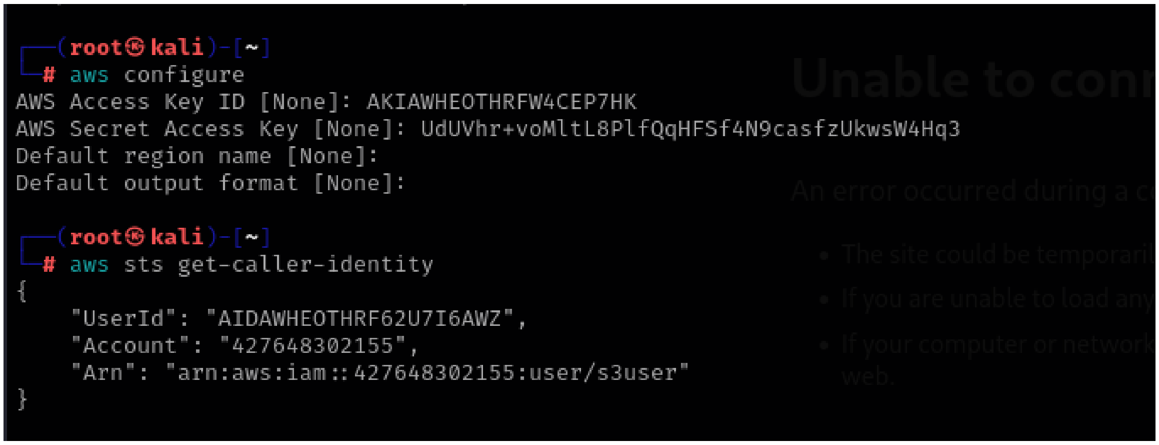
AWS S3 Enumeration Lab Report

1. AWS CLI Configuration & Caller Identity

This screenshot shows the configuration of the AWS CLI with provided credentials, followed by the `aws sts get-caller-identity` command. It reveals the UserID, Account number, and IAM ARN of the user. This is the IAM identity under which further actions are taken.



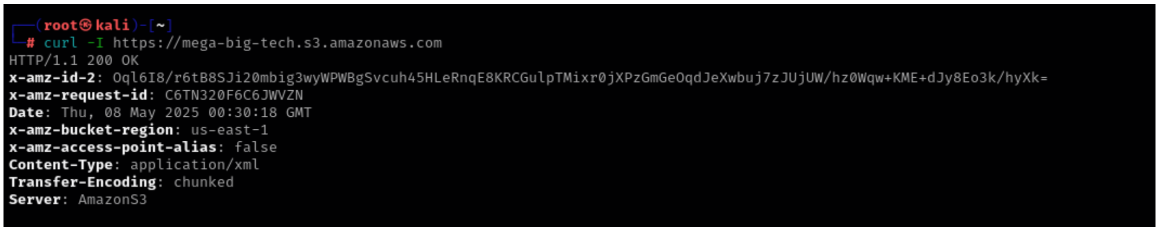
```
┌──(root㉿kali)-[~]
└─# aws configure
AWS Access Key ID [None]: AKIAWHEOTHRFW4CEP7HK
AWS Secret Access Key [None]: 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
Default region name [None]:
Default output format [None]:

┌──(root㉿kali)-[~]
└─# aws sts get-caller-identity
{
    "UserId": "AIDAWHEOTHRF62U7I6AWZ",
    "Account": "427648302155",
    "Arn": "arn:aws:iam::427648302155:user/s3user"
}
```

Unable to conn
An error occurred during a c
The site could be temporaril
If you are unable to load any
If your computer or network
web.

2. cURL Request to S3 Bucket

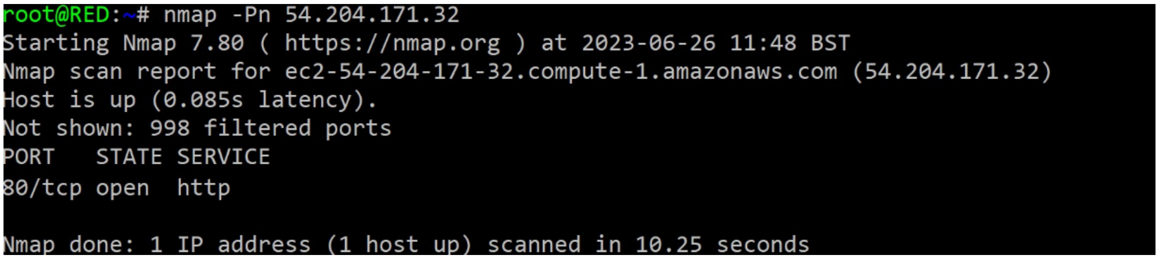
A `curl -I` command checks the headers of the target S3 bucket URL. The response confirms the bucket exists and reveals the region `us-east-1` through the `x-amz-bucket-region` header.



```
┌──(root㉿kali)-[~]
└─# curl -I https://mega-big-tech.s3.amazonaws.com
HTTP/1.1 200 OK
x-amz-id-2: Oql6I8/r6tB8SJi20mbig3wyWPWBgSvcuh45HLeRnqE8KRCGulpTMixr0jXPzGmGeOqdJeXwbuj7zJUjUW/hz0Wqw+KME+dJy8Eo3k/hyXk=
x-amz-request-id: C6TN320F6C6JWVZN
Date: Thu, 08 May 2025 00:30:18 GMT
x-amz-bucket-region: us-east-1
x-amz-access-point-alias: false
Content-Type: application/xml
Transfer-Encoding: chunked
Server: AmazonS3
```

3. Nmap Scan of Target IP

An Nmap scan is conducted on the target IP address, confirming port 80 is open. This indicates a web server is active and can be further investigated via a browser.

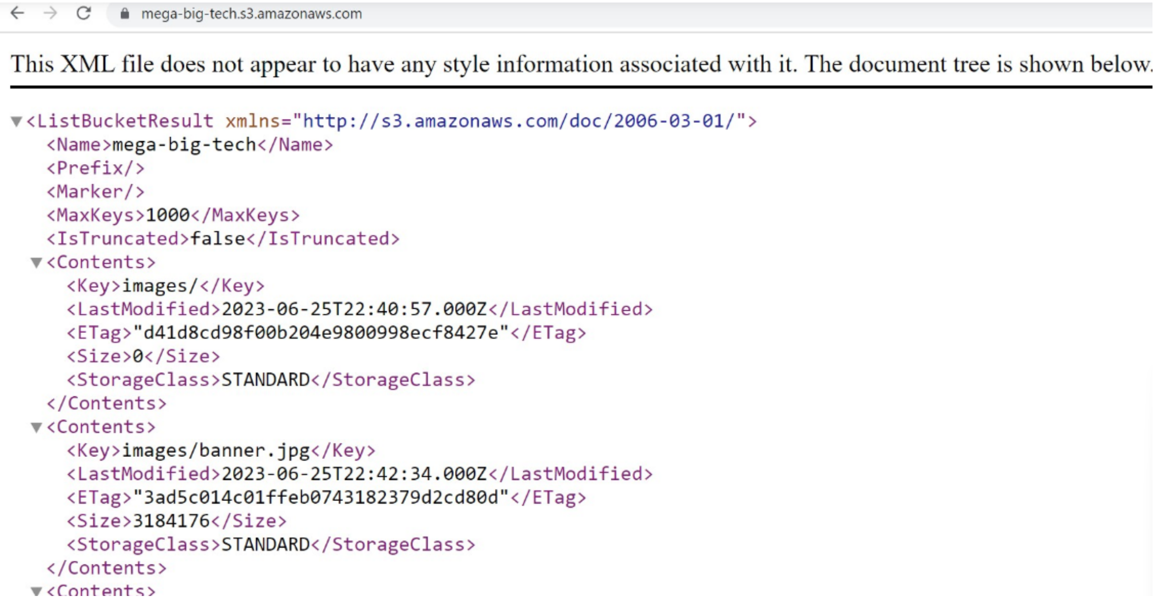


```
root@RED:~# nmap -Pn 54.204.171.32
Starting Nmap 7.80 ( https://nmap.org ) at 2023-06-26 11:48 BST
Nmap scan report for ec2-54-204-171-32.compute-1.amazonaws.com (54.204.171.32)
Host is up (0.085s latency).
Not shown: 998 filtered ports
PORT   STATE SERVICE
80/tcp open  http

Nmap done: 1 IP address (1 host up) scanned in 10.25 seconds
```

4. S3 Bucket Contents via Browser

Using a browser to access the S3 bucket confirms that directory listing is enabled. The files such as `images/banner.jpg` are publicly visible, indicating misconfigured bucket permissions.

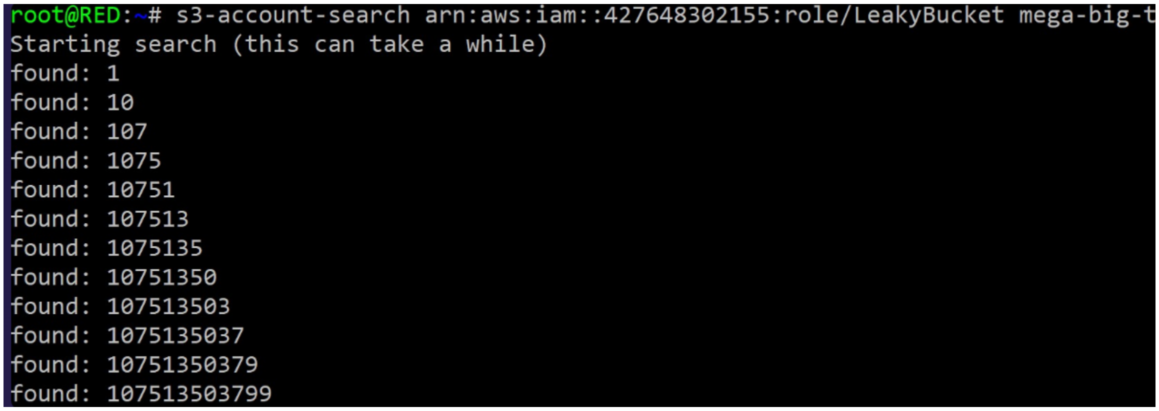


This XML file does not appear to have any style information associated with it. The document tree is shown below.

```
▼<ListBucketResult xmlns="http://s3.amazonaws.com/doc/2006-03-01/">
   <Name>mega-big-tech</Name>
   <Prefix/>
   <Marker/>
   <MaxKeys>1000</MaxKeys>
   <IsTruncated>false</IsTruncated>
 ▼<Contents>
     <Key>images/</Key>
     <LastModified>2023-06-25T22:40:57.000Z</LastModified>
     <ETag>"d41d8cd98f00b204e9800998ecf8427e"</ETag>
     <Size>0</Size>
     <StorageClass>STANDARD</StorageClass>
   </Contents>
 ▼<Contents>
     <Key>images/banner.jpg</Key>
     <LastModified>2023-06-25T22:42:34.000Z</LastModified>
     <ETag>"3ad5c014c01ffeb0743182379d2cd80d"</ETag>
     <Size>3184176</Size>
     <StorageClass>STANDARD</StorageClass>
   </Contents>
 ▼<Contents>
```

5. Brute Force AWS Account ID with s3-account-search

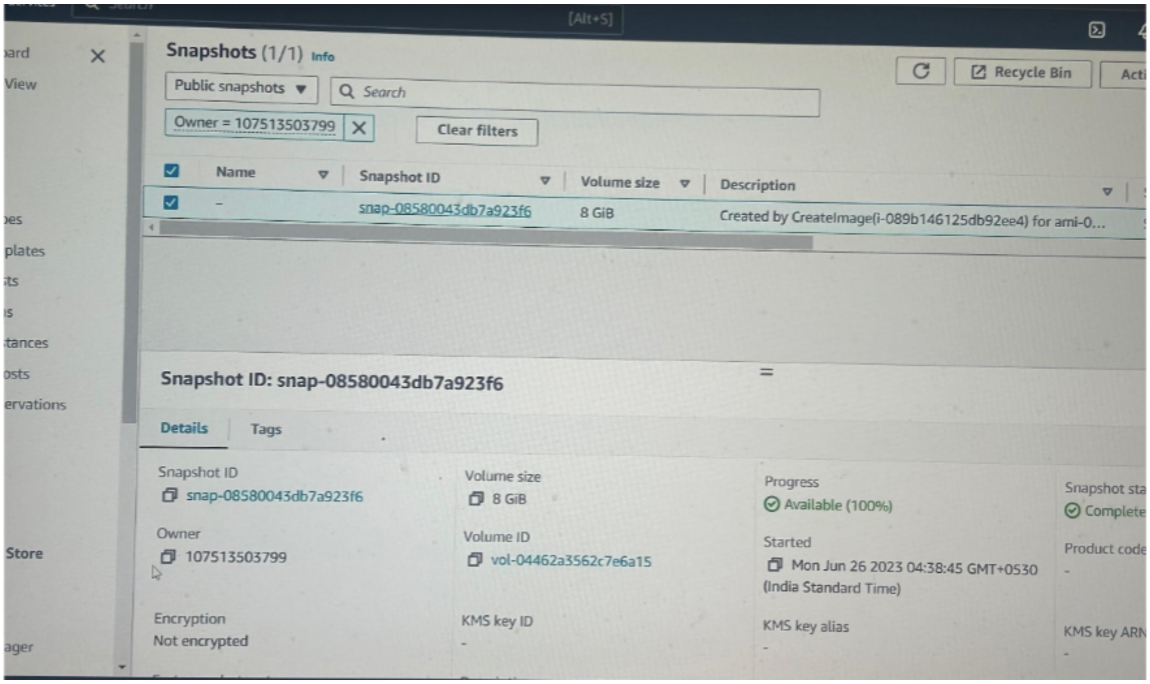
The tool `s3-account-search` is used to brute-force the AWS account ID linked to the S3 bucket using policy wildcard techniques. The account ID `107513503799` is successfully discovered.



```
root@RED:~# s3-account-search arn:aws:iam::427648302155:role/LeakyBucket mega-big-t
Starting search (this can take a while)
found: 1
found: 10
found: 107
found: 1075
found: 10751
found: 107513
found: 1075135
found: 10751350
found: 107513503
found: 1075135037
found: 10751350379
found: 107513503799
```

6. Discovery of Public EBS Snapshot

Logging into AWS and searching public snapshots with the discovered account ID reveals an 8 GiB EBS snapshot that is publicly accessible. This represents a significant data exposure.



Snapshots (1/1)

Public snapshots ▼

Owner = 107513503799

Clear filters

☑	Name	Snapshot ID	Volume size	Description
☑	-	snap-08580043db7a923f6	8 GiB	Created by CreateImage(i-089b146125db92ee4) for ami-0...

Snapshot ID: snap-08580043db7a923f6

Snapshot ID: snap-08580043db7a923f6

Owner: 107513503799

Encryption: Not encrypted

Volume size: 8 GiB

Volume ID: vol-04462a3562c7e6a15

KMS key ID: -

Progress: Available (100%)

Started: Mon Jun 26 2023 04:38:45 GMT+0530 (India Standard Time)

KMS key alias: -

Snapshot sta: Complete

Product code: -

KMS key ARN: -

Final Reflection and Commentary

Interesting Screenshots and Commentary from Your Investigation:

Throughout the lab, we successfully exploited a misconfigured Amazon S3 bucket and publicly exposed EBS snapshots. The screenshots document every major stage, from AWS CLI configuration to identifying the S3 bucket, brute-forcing the AWS account ID, and discovering an unencrypted EBS snapshot available to the public. These highlight both common misconfigurations and the power of enumeration tools in offensive cloud security.

What Was Learned About Security, Accounts, and S3 Bucket Exploits:

1**. S3 Buckets Can Be Publicly Accessible:**
 - If not properly configured, S3 buckets can leak sensitive files and expose organizations to data breaches. Misconfigured access policies allowed us to list and access objects directly in the browser.

2**. AWS Account ID Enumeration:**
 - Tools like `s3-account-search` can exploit policy wildcard behavior to enumerate account IDs. These IDs are often considered non-sensitive but can be leveraged to find further exposed resources like public snapshots.

3**. Snapshot Exposure**:
 - Public EBS snapshots can be copied and analyzed offline, potentially leaking sensitive disk contents. In this lab, we found one such snapshot belonging to the compromised AWS account.

How to Prevent This From Happening Again:

1. **Enable Proper Bucket Policies**:
 - Make use of bucket policies and ACLs to ensure that only authorized users can list and access objects. Use `Block Public Access` settings aggressively.

2. **Implement S3 Access Logging:**
 - Track who is accessing what and when. Enable data events in CloudTrail for deeper visibility into S3 actions.

3**. Avoid Public Snapshots:**
 - Never set EBS or RDS snapshots to public unless absolutely required. Audit these settings regularly with automation or AWS Config rules.

4. **Limit IAM Role Assumption**:
 - Use the principle of least privilege for IAM users and roles. Avoid granting broad AssumeRole permissions and monitor role usage.

5. **Regular Security Audits:**
  - Continuously review access configurations, logs, and snapshot visibility using tools like AWS Trusted Advisor, GuardDuty, and third-party security platforms.

This lab underscored the importance of defense-in-depth cloud security practices and the risks of misconfiguration in a cloud-first infrastructure.