TP AWS IAM – 23/05/2024

1. Secret access key

```
aws iam create-access-key
```

```
aws iam list-access-keys
```

```
PS C:\Users\Tom\Documents\ensibs\s6\securite_cloud\tp2> aws iam list-access-keys
{
    "AccessKeyMetadata": [
        {
            "UserName": "Falcoz.Tom",
            "AccessKeyId": "AKIA2CGK3UZ7OCI2BZDJ",
            "Status": "Active",
            "CreateDate": "2024-05-23T11:41:49+00:00"
        },
        {
            "UserName": "Falcoz.Tom",
            "AccessKeyId": "AKIA2CGK3UZ7LLOESUXX",
            "Status": "Active",
            "CreateDate": "2024-05-23T11:51:34+00:00"
        }
    ]
}
```

2. Create IAM account

```
aws iam create-user --user-name Falcoz.Tom.subaccount
```

```
aws iam create-access-key --user-name Falcoz.Tom.subaccount
```

3. Describing process

Dans un premier temps, nous allons créer une politique qui permet d'assumer des rôles et l'associer à l'utilisateur secondaire.

Ensuite, nous allons créer un rôle qui fait confiance à l'utilisateur secondaire.

Enfin, nous allons créer une autre politique qui autorise de lister les clés secrètes de l'utilisateur principal et l'associer au rôle précédemment créé.

Et pour terminer, nous allons nos connecter au rôle précédemment créé pour lister les clés secrètes du compte principal.

4. Policy – ASSUME ROLE

```
aws iam create-policy --policy-name falcoz-tom-asmrole --policy-document
file://.\policies\falcoz-tom-asmrole-policy.json
```

aws iam attach-user-policy --user-name Falcoz.Tom.subaccount --policy-arn arn:aws:iam::691915171454:policy/falcoz-tom-asmrole

aws iam list-attached-user-policies --user-name Falcoz.Tom.subaccount

```
PS C:\Users\Tom\Documents\ensibs\s6\securite_cloud\tp2> aws iam list-attached-user-policies --user-name Falcoz.Tom.subaccount
{
    "AttachedPolicies": [
        {
            "PolicyName": "falcoz-tom-asmrole",
            "PolicyArn": "arn:aws:iam::691915171454:policy/falcoz-tom-asmrole"
        }
    ]
}
```

5. Role

aws iam create-role --role-name falcoz-tom-role --assume-role-policy-document file://.\policies\falcoz-tom-role-policy.json

aws iam list-roles

```
➜  trust aws iam list-roles | grep falcoz-tom-role -B2 -A16 --color=never
        {
            "Path": "/",
            "RoleName": "falcoz-tom-role",
            "RoleId": "AROA2CGK3UZ7K6BA427XS",
            "Arn": "arn:aws:iam::691915171454:role/falcoz-tom-role",
            "CreateDate": "2024-05-23T12:51:56+00:00",
            "AssumeRolePolicyDocument": {
                "Version": "2012-10-17",
                "Statement": [
                    {
                        "Effect": "Allow",
                        "Principal": {
                            "AWS": "arn:aws:iam::691915171454:user/Falcoz.Tom.subaccount"
                        },
                        "Action": "sts:AssumeRole"
                    }
                ]
            },
            "MaxSessionDuration": 3600
        },
```

6. Policy – KEY LIST

aws iam create-policy --policy-name falcoz-tom-keylist --policy-document file://.\policies\falcoz-tom-keylist-policy.json

aws iam attach-role-policy --role-name falcoz-tom-role --policy-arn arn:aws:iam::691915171454:policy/falcoz-tom-keylist

aws iam list-attached-role-policies --role-name falcoz-tom-role

```
PS C:\Users\Tom\Documents\ensibs\s6\securite_cloud\tp2> aws iam list-attached-role-policies --role-name falcoz-tom-role
{
    "AttachedPolicies": [
        {
            "PolicyName": "falcoz-tom-keylist",
            "PolicyArn": "arn:aws:iam::691915171454:policy/falcoz-tom-keylist"
        }
    ]
}
```

7. List secret keys from subaccount

`aws configure`

`aws sts get-caller-identity`

```
PS C:\Users\Tom\Documents\ensibs\s6\securite_cloud\tp2> aws configure
AWS Access Key ID [****************BZDJ]: AKIA2CGK3UZ7JUHA7SXC
AWS Secret Access Key [****************5rbS]: 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
Default region name [None]:
Default output format [None]:
PS C:\Users\Tom\Documents\ensibs\s6\securite_cloud\tp2> aws sts get-caller-identity
{
    "UserId": "AIDA2CGK3UZ7IGN5RPKDM",
    "Account": "691915171454",
    "Arn": "arn:aws:iam::691915171454:user/Falcoz.Tom.subaccount"
}
```

`aws iam list-access-keys`

```
PS C:\Users\Tom\Documents\ensibs\s6\securite_cloud\tp2> aws iam list-access-keys

An error occurred (AccessDenied) when calling the ListAccessKeys operation: User: arn:aws:iam::691915171454:user/Falcoz.Tom.subaccount is not authorized to perform: iam
:ListAccessKeys on resource: user nullFalcoz.Tom.subaccount because no identity-based policy allows the iam:ListAccessKeys action
```

8. Assume IAM ROLE

`aws sts assume-role --role-arn arn:aws:iam::691915171454:role/falcoz-tom-role --role-session-name AWSCLI-Session`

9. Switch to created role

`aws sts get-caller-identity`

```
C:\Users\Tom\Documents\ensibs\s6\securite_cloud\tp2>aws sts get-caller-identity
{
    "UserId": "AROA2CGK3UZ7K6BA427XS:AWSCLI-Session",
    "Account": "691915171454",
    "Arn": "arn:aws:sts::691915171454:assumed-role/falcoz-tom-role/AWSCLI-Session"
}
```

`aws iam list-access-keys --user-name Falcoz.Tom`

```
C:\Users\Tom\Documents\ensibs\s6\securite_cloud\tp2>aws iam list-access-keys --user-name Falcoz.Tom
{
    "AccessKeyMetadata": [
        {
            "UserName": "Falcoz.Tom",
            "AccessKeyId": "AKIA2CGK3UZ7OCI2BZDJ",
            "Status": "Active",
            "CreateDate": "2024-05-23T11:41:49+00:00"
        },
        {
            "UserName": "Falcoz.Tom",
            "AccessKeyId": "AKIA2CGK3UZ7LLOESUXX",
            "Status": "Active",
            "CreateDate": "2024-05-23T11:51:34+00:00"
        }
    ]
}
```