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

Scenario Resources:

- 1 VPC with:
  - EC2 Snapshot x 1
  - Lambda x 1 (Used to restore the snapshot)
  - IAM user x 1

**Scenario: Unprotected EC2 Instance Snapshots

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 that has the ability to invoke an AWS Lambda function. The lambda function is misconfigured, and allows the user to change the EC2 instance snapshot to public.

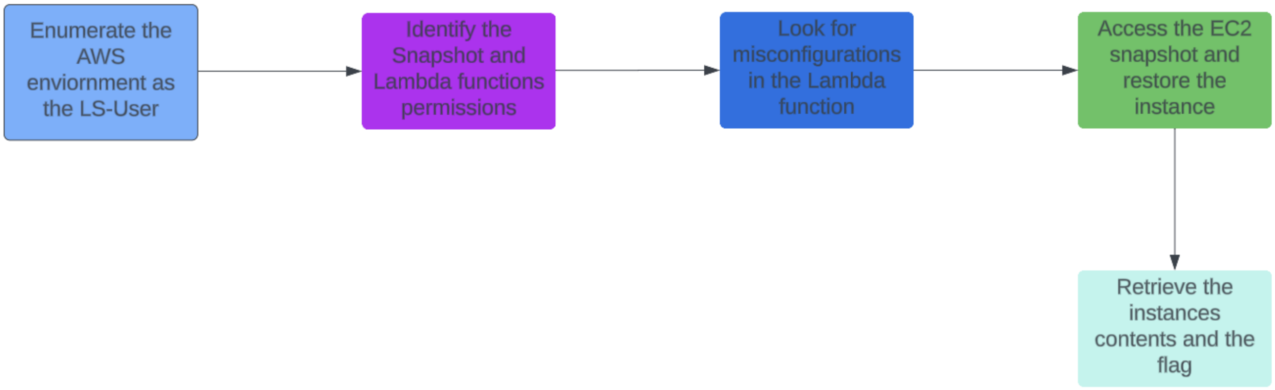
Scenario Start:

1. Participants begin with an IAM user role that lacks permissions to modify EC2 instance snapshots.

Scenario Flow:

1. **IAM User Reconnaissance:**
   - Participants need to perform reconnaissance as the IAM 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 changes the state of an EC2 instance snapshot to restore it.
3. **Snapshot to Public:**
   - The user will find that they do have permissions to use the lambda function that is provided to them
   - This lambda function can be invoked by the user, which will then restore the snapshot allowing the user to access its contents.

Exploitation Route


Enumerate the AWS enviornment as the LS-User
Identify the Snapshot and Lambda functions permissions
Look for misconfigurations in the Lambda function
Access the EC2 snapshot and restore the instance
Retrieve the instances contents and the flag