Command to launch: `$ ./cloudgoat.py create legacy_snapshot`

Scenario Resources:

- 1 VPC with:
  - EC2 Snapshot x 1
  - Lambda Function x 1
  - IAM user x 1
  - S3 Bucket x 1

Scenario: Unprotected EC2 Instance Snapshot

Background:
A company relies on AWS for hosting its infrastructure, including an EC2 instance for their server. You will start as an IAM user with insufficient permissions but with the ability to invoke an AWS Lambda function. The Lambda function is misconfigured and allows the user to restore an EC2 instance from a snapshot that they should not have access to.

Scenario Start:

- Participants begin with an IAM user role (ls-user) that lacks permissions to modify or restore EC2 instance snapshots directly.

Scenario Flow:

1. **IAM User Reconnaissance:**

- Participants need to perform reconnaissance as the ls-user to identify potential vulnerabilities or misconfigurations.
- Looking through the AWS environment will reveal that there is a snapshot of an EC2 instance that the user doesn't have permissions to restore.

2. **AWS Lambda Invocation:**

- Identify and invoke a misconfigured AWS Lambda function that, due to excessive permissions, allows the user to restore an EC2 instance from the snapshot.
- The ls-user will need to update the Lambda function code with the necessary resources discovered during the reconnaissance phase.

3. **Restore Snapshot:**

- The ls-user will find that they have permissions to invoke the provided Lambda function.
- This Lambda function can be invoked by the ls-user, which will restore the EC2 instance from the snapshot.

4. Accessing the Restored EC2 Instance:

- After invoking the Lambda function, the ls-user can connect to the restored EC2 instance using SSH.
- Inside the EC2 instance, the user will find a flag image, indicating the successful completion of the scenario.

Exploitation Route