Configuration Manual

MSc Research Project
MSc Cybersecurity

Piyush Nikam
21202931

School of Computing
National College of Ireland

Supervisor: Michael Prior

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Piyush Sanjay Nikam
Student ID:	21202931
Programme:	MSc Cybersecurity
Year:	Sep 2022 -Sep 2023
Module:	MSc Research Project
Supervisor:	Michael Prior
Submission Due Date:	14th Aug 2023
Project Title:	Customized Penetration Testing Framework for Assessing the Security of Amazon Web Services (AWS) Services
Word Count:	1604
Page Count:	17

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Piyush Sanjay Nikam

Date: 14th Aug 2023

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project	□

is lost or mislaid. It is not sufficient to keep a copy on computer.	

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Piyush Nikam

21202931

1 Introduction

The configuration guide offers a step-by-step walkthrough of the methodologies and practices showcased in the master's thesis. It provides readers with insights on how to set up, tweak, and utilize the tools essential for the research study. To start with, our setup process involves the establishment of a Kali Linux Virtual Machine. This ensures we have a versatile and powerful platform to conduct our desired experiments for the AWSNeo Framework, our main developed solution. Additionally, the guide delves into comprehensive details about setting up AWS and constructing an AWS infrastructure using CloudGoat. This aids in effectively testing the AWSNeo Framework.

2 System Specifications

2.1 For a local setup, here's the hardware and OS you'll need:

- CPU: AMD Ryzen 7 4800H, running at 2900 Mhz with 8 cores and Radeon Graphics.
- Memory: A 16GB DDR4 RAM clocked at 3200MHz.
- Drive: An SSD with a capacity of 512GB.
- OS: 64-bit version of Windows 11, Kali 2022.3

2.2 Software Packages and Tools for the local run

- Terraform >= 0.14
- AWS CLI
- Python 3.11.4
- IAM-Flaws
- weirdAAL
- CloudGoat

2.3 AWS Requirement:

- AWS account with Administrator Access

3 Configuring VirtualBox and Kali Linux

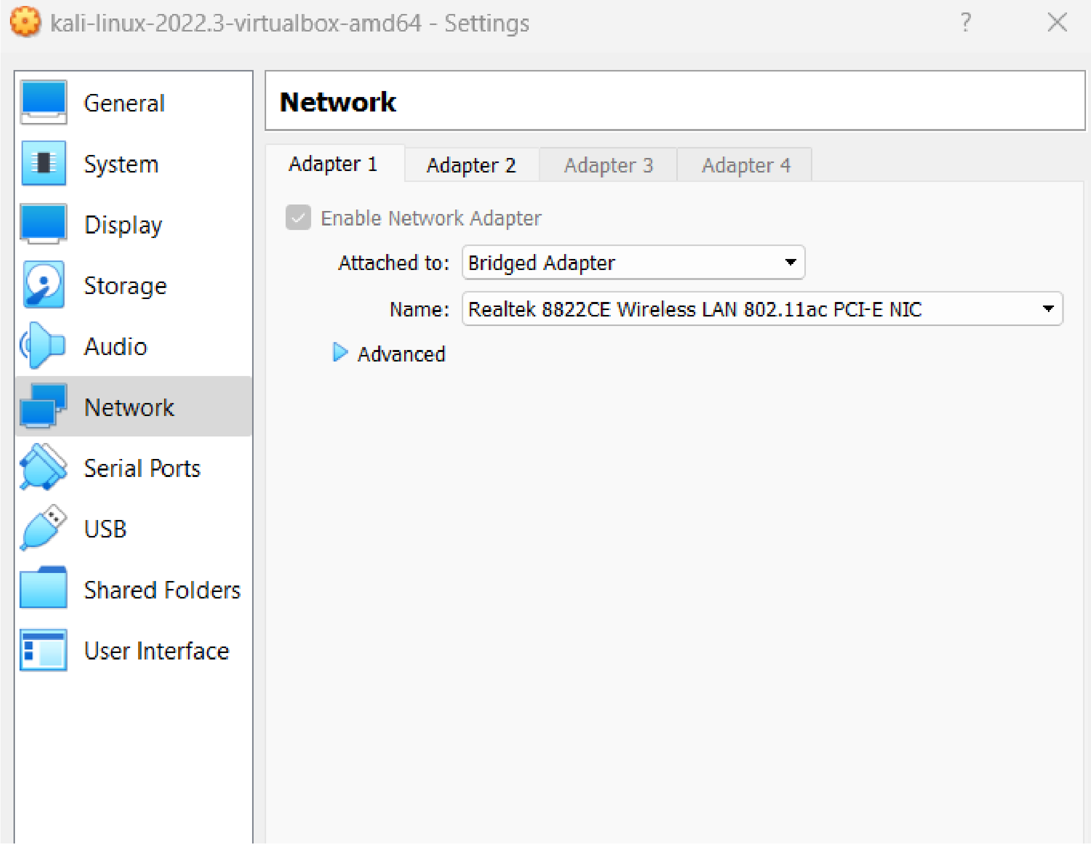
1. VirtualBox Installation: Follow the official VirtualBox documentation to install it (Krishnaraj, 2021).

2. Networking Method: For the project, use the 'Bridged Adapter' method in VirtualBox. This ensures the virtual machine operates as a unique entity on your network (Mucci, 2021).

3. Downloading Kali Linux Image: Obtain the appropriate Kali Linux virtual machine image from its official website.

4. Kali Configuration on VirtualBox(Rutger, 2021)**:**

- Launch VirtualBox and initiate a new virtual machine.
- Designate the Kali Linux image you downloaded as the installation source.
- Allocate an 80 GB virtual hard drive for Kali.
- Assign 5 CPU processors.
- Dedicate 8 GB of RAM.
- Ensure that the 'Bridged Adapter' option is selected in the network settings.



kali-linux-2022.3-virtualbox-amd64 - Settings

Network

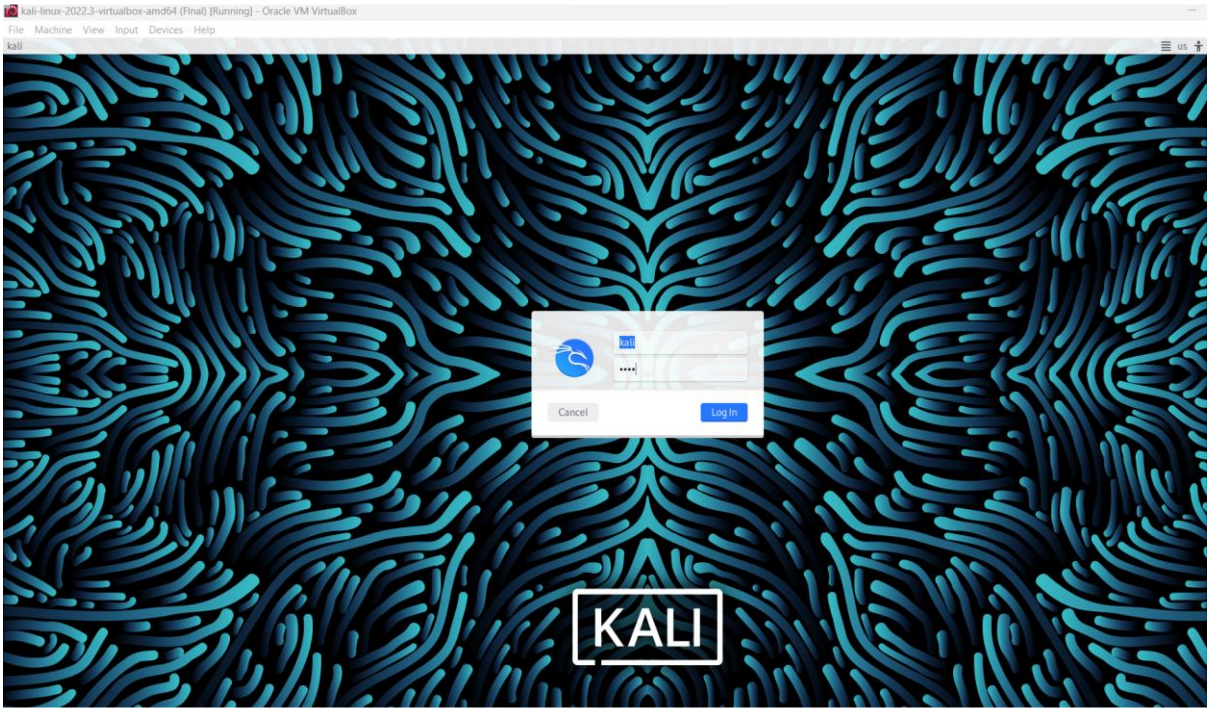
☑ Enable Network Adapter

Attached to: Bridged Adapter

Name: Realtek 8822CE Wireless LAN 802.11ac PCI-E NIC

5. Enabling Hardware Virtualization: Activate hardware virtualization in your system's BIOS/UEFI settings to optimize the performance of your Kali virtual machine within VirtualBox.

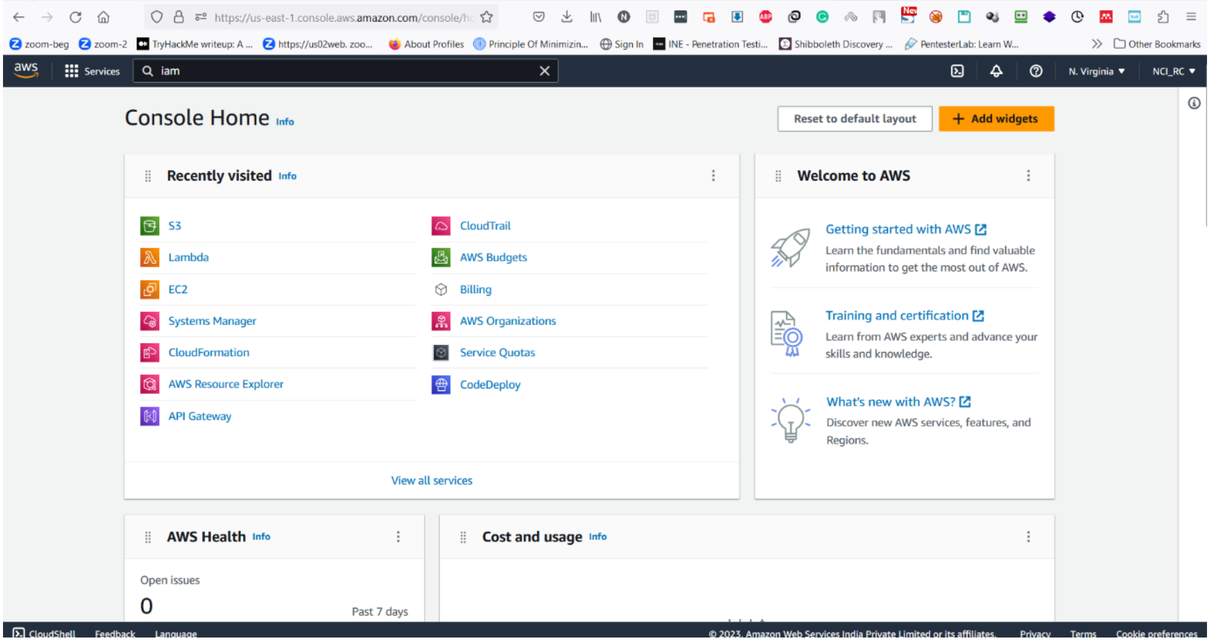
6. Booting Up Kali Linux: Start your Kali Linux virtual machine. When prompted, log in using the default Kali Linux credentials or those provided on the download page


kali-linux-2022.3-virtualbox-amd64 (Final) [Running] - Oracle VM VirtualBox

4 Configuration of AWS Account and Services and Tools

4.1 Setting Up Your AWS Account (kuppusamy, 2022)

1. AWS Account Creation: Refer to the AWS official documentation to set up an AWS account.


https://us-east-1.console.aws.amazon.com/console/h

Console Home

Recently visited: S3, Lambda, EC2, Systems Manager, CloudFormation, AWS Resource Explorer, API Gateway, CloudTrail, AWS Budgets, Billing, AWS Organizations, Service Quotas, CodeDeploy

Welcome to AWS

Getting started with AWS — Learn the fundamentals and find valuable information to get the most out of AWS.

Training and certification — Learn from AWS experts and advance your skills and knowledge.

What's new with AWS? — Discover new AWS services, features, and Regions.

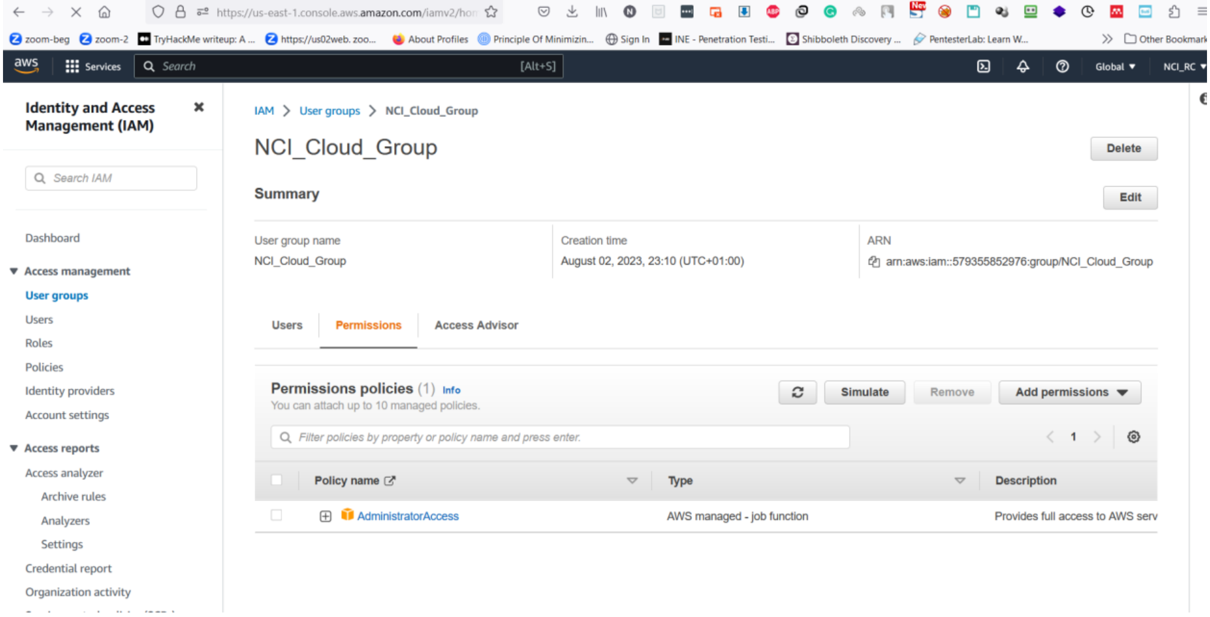
AWS Health — Open issues: 0, Past 7 days

Cost and usage

© 2023, Amazon Web Services India Private Limited or its affiliates.

2. Setting Up an Admin Group: Navigate to the IAM (Identity and Access Management) section and create a group for administrative access (Rosales, 2023).

3. Granting Permissions: In the permissions section, attach the predefined 'AdministratorAccess' policy. And



NCI_Cloud_Group

Summary

User group name: NCI_Cloud_Group

Creation time: August 02, 2023, 23:10 (UTC+01:00)

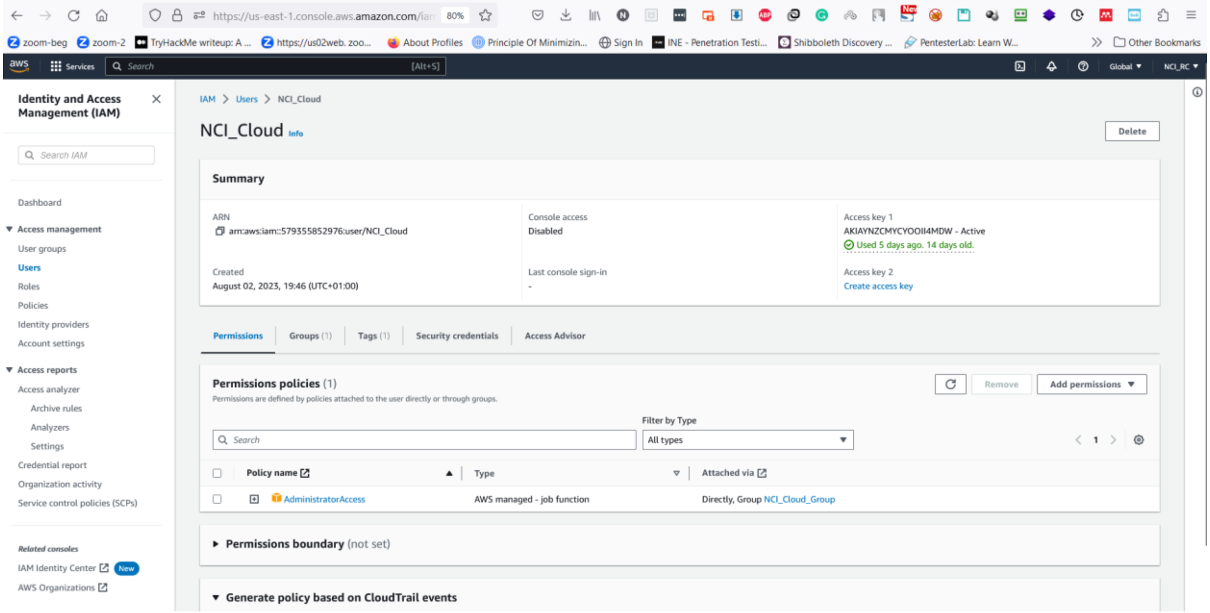
ARN: arn:aws:iam::579355852976:group/NCI_Cloud_Group

Permissions policies (1)

You can attach up to 10 managed policies.

Policy name	Type	Description
AdministratorAccess	AWS managed - job function	Provides full access to AWS serv

4. Creating an IAM User: Create a new IAM user with console access and access key and secret and add them to the 'Administrator' group.



NCI_Cloud

Summary

ARN: arn:aws:iam::579355852976:user/NCI_Cloud

Created: August 02, 2023, 19:46 (UTC+01:00)

Console access: Disabled

Last console sign-in: -

Access key 1: AKIAYNZCMYCYOOII4MDW - Active

Used 5 days ago. 14 days old.

Access key 2: Create access key

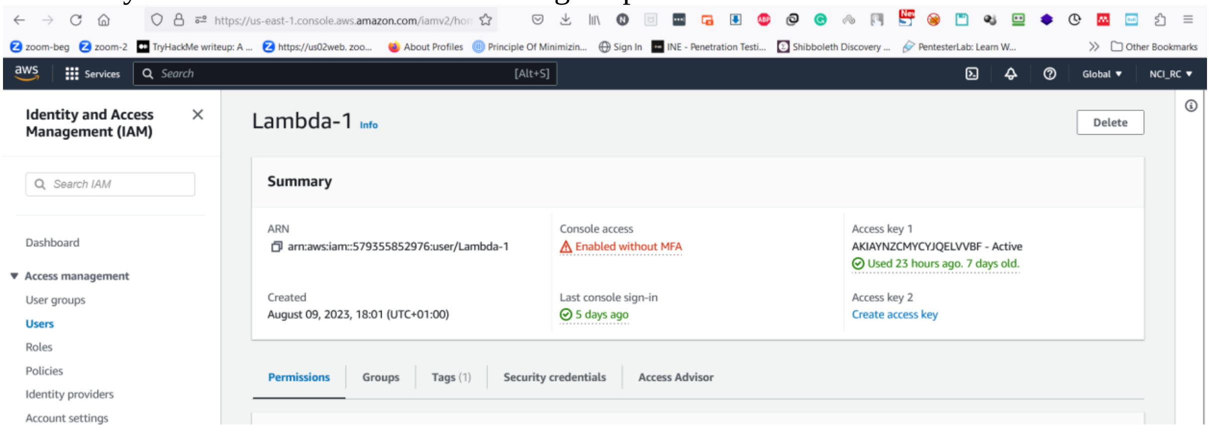
Permissions policies (1)

Permissions are defined by policies attached to the user directly or through groups.

Filter by Type: All types

Policy name	Type	Attached via
AdministratorAccess	AWS managed - job function	Directly, Group NCI_Cloud_Group

5. **Creating IAM User for Testing:** Now create another IAM User with console access and access key and secret for Framework testing Purpose



Identity and Access Management (IAM)

Lambda-1 Info

Summary

ARN
arn:aws:iam::579355852976:user/Lambda-1

Created
August 09, 2023, 18:01 (UTC+01:00)

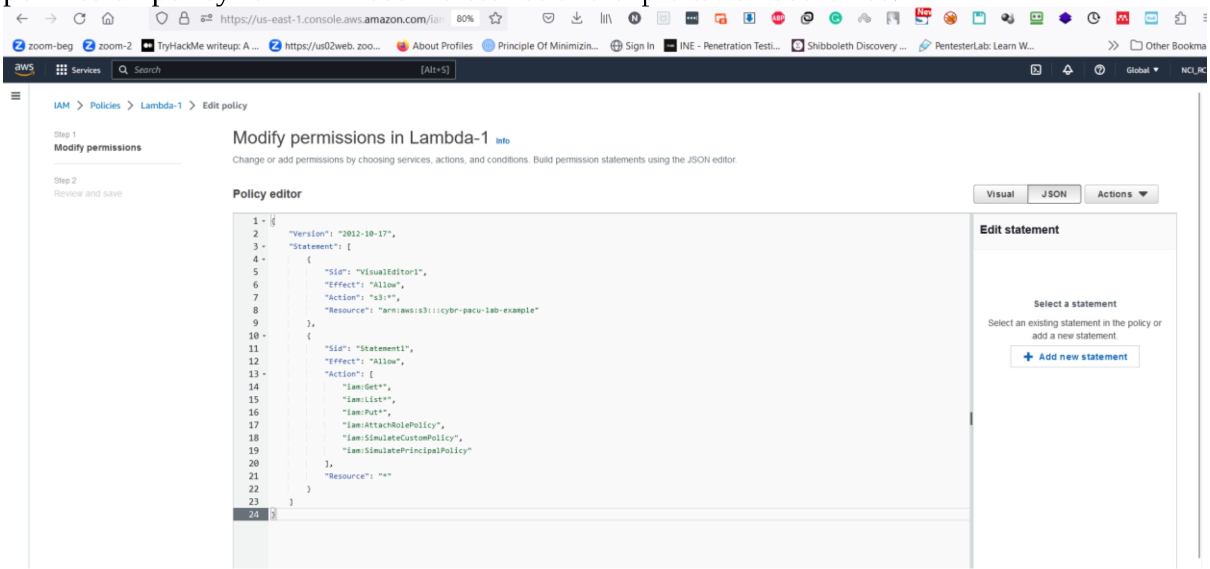
Console access
Enabled without MFA

Last console sign-in
5 days ago

Access key 1
AKIAYNZCMYCYJQELVVBF - Active
Used 23 hours ago. 7 days old.

Access key 2
Create access key

5. Custom IAM User Configuration: Create another IAM user and assign them a custom permission policy for IAM reconnaissance and exploitation scenarios.



IAM > Policies > Lambda-1 > Edit policy

Modify permissions in Lambda-1 Info

Change or add permissions by choosing services, actions, and conditions. Build permission statements using the JSON editor.

Policy editor

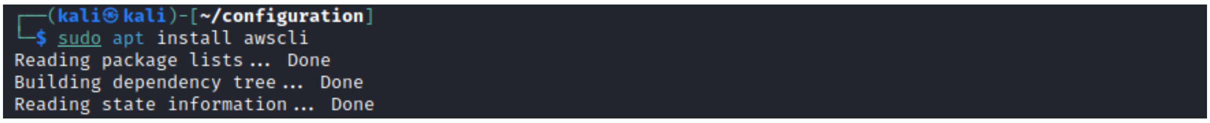
```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor1",
            "Effect": "Allow",
            "Action": "s3:*",
            "Resource": "arn:aws:s3:::cybr-pacu-lab-example"
        },
        {
            "Sid": "Statement1",
            "Effect": "Allow",
            "Action": [
                "iam:Get*",
                "iam:List*",
                "iam:Put*",
                "iam:AttachRolePolicy",
                "iam:SimulateCustomPolicy",
                "iam:SimulatePrincipalPolicy"
            ],
            "Resource": "*"
        }
    ]
}
```

6. **Configuring AWS CLI:** Configure AWS Administrator user into AWS CLI on Kali Linux using AWS configure command

4.2 Configuring tools

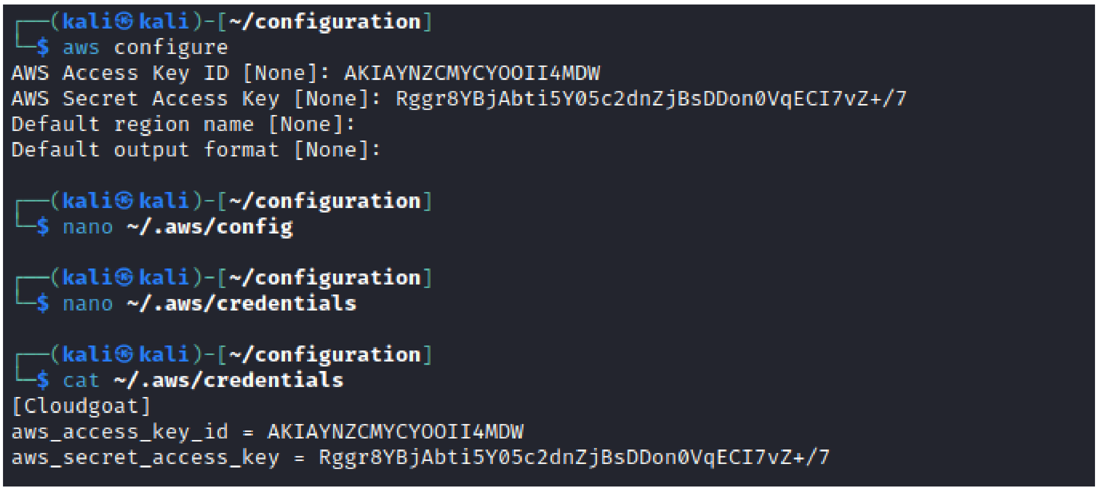
4.2.1 AWS CLI

1. Install awscli from Kali Linux CLI



```
┌──(kali㉿kali)-[~/configuration]
└─$ sudo apt install awscli
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
```

2. Configure AWS Administrator user into AWS CLI on Kali Linux using AWS configure command.



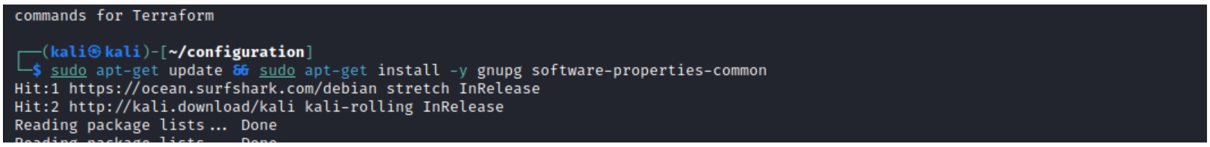
```
┌──(kali㉿kali)-[~/configuration]
└─$ aws configure
AWS Access Key ID [None]: AKIAYNZCMYCYOOII4MDW
AWS Secret Access Key [None]: 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
Default region name [None]:
Default output format [None]:

┌──(kali㉿kali)-[~/configuration]
└─$ nano ~/.aws/config

┌──(kali㉿kali)-[~/configuration]
└─$ nano ~/.aws/credentials

┌──(kali㉿kali)-[~/configuration]
└─$ cat ~/.aws/credentials
[Cloudgoat]
aws_access_key_id = AKIAYNZCMYCYOOII4MDW
aws_secret_access_key = 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
```

4.2.2 CloudGoat

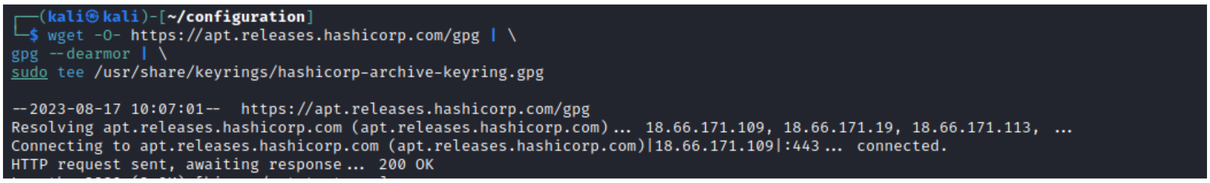
1. Cloning the Repository: First, head over to the Rhino Security Labs GitHub page and find the CloudGoat repository(Cloudgoat, 2023). Clone it onto your machine, so you've got your own local copy.

2. Navigate to the Directory: Once you've cloned it, navigate into the CloudGoat folder. This will be our workspace for the next steps.

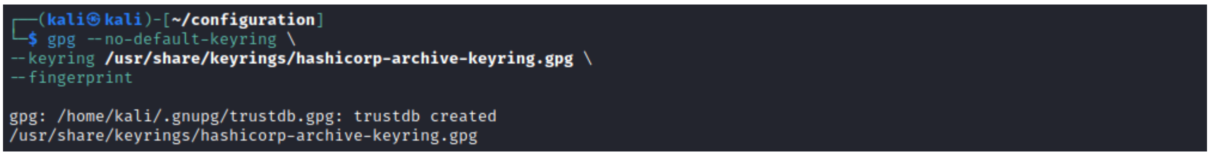
3. Installing Terraform: Before we proceed, ensure you've got Terraform (version 0.14 or above) installed. It's a key tool for CloudGoat.



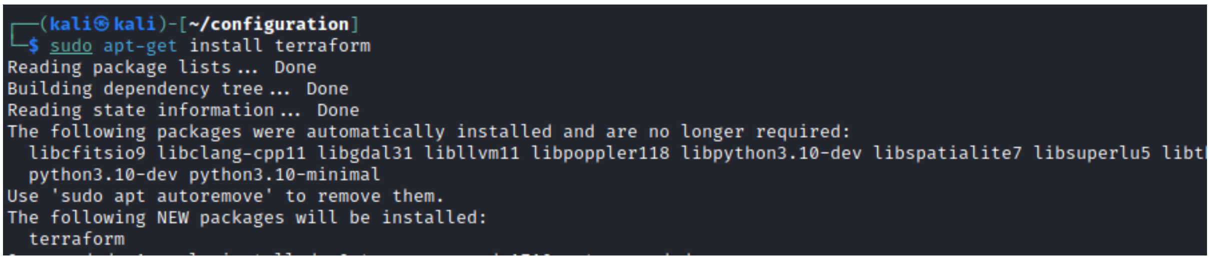
```
commands for Terraform

┌──(kali㉿kali)-[~/configuration]
└─$ sudo apt-get update && sudo apt-get install -y gnupg software-properties-common
Hit:1 https://ocean.surfshark.com/debian stretch InRelease
Hit:2 http://kali.download/kali kali-rolling InRelease
Reading package lists... Done
```



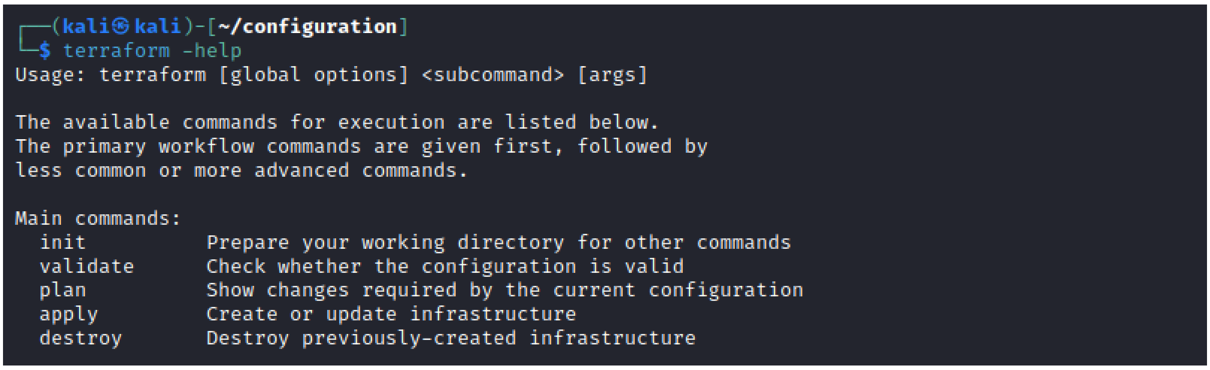
```
┌──(kali㉿kali)-[~/configuration]
└─$ wget -O- https://apt.releases.hashicorp.com/gpg | \
gpg --dearmor | \
sudo tee /usr/share/keyrings/hashicorp-archive-keyring.gpg

--2023-08-17 10:07:01--  https://apt.releases.hashicorp.com/gpg
Resolving apt.releases.hashicorp.com (apt.releases.hashicorp.com)... 18.66.171.109, 18.66.171.19, 18.66.171.113, ...
Connecting to apt.releases.hashicorp.com (apt.releases.hashicorp.com)|18.66.171.109|:443... connected.
HTTP request sent, awaiting response... 200 OK
```



```
┌──(kali㉿kali)-[~/configuration]
└─$ gpg --no-default-keyring \
--keyring /usr/share/keyrings/hashicorp-archive-keyring.gpg \
--fingerprint

gpg: /home/kali/.gnupg/trustdb.gpg: trustdb created
/usr/share/keyrings/hashicorp-archive-keyring.gpg
```



```
┌──(kali㉿kali)-[~/configuration]
└─$ sudo apt-get install terraform
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following packages were automatically installed and are no longer required:
  libcfitsio9 libclang-cpp11 libgdal31 libllvm11 libpoppler118 libpython3.10-dev libspatialite7 libsuperlu5 libt
  python3.10-dev python3.10-minimal
Use 'sudo apt autoremove' to remove them.
The following NEW packages will be installed:
  terraform
```



```
┌──(kali㉿kali)-[~/configuration]
└─$ terraform -help
Usage: terraform [global options] <subcommand> [args]

The available commands for execution are listed below.
The primary workflow commands are given first, followed by
less common or more advanced commands.

Main commands:
  init          Prepare your working directory for other commands
  validate      Check whether the configuration is valid
  plan          Show changes required by the current configuration
  apply         Create or update infrastructure
  destroy       Destroy previously-created infrastructure
```

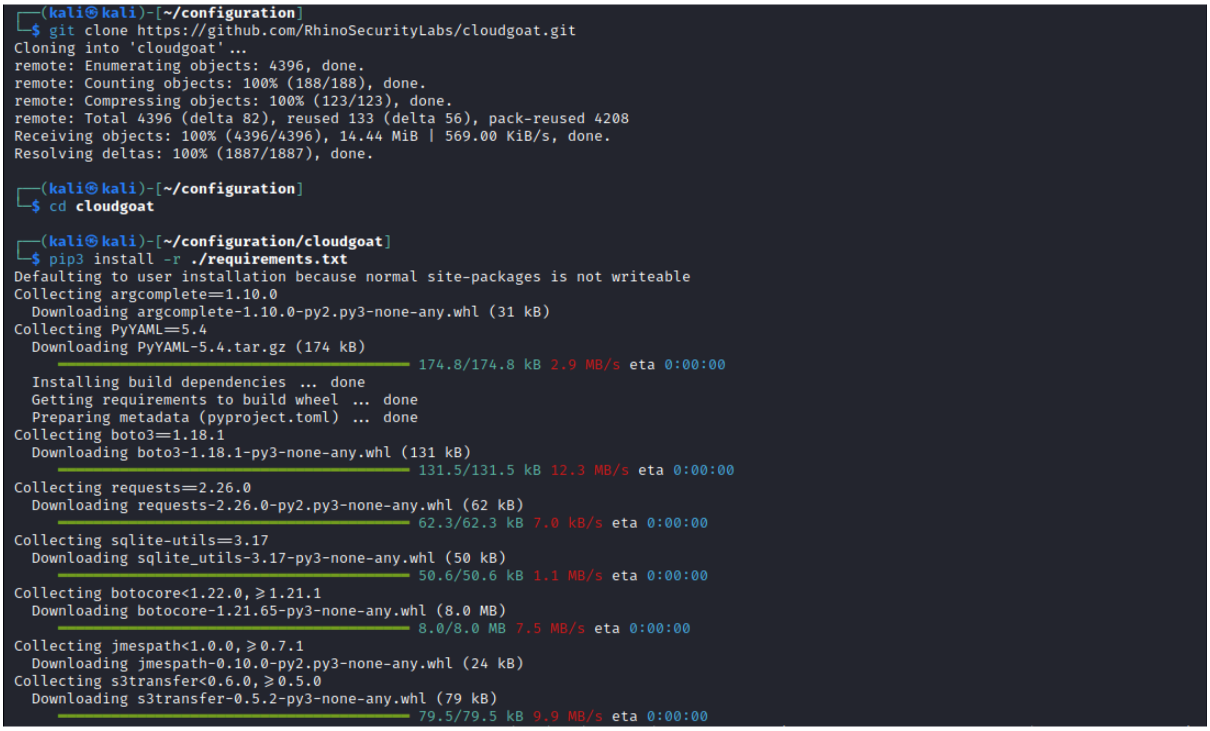
4. Installation of Dependencies: Inside the CloudGoat directory, you'll find a list of requirements to run CloudGoat smoothly. Install all of them.

5. Assign Execution Permissions: By default, CloudGoat not have the execution permissions. So, give it the green light to run on your system.

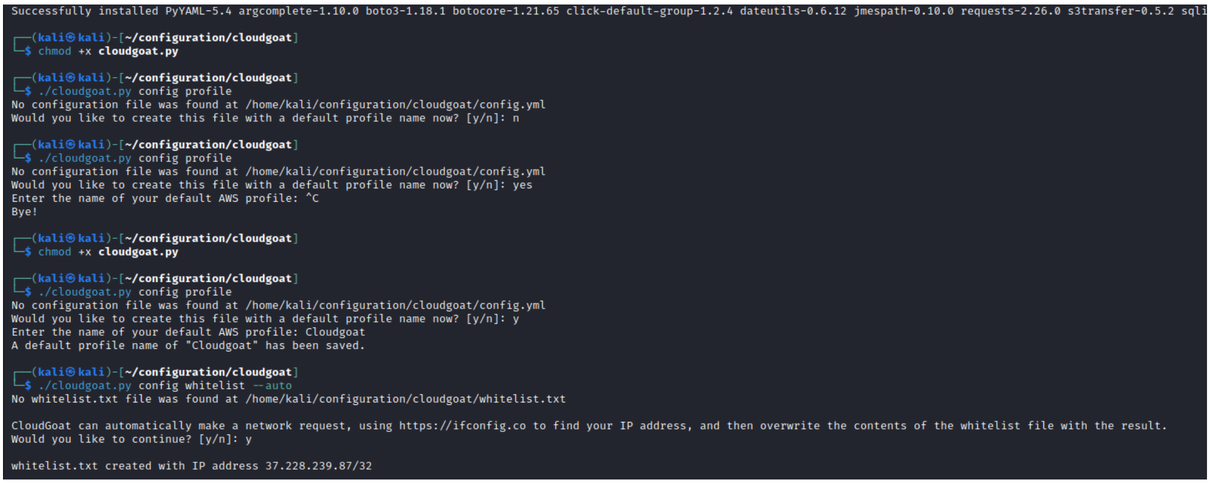
6. AWS Integration: Now, it's time to connect the dots. Integrate your AWS setup with CloudGoat with same profile name set in the 6th step while configuring AWS.

7. Whitelisting Your IP: To ensure smooth operation, add your IP to CloudGoat's whitelist.

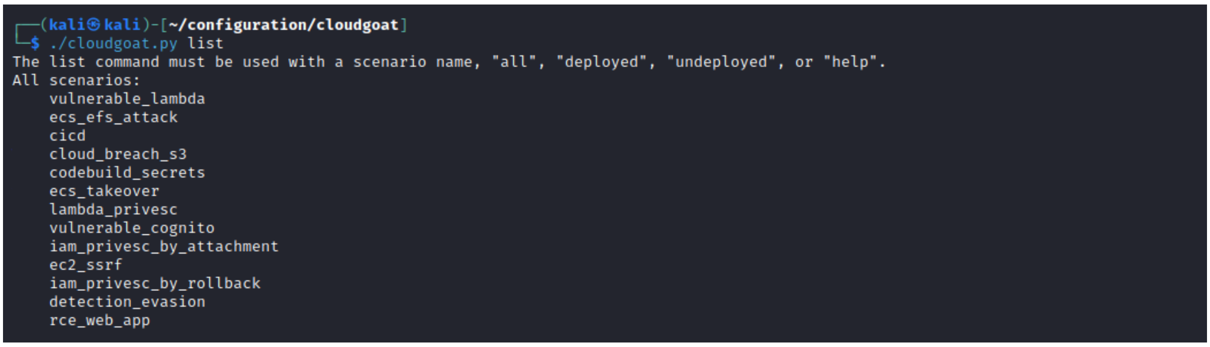
8. Test Execution: Take CloudGoat for a quick spin. Run it and ensure everything looks good.



```
┌──(kali㉿kali)-[~/configuration]
└─$ git clone https://github.com/RhinoSecurityLabs/cloudgoat.git
Cloning into 'cloudgoat'...
remote: Enumerating objects: 4396, done.
remote: Counting objects: 100% (188/188), done.
remote: Compressing objects: 100% (123/123), done.
remote: Total 4396 (delta 82), reused 133 (delta 56), pack-reused 4208
Receiving objects: 100% (4396/4396), 14.44 MiB | 569.00 KiB/s, done.
Resolving deltas: 100% (1887/1887), done.

┌──(kali㉿kali)-[~/configuration]
└─$ cd cloudgoat

┌──(kali㉿kali)-[~/configuration/cloudgoat]
└─$ pip3 install -r ./requirements.txt
Defaulting to user installation because normal site-packages is not writeable
Collecting argcomplete==1.10.0
  Downloading argcomplete-1.10.0-py2.py3-none-any.whl (31 kB)
Collecting PyYAML==5.4
  Downloading PyYAML-5.4.tar.gz (174 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 174.8/174.8 kB 2.9 MB/s eta 0:00:00
  Installing build dependencies ... done
  Getting requirements to build wheel ... done
  Preparing metadata (pyproject.toml) ... done
Collecting boto3==1.18.1
  Downloading boto3-1.18.1-py3-none-any.whl (131 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 131.5/131.5 kB 12.3 MB/s eta 0:00:00
Collecting requests==2.26.0
  Downloading requests-2.26.0-py2.py3-none-any.whl (62 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 62.3/62.3 kB 7.0 kB/s eta 0:00:00
Collecting sqlite-utils==3.17
  Downloading sqlite_utils-3.17-py3-none-any.whl (50 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 50.6/50.6 kB 1.1 MB/s eta 0:00:00
Collecting botocore<1.22.0,>=1.21.1
  Downloading botocore-1.21.65-py3-none-any.whl (8.0 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 8.0/8.0 MB 7.5 MB/s eta 0:00:00
Collecting jmespath<1.0.0,>=0.7.1
  Downloading jmespath-0.10.0-py2.py3-none-any.whl (24 kB)
Collecting s3transfer<0.6.0,>=0.5.0
  Downloading s3transfer-0.5.2-py3-none-any.whl (79 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 79.5/79.5 kB 9.9 MB/s eta 0:00:00
```



```
Successfully installed PyYAML-5.4 argcomplete-1.10.0 boto3-1.18.1 botocore-1.21.65 click-default-group-1.2.4 dateutils-0.6.12 jmespath-0.10.0 requests-2.26.0 s3transfer-0.5.2 sqli

┌──(kali㉿kali)-[~/configuration/cloudgoat]
└─$ chmod +x cloudgoat.py

┌──(kali㉿kali)-[~/configuration/cloudgoat]
└─$ ./cloudgoat.py config profile
No configuration file was found at /home/kali/configuration/cloudgoat/config.yml
Would you like to create this file with a default profile name now? [y/n]: n

┌──(kali㉿kali)-[~/configuration/cloudgoat]
└─$ ./cloudgoat.py config profile
No configuration file was found at /home/kali/configuration/cloudgoat/config.yml
Would you like to create this file with a default profile name now? [y/n]: yes
Enter the name of your default AWS profile: ^C
Bye!

┌──(kali㉿kali)-[~/configuration/cloudgoat]
└─$ chmod +x cloudgoat.py

┌──(kali㉿kali)-[~/configuration/cloudgoat]
└─$ ./cloudgoat.py config profile
No configuration file was found at /home/kali/configuration/cloudgoat/config.yml
Would you like to create this file with a default profile name now? [y/n]: y
Enter the name of your default AWS profile: Cloudgoat
A default profile name of "Cloudgoat" has been saved.

┌──(kali㉿kali)-[~/configuration/cloudgoat]
└─$ ./cloudgoat.py config whitelist --auto
No whitelist.txt file was found at /home/kali/configuration/cloudgoat/whitelist.txt

CloudGoat can automatically make a network request, using https://ifconfig.co to find your IP address, and then overwrite the contents of the whitelist file with the result.
Would you like to continue? [y/n]: y

whitelist.txt created with IP address 37.228.239.87/32
```



```
┌──(kali㉿kali)-[~/configuration/cloudgoat]
└─$ ./cloudgoat.py list
The list command must be used with a scenario name, "all", "deployed", "undeployed", or "help".
All scenarios:
    vulnerable_lambda
    ecs_efs_attack
    cicd
    cloud_breach_s3
    codebuild_secrets
    ecs_takeover
    lambda_privesc
    vulnerable_cognito
    iam_privesc_by_attachment
    ec2_ssrf
    iam_privesc_by_rollback
    detection_evasion
    rce_web_app
```

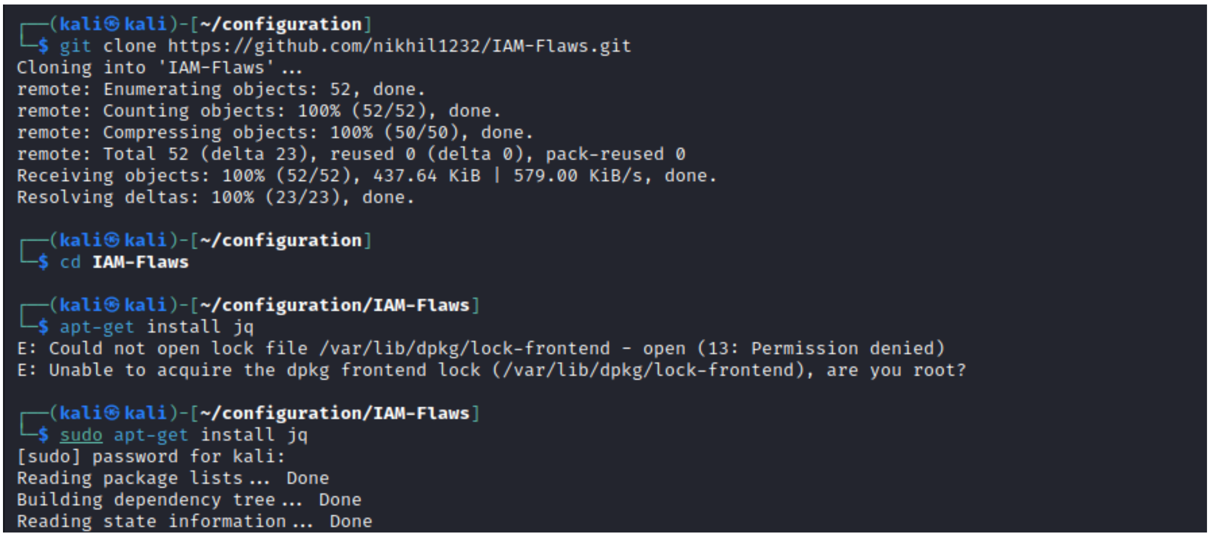
4.2.3 IAM-Flaws Installation & Configuration Guide(Nikhil, 2021)

1. Cloning the Repository: Begin by visiting the GitHub page where IAM-Flaws is hosted. Once there, clone the repository to your local system to get your very own copy.

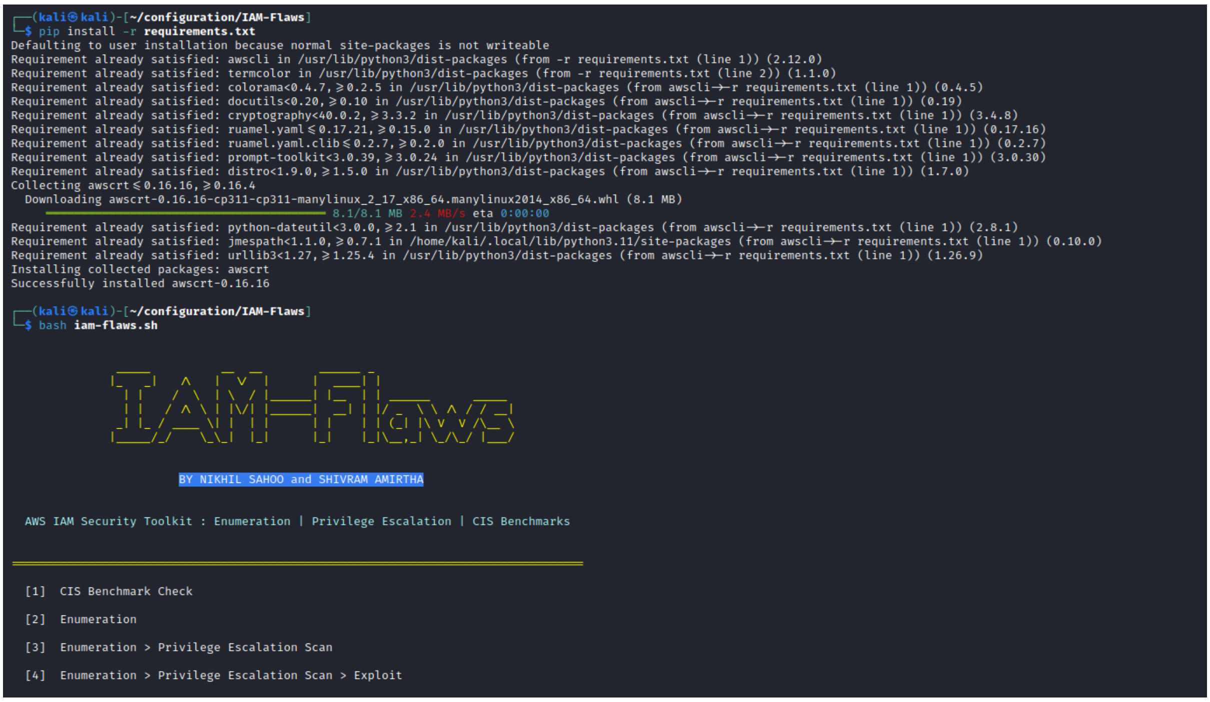
2. Navigate to the Directory: After successfully cloning, navigate into the IAM-Flaws folder.

3. Installing jq: Ensure jq is installed on your system. It's an essential tool for working with JSON data from the command line.

4. Installation of Dependencies: Within the IAM-Flaws directory, there's a list showcasing all the necessary components to run IAM-Flaws . Make sure to get each of them set up.



```
┌──(kali㉿kali)-[~/configuration]
└─$ git clone https://github.com/nikhil1232/IAM-Flaws.git
Cloning into 'IAM-Flaws' ...
remote: Enumerating objects: 52, done.
remote: Counting objects: 100% (52/52), done.
remote: Compressing objects: 100% (50/50), done.
remote: Total 52 (delta 23), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (52/52), 437.64 KiB | 579.00 KiB/s, done.
Resolving deltas: 100% (23/23), done.

┌──(kali㉿kali)-[~/configuration]
└─$ cd IAM-Flaws

┌──(kali㉿kali)-[~/configuration/IAM-Flaws]
└─$ apt-get install jq
E: Could not open lock file /var/lib/dpkg/lock-frontend - open (13: Permission denied)
E: Unable to acquire the dpkg frontend lock (/var/lib/dpkg/lock-frontend), are you root?

┌──(kali㉿kali)-[~/configuration/IAM-Flaws]
└─$ sudo apt-get install jq
[sudo] password for kali:
Reading package lists ... Done
Building dependency tree ... Done
Reading state information ... Done
```



```
┌──(kali㉿kali)-[~/configuration/IAM-Flaws]
└─$ pip install -r requirements.txt
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: awscli in /usr/lib/python3/dist-packages (from -r requirements.txt (line 1)) (2.12.0)
Requirement already satisfied: termcolor in /usr/lib/python3/dist-packages (from -r requirements.txt (line 2)) (1.1.0)
Requirement already satisfied: colorama<0.4.7,⩾0.2.5 in /usr/lib/python3/dist-packages (from awscli→-r requirements.txt (line 1)) (0.4.5)
Requirement already satisfied: docutils<0.20,⩾0.10 in /usr/lib/python3/dist-packages (from awscli→-r requirements.txt (line 1)) (0.19)
Requirement already satisfied: cryptography<40.0.2,⩾3.3.2 in /usr/lib/python3/dist-packages (from awscli→-r requirements.txt (line 1)) (3.4.8)
Requirement already satisfied: ruamel.yaml⩽0.17.21,⩾0.15.0 in /usr/lib/python3/dist-packages (from awscli→-r requirements.txt (line 1)) (0.17.16)
Requirement already satisfied: ruamel.yaml.clib⩽0.2.7,⩾0.2.0 in /usr/lib/python3/dist-packages (from awscli→-r requirements.txt (line 1)) (0.2.7)
Requirement already satisfied: prompt-toolkit<3.0.39,⩾3.0.24 in /usr/lib/python3/dist-packages (from awscli→-r requirements.txt (line 1)) (3.0.30)
Requirement already satisfied: distro<1.9.0,⩾1.5.0 in /usr/lib/python3/dist-packages (from awscli→-r requirements.txt (line 1)) (1.7.0)
Collecting awscrt⩽0.16.16,⩾0.16.4
  Downloading awscrt-0.16.16-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (8.1 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 8.1/8.1 MB 2.4 MB/s eta 0:00:00
Requirement already satisfied: python-dateutil<3.0.0,⩾2.1 in /usr/lib/python3/dist-packages (from awscli→-r requirements.txt (line 1)) (2.8.1)
Requirement already satisfied: jmespath<1.1.0,⩾0.7.1 in /home/kali/.local/lib/python3.11/site-packages (from awscli→-r requirements.txt (line 1)) (0.10.0)
Requirement already satisfied: urllib3<1.27,⩾1.25.4 in /usr/lib/python3/dist-packages (from awscli→-r requirements.txt (line 1)) (1.26.9)
Installing collected packages: awscrt
Successfully installed awscrt-0.16.16

┌──(kali㉿kali)-[~/configuration/IAM-Flaws]
└─$ bash iam-flaws.sh


               _____          __  __       _____ _
              |_   _|   /\   |  \/  |     |  ___| |
                | |    /  \  | \  / |_____| |_  | | __ ___      _____
                | |   / /\ \ | |\/| |_____|  _| | |/ _` \ \ /\ / / __|
               _| |_ / ____ \| |  | |     | |   | | (_| |\ V  V /\__ \
              |_____/_/    \_\_|  |_|     |_|   |_|\__,_| \_/\_/ |___/


                         BY NIKHIL SAHOO and SHIVRAM AMIRTHA


  AWS IAM Security Toolkit : Enumeration | Privilege Escalation | CIS Benchmarks


═══════════════════════════════════════════════════════════════════════════════


  [1]  CIS Benchmark Check

  [2]  Enumeration

  [3]  Enumeration > Privilege Escalation Scan

  [4]  Enumeration > Privilege Escalation Scan > Exploit
```

4.2.4 Pacu Installation & Configuration Guide(RhinoSecurityLabs, 2023.)

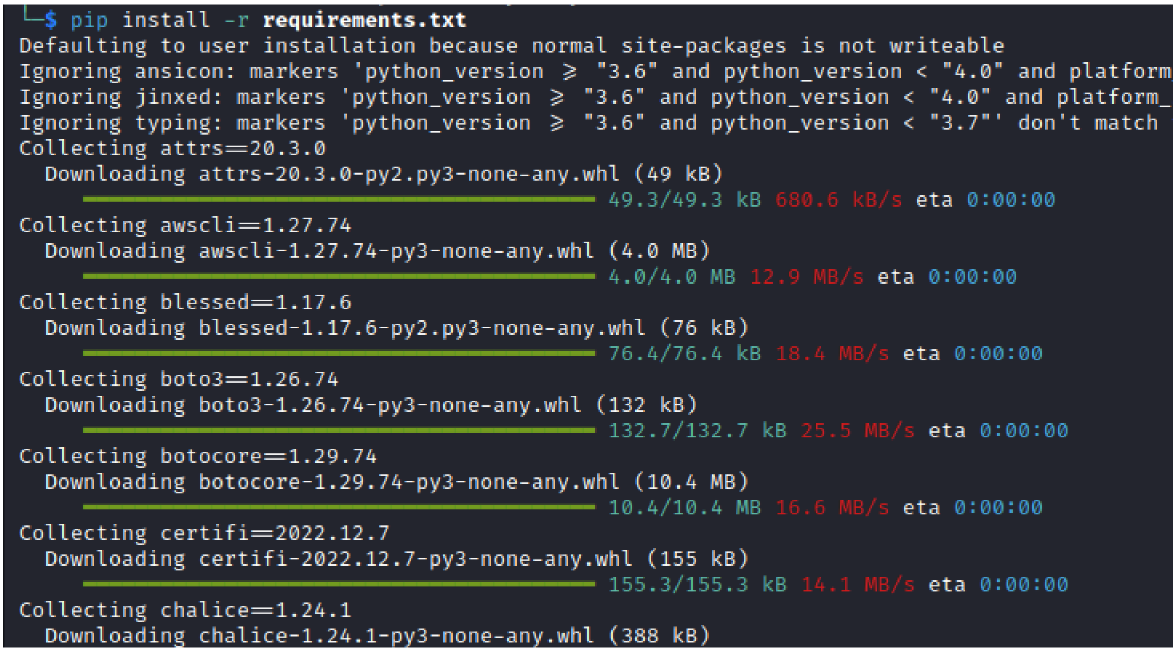
1. Navigating to the Code Artifact: Access the provided code artifact for Pacu Folder.

2. Navigate to the Directory Move to 'pacu' folder. Make this your current working directory.

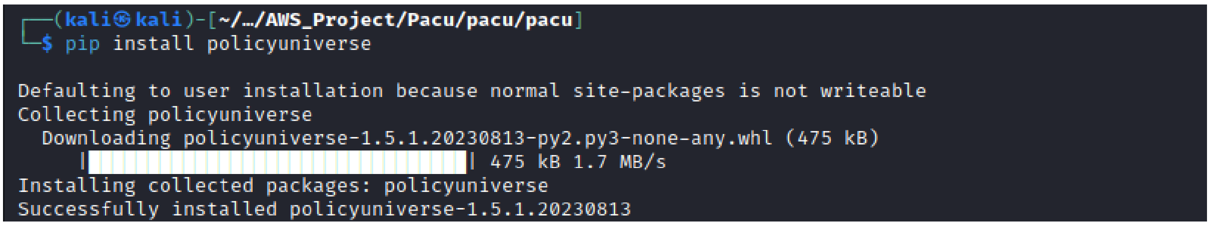
3. Installation of Requirements: Inside the CloudGoat directory, you'll find a list of requirements to run CloudGoat. Install all of them.

4. Resolving Dependencies: During your installation, should you encounter any issues or errors, particularly with policyuniverse, ensure it's installed or updated. Sometimes, specific dependencies can cause issues.

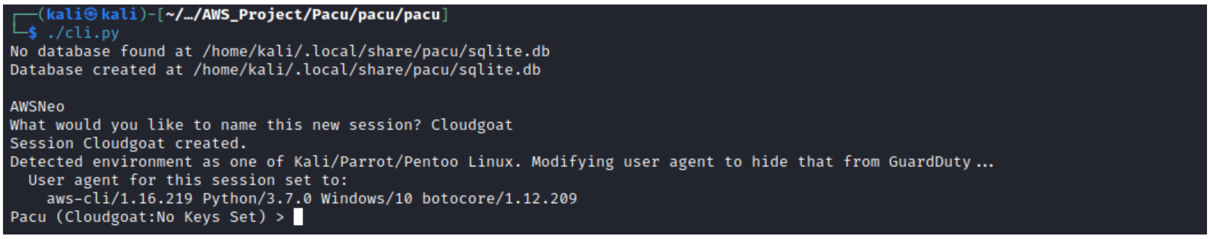
5. Initializing Pacu: With everything set up, it's time to bring Pacu to life. Launch it and when prompted, input the key for a user that doesn't possessAWS IAM administrator privileges.



```
└─$ pip install -r requirements.txt
Defaulting to user installation because normal site-packages is not writeable
Ignoring ansicon: markers 'python_version ≥ "3.6" and python_version < "4.0" and platform
Ignoring jinxed: markers 'python_version ≥ "3.6" and python_version < "4.0" and platform_
Ignoring typing: markers 'python_version ≥ "3.6" and python_version < "3.7"' don't match
Collecting attrs==20.3.0
  Downloading attrs-20.3.0-py2.py3-none-any.whl (49 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 49.3/49.3 kB 680.6 kB/s eta 0:00:00
Collecting awscli==1.27.74
  Downloading awscli-1.27.74-py3-none-any.whl (4.0 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 4.0/4.0 MB 12.9 MB/s eta 0:00:00
Collecting blessed==1.17.6
  Downloading blessed-1.17.6-py2.py3-none-any.whl (76 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 76.4/76.4 kB 18.4 MB/s eta 0:00:00
Collecting boto3==1.26.74
  Downloading boto3-1.26.74-py3-none-any.whl (132 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 132.7/132.7 kB 25.5 MB/s eta 0:00:00
Collecting botocore==1.29.74
  Downloading botocore-1.29.74-py3-none-any.whl (10.4 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 10.4/10.4 MB 16.6 MB/s eta 0:00:00
Collecting certifi==2022.12.7
  Downloading certifi-2022.12.7-py3-none-any.whl (155 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 155.3/155.3 kB 14.1 MB/s eta 0:00:00
Collecting chalice==1.24.1
  Downloading chalice-1.24.1-py3-none-any.whl (388 kB)
```



```
┌──(kali㉿kali)-[~/…/AWS_Project/Pacu/pacu/pacu]
└─$ pip install policyuniverse

Defaulting to user installation because normal site-packages is not writeable
Collecting policyuniverse
  Downloading policyuniverse-1.5.1.20230813-py2.py3-none-any.whl (475 kB)
     |████████████████████████████████| 475 kB 1.7 MB/s
Installing collected packages: policyuniverse
Successfully installed policyuniverse-1.5.1.20230813
```



```
┌──(kali㉿kali)-[~/…/AWS_Project/Pacu/pacu/pacu]
└─$ ./cli.py
No database found at /home/kali/.local/share/pacu/sqlite.db
Database created at /home/kali/.local/share/pacu/sqlite.db

AWSNeo
What would you like to name this new session? Cloudgoat
Session Cloudgoat created.
Detected environment as one of Kali/Parrot/Pentoo Linux. Modifying user agent to hide that from GuardDuty ...
  User agent for this session set to:
    aws-cli/1.16.219 Python/3.7.0 Windows/10 botocore/1.12.209
Pacu (Cloudgoat:No Keys Set) > █
```

4.2.5 AWSNeo Setup, Configuration & Execution Guide

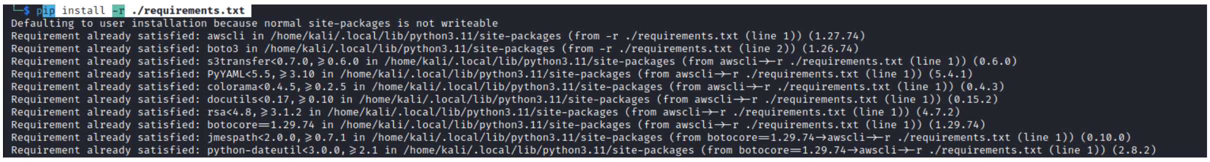
1. Navigate to the AWSNeo Directory: Inside the code artifact you've received, there's a folder specifically dedicated to AWSNeo. Make it current directory.

2. Installation of Requirements: Inside the AWSNeo directory, you'll find a list of requirements to run AWSNeo. Install all of them.

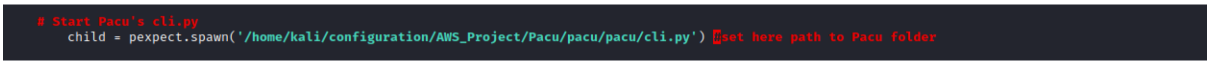
3. SET Path for Pacu: set path for pacu in lib/awsneo_shell.py it can be intialize

3. Launching AWSNeo: With the prerequisites in place, it's time to get AWSNeo running. Start the tool as directed, usually with a specific command or by executing the main script.

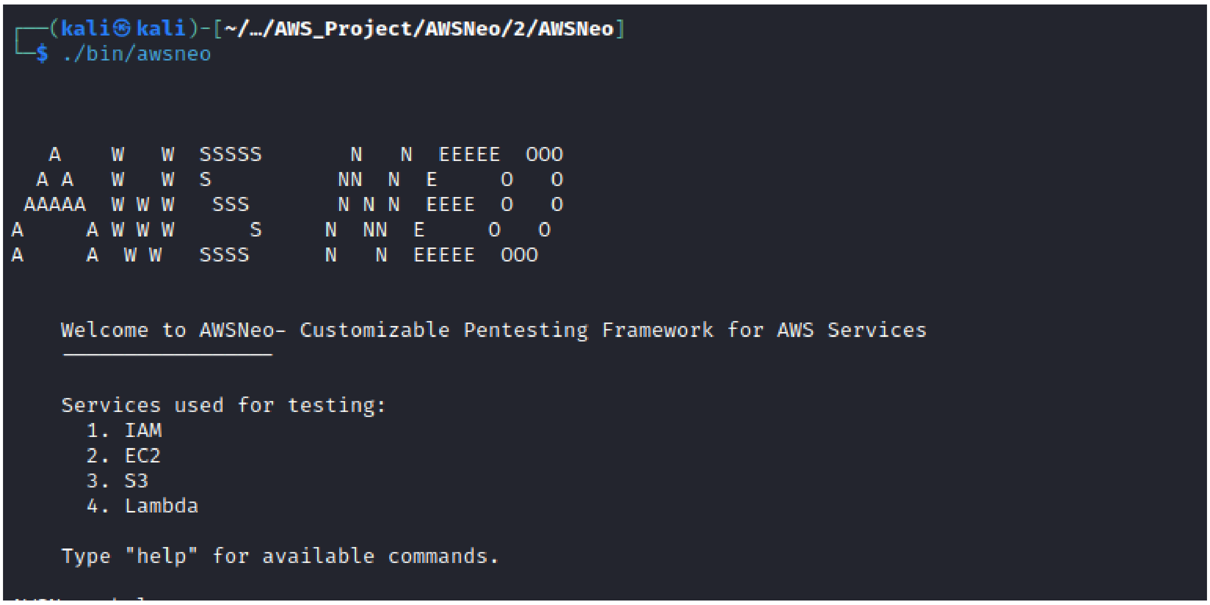
4. Initializing AWSNeo: To make sure AWSNeo is functioning as expected, test it out by executing `help` and `exit` command.



```
└─$ pip install -r ./requirements.txt
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: awscli in /home/kali/.local/lib/python3.11/site-packages (from -r ./requirements.txt (line 1)) (1.27.74)
Requirement already satisfied: boto3 in /home/kali/.local/lib/python3.11/site-packages (from -r ./requirements.txt (line 2)) (1.26.74)
Requirement already satisfied: s3transfer<0.7.0,⩾0.6.0 in /home/kali/.local/lib/python3.11/site-packages (from awscli→-r ./requirements.txt (line 1)) (0.6.0)
Requirement already satisfied: PyYAML<5.5,⩾3.10 in /home/kali/.local/lib/python3.11/site-packages (from awscli→-r ./requirements.txt (line 1)) (5.4.1)
Requirement already satisfied: colorama<0.4.5,⩾0.2.5 in /home/kali/.local/lib/python3.11/site-packages (from awscli→-r ./requirements.txt (line 1)) (0.4.3)
Requirement already satisfied: docutils<0.17,⩾0.10 in /home/kali/.local/lib/python3.11/site-packages (from awscli→-r ./requirements.txt (line 1)) (0.15.2)
Requirement already satisfied: rsa<4.8,⩾3.1.2 in /home/kali/.local/lib/python3.11/site-packages (from awscli→-r ./requirements.txt (line 1)) (4.7.2)
Requirement already satisfied: botocore==1.29.74 in /home/kali/.local/lib/python3.11/site-packages (from awscli→-r ./requirements.txt (line 1)) (1.29.74)
Requirement already satisfied: jmespath<2.0.0,⩾0.7.1 in /home/kali/.local/lib/python3.11/site-packages (from botocore==1.29.74→awscli→-r ./requirements.txt (line 1)) (0.10.0)
Requirement already satisfied: python-dateutil<3.0.0,⩾2.1 in /home/kali/.local/lib/python3.11/site-packages (from botocore==1.29.74→awscli→-r ./requirements.txt (line 1)) (2.8.2)
```



```
    # Start Pacu's cli.py
        child = pexpect.spawn('/home/kali/configuration/AWS_Project/Pacu/pacu/pacu/cli.py') #set here path to Pacu folder
```

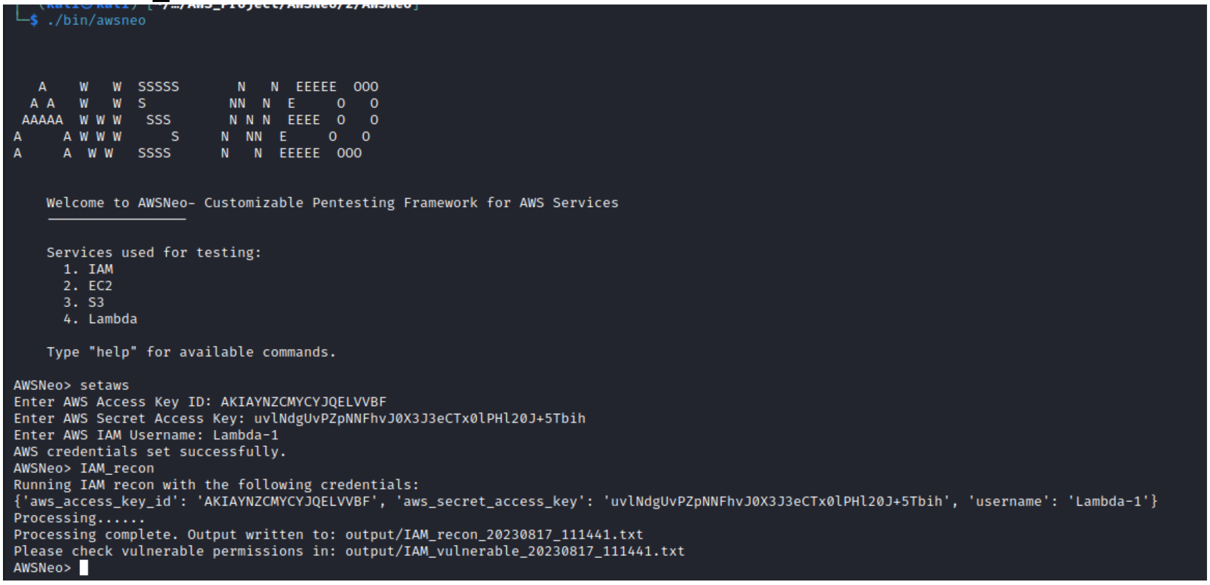


```
┌──(kali㉿kali)-[~/…/AWS_Project/AWSNeo/2/AWSNeo]
└─$ ./bin/awsneo



   A    W   W  SSSSS       N   N  EEEEE  OOO
  A A   W   W  S          NN   N  E     O   O
 AAAAA  W W W   SSS       N N N  EEEE   O   O
A     A W W W      S     N  NN  E     O   O
A     A  W W   SSSS      N   N  EEEEE  OOO


    Welcome to AWSNeo- Customizable Pentesting Framework for AWS Services
    ________________

    Services used for testing:
      1. IAM
      2. EC2
      3. S3
      4. Lambda

    Type "help" for available commands.
```

5 Scenarios

5.1 Scenario 1: IAM _Recon

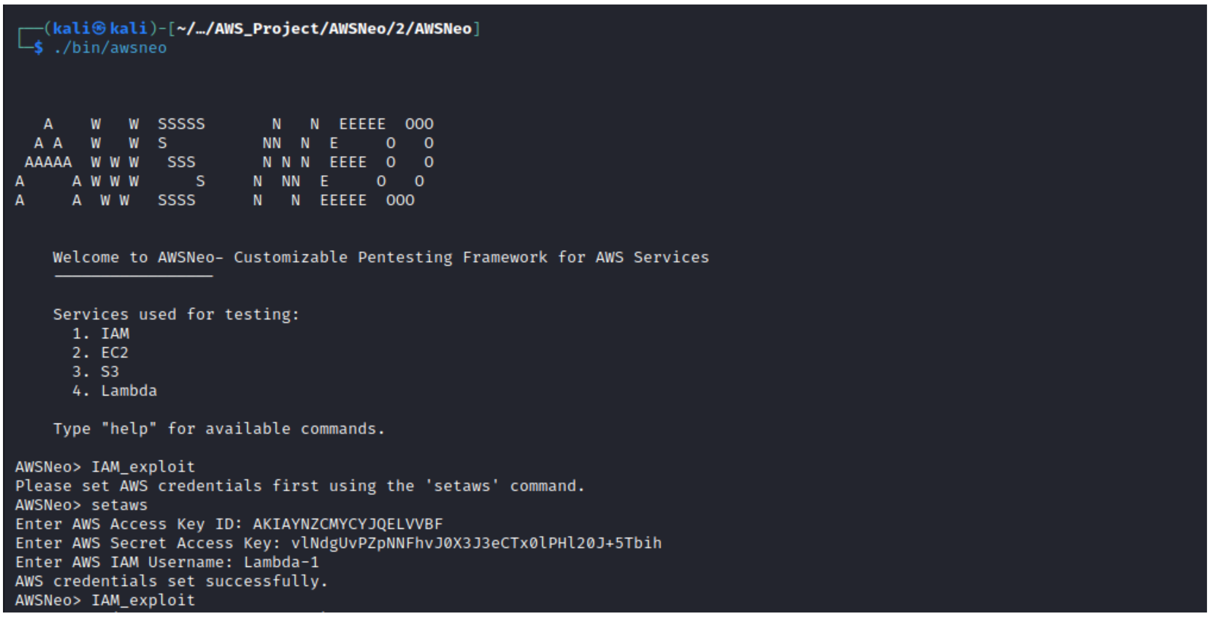
- Start the AWSNeo
- Run ` setaws`
- Configure AWS key
- Run ` IAM_recon` command



```
┌──(kali㉿kali)-[~/…/AWS_Project/AWSNeo/2/AWSNeo]
└─$ ./bin/awsneo



   A    W   W  SSSSS       N   N  EEEEE  OOO
  A A   W   W  S          NN  N  E     O   O
 AAAAA  W W W   SSS       N N N  EEEE  O   O
A     A W W W      S     N  NN  E     O   O
A     A  W W   SSSS      N   N  EEEEE  OOO


    Welcome to AWSNeo- Customizable Pentesting Framework for AWS Services
    ─────────────────

    Services used for testing:
      1. IAM
      2. EC2
      3. S3
      4. Lambda

    Type "help" for available commands.

AWSNeo> setaws
Enter AWS Access Key ID: AKIAYNZCMYCYJQELVVBF
Enter AWS Secret Access Key: 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
Enter AWS IAM Username: Lambda-1
AWS credentials set successfully.
AWSNeo> IAM_recon
Running IAM recon with the following credentials:
{'aws_access_key_id': 'AKIAYNZCMYCYJQELVVBF', 'aws_secret_access_key': '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', 'username': 'Lambda-1'}
Processing......
Processing complete. Output written to: output/IAM_recon_20230817_111441.txt
Please check vulnerable permissions in: output/IAM_vulnerable_20230817_111441.txt
AWSNeo> █
```

5.2 Scenario 2: IAM _Exploit

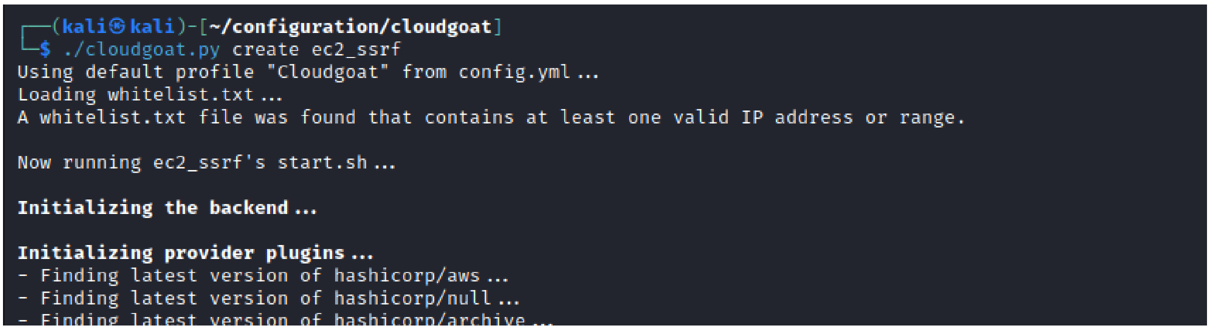
- Start the AWSNeo
- Run ` setaws`
- Configure AWS key
- Run ` IAM_exploit` command



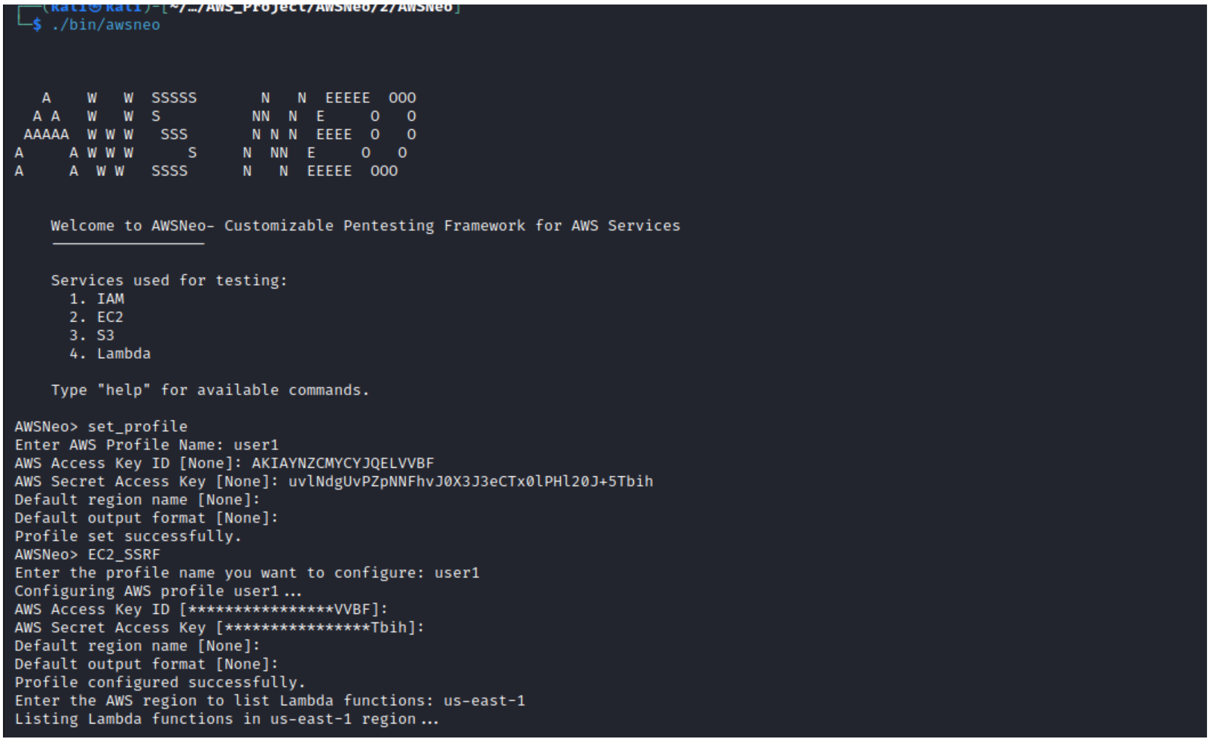
```
┌──(kali㉿kali)-[~/…/AWS_Project/AWSNeo/2/AWSNeo]
└─$ ./bin/awsneo



   A    W   W  SSSSS       N   N  EEEEE  OOO
  A A   W   W  S          NN  N  E     O   O
 AAAAA  W W W   SSS       N N N  EEEE  O   O
A     A W W W      S     N  NN  E     O   O
A     A  W W   SSSS      N   N  EEEEE  OOO


    Welcome to AWSNeo- Customizable Pentesting Framework for AWS Services
    ─────────────────

    Services used for testing:
      1. IAM
      2. EC2
      3. S3
      4. Lambda

    Type "help" for available commands.

AWSNeo> IAM_exploit
Please set AWS credentials first using the 'setaws' command.
AWSNeo> setaws
Enter AWS Access Key ID: AKIAYNZCMYCYJQELVVBF
Enter AWS Secret Access Key: vlNdgUvPZpNNFhvJ0X3J3eCTx0lPHl20J+5Tbih
Enter AWS IAM Username: Lambda-1
AWS credentials set successfully.
AWSNeo> IAM_exploit
```

5.3 Scenario 3: EC2_SSRF

- Create AWS Infrastructure using the CloudGoat.
- Start the AWSNeo
- Run ` set_profile`
- Configure AWS key
- Run ` EC2_SSRF` command



```
┌──(kali㉿kali)-[~/configuration/cloudgoat]
└─$ ./cloudgoat.py create ec2_ssrf
Using default profile "Cloudgoat" from config.yml ...
Loading whitelist.txt ...
A whitelist.txt file was found that contains at least one valid IP address or range.

Now running ec2_ssrf's start.sh ...

Initializing the backend ...

Initializing provider plugins ...
- Finding latest version of hashicorp/aws ...
- Finding latest version of hashicorp/null ...
- Finding latest version of hashicorp/archive ...
```



```
┌──(kali㉿kali)-[~/…/AWS_Project/AWSNeo/2/AWSNeo]
└─$ ./bin/awsneo



   A    W   W  SSSSS       N   N  EEEEE  OOO
  A A   W   W  S          NN  N  E     O   O
 AAAAA  W W W   SSS       N N N  EEEE  O   O
A     A W W W      S     N  NN  E     O   O
A     A  W W   SSSS      N   N  EEEEE  OOO


    Welcome to AWSNeo- Customizable Pentesting Framework for AWS Services
    ────────────────


    Services used for testing:
      1. IAM
      2. EC2
      3. S3
      4. Lambda

    Type "help" for available commands.

AWSNeo> set_profile
Enter AWS Profile Name: user1
AWS Access Key ID [None]: AKIAYNZCMYCYJQELVVBF
AWS Secret Access Key [None]: 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
Default region name [None]:
Default output format [None]:
Profile set successfully.
AWSNeo> EC2_SSRF
Enter the profile name you want to configure: user1
Configuring AWS profile user1 ...
AWS Access Key ID [****************VVBF]:
AWS Secret Access Key [****************Tbih]:
Default region name [None]:
Default output format [None]:
Profile configured successfully.
Enter the AWS region to list Lambda functions: us-east-1
Listing Lambda functions in us-east-1 region ...
```

References

Cloudgoat. (2023) *RhinoSecurityLabs/Cloudgoat: CloudGoat Is Rhino Security Labs' 'Vulnerable by Design' AWS Deployment Tool*. Available at: https://github.com/RhinoSecurityLabs/cloudgoat (Accessed: 5 August 2023).

Krishnaraj, N. (2021) *Part 1: How to Set Up a Virtual Machine on Windows Using VirtualBox | by Nithil Krishnaraj | TechTalkers | Medium*. *Medium*. Available at: https://medium.com/techtalkers/part-1-how-to-set-up-a-virtual-machine-on-windows-using-virtualbox-d169088e6baa (Accessed: 5 August 2023).

kuppusamy, S. (2022) *How to Create an AWS Account?. The Creation Process of an AWS Account. | by Selvaraj Kuppusamy | Medium*. *Medium*. Available at: https://medium.com/@selvarajk/how-to-create-an-aws-account-3a6682818355 (Accessed: 5 August 2023).

Mucci, T. (2021) *VirtualBox → How to Properly Set up Your VM's Network for Various Network Situations | by Tony Mucci | Code Kings | Medium*. *Medium*. Available at: https://medium.com/code-kings/virtualbox-how-to-properly-set-up-your-vms-network-for-various-network-situations-f589ee3a8f8 (Accessed: 6 August 2023).

Nikhil. (2021) *Nikhil1232/IAM-Flaws: AWS IAM Security Toolkit: CIS Benchmarks | Enumeration | Privilege Escalation*. Available at: https://github.com/nikhil1232/IAM-Flaws (Accessed: 6 August 2023).

RhinoSecurityLabs. *Pacu: The AWS Exploitation Framework, Designed for Testing the Security of Amazon Web Services Environments.* Available at: https://github.com/RhinoSecurityLabs/pacu (Accessed: 6 August 2023).
Rosales, I. (2023) *AWS IAM Tutorial. Identity and Access Management . Medium*. Available at: https://blog.devgenius.io/aws-iam-tutorial-a030d3792412 (Accessed: 6 August 2023).

Rutger. (2021) *How to Install Kali Linux Using the VirtualBox Image Provided by Kali | by Rutger | Medium*. *Medium*. Available at: https://rutger-t.medium.com/how-to-install-kali-linux-using-the-virtualbox-image-provided-by-kali-746d82df9bdd (Accessed: 6 August 2023).