Module 19: Cloud Computing

Scenario

Cloud computing is an emerging technology that delivers computing services such as online business applications, online data storage, and webmail over the Internet. Cloud implementation enables a distributed workforce, reduces organization expenses, provides data security, etc. As enterprises are increasingly adopting cloud services, cloud systems have emerged as targets for attackers to gain unauthorized access to the valuable data stored in them. Therefore, it is essential to regularly perform pen testing on cloud systems to monitor their security posture.

Security administrators claim that cloud systems are more vulnerable to DoS assaults, because they involves numerous individuals or clients, making DoS assaults potentially very harmful. Because of the high workload on a flooded service, these systems attempt to provide additional computational power (more virtual machines, more service instances) to cope with the workload, and they will eventually fail.

Although cloud systems try to thwart attackers by providing additional computational power, they inadvertently aid attackers by allowing the most significant possible damage to the availability of a service—a process that starts from a single flooding-attack entry point. Thus, attackers need not flood all servers that provide a particular service but merely flood a single, cloud-based address to a service that is unavailable. Thus, adequate security is vital in this context, because cloud-computing services are based on sharing.

As an ethical hacker and penetration tester, you must have sound knowledge of hacking cloud platforms using various tools and techniques. The labs in this module will provide you with real-time experience in exploiting the underlying vulnerabilities in a target cloud platform using various hacking methods and tools. However, hacking the cloud platform may be illegal depending on the organization's policies and any laws that are in effect. As an ethical or pen tester, you should always acquire proper authorization before performing system hacking.

Objective

The objective of the lab is to perform cloud platform hacking and other tasks that include, but are not limited to:

- Performing S3 bucket enumeration
- Exploiting misconfigured S3 buckets
- Escalating privileges of a target IAM user account by exploiting misconfigurations in a user policy

Overview of Cloud Computing

Cloud computing refers to on-demand delivery of IT capabilities, in which IT infrastructure and applications are provided to subscribers as metered services over a network. Cloud services are

classified into three categories, namely infrastructure-as-a-service (IaaS), platform-as-a-service (PaaS), and software-as-a-service (SaaS), which offer different techniques for developing cloud.

Lab Tasks

Ethical hackers or pen testers use numerous tools and techniques to hack the target cloud platform. Recommended labs that will assist you in learning various cloud platform hacking techniques include:

1. Perform Reconnaissance on Azure
    - Azure reconnaissance with AADInternals
2. Exploit S3 buckets
    - Exploit open S3 buckets using AWS CLI
3. Perform privilege escalation to gain higher privileges
    - Escalate IAM user privileges by exploiting misconfigured user policy
4. Perform vulnerability assessment on Docker images
    - Vulnerability assessment on Docker images using Trivy

Lab 1: Perform Reconnaissance on Azure

Lab Scenario

As an ethical hacker, you need to know how to utilize PowerShell command-based scripting tools for conducting reconnaissance and gathering information. This information can then be used to assess the security posture of other systems within the network.

Lab Objectives

- Azure Reconnaissance with AADInternals

Overview of Reconnaissance Tools

Reconnaissance tools serve as indispensable assets for attackers in cloud hacking, providing them with the essential information and insights needed to orchestrate successful attacks against cloud environments.

Task 1: Azure Reconnaissance with AADInternals

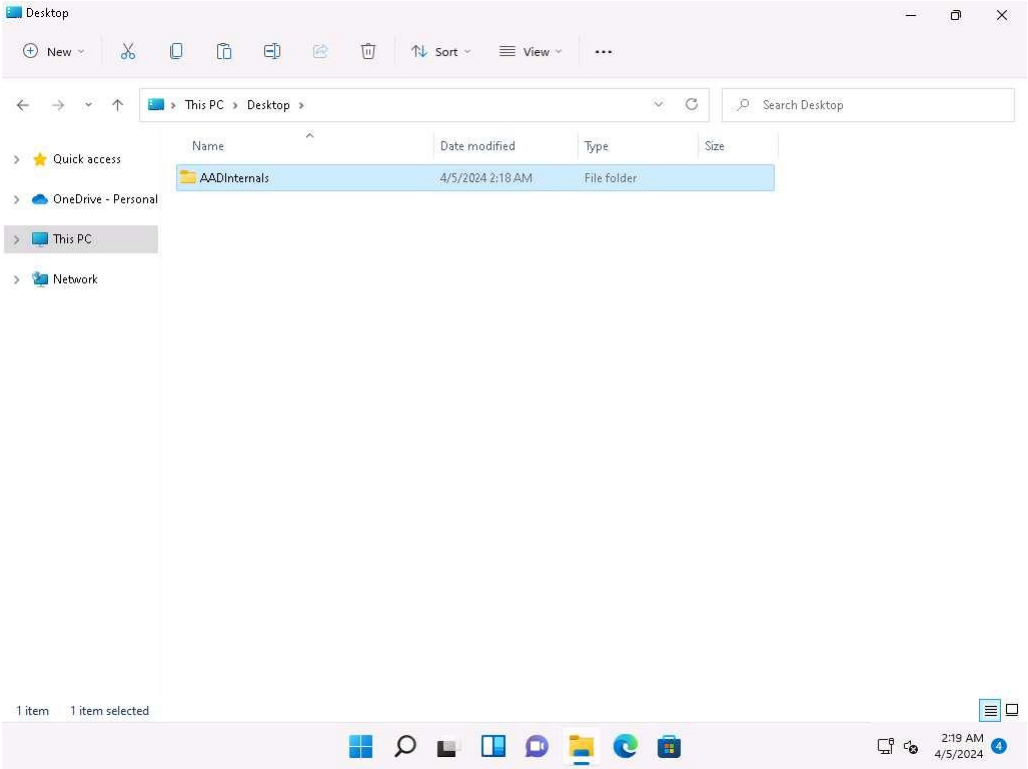
AADInternals is primarily focused on auditing and attacking Azure Active Directory (AAD) environments, it can still be utilized as part of a broader cloud reconnaissance effort. This tool

has several features such as user enumeration, credential extraction, token extraction and manipulation, privilege escalation, etc.

In this lab we will perform Azure Active Directory reconnaissance as an outsider.

1. Click Windows 11 to switch to the **Windows 11** machine.
   Click Ctrl+Alt+Delete to activate the machine and login with **Admin/Pa$$w0rd**.

2. Navigate to **E:\CEH-Tools\CEHv13 Module 19 Cloud Computing\GitHub Tools** and copy **AADInternals** folder and paste it on **Desktop**.



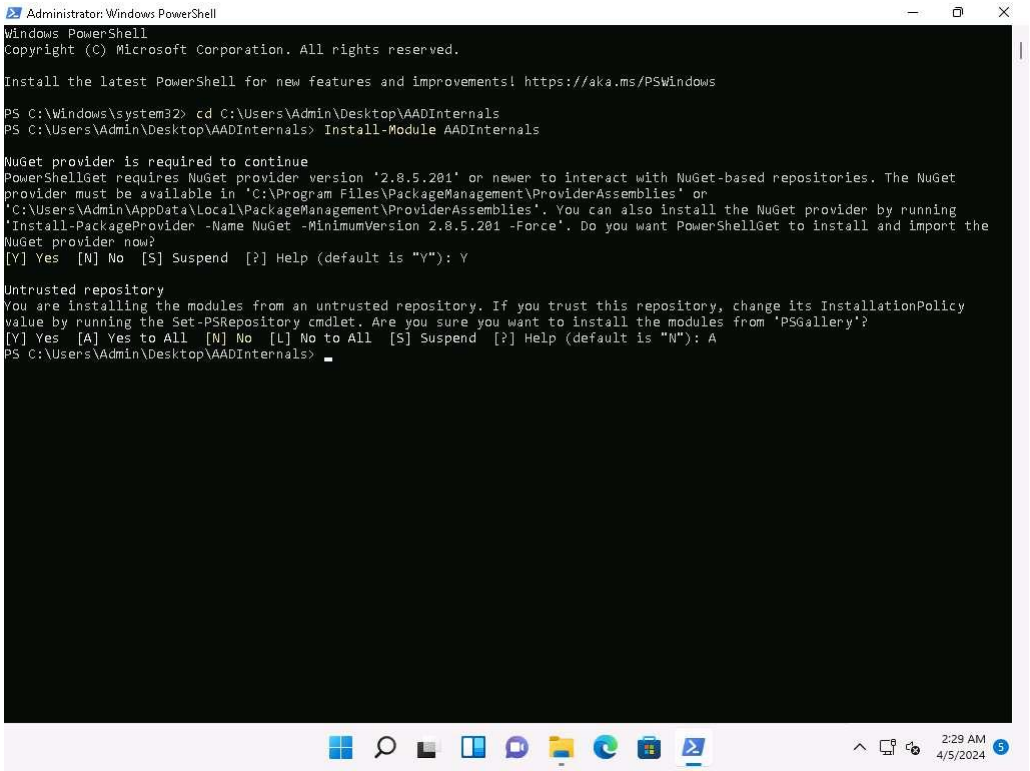
3. In the Windows search type **powershell** and under **PowerShell** click on **Run as Administrator** to open an administrator PowerShell window.

   If a **User Account Control** window appears, click **Yes**.

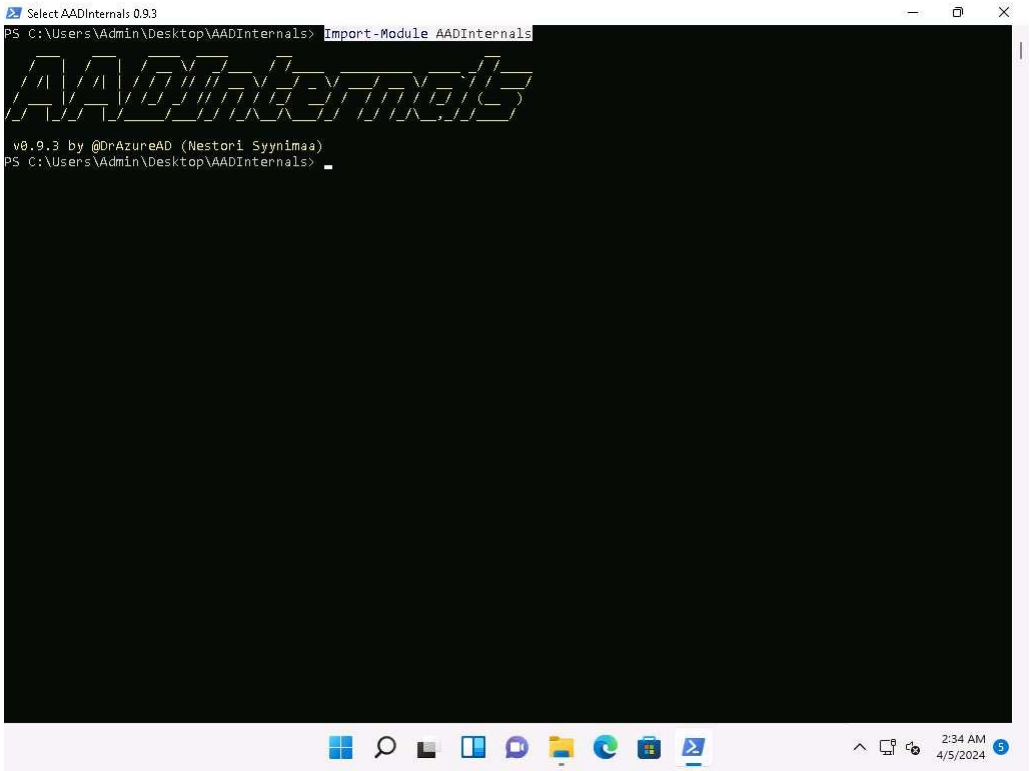
4. In the PowerShell window run **cd C:\Users\Admin\Desktop\AADInternals** command to navigate to **AADInternals** folder.

5. In the PowerShell window run **Install-Module AADInternals** command to install AADInternals module.

   In the **Do you want PowerShellGet to install and import the NuGet provider now?** Question type **Y** and press **Enter**. In the Are you sure you want to install the modules from "PSGallery"? question type **A** and press **Enter**.



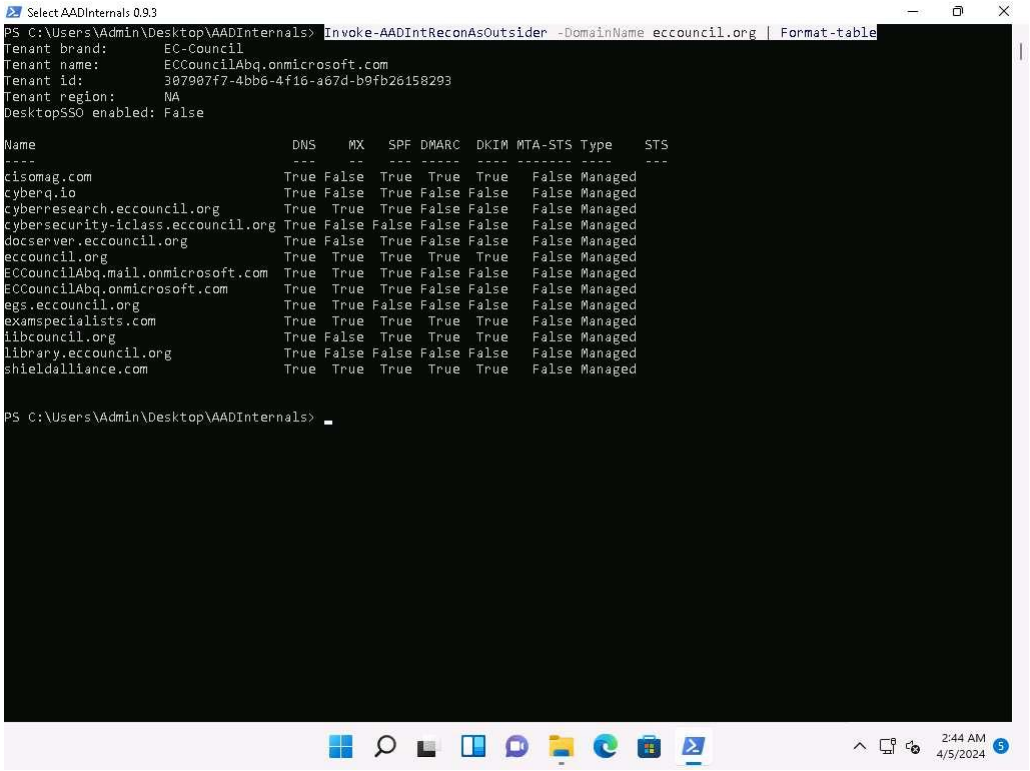
6. Now, run **Import-Module AADInternals** command, to import **AADInternals** module.



7. Now, we will gather the publicly available information of a target Azure AD such as Tenant brand, Tenant name, Tenant ID along with the names of the verified domains.

8. In the PowerShell window run **Invoke-AADIntReconAsOutsider -DomainName company.com | Format-table** command.

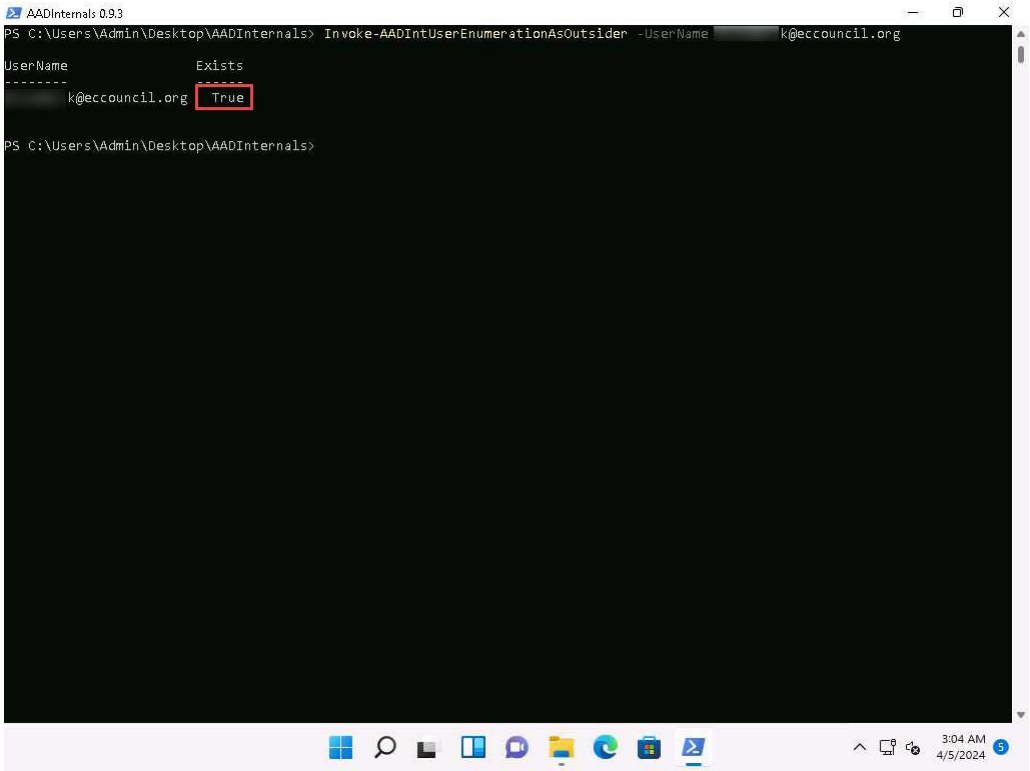
   In the above command replace the company.com with the target company's domain (here, we are using eccouncil.org).



9. From the above screenshot we can gather information such as **DNS**, **MX**, **SPF**, **DMARC**, **DKIM** etc.

10. Now, we will perform user enumeration in Azure AD, in the PowerShell window type **Invoke-AADIntUserEnumerationAsOutsider -UserName user@company.com** and press **Enter**.

    In the above command replace the user@company.com with the target users email address.

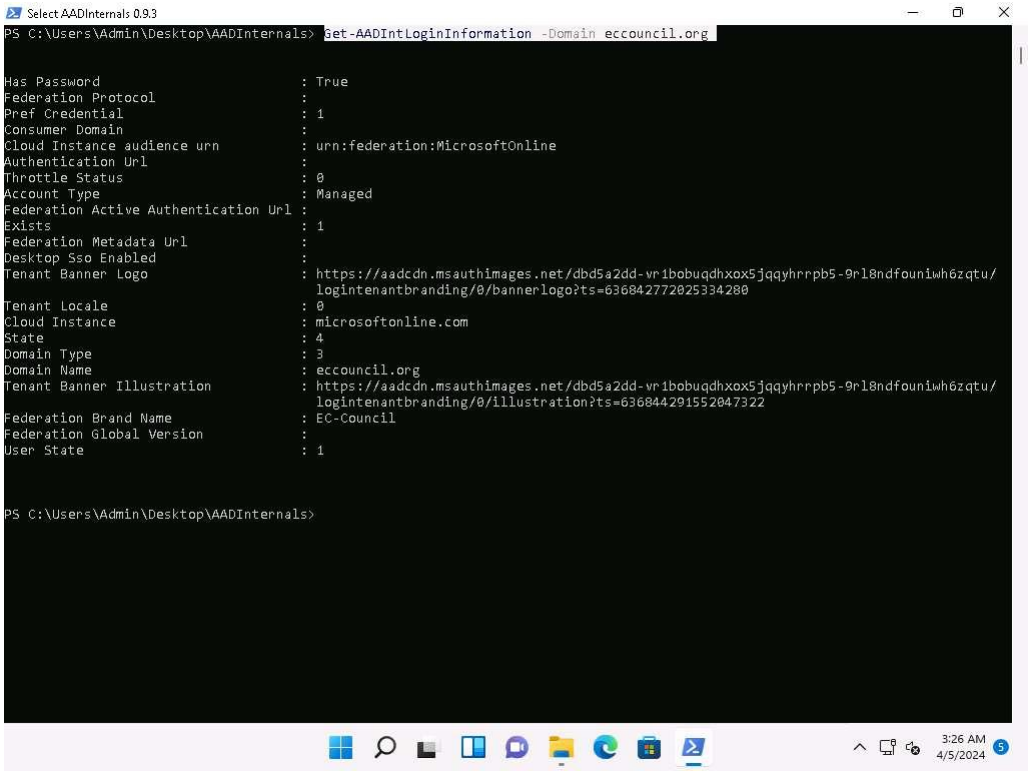


11. We can see that the result appears, **True** under **Exists** field which implies that the Azure account with the given username exists and the attacker can perform further attacks.

12. We can also perform the user enumeration by placing the usernames in a text file, by running **Get-Content .\users.txt | Invoke-AADIntUserEnumerationAsOutsider -Method Normal**. Where the users.txt file contains the target email addresses.

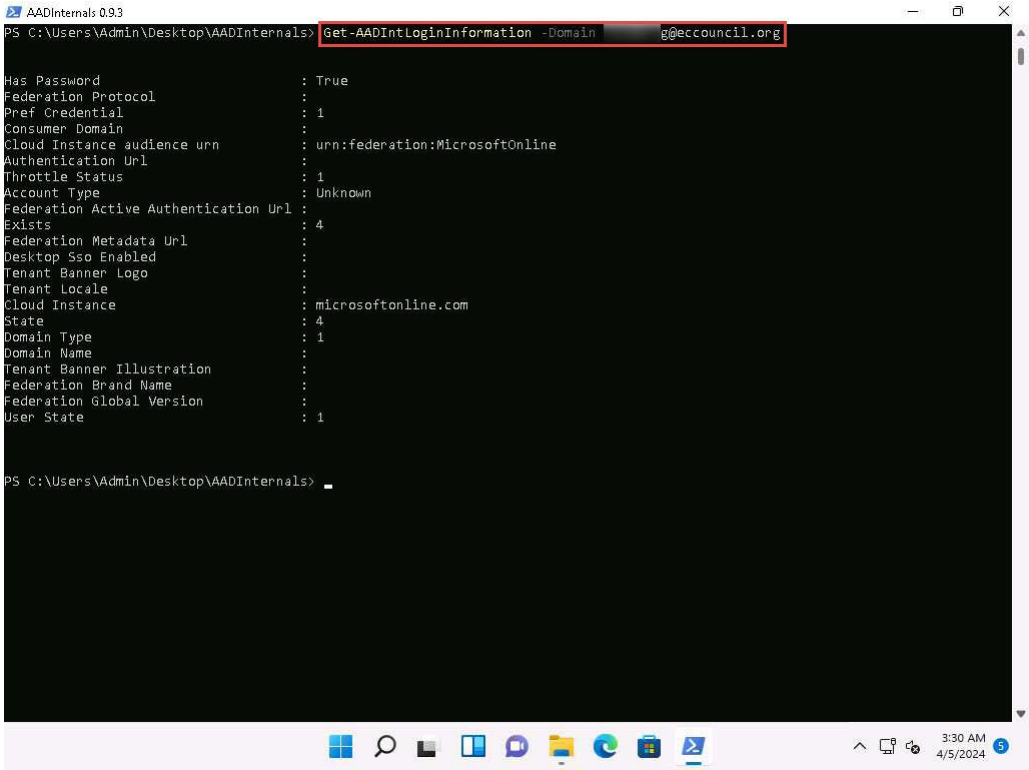
13. Now, to get login information for a domain type **Get-AADIntLoginInformation -Domain company.com** and press **Enter**.

    In the above command replace the company.com with the target company's domain (here, we are using eccouncil.org).



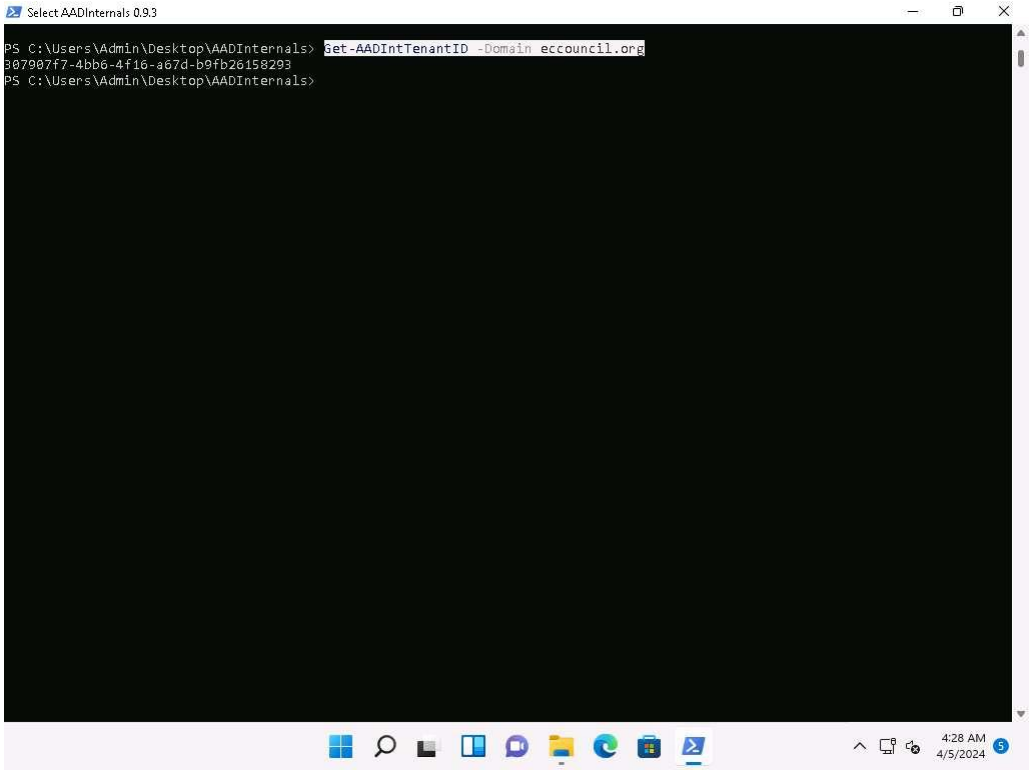
14. Now, to get login information for a user type **Get-AADIntLoginInformation -Domain user@company** and press **Enter**.

    In the above command replace the user@company.com with the target users email address.



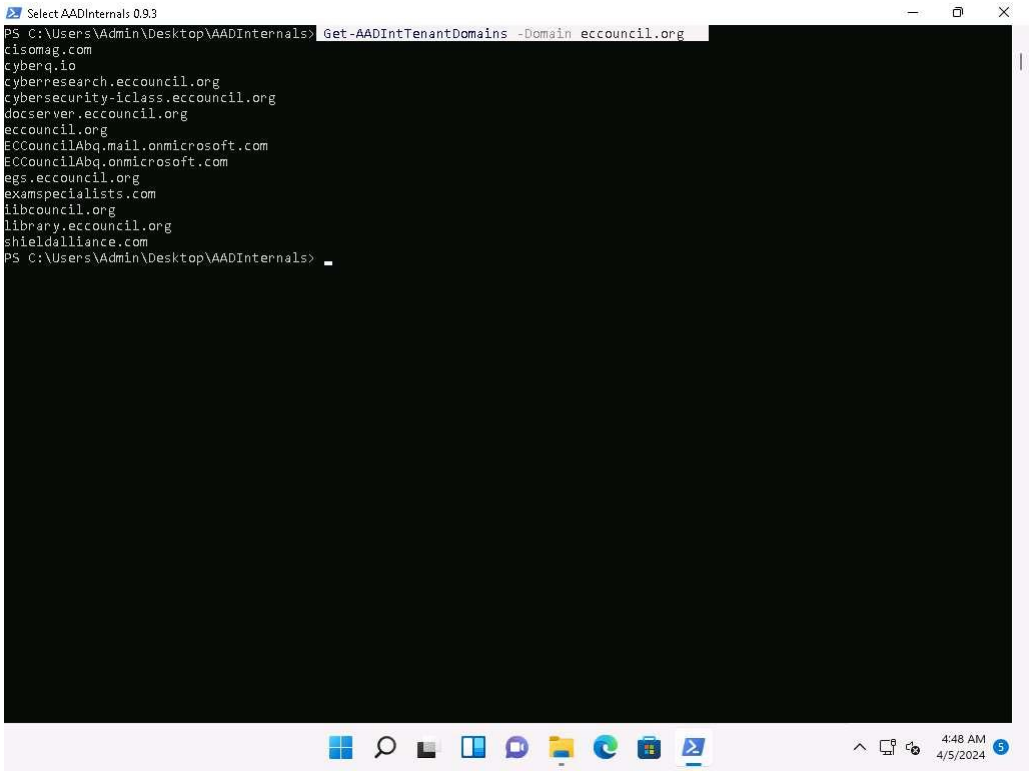
15. To get the tenant ID for the given user, domain, or Access Token, type **Get-AADIntTenantID -Domain company.com**.

    In the above command replace the company.com with the target company's domain (here, we are using eccouncil.org).



16. To get registered domains from the tenant of the given domain **Get-AADIntTenantDomains -Domain company.com**

    In the above command replace the company.com with the target company's domain (here, we are using eccouncil.org).



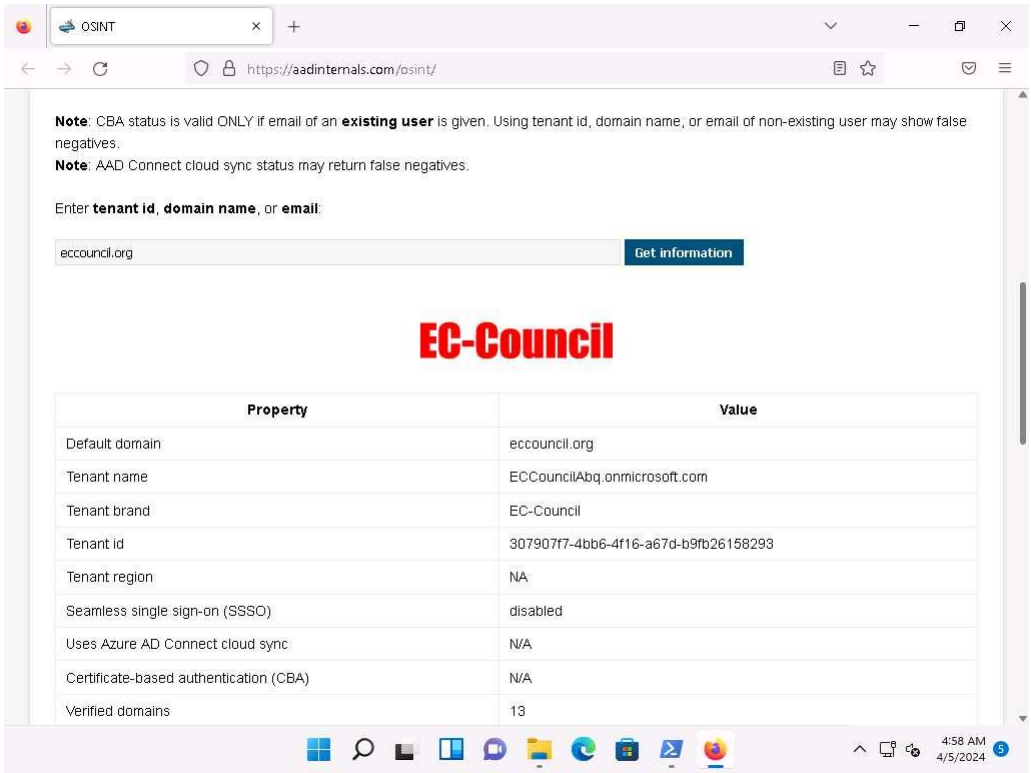
17. We can see that all the domains associated with the tenant will be listed.

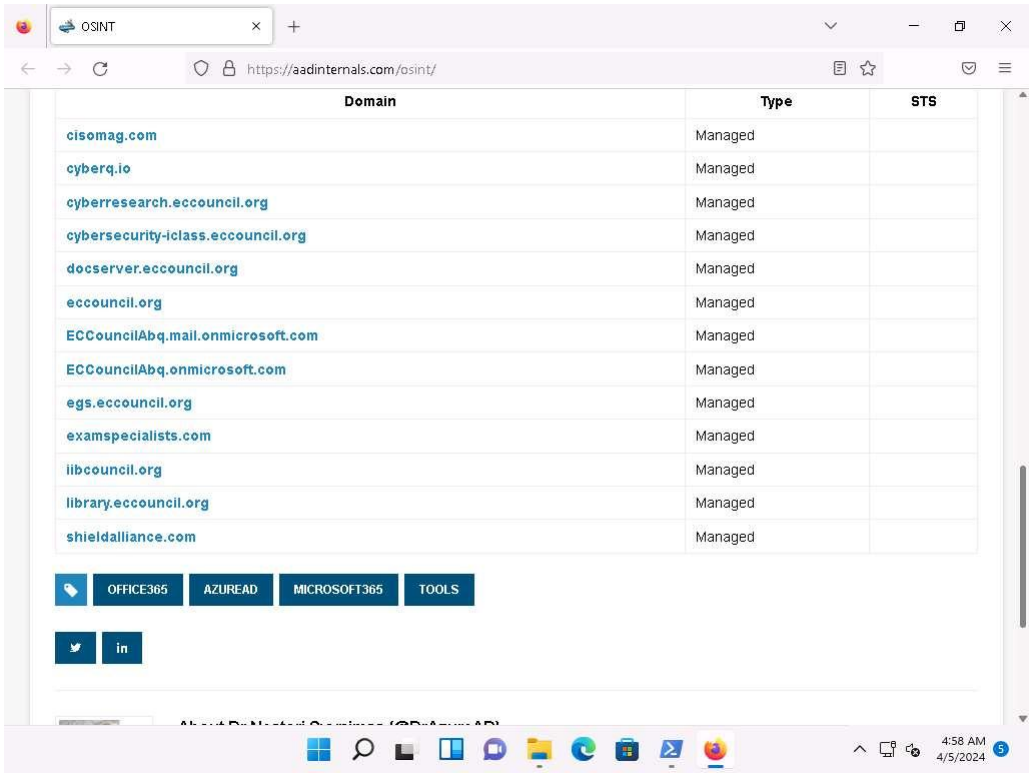
18. Alternatively you can visit **https://aadinternals.com/osint/** site and type the tenant ID, domain name, or email to get the openly available information for the given tenant.

19. Launch Firefox browser and go to **https://aadinternals.com/osint/** and type the **domain name** in the search box and click on **Get information** button.

    Here we are giving the domain name as eccouncil.org.

20. We will get the Domain information and the list of domains connected with the provided domain name.





21. In similar way you can enter the tenant ID and email in the search field to view the information regarding the tenant and the user.

22. This concludes the demonstration of Azure reconnaissance with AADInternals.

23. Close all open windows and document all acquired information.

Question 19.1.1.1

On windows 11 machine use AADIntenals tool located at E:\CEH-Tools\CEHv13 Module 19 Cloud Computing\GitHub Tools\ to perform Reconnaissance on Azure AD.While performing user enumeration in Azure AD what does the Exists field display if the user exists.

Lab 2: Exploit S3 Buckets

Lab Scenario

As a professional ethical hacker or pen tester, you must have sound knowledge of enumerating S3 buckets. Using various techniques, you can exploit misconfigurations in bucket implementation and breach the security mechanism to compromise data privacy. Leaving the S3 bucket session running enables you to modify files such as JavaScript or related code and inject malware into the bucket files. Furthermore, finding the bucket's location and name will help you in testing its security and identifying vulnerabilities in the implementation.

Lab Objectives

- Exploit open S3 buckets using AWS CLI

Overview of S3 Buckets

S3 buckets are used by customers and end users to store text documents, PDFs, videos, images, etc. To store all these data, the user needs to create a bucket with a unique name.

Listed below are several techniques that can be adopted to identify AWS S3 Buckets:

- **Inspecting HTML**: Analyze the source code of HTML web pages in the background to find URLs to the target S3 buckets
- **Brute-Forcing URL**: Use Burp Suite to perform a brute-force attack on the target bucket's URL to identify its correct URL
- **Finding subdomains**: Use tools such as Findsubdomains and Robtex to identify subdomains related to the target bucket
- **Reverse IP Search**: Use search engines such as Bing to perform reverse IP search to identify the domains of the target S3 buckets
- **Advanced Google hacking**: Use advanced Google search operators such as **"inurl"** to search for URLs related to the target S3 buckets

Task 1: Exploit Open S3 Buckets using AWS CLI

The AWS command line interface (CLI) is a unified tool for managing AWS services. With just one tool to download and configure, you can control multiple AWS services from the command line and automate them through scripts.

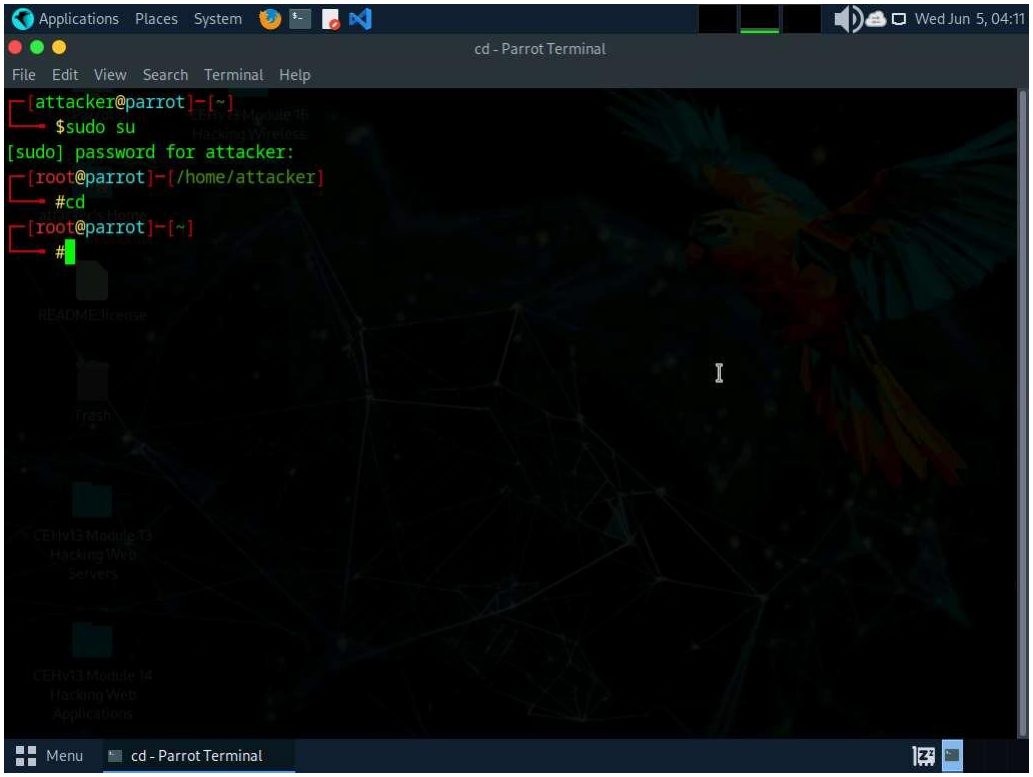
Before starting this task, you must create your AWS account (**https://aws.amazon.com**).

1. In the **Parrot Security** machine, click the **MATE Terminal** icon in the menu to launch the terminal.

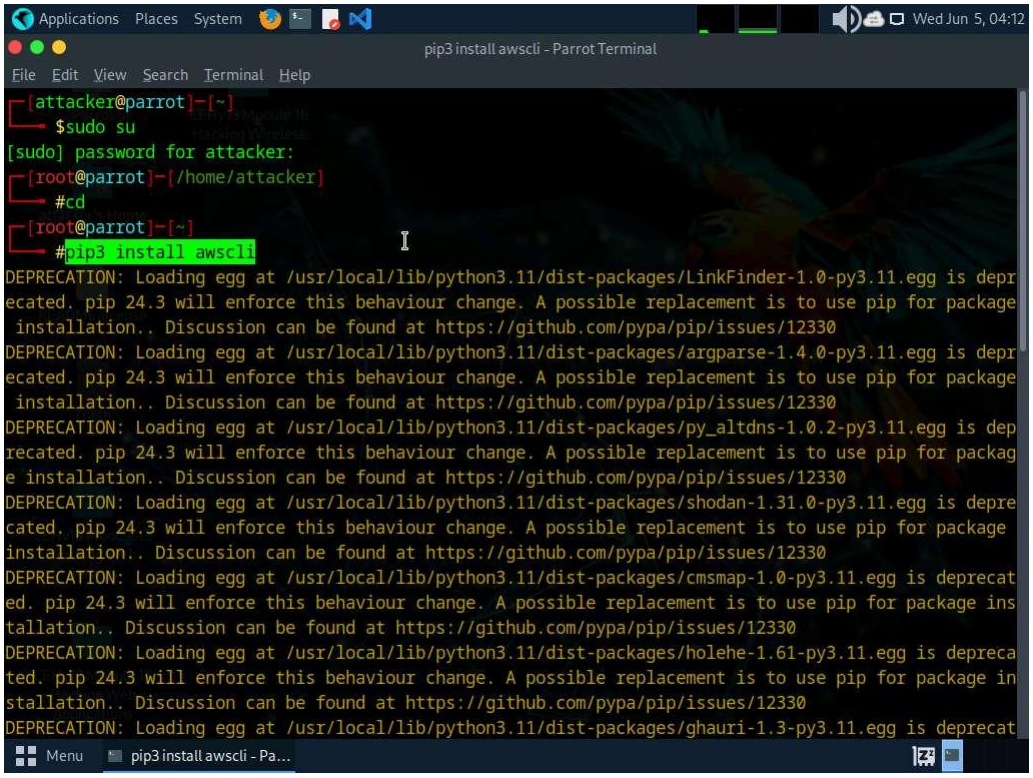
2. A **Parrot Terminal** window appears. In the terminal window, type **sudo su** and press **Enter** to run the programs as a root user use **toor** as password.

   The password that you type will not be visible.

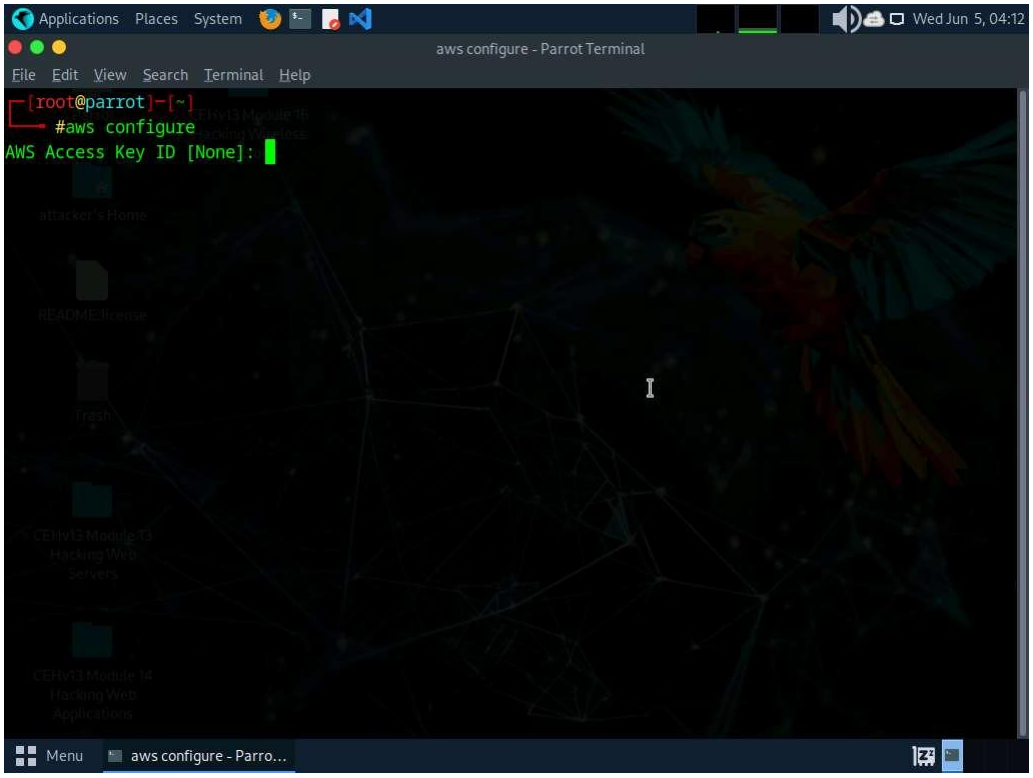
3. Now, type **cd** and press **Enter** to jump to the root directory.



4. In the terminal window, type **pip3 install awscli** and press **Enter** to install AWS CLI.



5. Now, we need to configure AWS CLI. To configure AWS CLI in the terminal window, type **aws configure** and press **Enter**.



6. It will ask for the following details:

    - AWS Access Key ID

- AWS Secret Access Key
- Default region name
- Default output format

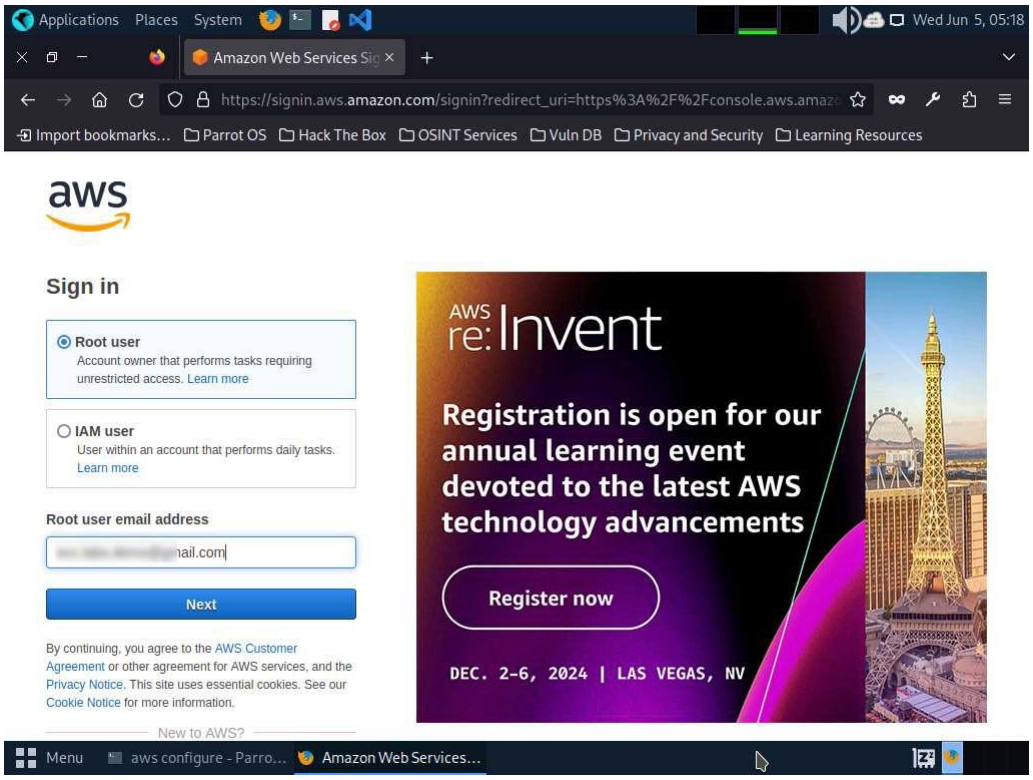
7. To provide these details, you need to login to your AWS account.

8. Click **Firefox** icon from the top-section of the **Desktop**.

9. Login to your AWS account that you created at the beginning of this task. Click the **Firefox** browser icon in the menu, type **https://console.aws.amazon.com** in the address bar, and press **Enter**.

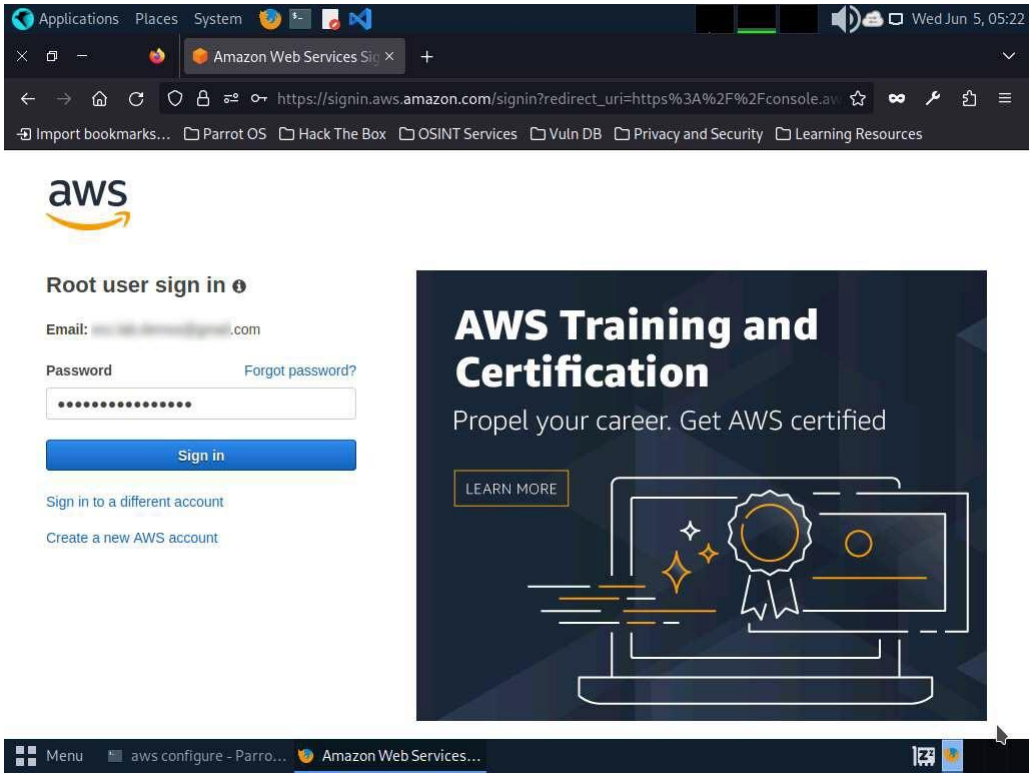
   If you do not have an AWS account, create one with the Basic Free Plan, and then proceed with the tasks.

10. The **Amazon Web Services Sign-In** page appears; type your email account in the **Root user email address** field and click **Next**.

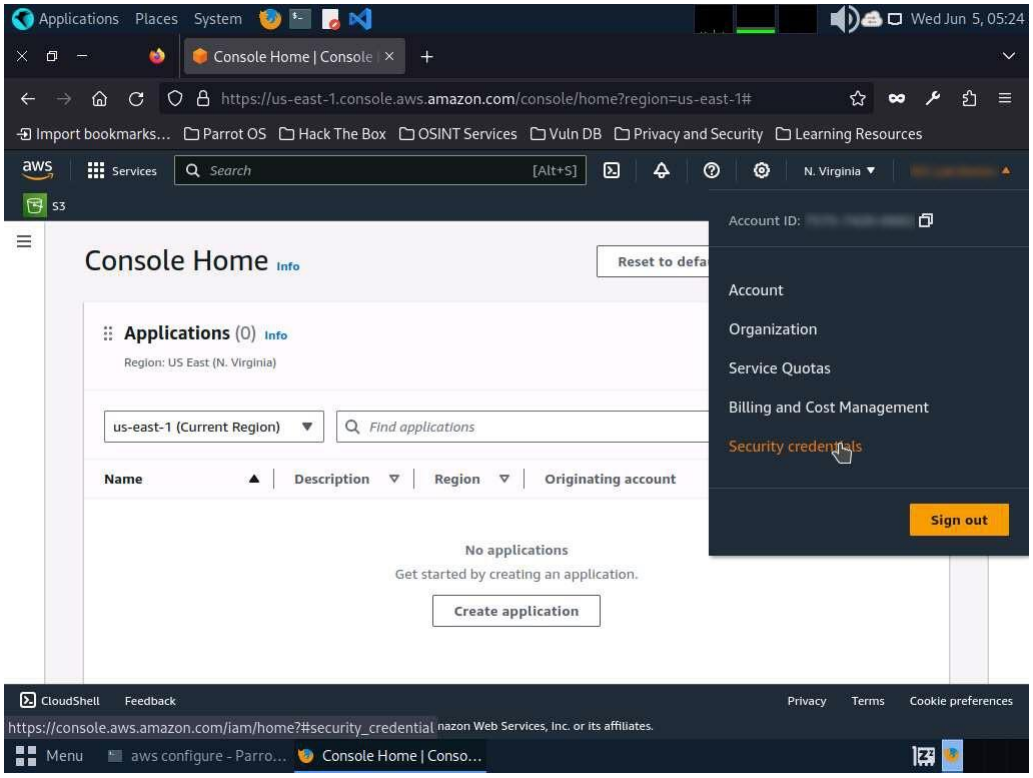


11. Type your AWS account password in the **Password** field and click **Sign in**.

    If a **Security check** window appears, enter the captcha and click on **Submit**.

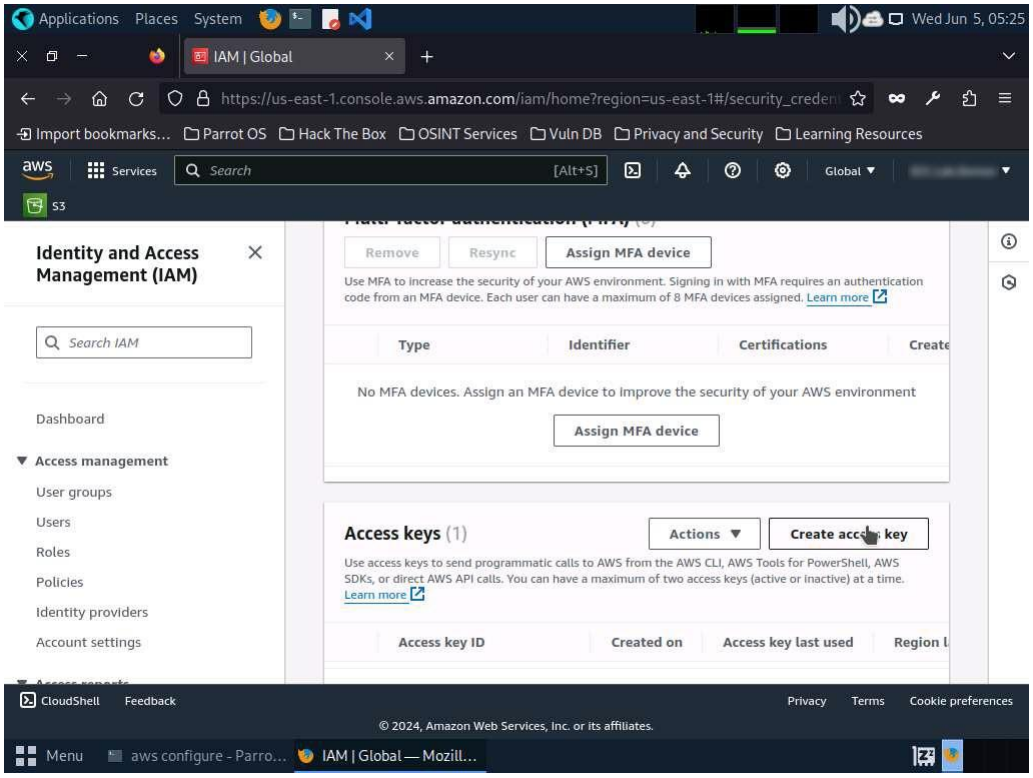


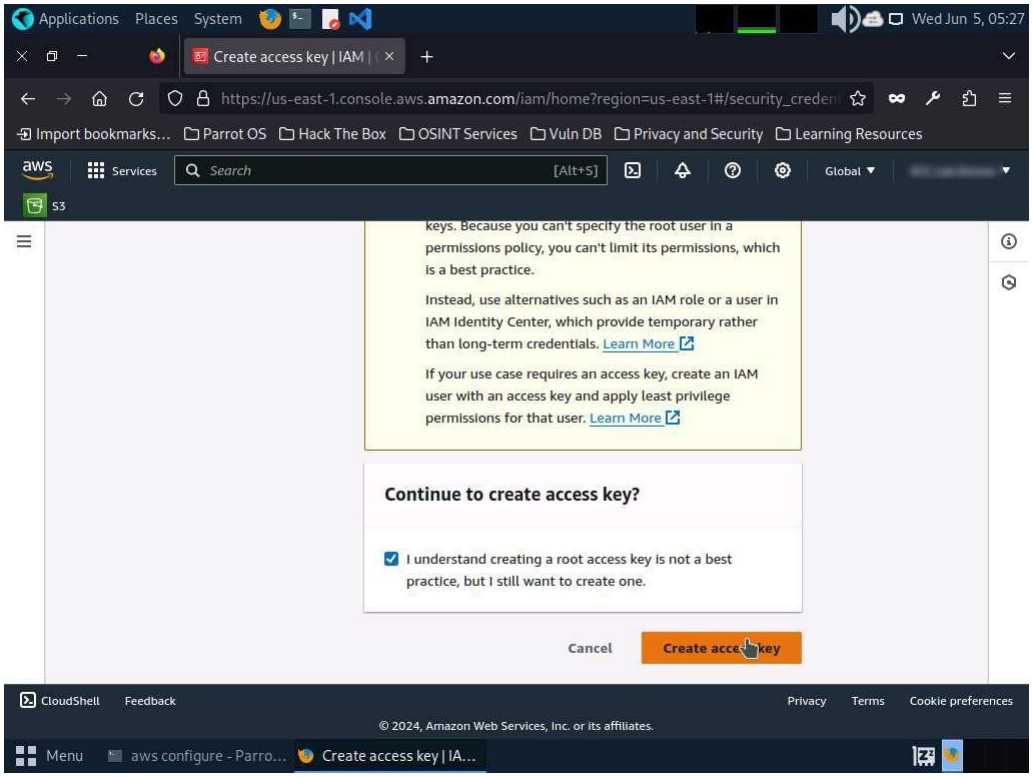
12. Click the AWS account drop-down menu and click **Security credentials**, as shown in the screenshot.



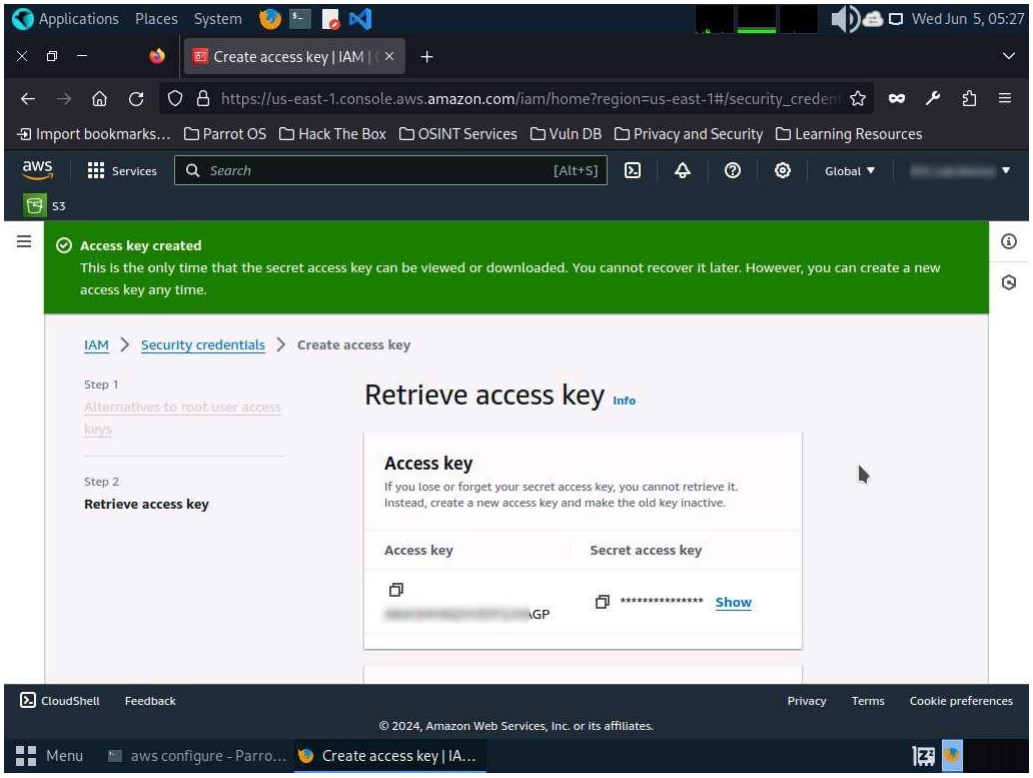
13. Scroll down to **Access Keys** section.

14. Click the **Create Access Key** button. In **Continue to create access key?**; check the check box and click **Create access key**.





15. Copy the **Access Key** and switch to the **Terminal** window.



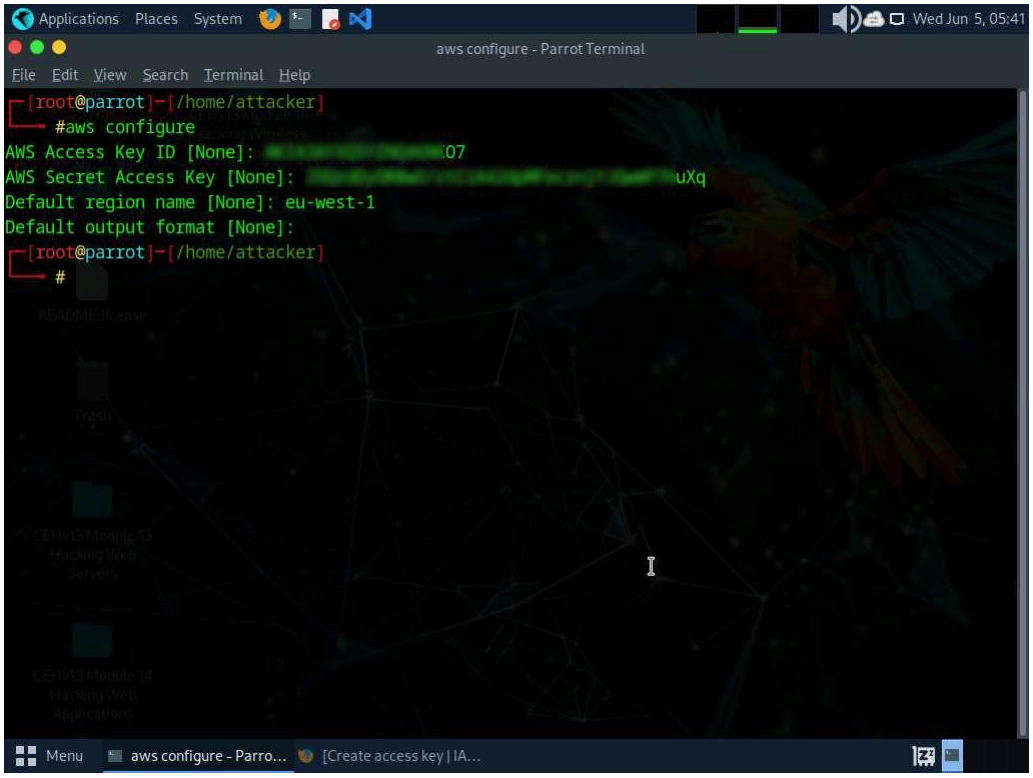
16. In the terminal window, right-click your mouse; select **Paste** from the context menu to paste the copied **AWS Access Key ID** and press **Enter**. It will prompt you to the **AWS Secret Access Key**. Switch to your AWS Account in the browser.

17. Copy the **Secret Access Key** and minimize the browser window. Switch to the **Terminal** window.

18. In the terminal window, right-click your mouse, select **Paste** from the context menu to paste the copied **Secret Access Key** and press **Enter**. It will prompt you for the default region name.

19. In the **Default region name** field, type **eu-west-1** and press **Enter**.

20. The **Default output format** prompt appears; leave it as default and press **Enter**.



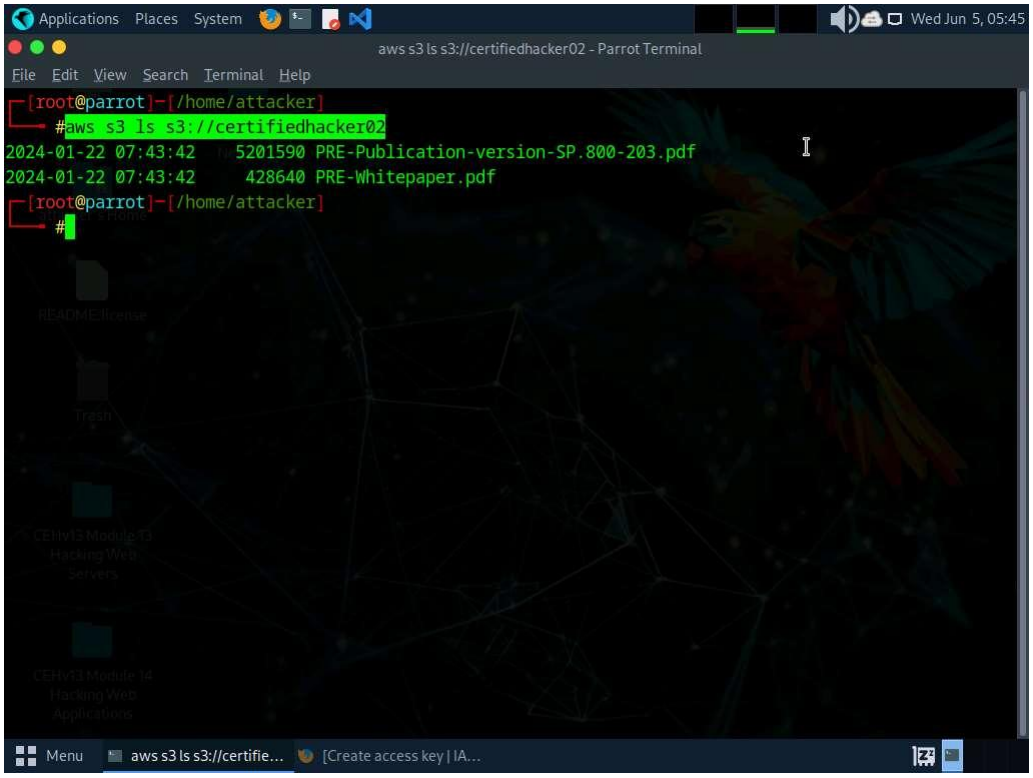
21. For demonstration purposes, we have created an open S3 bucket with the name **certifiedhacker02** in the AWS service. We are going to use that bucket in this task.

    The public S3 buckets can be found during the enumeration phase.

22. Let us list the directories in the certifiedhacker02 bucket. In the terminal window, type **aws s3 ls s3://[Bucket Name]** (here, Bucket Name is **certifiedhacker02**) and press **Enter**.

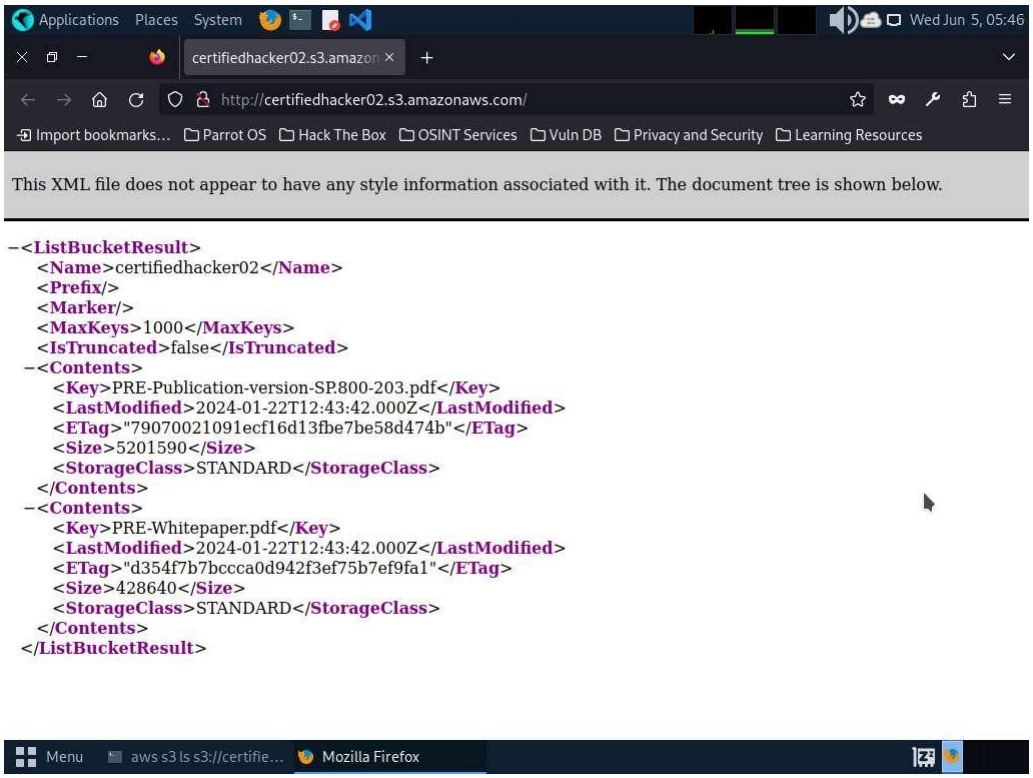
    The bucket name may be different in your lab environment depending on the bucket you are targeting.

23. This will show you the list of directories in the **certifiedhacker02** S3 bucket, as shown in the screenshot.



24. Now, maximize the browser window, type **certifiedhacker02.s3.amazonaws.com** in the address bar, and press **Enter**.

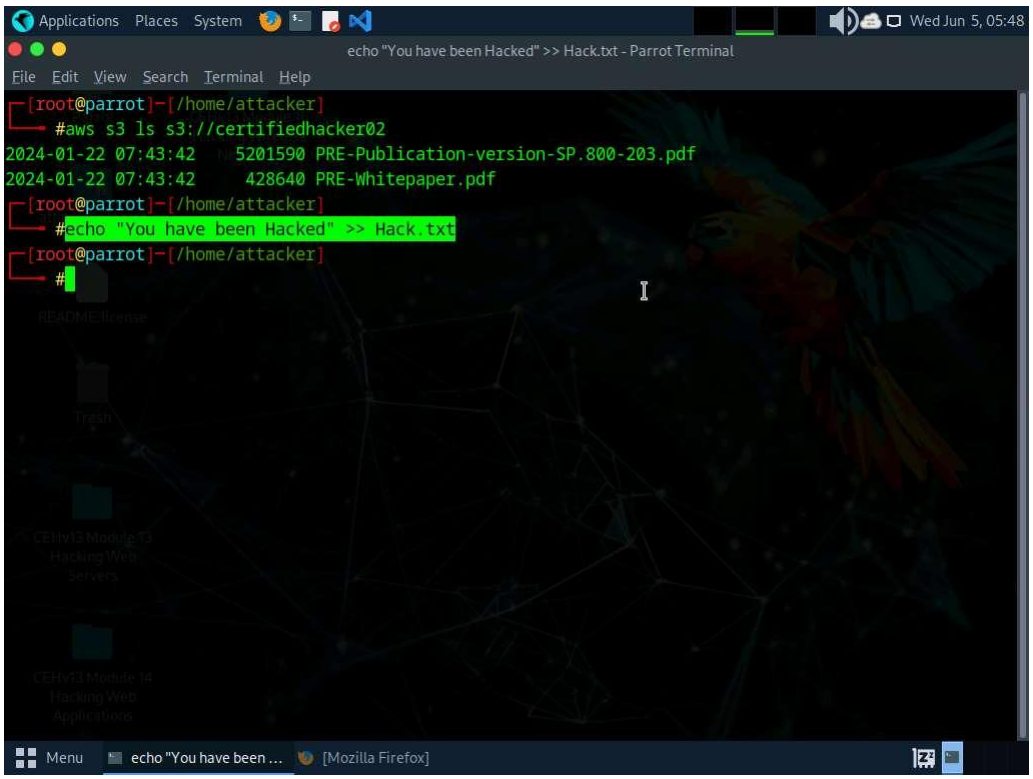
25. This will show you the complete list of directories and files available in this bucket.



26. Minimize the browser window and switch to **Terminal**.

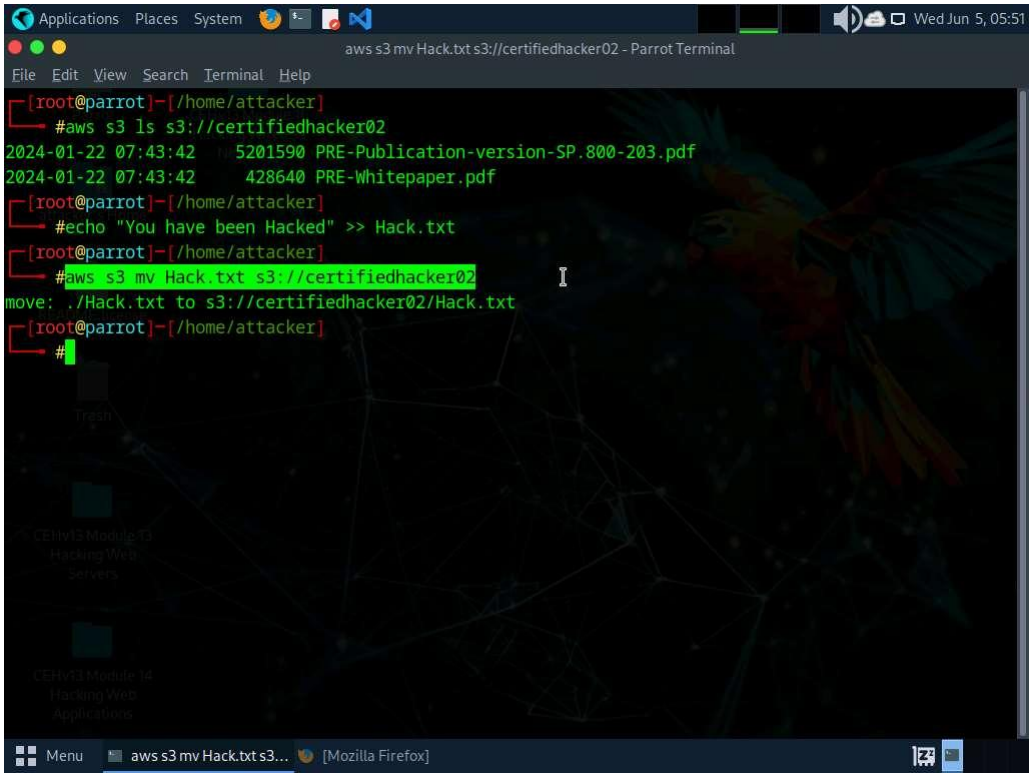
27. Let us move some files to the certifiedhacker02 bucket. To do this, in the terminal window, type **echo "You have been hacked" >> Hack.txt** and press **Enter**.

28. By issuing this command, you are creating a file named **Hack.txt**.



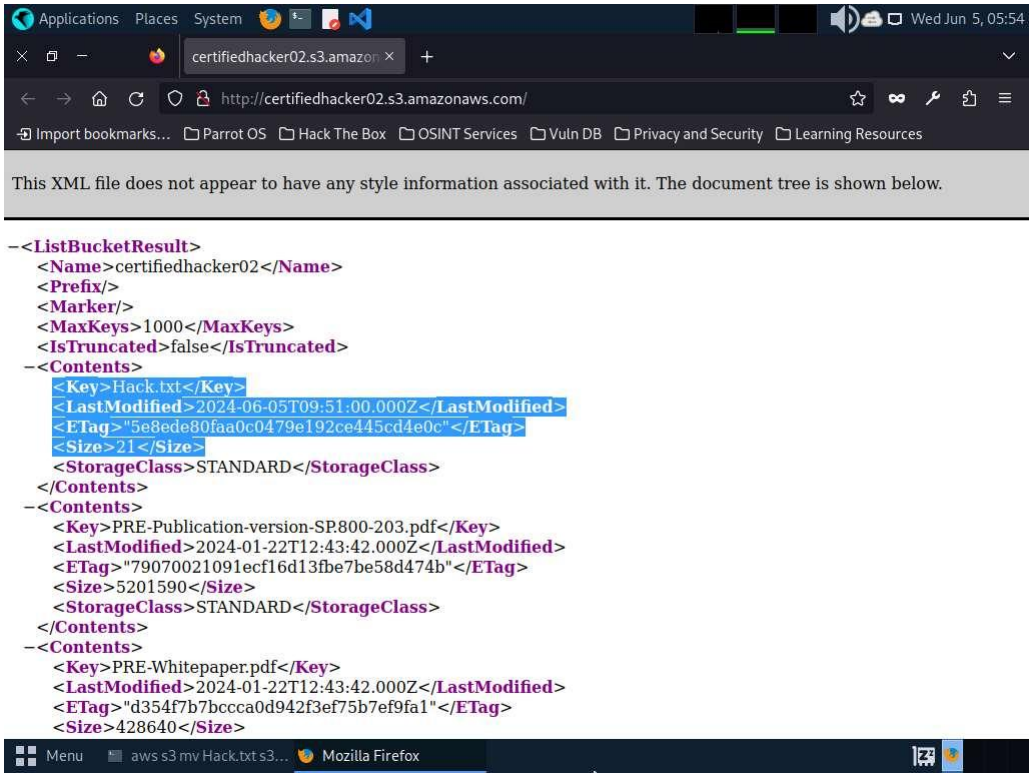
29. Let us try to move the **Hack.txt** file to the **certifiedhacker02** bucket. In the terminal window, type **aws s3 mv Hack.txt s3://certifiedhacker02** and press **Enter**.

30. You have successfully moved the **Hack.txt** file to the **certifiedhacker02** bucket.



31. To verify whether the file is moved, switch to the browser window and maximize it. Reload the page.

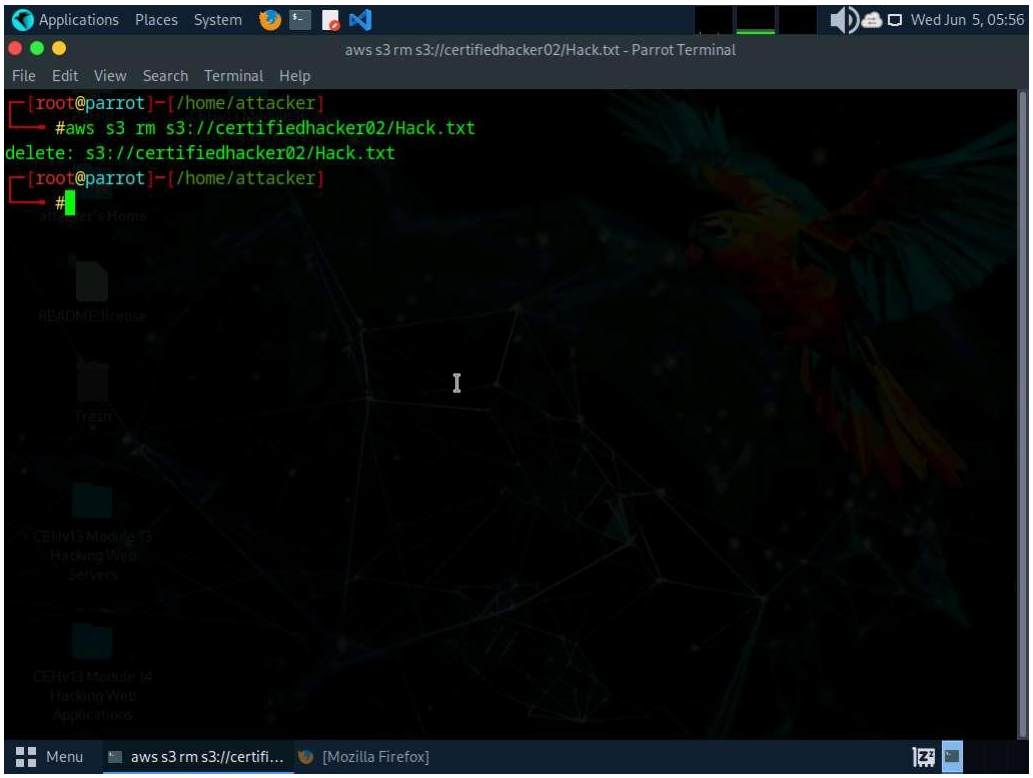
32. You can observe that the **Hack.txt** file is moved to the certifiedhacker02 bucket, as shown in the screenshot.



33. Minimize the browser window and switch to the **Terminal** window.

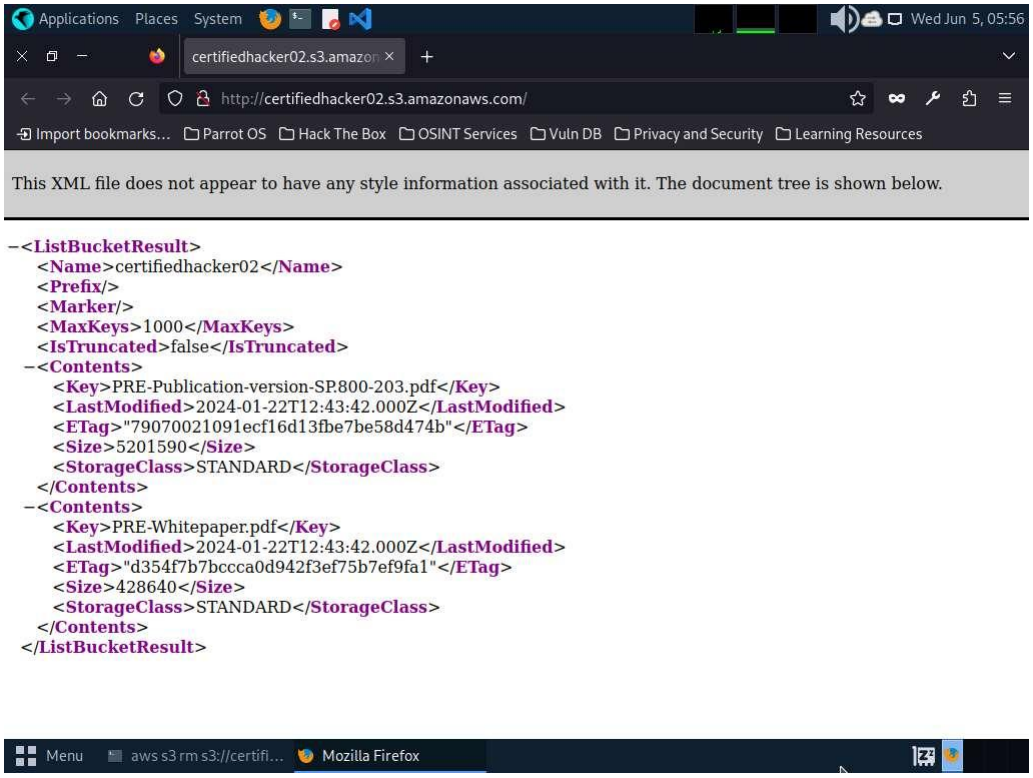
34. Let us delete the **Hack.txt** file from the **certifiedhacker02** bucket. In the terminal window, type **aws s3 rm s3://certifiedhacker02/Hack.txt** and press **Enter**.

35. By issuing this command, you have successfully deleted the **Hack.txt** file from the **certifiedhacker02** bucket.



36. To verify whether the file is deleted, switch to the browser window and reload the page.

37. The **Hack.txt** file is deleted from the **certifiedhacker02** bucket.



38. Thus, you can add or delete files from open S3 buckets.

39. This concludes the demonstration of exploiting public S3 buckets.

40. Do not end the lab as we will be continuing it in next #Task.

Question 19.2.1.1

Use the AWS CLI tool to exploit open S3 buckets (certifiedhacker1) in the AWS service. Find the total number of files available in the “certifiedhacker1” S3 bucket. Note: You must create an AWS account (https://aws.amazon.com) to perform this task. Enter the command that was used in this lab to list the contents of certifiedhacker02 bucket.

Lab 3: Perform Privilege Escalation to Gain Higher Privileges

Lab Scenario

As a professional ethical hacker or pen tester, you must try to escalate privileges by employing a user account access key and secret access key obtained using various social engineering techniques. In privilege escalation, you attempt to gain complete access to the target IAM user’s account and, then try to attain higher-level privileges in the AWS environment.

In the cloud platform, owing to mistakes in the access allocation system such as coding errors and design flaws, a customer, a third party, or an employee can obtain higher access rights

than those that they are authorized to use. This threat arises, because of authentication, authorization, and accountability (AAA) vulnerabilities, user provisioning and de-provisioning vulnerabilities, hypervisor vulnerabilities, unclear roles and responsibilities, misconfiguration, etc.

In this lab, we will exploit a misconfigured user permission policy to escalate privileges to the administrator level.

Lab Objectives

- Escalate IAM user privileges by exploiting misconfigured user policy

Overview of Privilege Escalation

Privileges are security roles assigned to users for using specific programs, features, OSes, functions, files, code, etc. to limit access depending on the type of user. Privilege escalation is required when you want to access system resources that you are not authorized to access. It takes place in two forms: vertical and horizontal.

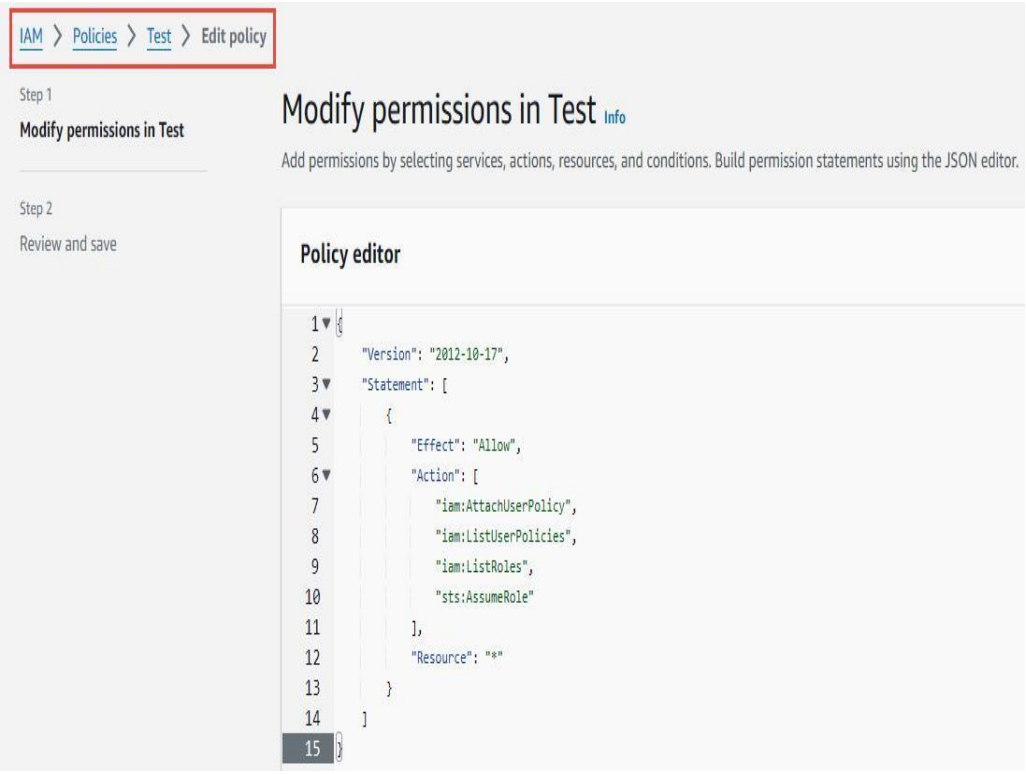
- **Horizontal Privilege Escalation**: An unauthorized user tries to access the resources, functions, and other privileges of an authorized user who has similar access permissions
- **Vertical Privilege Escalation**: An unauthorized user tries to access the resources and functions of a user with higher privileges such as application or site administrators

Task 1: Escalate IAM User Privileges by Exploiting Misconfigured User Policy

A policy is an entity that, when attached to an identity or resource, defines its permissions. You can use the AWS Management Console, AWS CLI, or AWS API to create customer-managed policies in IAM. Customer-managed policies are standalone policies that you administer in your AWS account. You can then attach the policies to the identities (users, groups, and roles) in your AWS account. If the user policies are not configured properly, they can be exploited by attackers to gain full administrator access to the target user's AWS account.

You need to configure aws cli for this lab refer to **Lab 2: Exploit S3 Buckets, Task 1: Exploit Open S3 Buckets using AWS CLI, Steps#1-20.**

Before starting this task, create an **IAM** user (**Test**) with default settings and create a policy (**Test**) with permissions including, iam:AttachUserPolicy, iam:ListUserPolicies, sts:AssumeRole, and iam:ListRoles, as shown in the below screenshot. These policies can be exploited by attackers to gain administrator-level privileges.



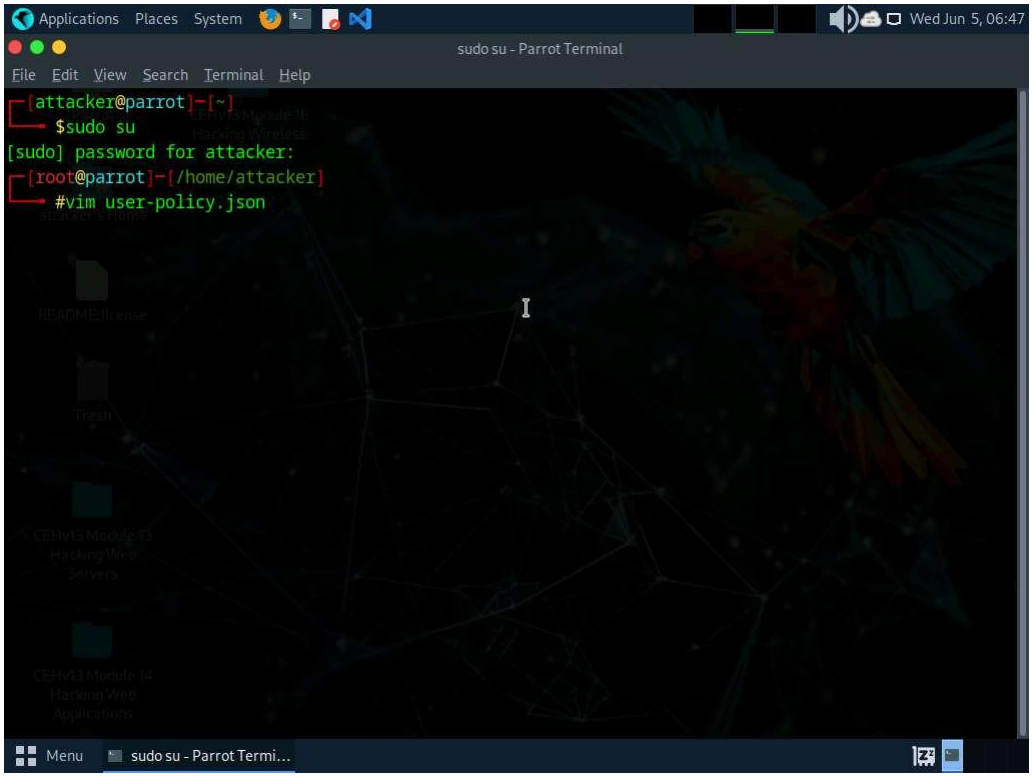
1. In the **Parrot Security** machine, click the **MATE Terminal** icon in the menu to launch the terminal.

2. A **Parrot Terminal** window appears. In the terminal window, type **sudo su** and press **Enter** to run the programs as a root user and user **toor** as password.

3. After configuring the AWS CLI, we create a user policy and attach it to the target IAM user account to escalate the privileges.

4. In the terminal window, type **vim user-policy.json** and press **Enter**.

   This command will create a file named **user-policy** in the **attacker** directory.



5. A command line text editor appears; press **I** and type the script given below:

   {

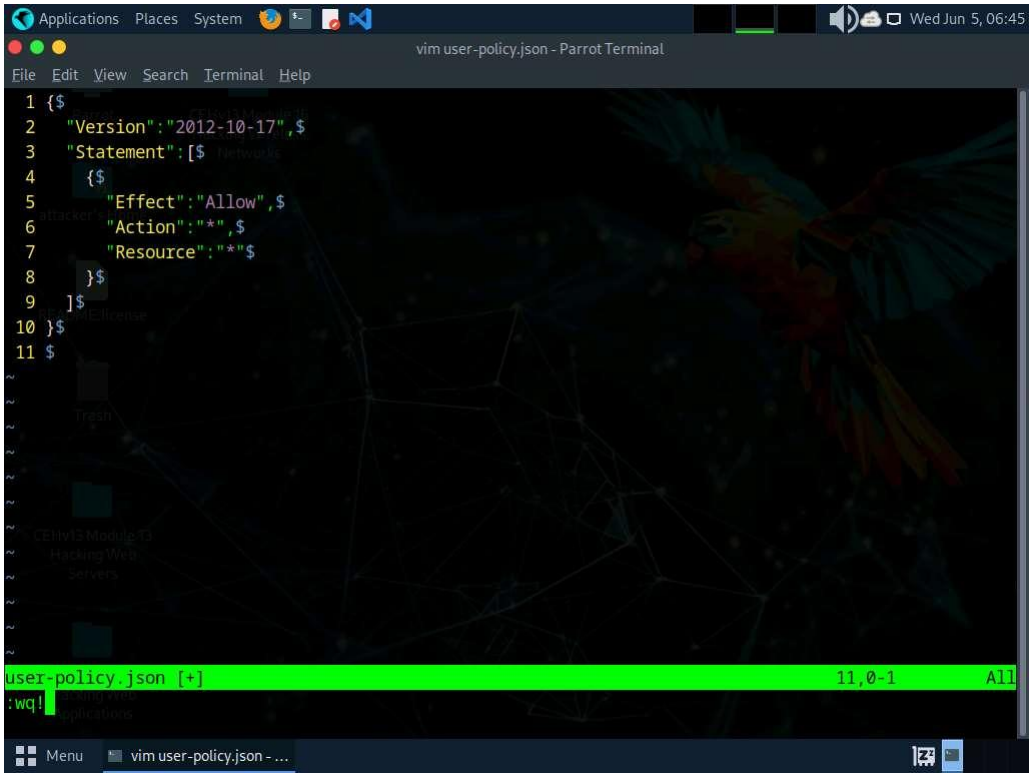
   TypeCopy
   ```
   "Version":"2012-10-17",

   "Statement": [

       "Effect":"Allow",

       "Action":"*",

       "Resource":"*"

   }
   ]
   ```

   }

   This is an AdministratorAccess policy that gives administrator access to the target IAM user.

   Ignore the $ symbols in the script.

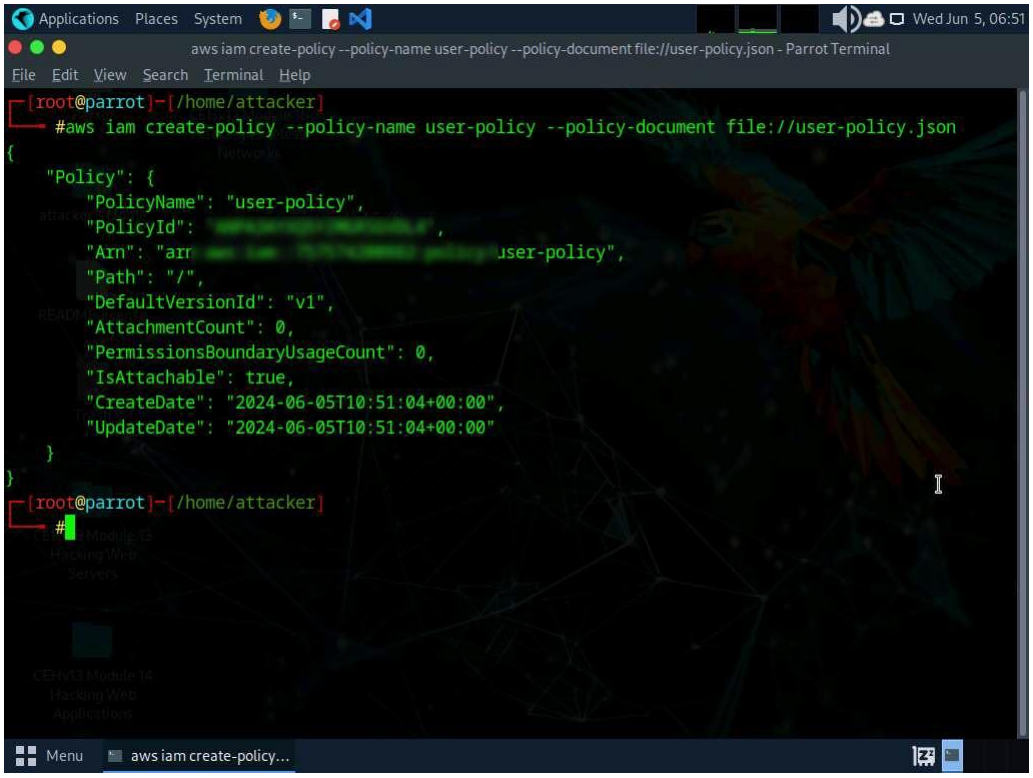
6. After entering the script given in the previous step, press the **Esc** button. Then, type **:wq!** and press **Enter** to save the text document.



7. Now, we will attach the created policy (**user-policy**) to the target IAM user's account. To do so, type **aws iam create-policy --policy-name user-policy --policy-document file://user-policy.json** and press **Enter**.

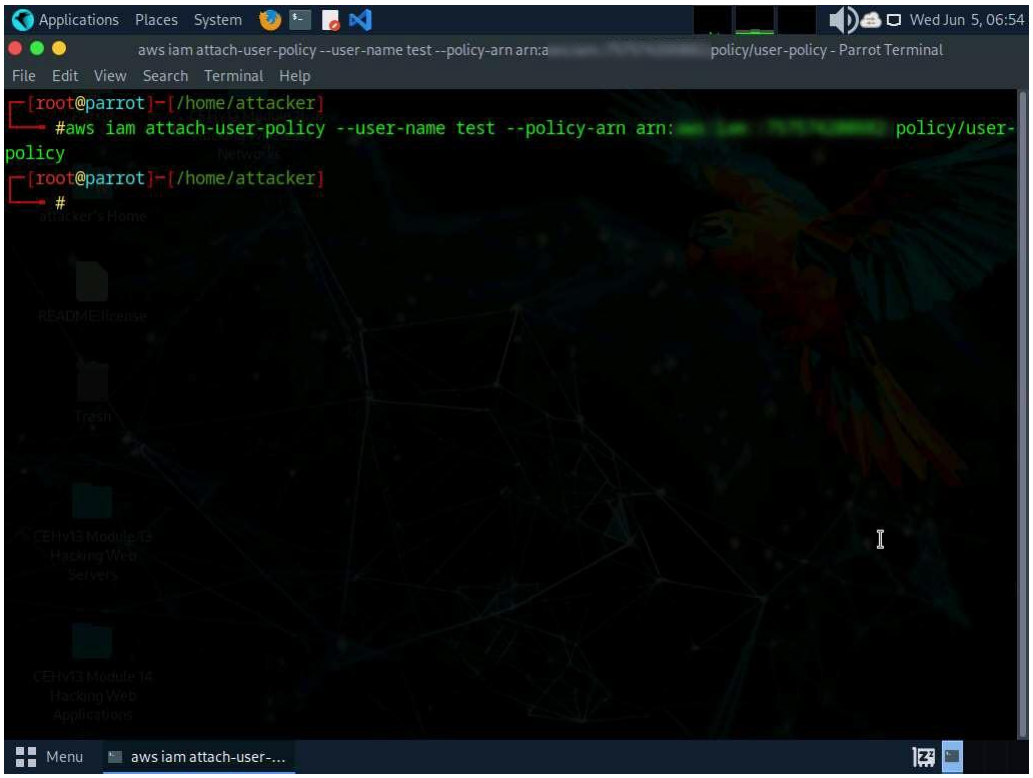
   If you receive an error that policy already exists, rename the file and try again.

8. The created user policy is displayed, showing various details such as **PolicyName**, **PolicyId**, and **Arn**.



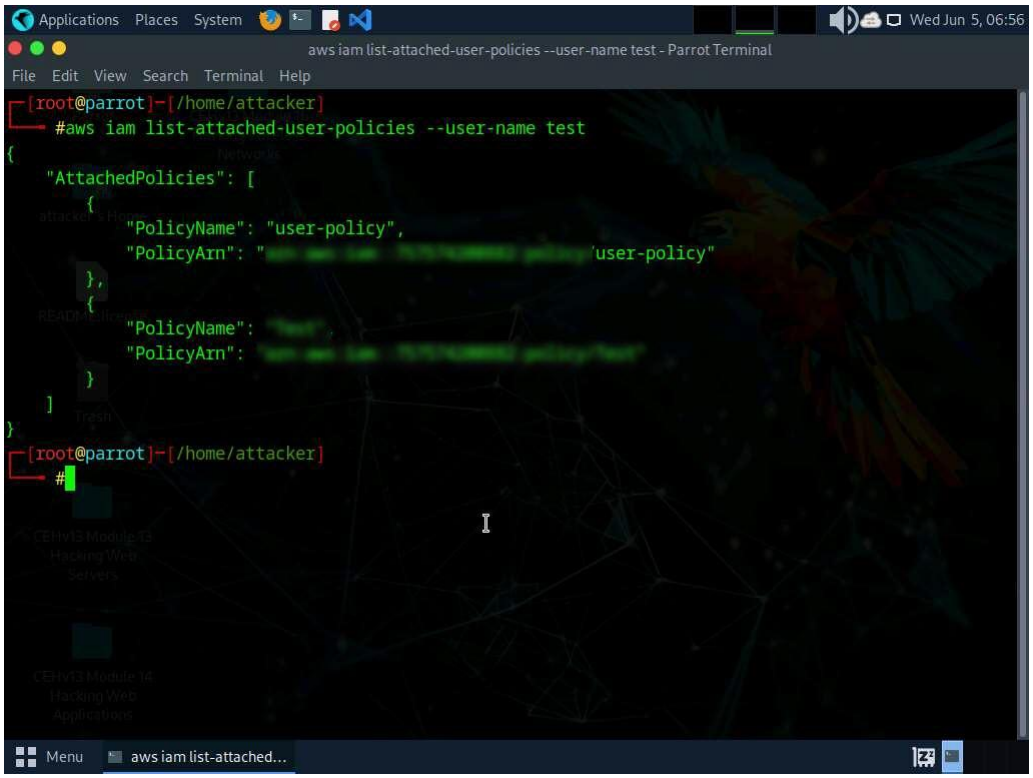
9. In the terminal, type **aws iam attach-user-policy --user-name [Target Username] --policy-arn arn:aws:iam::[Account ID]:policy/user-policy** and press **Enter**.

10. The above command will attach the policy (**user-policy**) to the target IAM user account (here, **test**).



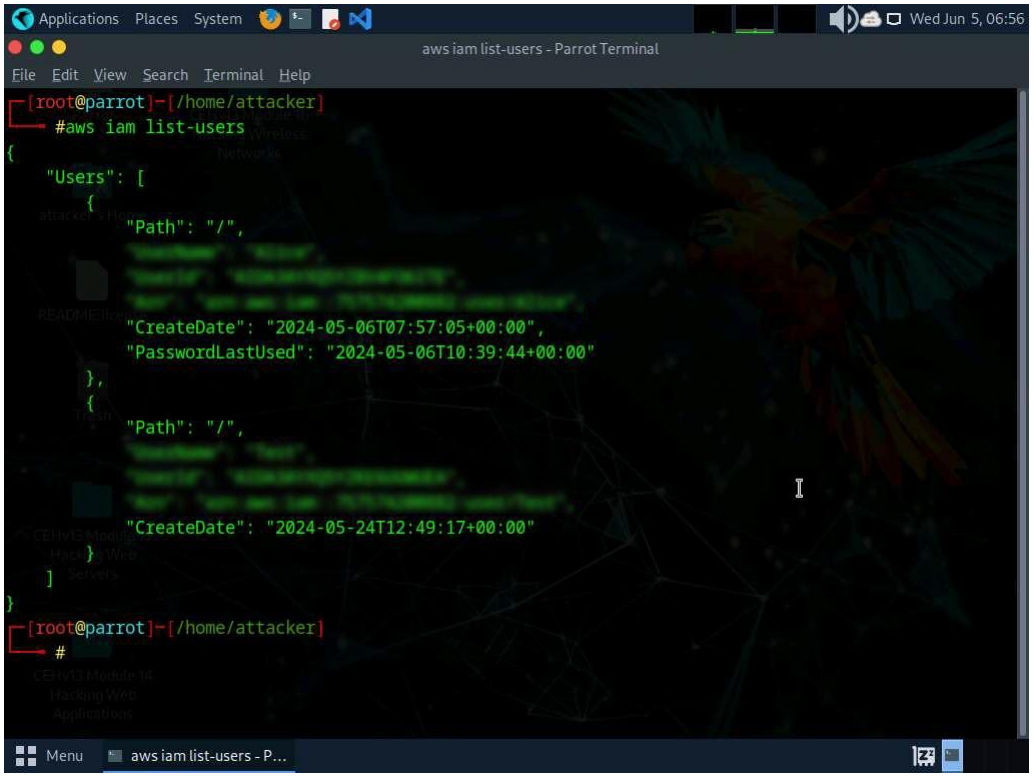
11. Now, type **aws iam list-attached-user-policies --user-name [Target Username]** and press **Enter** to view the attached policies of the target user (here, **test**).

12. The result appears, displaying the attached policy name (**user-policy**), as shown in the screenshot.



13. Now that you have successfully escalated the privileges of the target IAM user account, you can list all the IAM users in the AWS environment. To do so, type **aws iam list-users** and press **Enter**.

14. The result appears, displaying the list of IAM users, as shown in the screenshot.



15. Similarly, you can use various commands to obtain complete information about the AWS environment such as the list of S3 buckets, user policies, role policies, and group policies, as well as to create a new user.

    - List of S3 buckets: **aws s3api list-buckets --query "Buckets[].Name"**
    - User Policies: **aws iam list-user-policies**
    - Role Policies: **aws iam list-role-policies**
    - Group policies: **aws iam list-group-policies**
    - Create user: **aws iam create-user**

16. This concludes the demonstration of escalating IAM user privileges by exploiting a misconfigured user policy.

17. Close all open windows and document all acquired information.

Question 19.3.1.1

Escalate IAM user privileges by exploiting a misconfigured user policy. Which aws command will list all user policies?

Lab 4: Perform Vulnerability Assessment on Docker Images

Lab Scenario

As a professional ethical hacker or pen tester, expertise in Docker vulnerability assessment is crucial. By leveraging tools like Trivy, you can analyze Docker images, identifying and exploiting vulnerabilities. Active scanning and manual inspection reveal weak configurations, enabling you to breach security and implant malicious code, while understanding image location aids in comprehensive security testing and mitigation.

Lab Objectives

- Vulnerability assessment on Docker images using Trivy

Overview of Docker Images

Docker images are lightweight, standalone, executable packages that contain everything needed to run a software application, including the code, runtime, libraries, and dependencies. They enable consistent deployment across various environments, simplify software distribution, and facilitate scalability and reproducibility in containerized environments.

Task 1: Vulnerability Assessment on Docker Images using Trivy

Trivy is a powerful security scanner that detects vulnerabilities and misconfigurations across a wide range of targets, including container images, file systems, Git repositories, virtual machine images, Kubernetes, and AWS. With its comprehensive scanners, Trivy identifies OS package vulnerabilities, sensitive information, IaC issues, and more, providing a robust security solution for your infrastructure.

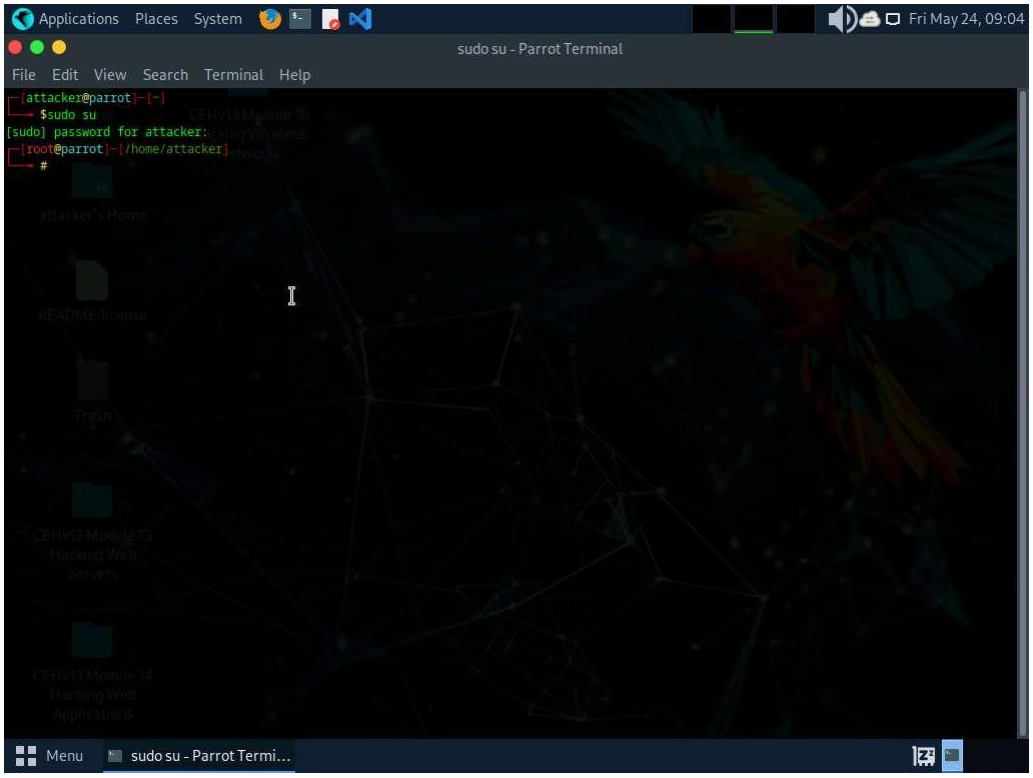
1. In the **Parrot Security** machine, click the **MATE Terminal** icon in the menu to launch the terminal.

2. A **Parrot Terminal** window appears. In the terminal window, type **sudo su** and press **Enter** to run the programs as a root user.

3. In the **[sudo] password for attacker** field, type **toor** as a password and press **Enter**.

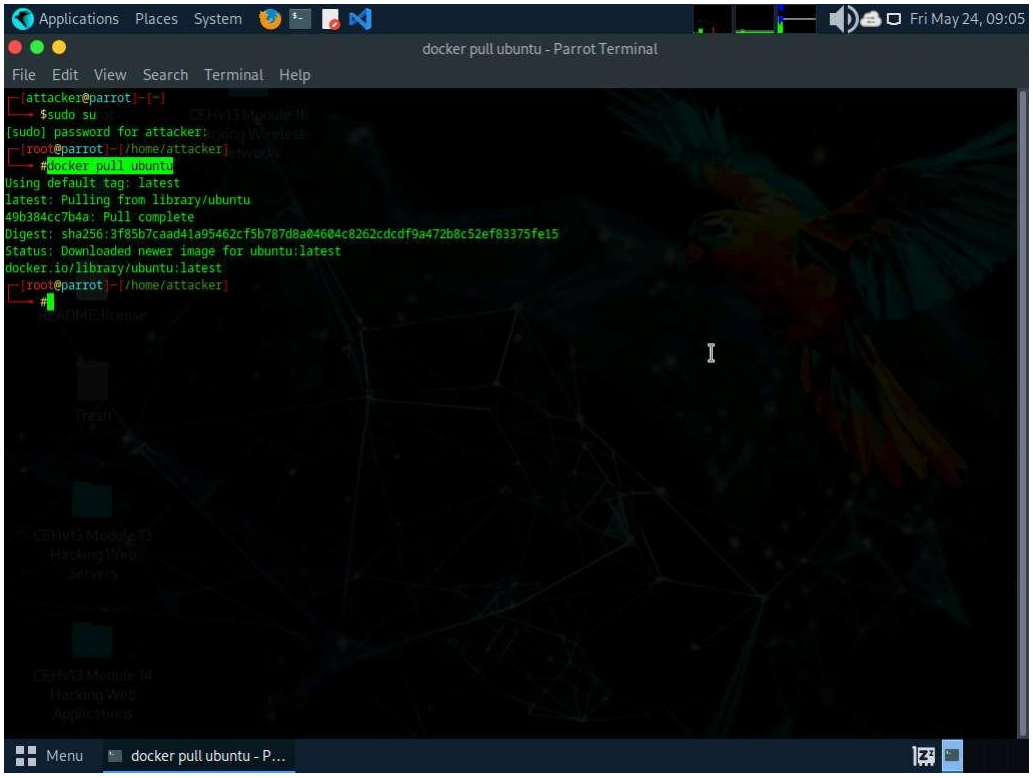
   The password that you type will not be visible.

   Minimise the terminal for better view of output

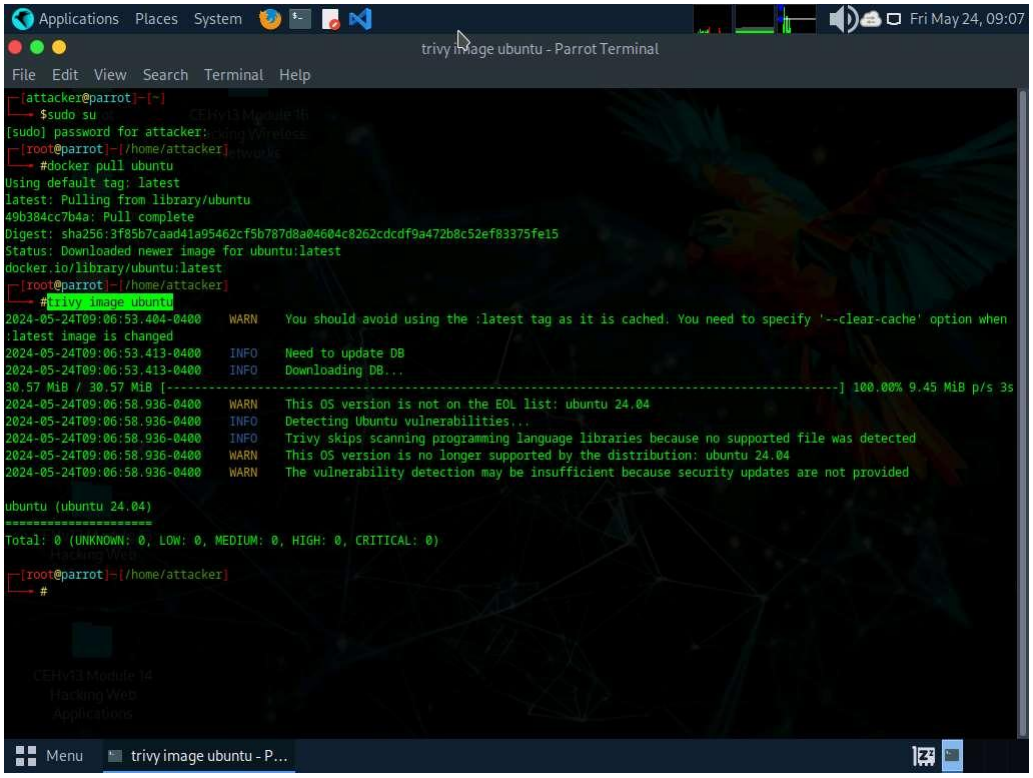


4. In this lab we will be scanning two docker images, first the secure one and second the vulnerable one.

5. Execute command **docker pull ubuntu** to install the first docker image.

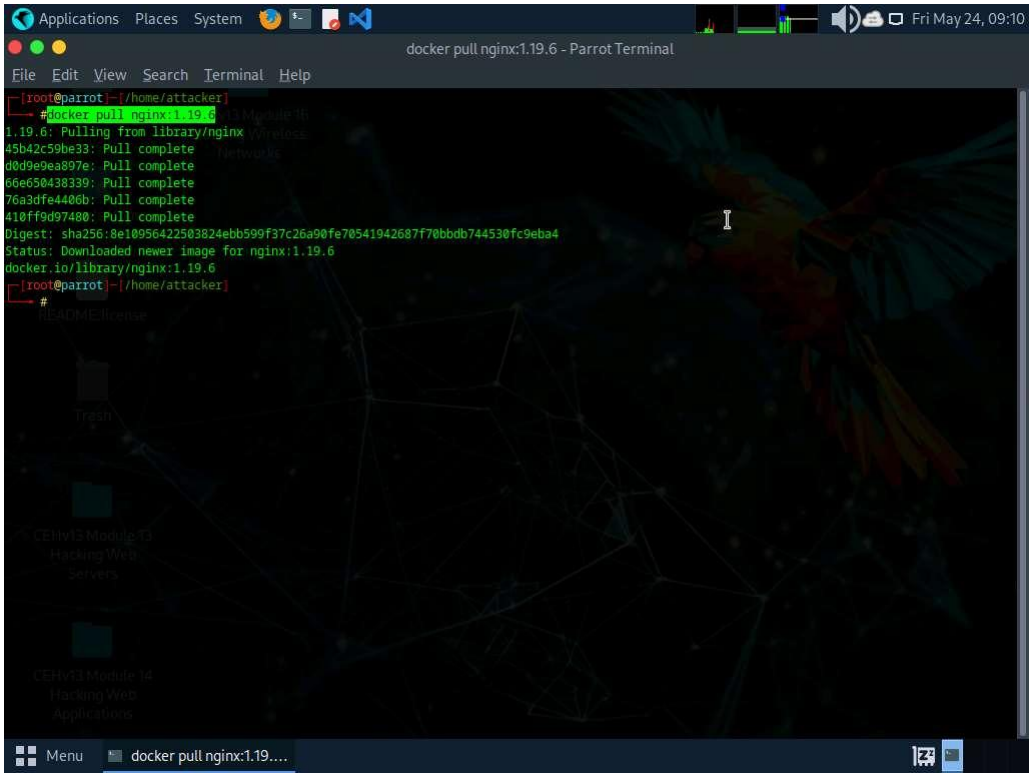


6. Once the image is pulled we will be performing vulnerability assessment. Execute command **trivy image ubuntu**.

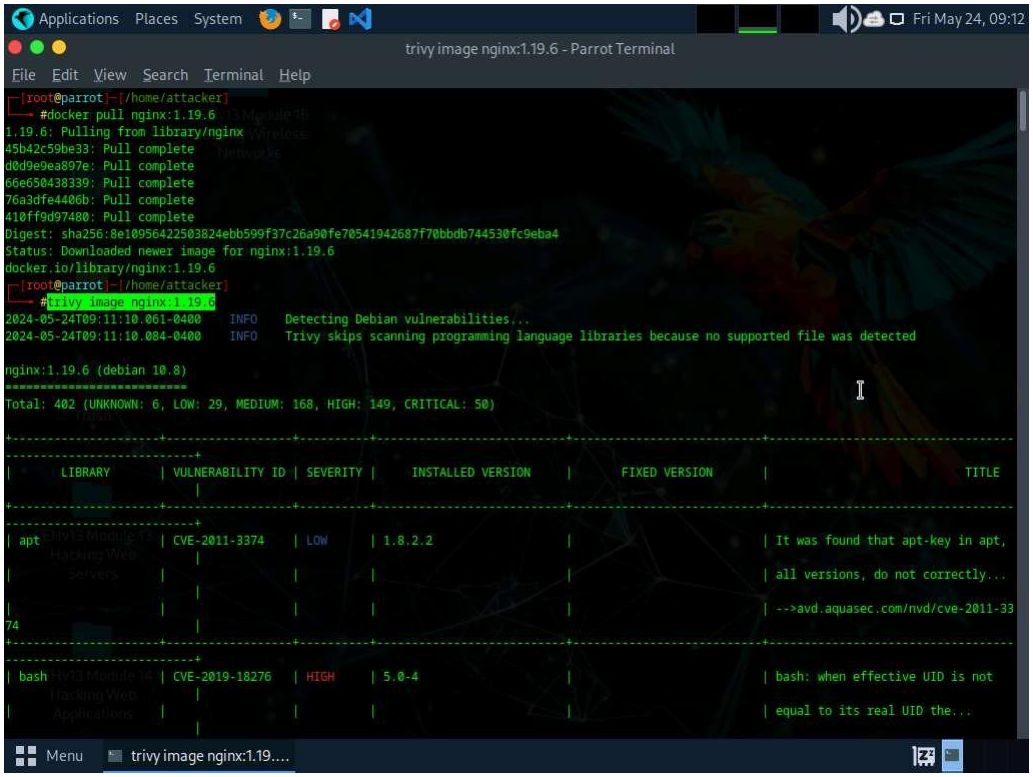


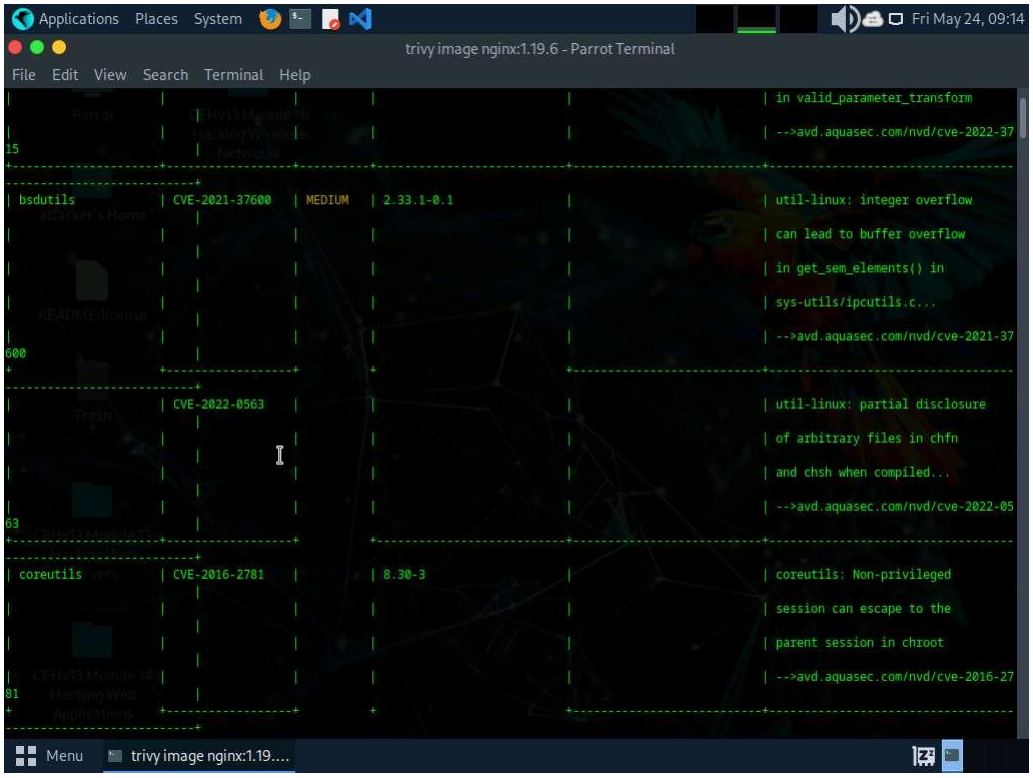
7. In the above screenshot, we can observe that we have total **0** vulnerability and it's completely secure.

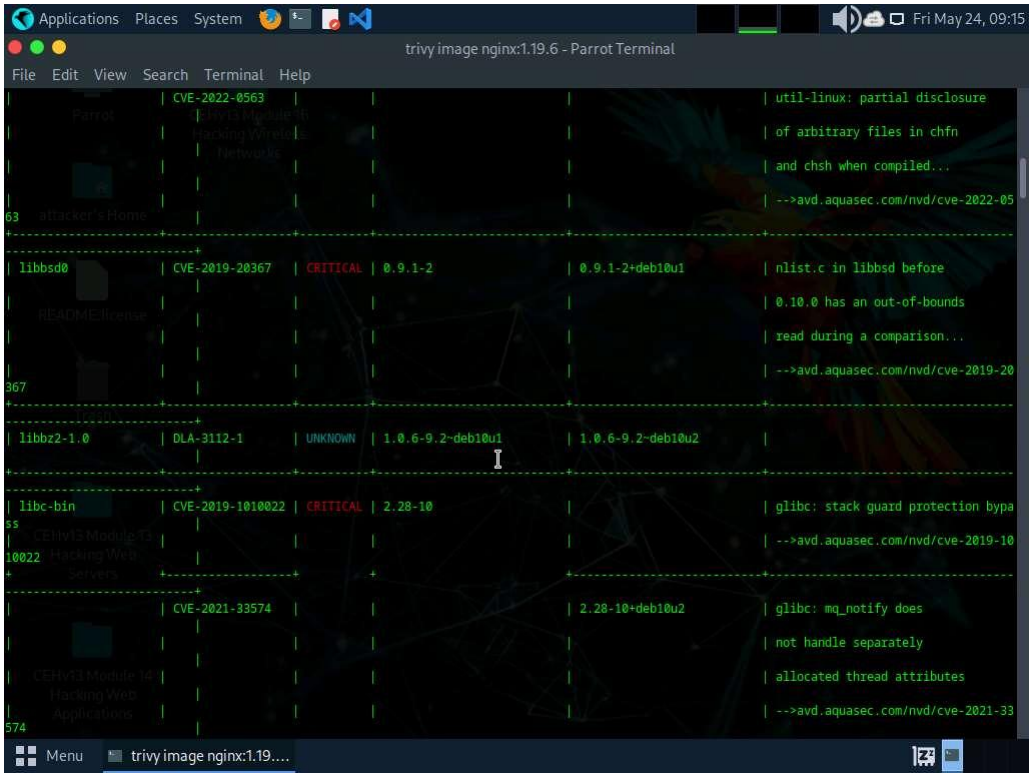
8. Now, we will analyse the vulnerbale image. execute command **docker pull nginx:1.19.6** to pull the vulnerable image.



9. Execute command **trivy image nginx:1.19.6** to scan the image.







10. In the above screenshot we can see that we have total **401** vulnerabilities which is categorized as well along with **CVEs** mentioned.

11. This concludes the demonstration of vulnerability assessment on docker images using Trivy

12. Close all open windows and document all acquired information.

Question 19.4.1.1

In Parrot machine install ubuntu and nginx:1.19.6 images and scan with trivy security scanner. Enter the severity level that can be observed for bsdutils vulnerability of nginx:1.19.6 docker image.