☐ HTTP ⓘ
☐ Mock ⓘ
☐ AWS Service ⓘ
☐ VPC Link ⓘ

Use Lambda Proxy integration ☒ ⓘ

Lambda Region us-east-1 ✎

Lambda Function demo-lambda ✎

Execution role ✎

Invoke with caller credentials ☐ ⓘ

Credentials cache Do not add caller credentials to cache key ✎

Use Default Timeout ☒ ⓘ

Step 6: Click on Stages on the left panel to enumerate stages for API.

Amazon API Gateway

APIs > api-key-based-auth (16to6xfsd9) > Stages > dev > /v1 > POST

APIs

Custom Domain Names

VPC Links

API: api-key-based-...

Resources

Stages

Stages Create

dev

/

/v1

POST

dev - POST - /v1

Invoke URL: <https://16to6xfsd9.execute-api.us-east-1.amazonaws.com/dev/v1>

Use this page to override the dev stage settings for the POST to /v1 method.

Settings ☒ Inherit from stage
☐ Override for this method

Step 7: Click on authorizers to check authorizers for the API.

VPC Links

API: api-key-based-...

Resources

Stages

| Authorizers

Gateway Responses

Authorizers

Authorizers enable you to control access to your APIs using Amazon Cognito User Pools or a Lambda function.

+ Create New Authorizer

Step 8: Click on Resources on the left panel to enumerate resources for API.

APIs

Custom Domain Names

VPC Links

API: api-key-based-...

Resources

Stages

Authorizers

Gateway Responses

Models

| Resource Policy

Resource Policy

Configure access control to this private API using a Resource Policy. Access can be controlled by IAM condition elements, IP range. If the Principal in the policy is set to *, other authorization types can be used alongside the resource policy. If the P AWS_IAM auth, including unsecured resources. Changes to this policy require a deployment to take effect. [Learn more.](#)

1

Step 9: Check API's usage plans and check API keys. Go to the API Keys section in the left panel. Click on the API Key name listed.

API Keys Actions ▾

Search...

Lab1GW2

ID cjppyvuz4b

Name Lab1GW2

API key [Show](#)

Description Managed by Terraform

Enabled Enabled ⓘ

Associated Usage Plans

[Add to Usage Plan](#)

| Usage Plan | API |
|-------------------------------|------------------------------------|
| my_usage_plan | api-key-based-auth |

Lab1GW2

ID cjppyvuz4b

Name Lab1GW2

API key TZiZRdZ7ga4g7wkiSzDVA5dCbFsulAcu8em9wMph

Description Managed by Terraform

Enabled Enabled ⓘ

Step 10: Similarly enumerate other apis on the AWS account.

CLI Based Enumeration

Step 1: Click on the lab link to get AWS access credentials.

Access Credentials to your AWS lab Account

| | |
|-------------------|---|
| Login URL | https://367163872066.signin.aws.amazon.com/console |
| Region | US East (N. Virginia) us-east-1 |
| Username | student |
| Password | Ad63Q9Q0vNQUEGui |
| Access Key ID | AKIAVK7FMCNBB2Z4VRPY |
| Secret Access Key | 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 |

Step 2: Configure AWS CLI with AWS access keys.

Command: aws configure

```
< root@Kali ~# aws configure
AWS Access Key ID [*****VTGY]: AKIA2DM3DTVBTSD3IG4
AWS Secret Access Key [*****FAMU]: som72hhSpPMmSvVUcf4NNJN85sH5QuYwyctB0txH
Default region name [us-east-1]:
Default output format [None]:
< root@Kali ~#
```

Step 3: Check API settings for the account.

Command: aws apigateway get-account

```
< root@Kali ~ ➤ aws apigateway get-account
{
  "throttleSettings": {
    "burstLimit": 5000,
    "rateLimit": 10000.0
  },
  "features": [
    "UsagePlans"
  ],
  "apiKeyVersion": "4"
}
< root@Kali ~ ➤
```

Step 4: List API's on the AWS account.

Command: aws apigateway get-rest-apis

```
< root@Kali ~ ➤ aws apigateway get-rest-apis
{
  "items": [
    {
      "id": "4xn6jt50a6",
      "name": "iam-based-auth",
      "createdDate": 1613658554,
      "apiKeySource": "HEADER",
      "endpointConfiguration": {
        "types": [
          "EDGE"
        ]
      },
      "disableExecuteApiEndpoint": false
    },
    {
      "id": "57z6f0kgxi",
      "name": "demo",
      "createdDate": 1613658554,
      "apiKeySource": "HEADER",
      "endpointConfiguration": {
        "types": [
          "EDGE"
        ]
      }
    }
  ]
}
```

Step 5: Check API resources

Command: aws apigateway get-resources --rest-api-id 4xn6jt50a6

```
< root@Kali ~# aws apigateway get-resources --rest-api-id 4xn6jt50a6
{
  "items": [
    {
      "id": "ivmxmfydif",
      "path": "/"
    },
    {
      "id": "lmxnuv",
      "parentId": "ivmxmfydif",
      "pathPart": "v1",
      "path": "/v1",
      "resourceMethods": {
        "GET": {}
      }
    }
  ]
}
```

Step 6: Enumerate API resource HTTP methods.

Command: aws apigateway get-method --rest-api-id 4xn6jt50a6 --http-method GET --resource-id lmxnuv

```
< root@Kali ~# aws apigateway get-method --rest-api-id 4xn6jt50a6 --http-method GET --resource-id lmxnuv
{
  "httpMethod": "GET",
  "authorizationType": "AWS_IAM",
  "apiKeyRequired": false,
  "methodIntegration": {
    "type": "AWS_PROXY",
    "httpMethod": "POST",
    "uri": "arn:aws:apigateway:us-east-1:lambda:path/2015-03-31/functions/arn:aws:lambda:us-east-1:694499384643:function:demo-lambda/invocations",
    "requestParameters": {},
    "requestTemplates": {},
    "passthroughBehavior": "WHEN_NO_MATCH",
    "timeoutInMillis": 29000,
    "cacheNamespace": "lmxnuv",
    "cacheKeyParameters": []
  }
}
```

Step 7: Check authorizers for API.

Command: aws apigateway get-authorizers --rest-api-id yz2duibd7a

```

root@Kali ~$ aws apigateway get-authorizers --rest-api-id yz2duibd7a
{
  "items": []
}
root@Kali ~$

```

Step 8: Check API keys for the AWS account.

Command: aws apigateway get-api-keys

```

root@Kali ~$ aws apigateway get-api-keys
{
  "items": [
    {
      "id": "pex0pljl5h",
      "name": "Lab1GW2",
      "description": "Managed by Terraform",
      "enabled": true,
      "createdDate": 1613658554,
      "lastUpdatedDate": 1613658554,
      "stageKeys": []
    }
  ]
}
root@Kali ~$

```

Step 9: Get API key details.

Command: aws apigateway get-api-key --api-key pex0pljl5h

```

root@Kali ~$ aws apigateway get-api-key --api-key pex0pljl5h
{
  "id": "pex0pljl5h",
  "name": "Lab1GW2",
  "description": "Managed by Terraform",
  "enabled": true,
  "createdDate": 1613658554,
  "lastUpdatedDate": 1613658554,
  "stageKeys": [],
  "tags": {}
}
root@Kali ~$

```

| | |
|-----------|---------------|
| Login URL | https://0044 |
| Region | US East (N. V |
| Username | student |

Step 10: Get API key values.

Command: aws apigateway get-api-key --api-key pex0pljl5h --include-value

```
> root@Kali ~# aws apigateway get-api-key --api-key pex0pljl5h --include-value
{
  "id": "pex0pljl5h",
  "value": "2UkSM508yl8UoD0wUSZFR4n0e5tdBU28Cxuj2f2",
  "name": "Lab1GW2",
  "description": "Managed by Terraform",
  "enabled": true,
  "createdDate": 1613658554,
  "lastUpdatedDate": 1613658554,
  "stageKeys": [],
  "tags": {}
}
> root@Kali ~#
```

Step 11: Enumerate API client certificates.

Command: aws apigateway get-client-certificates

```
File  Actions  Edit  View  Help
> root@Kali ~# aws apigateway get-client-certificates
{
  "items": []
}
> root@Kali ~#
```

References:

1. AWS CLI (<https://docs.aws.amazon.com/cli/latest/reference/>)