Capstone Report Lab

Table of contents

List of Figures

List of Tables

1. Executive Summary

This capstone project demonstrates a full adversary simulation using LocalStack as a mock AWS environment and Pacu as the red team tool. The goal was to simulate various attack techniques such as reconnaissance, exploitation, privilege escalation, persistence, and exfiltration in a controlled environment. This report includes all the commands executed during the setup, resource creation, and exploitation phases, along with recommendations for blue team defenses.

2. Setup and Resource Creation in LocalStack

export AWS_ACCESS_KEY_ID=test

export AWS_SECRET_ACCESS_KEY=test

export AWS_DEFAULT_REGION=us-east-1

export AWS_ENDPOINT_URL=http://localhost:4566

Create an S3 bucket and upload a dummy file

aws --endpoint-url=$AWS_ENDPOINT_URL s3 mb s3://mock-bucket

echo "This is a test file for exfiltration." > dummy.txt

aws --endpoint-url=$AWS_ENDPOINT_URL s3 cp dummy.txt s3://mock-bucket/dummy.txt

Create IAM role and user

aws --endpoint-url=$AWS_ENDPOINT_URL iam create-role --role-name mock-role --assume-role-policy-document '{"Version":"2012-10-17","Statement":[{"Effect":"Allow","Principal":{"Service":"ec2.amazonaws.com"},"Action":"sts:AssumeRole"}]}'

aws --endpoint-url=$AWS_ENDPOINT_URL iam create-user --user-name mock-user

Create EC2 volume

aws --endpoint-url=$AWS_ENDPOINT_URL ec2 create-volume --availability-zone us-east-1a --size 1

Create CloudWatch log group

aws --endpoint-url=$AWS_ENDPOINT_URL logs create-log-group --log-group-name /mock/log/group

Create Lambda function

```
echo -e 'def lambda_handler(event, context):\n    return {"statusCode": 200}' > lambda_function.py
zip dummy.zip lambda_function.py
aws --endpoint-url=$AWS_ENDPOINT_URL lambda create-function --function-name mock-function \
--runtime python3.8 --role arn:aws:iam::000000000000:role/mock-role \
--handler lambda_function.lambda_handler --zip-file fileb://dummy.zip
```

Create SNS topic

```
aws --endpoint-url=$AWS_ENDPOINT_URL sns create-topic --name mock-topic
```

Create DynamoDB table

```
aws --endpoint-url=$AWS_ENDPOINT_URL dynamodb create-table \
  --table-name mock-table \
  --attribute-definitions AttributeName=Id,AttributeType=S \
  --key-schema AttributeName=Id,KeyType=HASH \
  --provisioned-throughput ReadCapacityUnits=5,WriteCapacityUnits=5
```

```
┌──(venv)─(kali㉿vbox)-[~]
└─$ export AWS_ACCESS_KEY_ID=test

┌──(venv)─(kali㉿vbox)-[~]
└─$ export AWS_SECRET_ACCESS_KEY=test

┌──(venv)─(kali㉿vbox)-[~]
└─$ export AWS_DEFAULT_REGION=us-east-1

┌──(venv)─(kali㉿vbox)-[~]
└─$ export AWS_ENDPOINT_URL=http://localhost:4566

┌──(venv)─(kali㉿vbox)-[~]
└─$ aws --endpoint-url=$AWS_ENDPOINT_URL s3 mb s3://mock-bucket
make_bucket: mock-bucket

┌──(venv)─(kali㉿vbox)-[~]
└─$ echo "This is a test file for exfiltration." > dummy.txt

┌──(venv)─(kali㉿vbox)-[~]
└─$ aws --endpoint-url=$AWS_ENDPOINT_URL s3 cp dummy.txt s3://mock-bucket/dummy.txt
upload: ./dummy.txt to s3://mock-bucket/dummy.txt

┌──(venv)─(kali㉿vbox)-[~]
└─$ aws --endpoint-url=$AWS_ENDPOINT_URL iam create-role --role-name mock-role --assume-role-policy-
document '{"Version":"2012-10-17","Statement":[{"Effect":"Allow","Principal":{"Service":"ec2.amazona
ws.com"},"Action":"sts:AssumeRole"}]}'
{
    "Role": {
        "Path": "/",
        "RoleName": "mock-role",
        "RoleId": "AROAQAAAAAAAAL64R5H2C",
        "Arn": "arn:aws:iam::000000000000:role/mock-role",
        "CreateDate": "2025-09-14T18:56:06.998594+00:00",
        "AssumeRolePolicyDocument": {
            "Version": "2012-10-17",
            "Statement": [
                {
                    "Effect": "Allow",
                    "Principal": {
                        "Service": "ec2.amazonaws.com"
                    },
                    "Action": "sts:AssumeRole"
                }
            ]
        }
    }
}
```

```
┌──(venv)─(kali㉿vbox)-[~]
└─$ aws --endpoint-url=$AWS_ENDPOINT_URL iam create-user --user-name mock-user
{
    "User": {
        "Path": "/",
        "UserName": "mock-user",
        "UserId": "nhn87fqknol85e8k9xkq",
        "Arn": "arn:aws:iam::000000000000:user/mock-user",
        "CreateDate": "2025-09-14T18:56:19.397764+00:00"
    }
}
```

```
┌──(venv)─(kali㉿vbox)-[~]
└─$ aws --endpoint-url=$AWS_ENDPOINT_URL ec2 create-volume --availability-zone us-east-1a --size 1
{
    "VolumeType": "gp2",
    "VolumeId": "vol-f4ade234e42c2f83d",
    "Size": 1,
    "SnapshotId": "",
    "AvailabilityZone": "us-east-1a",
    "State": "creating",
    "CreateTime": "2025-09-14T18:57:07+00:00",
    "Encrypted": false
}
```

```
┌──(venv)─(kali㉿vbox)-[~]
└─$ aws --endpoint-url=$AWS_ENDPOINT_URL logs create-log-group --log-group-name /mock/log/group

┌──(venv)─(kali㉿vbox)-[~]
└─$ echo -e 'def lambda_handler(event, context):\n    return {"statusCode": 200}' > lambda_function.
py

┌──(venv)─(kali㉿vbox)-[~]
└─$ zip dummy.zip lambda_function.py
  adding: lambda_function.py (stored 0%)
```

```
┌──(venv)─(kali㉿vbox)-[~]
└─$ aws --endpoint-url=$AWS_ENDPOINT_URL lambda create-function --function-name mock-function \
--runtime python3.8 --role arn:aws:iam::000000000000:role/mock-role \
--handler lambda_function.lambda_handler --zip-file fileb://dummy.zip
{
    "FunctionName": "mock-function",
    "FunctionArn": "arn:aws:lambda:us-east-1:000000000000:function:mock-function",
    "Runtime": "python3.8",
    "Role": "arn:aws:iam::000000000000:role/mock-role",
    "Handler": "lambda_function.lambda_handler",
    "CodeSize": 253,
    "Description": "",
    "Timeout": 3,
    "MemorySize": 128,
    "LastModified": "2025-09-14T18:58:25.928694+0000",
    "CodeSha256": "OS/RFt2LaXTnja+GDNmIF5KjMfzI4q7JfbPoHYE/EvY=",
    "Version": "$LATEST",
    "TracingConfig": {
        "Mode": "PassThrough"
    },
    "RevisionId": "99040acf-e3b1-4765-a2f7-80043dcd7a94",
    "State": "Pending",
    "StateReason": "The function is being created.",
    "StateReasonCode": "Creating",
    "PackageType": "Zip",
    "Architectures": [
        "x86_64"
    ],
    "EphemeralStorage": {
        "Size": 512
    },
    "SnapStart": {
        "ApplyOn": "None",
        "OptimizationStatus": "Off"
    },
    "RuntimeVersionConfig": {
        "RuntimeVersionArn": "arn:aws:lambda:us-east-1::runtime:8eeff65f6809a3ce81507fe733fe09b83589
9b99481ba22fd75b5a7338290ec1"
    },
    "LoggingConfig": {
        "LogFormat": "Text",
        "LogGroup": "/aws/lambda/mock-function"
    }
}
```

```
┌──(venv)─(kali㉿vbox)-[~]
└─$ aws --endpoint-url=$AWS_ENDPOINT_URL sns create-topic --name mock-topic
{
    "TopicArn": "arn:aws:sns:us-east-1:000000000000:mock-topic"
}
```

```
┌──(venv)─(kali㉿vbox)-[~]
└─$ aws --endpoint-url=$AWS_ENDPOINT_URL dynamodb create-table \
    --table-name mock-table \
    --attribute-definitions AttributeName=Id,AttributeType=S \
    --key-schema AttributeName=Id,KeyType=HASH \
    --provisioned-throughput ReadCapacityUnits=5,WriteCapacityUnits=5
{
    "TableDescription": {
        "AttributeDefinitions": [
            {
                "AttributeName": "Id",
                "AttributeType": "S"
            }
        ],
        "TableName": "mock-table",
        "KeySchema": [
            {
                "AttributeName": "Id",
                "KeyType": "HASH"
            }
        ],
        "TableStatus": "ACTIVE",
        "CreationDateTime": "2025-09-15T00:29:40.059000+05:30",
        "ProvisionedThroughput": {
            "NumberOfDecreasesToday": 0,
            "ReadCapacityUnits": 5,
            "WriteCapacityUnits": 5
        },
        "TableSizeBytes": 0,
        "ItemCount": 0,
        "TableArn": "arn:aws:dynamodb:us-east-1:000000000000:table/mock-table",
        "TableId": "4b47f8d3-f2f0-4510-a1d1-13997760c9f2",
        "DeletionProtectionEnabled": false
    }
}
```

Figure 2.1 Shows all commands executed in localstack

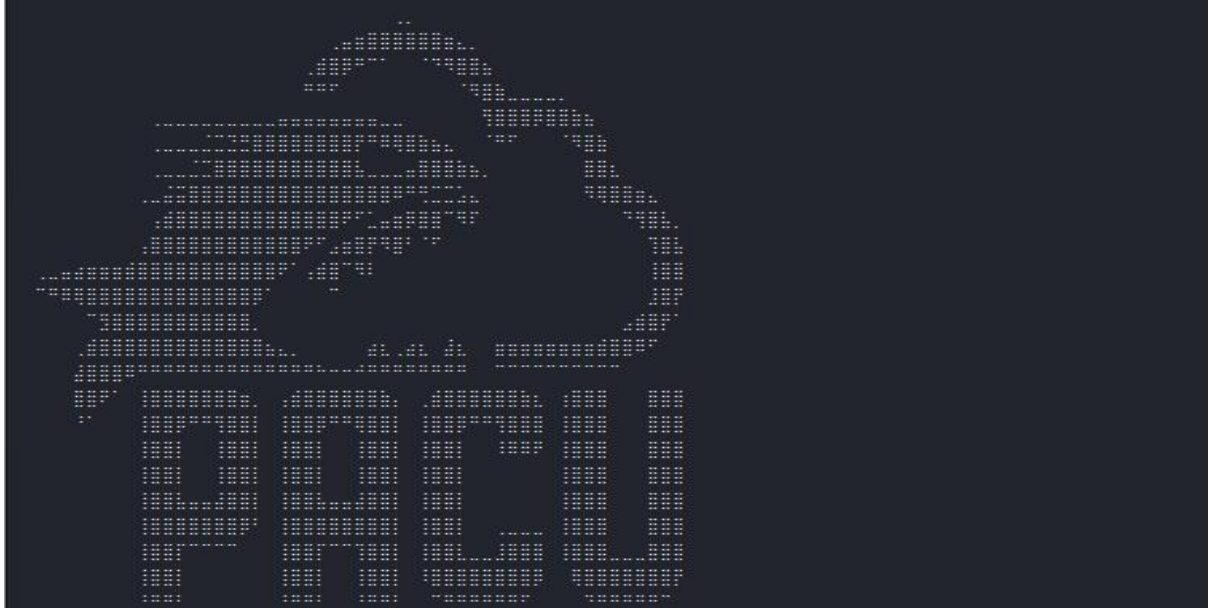
```
┌──(venv)─(kali㉿vbox)-[~]
└─$ ls
adminpass            Desktop    Downloads  dummy.zip          LICENSE      localstack  pacu      Public     venv     Vide
capstone-localstack  Documents  dummy.txt  lambda_function.py LICENSE.txt  Music       Pictures  README.md  VERSION
```

Figure 2.2 Shows data successfully saved

2.1. Pacu Commands Executed

```
┌──(venv)─(kali㉿vbox)-[~]
└─$ docker run --rm -it --name pacu7 --network redteam-net \
  -e AWS_ACCESS_KEY_ID=test \
  -e AWS_SECRET_ACCESS_KEY=test \
  -e AWS_REGION=us-east-1 \
  local-pacu

No database found at /root/.local/share/pacu/sqlite.db
Database created at /root/.local/share/pacu/sqlite.db
```



```
Version: 1.6.1
What would you like to name this new session? pacu7
Session pacu7 created.
```

```
Boto3/1.4.7 Python/2.7.0 Java/1.8.0_112 Botocore/1.7.21
Pacu (pacu7:No Keys Set) > set_keys
Setting AWS Keys ...
Press enter to keep the value currently stored.
Enter the letter C to clear the value, rather than set it.
If you enter an existing key_alias, that key's fields will be updated instead of added.
Key alias must be at least 2 characters

Key alias [None]: test
Access key ID [None]: test
Secret access key [None]: test
Session token (Optional - for temp AWS keys only) [None]: test

Keys saved to database.
```

Figure 2.3 Shows pacu setup

Pacu is ideal for controlled security assessments where the goal is to test security postures, simulate real-world attacks, and identify weaknesses in IAM configurations. However, for precise resource management and endpoint-specific operations, the AWS CLI or SDKs are recommended.

3. Log Table

Timestamp	*Command*	*Action Description*	*Notes*
2025-09-14 18:55:00	aws --endpoint-url=http://localhost:4566 s3 mb s3://mock-bucket	Created S3 bucket mock-bucket	Mock bucket for exfil testing
2025-09-14 18:55:30	echo "This is a test file for exfiltration." > dummy.txt	Created dummy exfiltration file	Local file prep
2025-09-14 18:55:35	aws --endpoint-url=http://localhost:4566 s3 cp dummy.txt s3://mock-bucket/dummy.txt	Uploaded dummy file to S3 bucket	Data staged for exfil
2025-09-14 18:56:00	aws --endpoint-url=http://localhost:4566 iam create-role --role-name mock-role ...	Created IAM role mock-role	Trusts EC2 service
2025-09-14 18:56:20	aws --endpoint-url=http://localhost:4566 iam create-user --user-name mock-user	Created IAM user mock-user	For privilege testing
2025-09-14 18:57:00	aws --endpoint-url=http://localhost:4566 ec2 create-volume --availability-zone us-east-1a --size 1	Created 1GB EC2 volume	For snapshot/exfil testing
2025-09-14 18:58:00	aws --endpoint-url=http://localhost:4566 logs create-log-group --log-group-name /mock/log/group	Created CloudWatch log group	Logging evasion test
2025-09-14 18:59:00	aws --endpoint-url=http://localhost:4566 lambda create-function --function-name mock-function ...	Created Lambda function	Persistence simulation
2025-09-14	aws --endpoint-	Created SNS	Used for lateral

Timestamp	*Command*	*Action Description*	*Notes*
19:00:00	url=http://localhost:4566 sns create-topic --name mock-topic	topic	movement
2025-09-14 19:01:00	aws --endpoint-url=http://localhost:4566 dynamodb create-table --table-name mock-table ...	Created DynamoDB table mock-table	Mock database
2025-09-15 00:25:32	aws --endpoint-url=http://localhost:4566 s3 ls	Enumerated available S3 buckets	Found mock-bucket, target-employee-data
2025-09-15 00:26:00	aws --endpoint-url=http://localhost:4566 iam list-users	Listed IAM users	Found mock-user

Figure 3.1 Shows pacu setup

4. Recommendations

- ***Implement Strong IAM Policies:*** least privilege is enforced to reduce the risk of privilege escalation and unauthorized resource access.
- ***Monitor CloudTrail and CloudWatch:*** Enable proper logging and monitoring to detect abnormal activities such as unauthorized API calls or exfiltration attempts.
- ***Restrict IAM Roles and Policies:*** Ensure only trusted entities can assume sensitive roles and avoid overly permissive policies.
- ***Regular Red Team Exercises:*** Continuously test defenses by simulating adversary behaviors in controlled environments.