Cloud Computing

Module 19

Cloud Computing

Cloud computing delivers various types of services and applications over the Internet. These services enable users to use software and hardware managed by third parties at remote locations. Some well-known cloud service providers are Google, Amazon, and Microsoft.

Lab Scenario

ICON KEY

Valuable information

Test your knowledge

Web exercise

Workbook review

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.

Tools demonstrated in this lab are available in E:\CEH-Tools\CEHv11 Module 19 Cloud Computing

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.

Lab Objectives

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

Lab Environment

To carry out this lab, you need:

- Parrot Security virtual machine
- Web browsers with an Internet connection
- Administrator privileges to run the tools

Lab Duration

Time: 30 Minutes

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:

Lab No.	Lab Exercise Name	Core*	Self-study**	iLabs ***
1	Perform S3 Bucket Enumeration using Various S3 Bucket Enumeration Tools	√		√
	1.1 Enumerate S3 Buckets using lazys3	√		√
	1.2 Enumerate S3 Buckets using S3Scanner	√		√
2	Exploit S3 Buckets	√		√
	2.1 Exploit Open S3 Buckets using AWS CLI	√		√
3	Perform Privilege Escalation to Gain Higher Privileges	√		√
	3.1 Escalate IAM User Privileges by Exploiting Misconfigured User Policy	√		√

Remark

EC-Council has prepared a considered amount of lab exercises for student to practice during the 5-day class and at their free time to enhance their knowledge and skill.

***Core** - Lab exercise(s) marked under Core are recommended by EC-Council to be practised during the 5-day class.

****Self-study** - Lab exercise(s) marked under self-study is for students to practise at their free time. Steps to access the additional lab exercises can be found in the first page of CEHv11 volume 1 book.

*****iLabs** - Lab exercise(s) marked under iLabs are available in our iLabs solution. iLabs is a cloud-based virtual lab environment preconfigured with vulnerabilities, exploits, tools and scripts, and can be accessed from anywhere with an Internet connection. If you are interested to learn more about our iLabs solution, please contact your training center or visit https://ilabs.eccouncil.org.

Lab Analysis

Analyze and document the results related to the lab exercise. Give your opinion on your target's security posture and exposure.

PLEASE TALK TO YOUR INSTRUCTOR IF YOU HAVE QUESTIONS RELATED TO THIS LAB.

Lab 1

Perform S3 Bucket Enumeration using Various S3 Bucket Enumeration Tools

Ethical hackers and penetration testers are aided in enumeration by various tools that make information gathering an easy task.

ICON KEY

Valuable Information

Test Your Knowledge

Web Exercise

Workbook Review

Lab Scenario

As an ethical hacker, you must try to obtain as much information as possible about the target cloud environment using various enumeration tools. This lab will demonstrate various S3 bucket enumeration tools that can help you in extracting the list of publicly available S3 buckets.

Lab Objectives

- Enumerate S3 buckets using lazys3
- Enumerate S3 buckets using S3Scanner

Lab Environment

Tools demonstrated in this lab are available in E:\CEH-Tools\CEHv11 Module 19 Cloud Computing

To carry out lab, you need:

- Parrot Security virtual machine
- Web browsers with an Internet connection
- Administrator privileges to run the tools

Lab Duration

Time: 10 Minutes

Overview of Enumeration Tools

Enumeration tools are used to collect detailed information about target systems to exploit them. Information collected by S3 enumeration tools consists of a list

of misconfigured S3 buckets that are available publicly. Attackers can exploit these buckets to gain unauthorized access to them. Moreover, they can modify, delete, and exfiltrate the bucket content.

Lab Tasks

TASK 1

Enumerate S3 Buckets using lazys3

lazys3 is a Ruby script tool that is used to brute-force AWS S3 buckets using different permutations. This tool obtains the publicly accessible S3 buckets and also allows you to search the S3 buckets of a specific company by entering the company name.

1. Launch the **Parrot Security** virtual machine. In the login page, the **attacker** username will be selected by default. Enter password as **toor** in the **Password** field and press **Enter** to log in to the machine.

Figure 1.1.1: Parrot Security Login

Note:

- If a **Parrot Updater** pop-up appears at the top-right corner of **Desktop**, ignore and close it.
- If a **Question** pop-up window appears asking you to update the machine, click **No** to close the window.

TASK 1.1

Clone lazys3 Repository

2. Click the **MATE Terminal** icon in the menu to launch the terminal.
3. A **Parrot Terminal** window appears. In the terminal window, type **sudo su** and press **Enter** to run the programs as a root user.
4. In the **[sudo] password for attacker** field, type **toor** as a password and press **Enter**.

Note: The password that you type will not be visible.

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

Figure 1.1.2: Running the programs as a root user

6. In the terminal window, type **git clone https://github.com/nahamsec/lazys3** and press **Enter** to install and clone the lazys3 tool.

```
Parrot Terminal
File  Edit  View  Search  Terminal  Help
┌─[root@parrot]─[~]
└──╼ #git clone https://github.com/nahamsec/lazys3
Cloning into 'lazys3'...
remote: Enumerating objects: 22, done.
remote: Total 22 (delta 0), reused 0 (delta 0), pack-reused 22
Unpacking objects: 100% (22/22), 4.92 KiB | 179.00 KiB/s, done.
```

Figure 1.1.3: Clone lazys3 repository

Note: You can also access the tool repository from the **CEH-Tools** folder available in **Windows 10** virtual machine, in case, the GitHub link does not exist, or you are unable to clone the tool repository. Follow the steps below in order to access **CEH-Tools** folder from the **Parrot Security** virtual machine:

- Open a windows explorer and press **Ctrl+L**. The **Location** field appears; type **smb://10.10.10.10** and press **Enter** to access **Windows 10** shared folders.
- The security pop-up appears; enter the **Windows 10** virtual machine credentials (**Username**: **Admin** and **Password**: **Pa$$w0rd**) and click **Connect**.
- The **Windows shares on 10.10.10.10** window appears; navigate to the location **CEH-Tools/CEHv11 Module 19 Cloud Computing/GitHub Tools/** and copy the **lazys3** folder.
- Paste the copied **lazys3** folder on the location **/home/attacker/**.
- In the terminal window, type **mv /home/attacker/lazys3 /root/**.

7. After the successful cloning of the lazys3 tool, in the terminal window, type **cd lazys3** and press **Enter** to navigate to the cloned repository.

Note: By default, the tool is cloned to the root directory.

8. In the lazys3 folder, type **ls** and press **Enter** to list the folder content.
9. The folder content is displayed; here, we will run the **lazys3.rb** script to find the public S3 buckets.

```
Parrot Terminal
File  Edit  View  Search  Terminal  Help
┌─[root@parrot]─[~]
└──╼ #cd lazys3
┌─[root@parrot]─[~/lazys3]
└──╼ #ls
common_bucket_prefixes.txt  lazys3.rb  README.md
┌─[root@parrot]─[~/lazys3]
└──╼ #
```

Figure 1.1.4: Navigating to the cloned folder and listing the folder content

TASK 1.2

Run lazys3 Script

10. Now, type **ruby lazys3.rb** and press **Enter**.
11. A list of public S3 buckets is displayed, as shown in the screenshot.

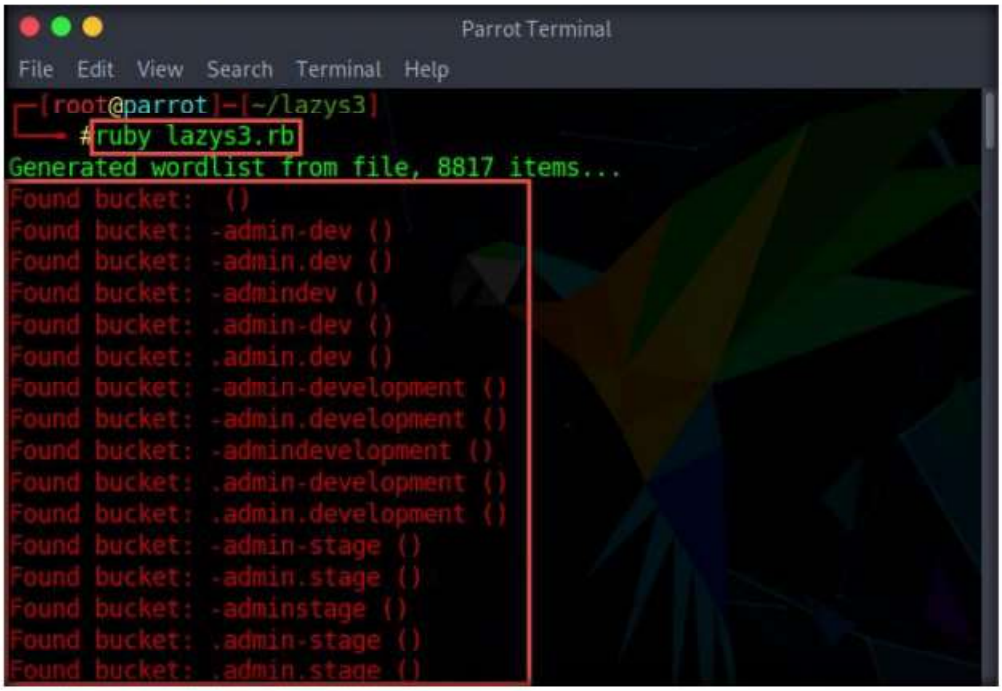

Figure 1.1.5: List of open S3 buckets

12. Press **Ctrl+Z** to stop the script.

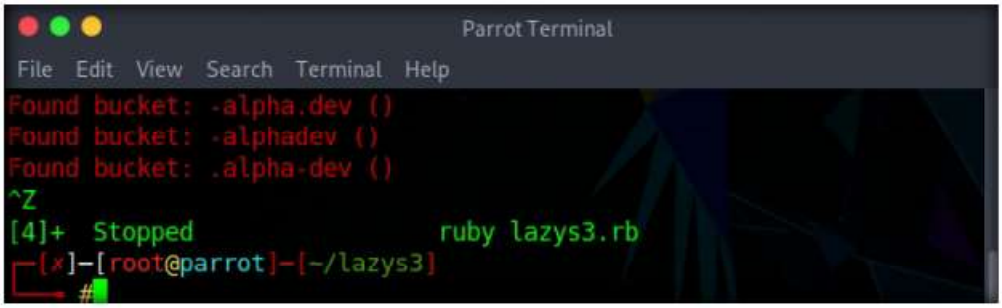

Figure 1.1.6: Stop the script

TASK 1.3

Run lazys3 Script for a Specific Company

13. You can search the S3 buckets of specific company. To do so, type **ruby lazys3.rb <Company>** and press **Enter**.

Note: Here, the target company name is HackerOne; you can enter the company name of your choice.

14. The result appears, showing the obtained list of S3 buckets of the specified company.

Note: It will take some time to obtain a complete list of the available S3 buckets.

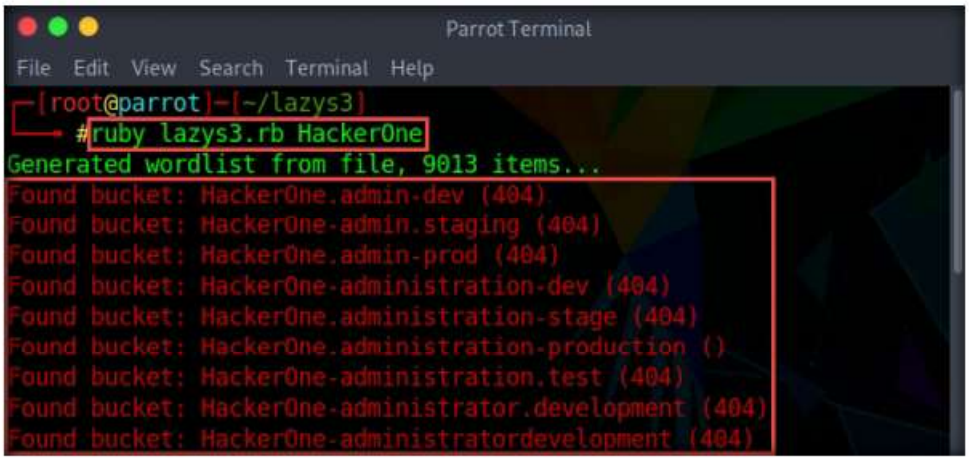

Figure 1.1.7: List of open S3 buckets of a specific company

15. Press **Ctrl+Z** to stop running the script.
16. This concludes the demonstration of enumerating public S3 buckets.
17. Close all open windows and document all the acquired information.

TASK 2

Enumerate S3 Buckets using S3Scanner

S3Scanner is a tool that finds the open S3 buckets and dumps their contents. It takes a list of bucket names to check as its input. The S3 buckets that are found are output to a file. The tool also dumps or lists the contents of "open" buckets locally.

Here, we will use the S3Scanner tool to enumerate open S3 buckets.

1. 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**.

Note: The password that you type will not be visible.

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

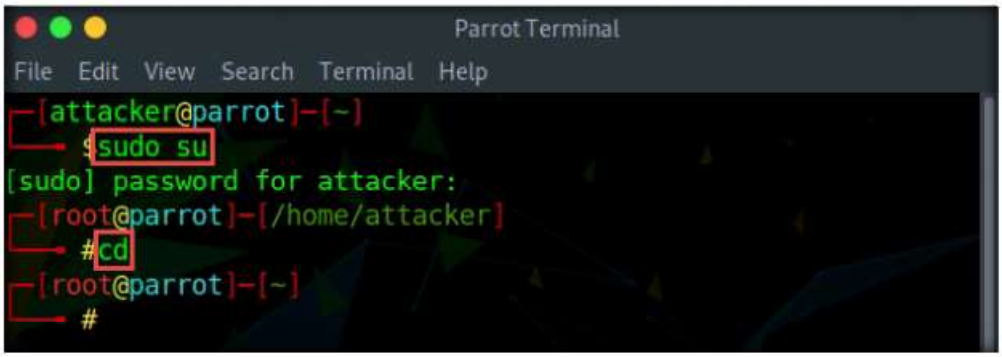

Figure 1.2.1: Running the programs as a root user

TASK 2.1

Clone and Install S3Scanner Tool

5. In the terminal window, type **git clone https://github.com/sa7mon/S3Scanner** and press **Enter** to install and clone the S3Scanner tool.

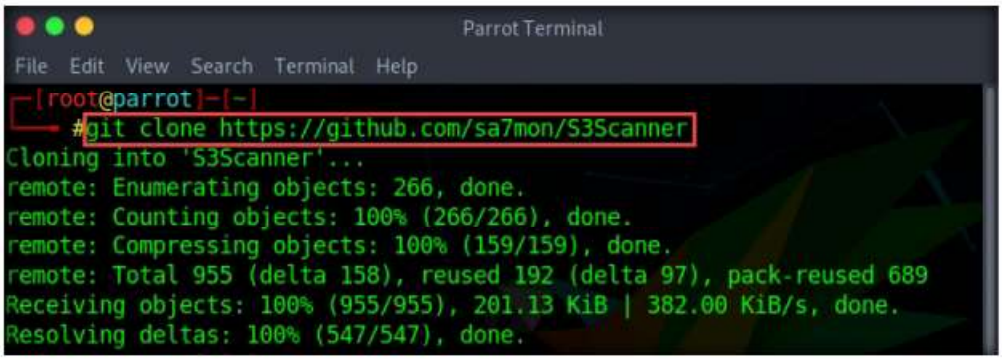

Figure 1.2.2: Cloning S3Scanner repository

Note: You can also access the tool repository from the **CEH-Tools** folder available in **Windows 10** virtual machine, in case, the GitHub link does not exist, or you are unable to clone the tool repository. Follow the steps below in order to access **CEH-Tools** folder from the **Parrot Security** virtual machine:

- Open a windows explorer and press **Ctrl+L**. The **Location** field appears; type **smb://10.10.10.10** and press **Enter** to access **Windows 10** shared folders.
- The security pop-up appears; enter the **Windows 10** virtual machine credentials (**Username**: **Admin** and **Password**: **Pa$$w0rd**) and click **Connect**.
- The **Windows shares on 10.10.10.10** window appears; navigate to the location **CEH-Tools/CEHv11 Module 19 Cloud Computing/GitHub Tools/** and copy the **S3Scanner** folder.
- Paste the copied **S3Scanner** folder on the location **/home/attacker/**.
- In the terminal window, type **mv /home/attacker/S3Scanner /root/**.

6. After the successful cloning of the S3Scanner tool, in the terminal window, type **cd S3Scanner** and press **Enter** to navigate to the cloned repository.

Note: By default, the tool is cloned to the root directory.

7. In the S3Scanner folder, type **pip3 install -r requirements.txt** and press **Enter** to install the required dependencies.

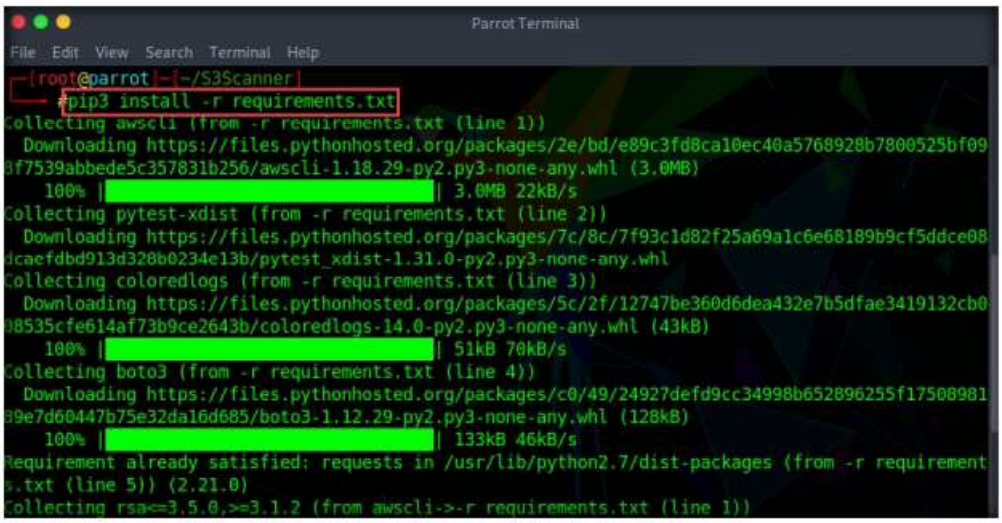

Figure 1.2.3: Installing dependencies

TASK 2.2

Scan for Open S3 Buckets

8. After the successful installation of the dependencies, in the terminal window, type **python ./s3scanner.py sites.txt** and press **Enter** to run the tool.

Note: Here, **sites.txt** is a text file containing the target website URL that is scanned for open S3 buckets. You can edit the **sites.txt** file to enter the target website URL of your choice.

9. The result appears, displaying a list of public S3 buckets, as shown in the screenshot.

Note: You might encounter the following error: "AWS credentials not configured." Ignore the error, as we will install and configure the AWS CLI in the next lab.

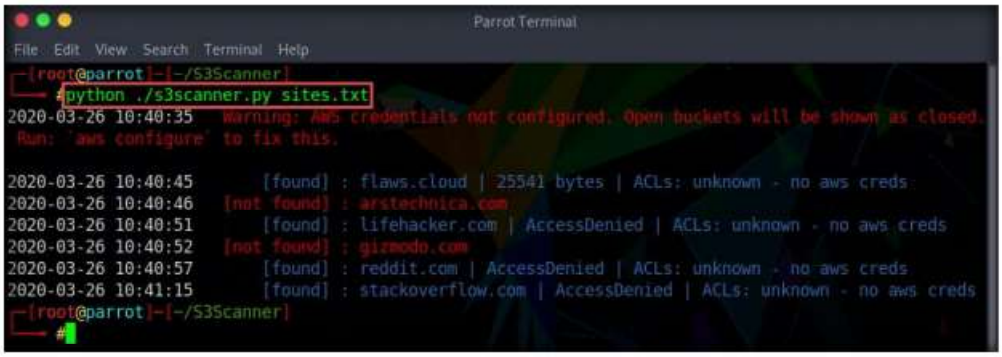

Figure 1.2.4: Running S3Scanner script for a list of websites

You can also use other S3 bucket enumeration tools such as **S3Inspector** (https://github.com), **s3-buckets-bruteforcer** (https://github.com), **Mass3** (https://github.com), **Bucket Finder** (https://digi.ninja), and **s3recon** (https://github.com) to perform S3 bucket enumeration for a target website or company.

10. Apart from the aforementioned command, you can use the S3Scanner tool to perform the following functions:
    - Dump all open buckets and log both open and closed buckets in found.txt:

      python ./s3scanner.py --include-closed --out-file found.txt --dump names.txt
    - Just log open buckets in the default output file (buckets.txt):

      python ./s3scanner.py names.txt
    - Save the file listings of all open buckets to a file:

      python ./s3scanner.py --list names.txt
11. This concludes the demonstration of enumerating S3 buckets using the S3Scanner tool.
12. Close all open windows and document all the acquired information.
13. Turn off the **Parrot Security** virtual machine.

Lab Analysis

Analyze and document all the results obtained in the lab exercise.

PLEASE TALK TO YOUR INSTRUCTOR IF YOU HAVE QUESTIONS ABOUT THIS LAB.

Internet Connection Required	
☑ Yes	☐ No
Platform Supported	
☑ Classroom	☑ iLabs

Lab

2

Exploit S3 Buckets

Simple Storage Service (S3) is a scalable cloud storage service offered by Amazon Web Services (AWS) whereby files, folders, and objects are stored via web APIs.

ICON KEY

Valuable Information

Test Your Knowledge

Web Exercise

Workbook Review

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

Lab Environment

Tools demonstrated in this lab are available in E:\CEH-Tools\CEHv11 Module 19 Cloud Computing

To carry out lab, you need:

- Parrot Security virtual machine
- Web browsers with an Internet connection
- Administrator privileges to run the tools

Lab Duration

Time: 10 Minutes

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

Lab Tasks

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.

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

1. Launch the **Parrot Security** virtual machine. In the login page, the **attacker** username will be selected by default. Enter password as **toor** in the **Password** field and press **Enter** to log in to the machine.

Figure 2.1.1: Parrot Security Login

Note:

- If a **Parrot Updater** pop-up appears at the top-right corner of **Desktop**, ignore and close it.
- If a **Question** pop-up window appears asking you to update the machine, click **No** to close the window.

2. Click the **MATE Terminal** icon in the menu to launch the terminal.
3. A **Parrot Terminal** window appears. In the terminal window, type **sudo su** and press **Enter** to run the programs as a root user.

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

Note: The password that you type will not be visible.

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

TASK 1.1

Install AWS CLI

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

Figure 2.1.2: Installing AWS CLI

7. Once the installation is completed, type **aws --help** and press **Enter** to check whether AWS CLI is properly installed.

Note: Ignore the errors (if you find any).

Figure 2.1.3: AWS CLI installed successfully

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

Figure 2.1.4: AWS CLI configuration

9. It will ask for the following details:
   - AWS Access Key ID
   - AWS Secret Access Key
   - Default region name
   - Default output format
10. To provide these details, you need to login to your AWS account.
11. 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**.

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

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

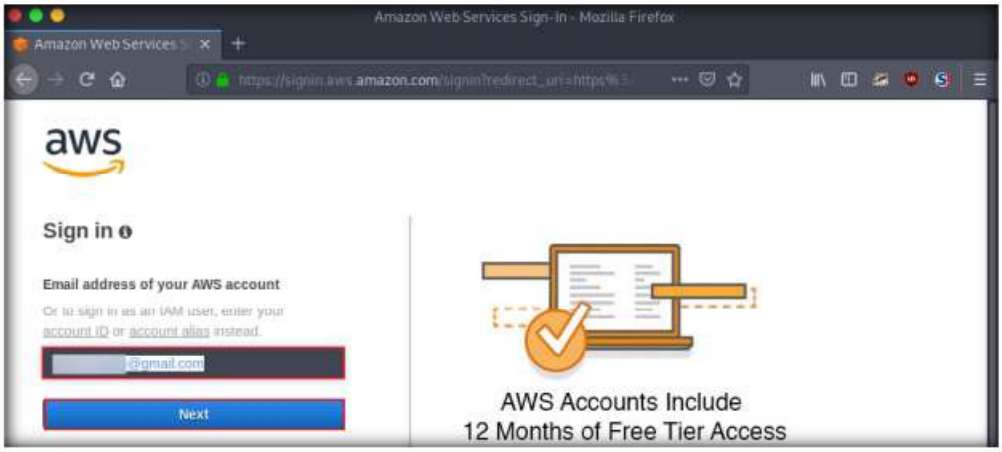

Figure 2.1.5: AWS Sign In page

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

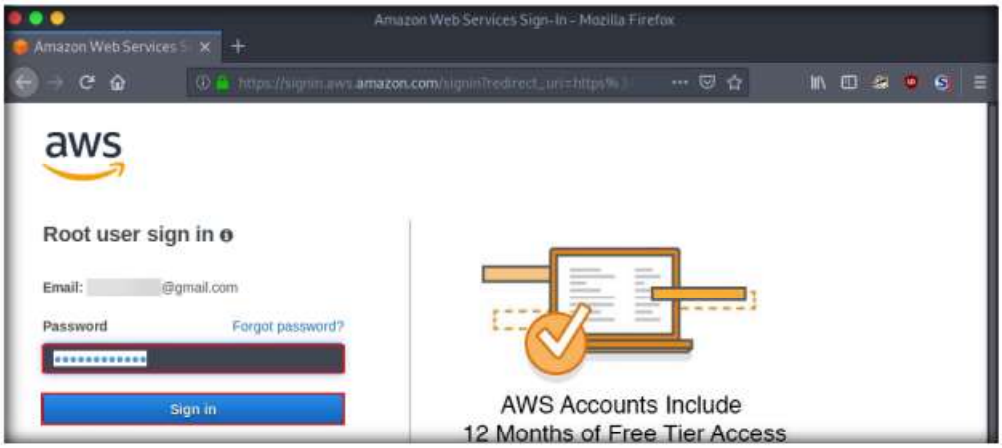

Figure 2.1.6: AWS Sign In page

14. Click the AWS account drop-down menu and click **My Security Credentials**, as shown in the screenshot.

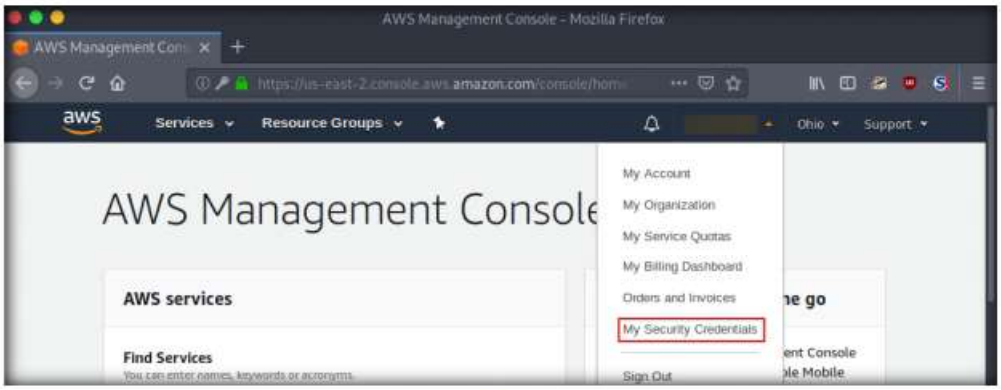

Figure 2.1.7: AWS Management Console

15. A pop-up appears; click the **X** icon to close the pop-up.

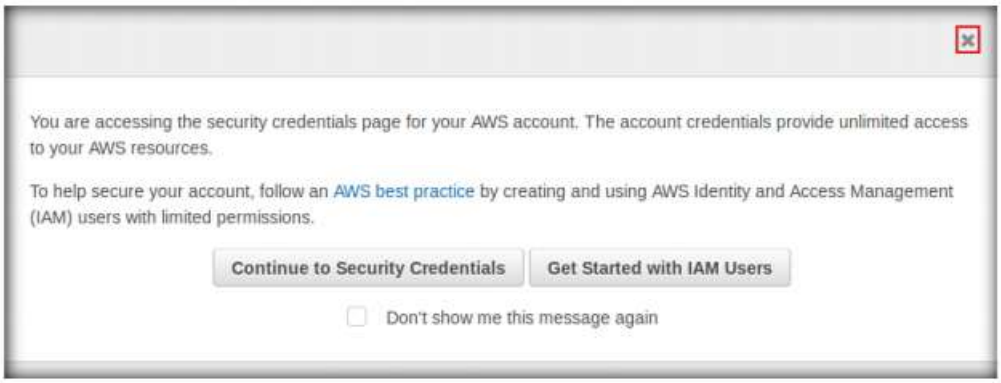

Figure 2.1.8: Security Credentials message

16. Click **Access keys (access key ID and secret access key)** in the **Your Security Credentials** section.

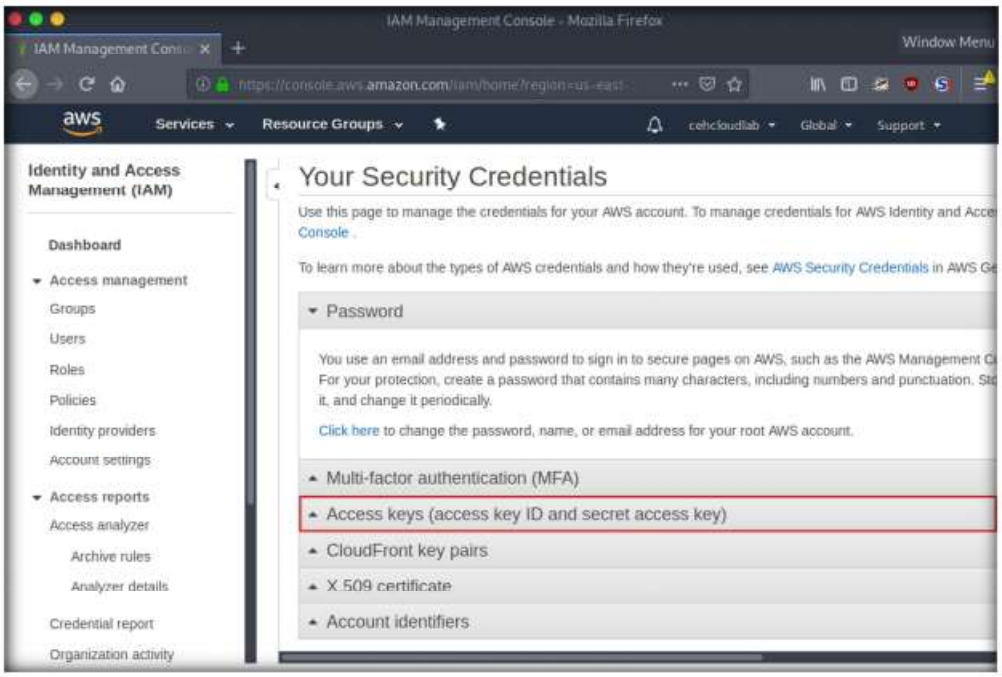

Figure 2.1.9: Security Credentials – Access Keys

17. Click the **Create New Access Key** button.

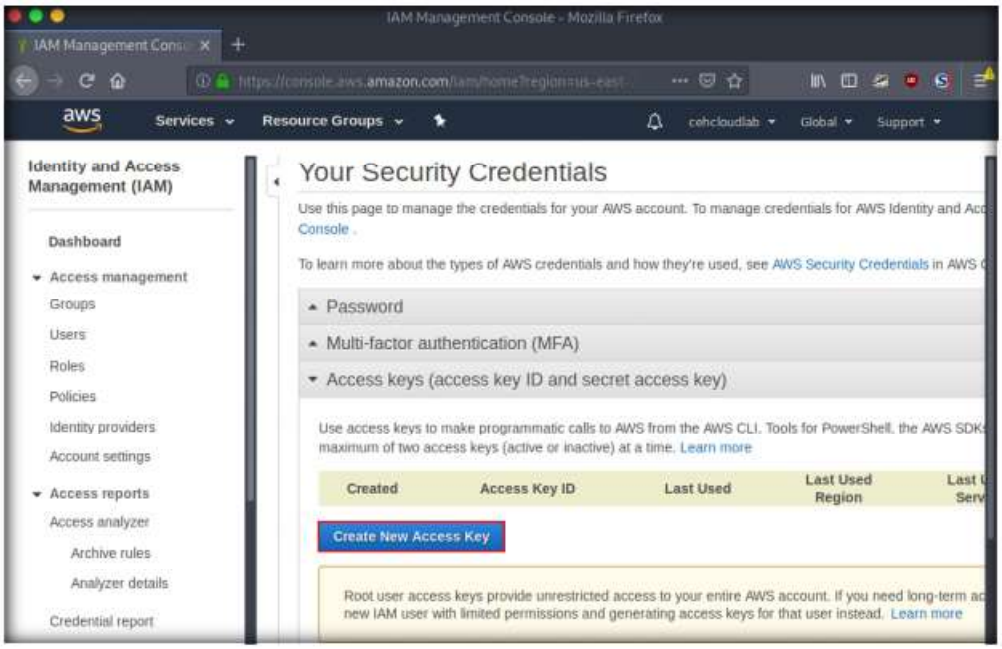

Figure 2.1.10: Create new access key

18. A **Create Access Key** pop-up appears, stating that your access key has been successfully created. Click the **Show Access Key** link to view the access key.

Figure 2.1.11: Access key created

19. Copy the **Access Key ID** displayed by pressing **Ctrl+C** on your keyboard and switch to the **Terminal** window.

Figure 2.1.12: Access keys

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

Figure 2.1.13: AWS configuring access key ID

21. In the **Create Access Key** pop-up, select the **Secret Access Key** displayed, copy it by pressing **Ctrl+C** on your keyboard, and minimize the browser window. Switch to the **Terminal** window.

22. 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.

Figure 2.1.14: AWS configuring secret access key

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

Figure 2.1.15: AWS configuring default region name

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

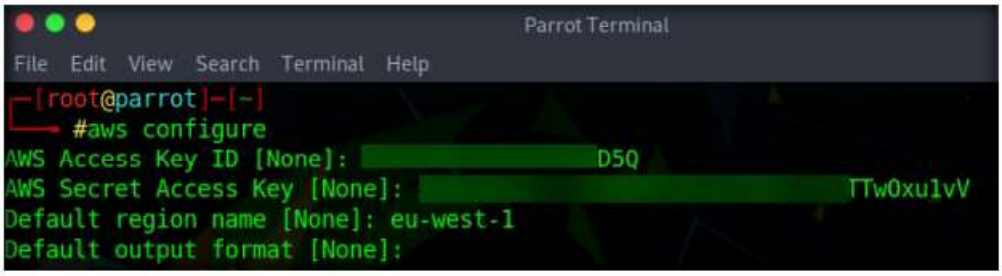

Figure 2.1.16: AWS configuring default output format

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

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

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

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

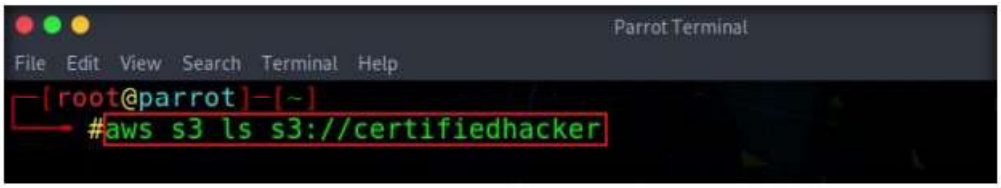

Figure 2.1.17: Listing directories in open S3 bucket

27. This will show you the list of directories in the certifiedhacker S3 bucket, as shown in the screenshot.

Figure 2.1.18: Directories list in open S3 bucket

28. Now, maximize the browser window, type **certifiedhacker.s3.amazonaws.com** in the address bar, and press **Enter**.
29. This will show you the complete list of directories and files available in this bucket.

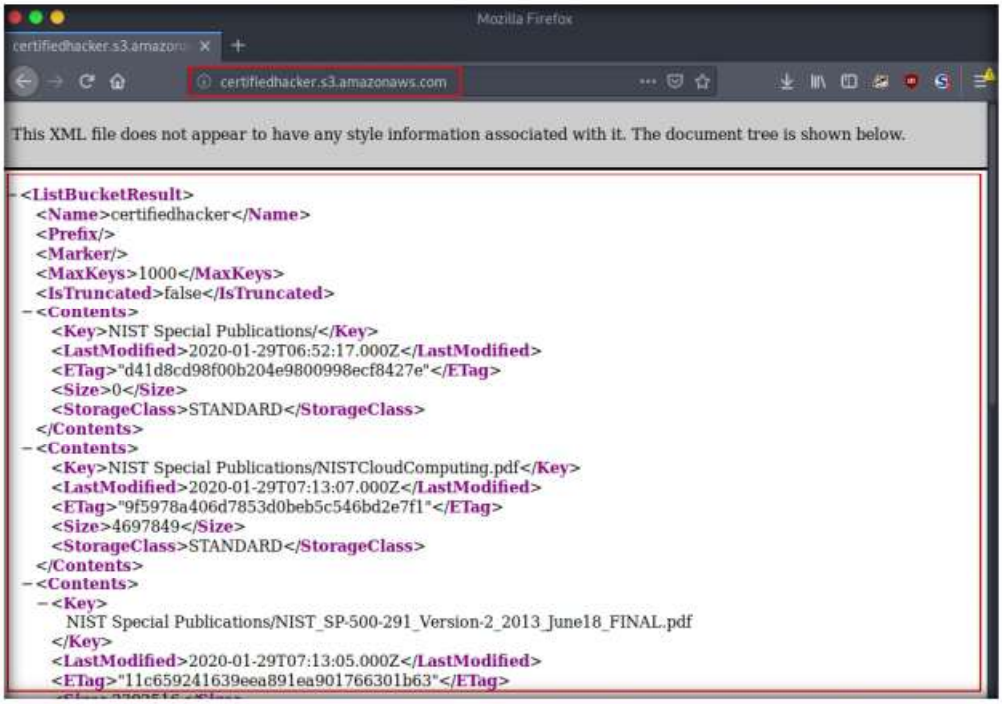

Figure 2.1.19: S3 bucket directories and files list

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

Figure 2.1.20: Creating Hack.txt file

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

34. You have successfully moved the **Hack.txt** file to the **certifiedhacker** bucket.

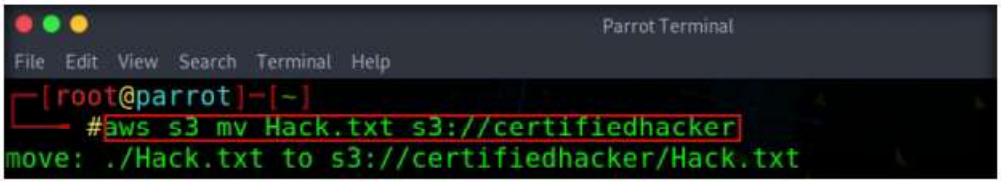

Figure 2.1.21: Moving Hack.txt file to S3 bucket

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

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

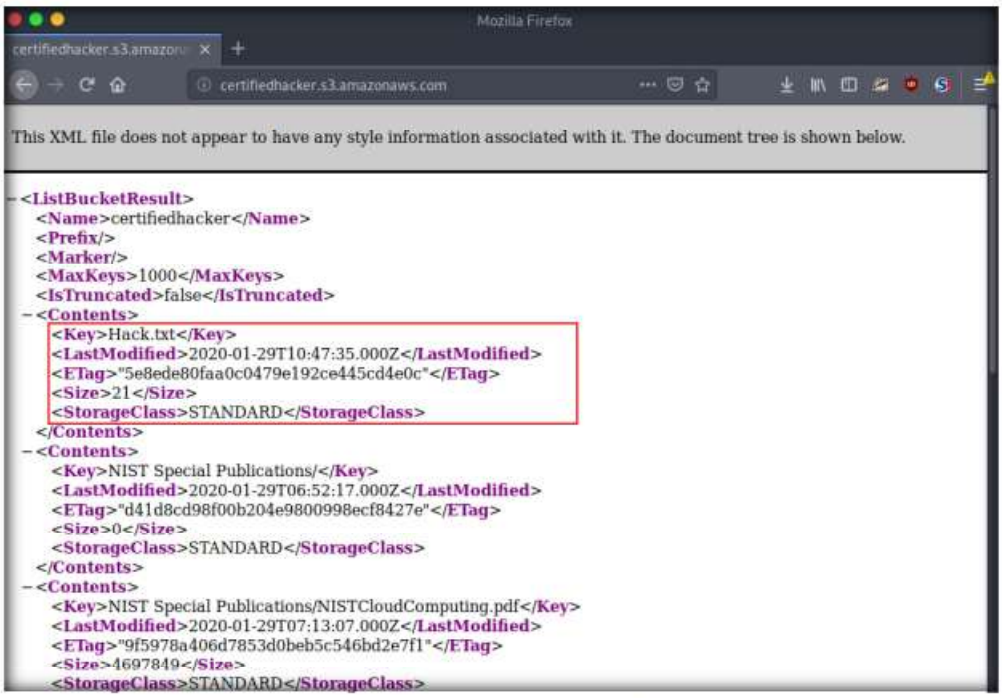

Figure 2.1.22: Hack.txt file moved to S3 bucket

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

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

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

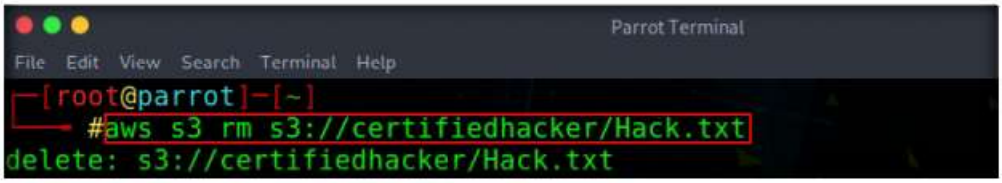

Figure 2.1.23: Deleting Hack.txt file from S3 bucket

40. To verify whether the file is deleted, switch to the browser window and reload the page.
41. The **Hack.txt** file is deleted from the **certifiedhacker** bucket.

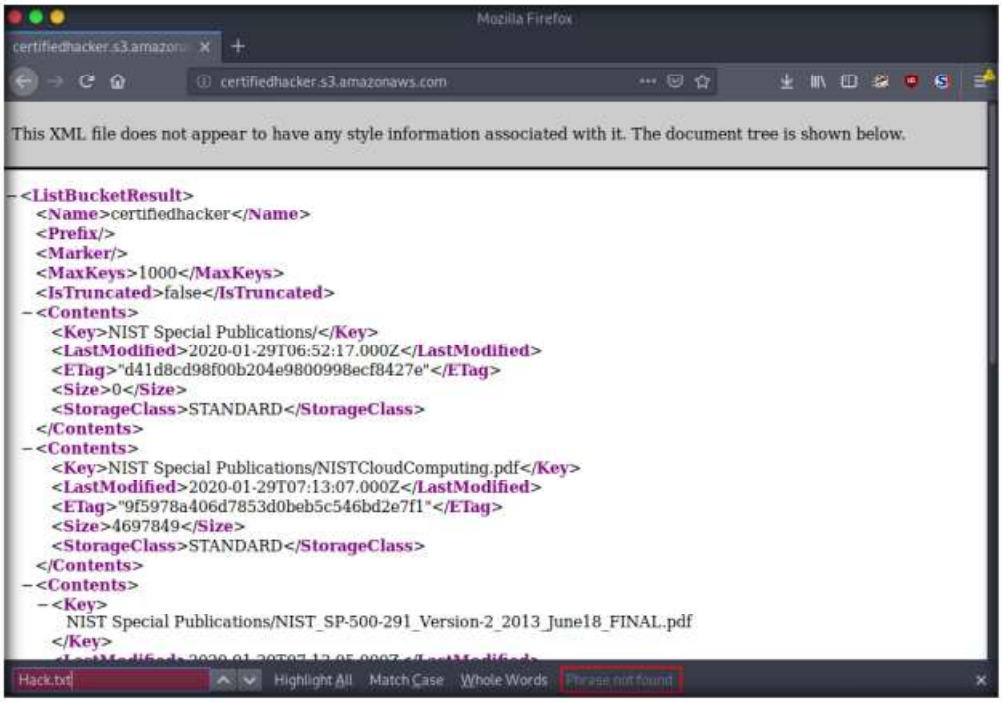

Figure 2.1.24: Hack.txt file deleted from S3 bucket.

42. Thus, you can add or delete files from open S3 buckets.
43. This concludes the demonstration of exploiting public S3 buckets.
44. Close all open windows and document all the acquired information.
45. Turn off the **Parrot Security** virtual machine.

Lab Analysis

Analyze and document all the results obtained in the lab exercise.

PLEASE TALK TO YOUR INSTRUCTOR IF YOU HAVE QUESTIONS ABOUT THIS LAB.

Internet Connection Required	
☑ Yes	☐ No
Platform Supported	
☑ Classroom	☑ iLabs

Lab

3

Perform Privilege Escalation to Gain Higher Privileges

Privilege escalation is the process of gaining higher-level or administrator-level privileges for the target system using a non-administrator user account.

ICON KEY

Valuable Information

Test Your Knowledge

Web Exercise

Workbook Review

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.

Tools demonstrated in this lab are available in E:\CEH-Tools\CEHv11 Module 19 Cloud Computing

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

Lab Environment

To carry out lab, you need:

- Parrot Security virtual machine
- Web browsers with an Internet connection
- Administrator privileges to run the tools

Lab Duration

Time: 10 Minutes

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

Lab Tasks

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.

Note: In this task, for demonstration purposes, we have created an IAM user account with permissions including iam:CreatePolicy, iam:AttachUserPolicy, iam:ListUserPolicies, sts:AssumeRole, and iam:ListRoles. This policy can be exploited by attackers to gain administrator-level privileges.

1. Launch the **Parrot Security** virtual machine. In the login page, the **attacker** username will be selected by default. Enter password as **toor** in the **Password** field and press **Enter** to log in to the machine.
2. Click the **MATE Terminal** icon in the menu to launch the terminal.
3. A **Parrot Terminal** window appears. In the terminal window, type **sudo su** and press **Enter** to run the programs as a root user.
4. In the **[sudo] password for attacker** field, type **toor** as a password and press **Enter**.

Note: The password that you type will not be visible.

5. Now, type **cd** and press **Enter** to jump to the root directory.
6. In the terminal window, type **aws configure** and press **Enter**.

TASK 1.1

Configure AWS

7. Enter the details of the target IAM user's access key in the **AWS Access Key ID** field and press **Enter**. Similarly, in the **AWS Secret Access Key** filed, enter the target IAM user's secret access key and press **Enter**.

Note: The **AWS Access Key ID** and **AWS Secret Access Key** of the target user's account can be obtained using various social engineering techniques, as discussed in **Module 09 Social Engineering**.

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.

8. In the **Default region name** field, type **us-east-2** and press **Enter**. In the **Default output format** field, type **json** and press **Enter**.

Figure 3.1.1: AWS CLI Configuration

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

TASK 1.2

Create a User Policy

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

Note: This command will create a file named **user-policy** in the **root** directory.

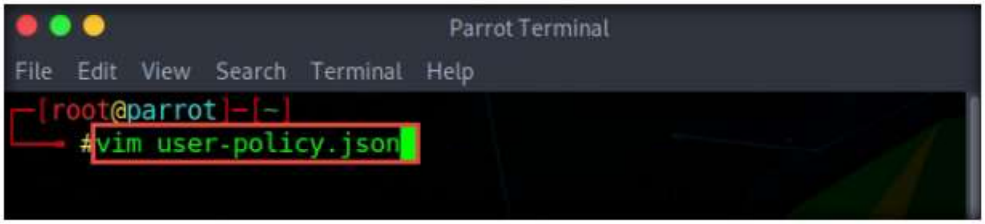

Figure 3.1.2: Open a text editor

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

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": "*",
      "Resource": "*"
    }
  ]
}
```

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

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

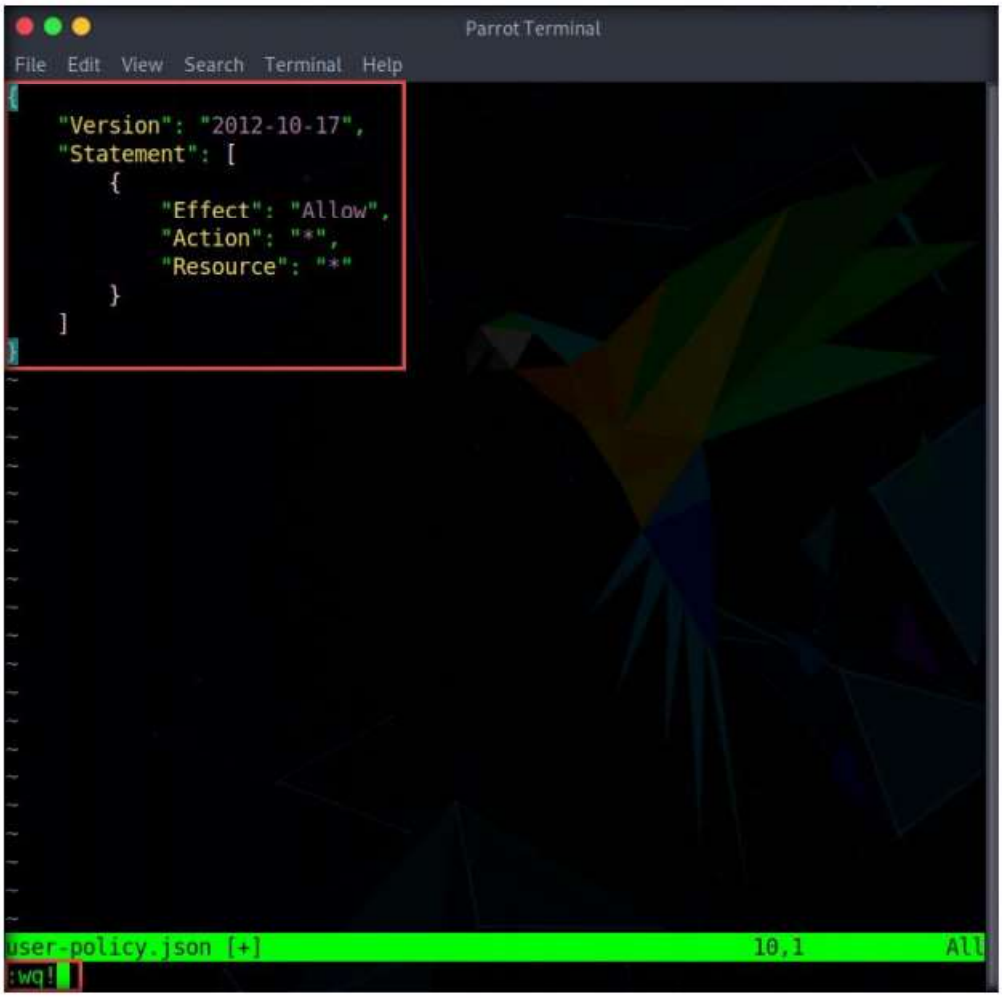

Figure 3.1.3: Writing a user policy

13. 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**.
14. The created user policy is displayed, showing various details such as **PolicyName**, **PolicyId**, and **Arn**.

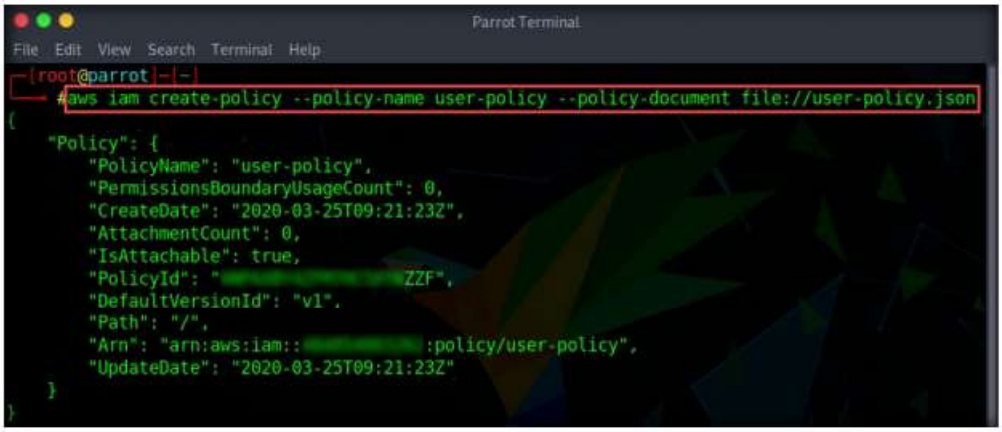

Figure 3.1.4: Creating a user policy

TASK 1.3

Attach User Policy to the Target User

15. 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**.
16. The above command will attach the policy (**user-policy**) to the target IAM user account (here, **test**).

Figure 3.1.5: Attaching a user policy

TASK 1.4

List the Attached User Policy

17. 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**).
18. The result appears, displaying the attached policy name (**user-policy**), as shown in the screenshot.

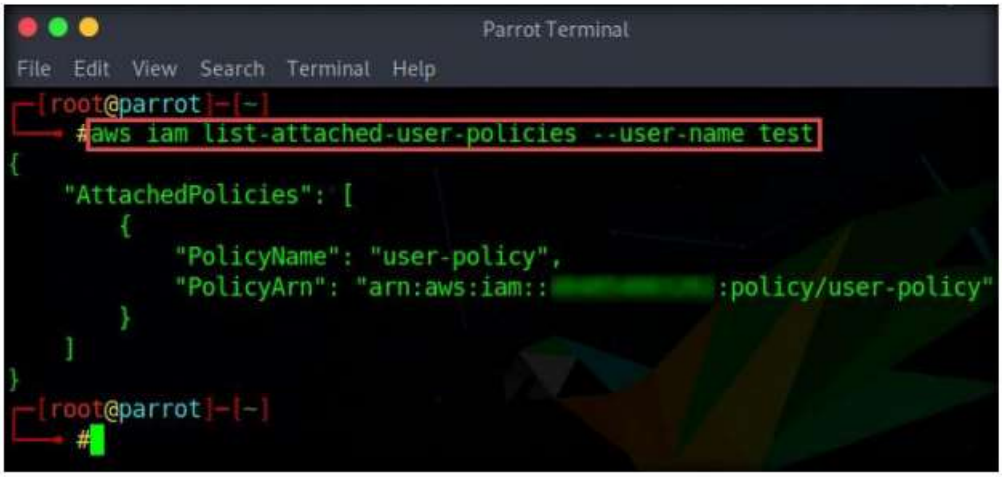

Figure 3.1.6: Listing the attached user policy

TASK 1.5

List the IAM Users

19. 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**.
20. The result appears, displaying the list of IAM users, as shown in the screenshot.

Note: The results may be different in your lab environment.

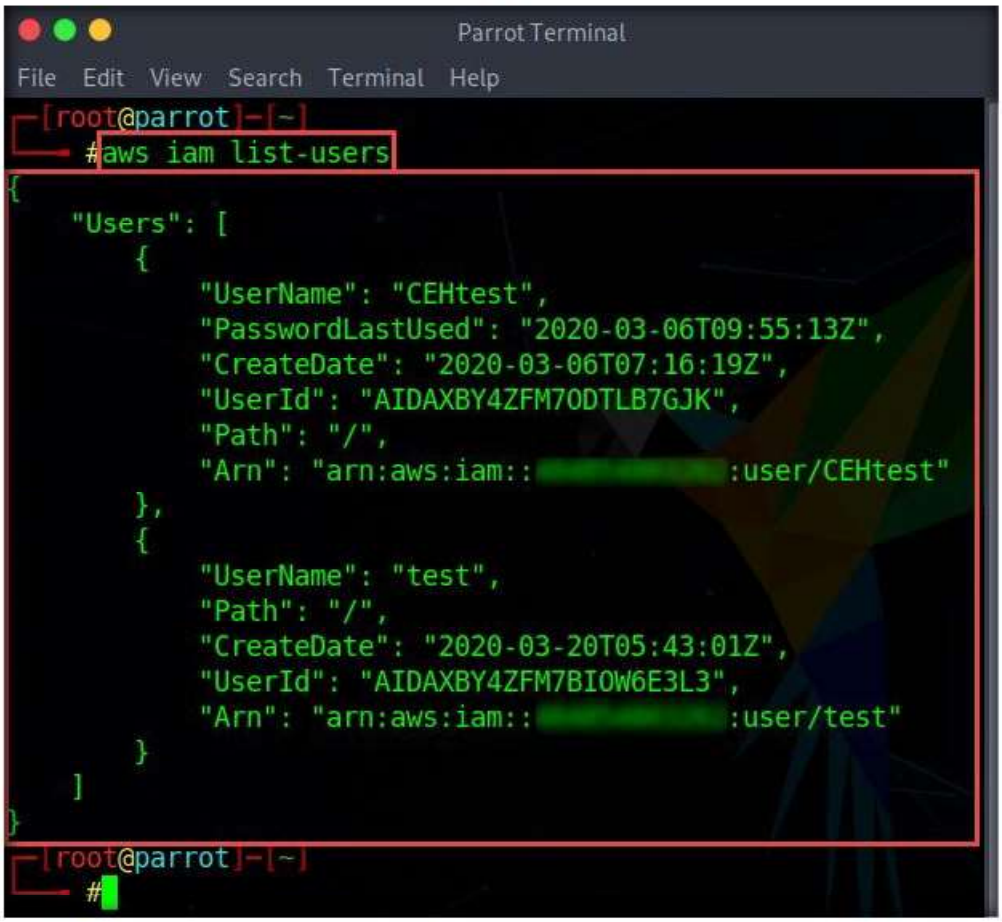

Figure 3.1.7: Listing IAM users

21. 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
  ```

22. This concludes the demonstration of escalating IAM user privileges by exploiting a misconfigured user policy.
23. Close all open windows and document all the acquired information.
24. Turn off the **Parrot Security** virtual machine.

Lab Analysis

Analyze and document all the results obtained in the lab exercise.

PLEASE TALK TO YOUR INSTRUCTOR IF YOU HAVE QUESTIONS ABOUT THIS LAB.

Internet Connection Required	
☑ Yes	☐ No
Platform Supported	
☑ Classroom	☑ iLabs