Week 8: Simulated Attack & Cloud Incident Response

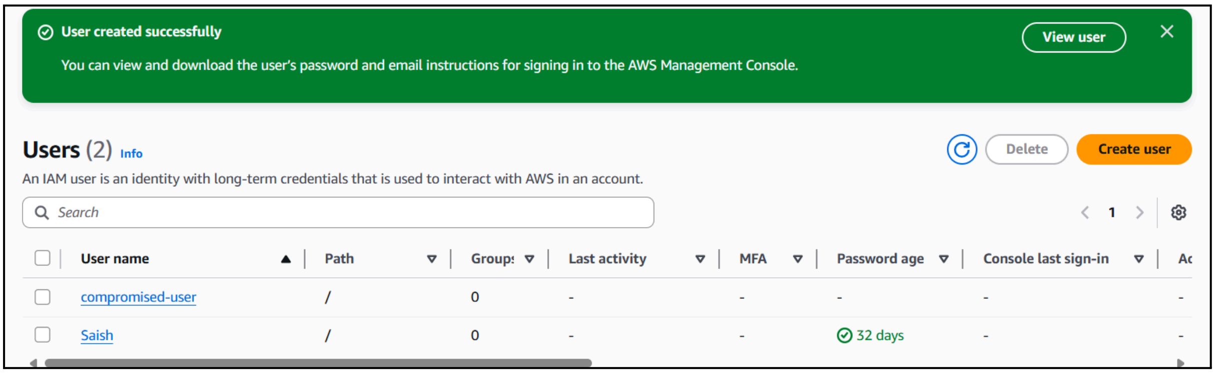
Name- Saish Dhiwar
PRN- 2124UCSM2011
Email- saish.dhiwar_24ucs@sanjivani.edu.in

Task- 1) Simulate a compromised account scenario.
2) Respond & secure the account.
3) Document the incident response steps.

Task 1- Simulate a Compromised Account.

1) Log in to AWS Console. Go to https://aws.amazon.com, sign in to the AWS Management Console.

2) Create a test IAM user named **`compromised-user`** with basic permissions (e.g., Amazon S3 read access).


User created successfully

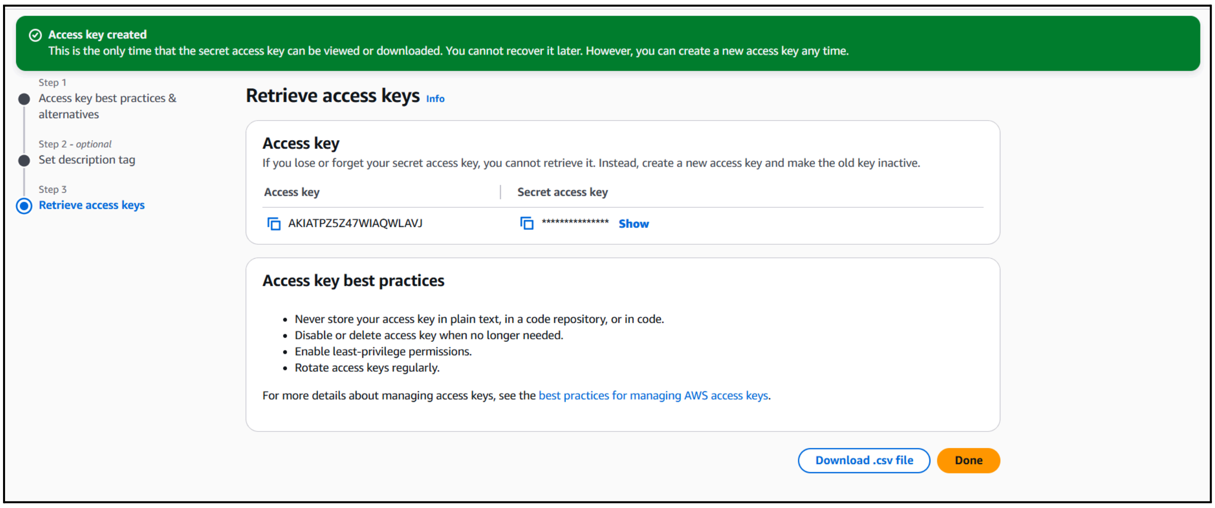
You can view and download the user's password and email instructions for signing in to the AWS Management Console.

Users (2) Info

An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.

User name	Path	Group:	Last activity	MFA	Password age	Console last sign-in	Ac
compromised-user	/	0	-	-	-	-	-
Saish	/	0	-	-	✓ 32 days	-	-

3) Generate **Access Keys** for the user.


Access key created
This is the only time that the secret access key can be viewed or downloaded. You cannot recover it later. However, you can create a new access key any time.

Step 1
Access key best practices & alternatives

Step 2 - *optional*
Set description tag

Step 3
Retrieve access keys

Retrieve access keys Info

Access key

If you lose or forget your secret access key, you cannot retrieve it. Instead, create a new access key and make the old key inactive.

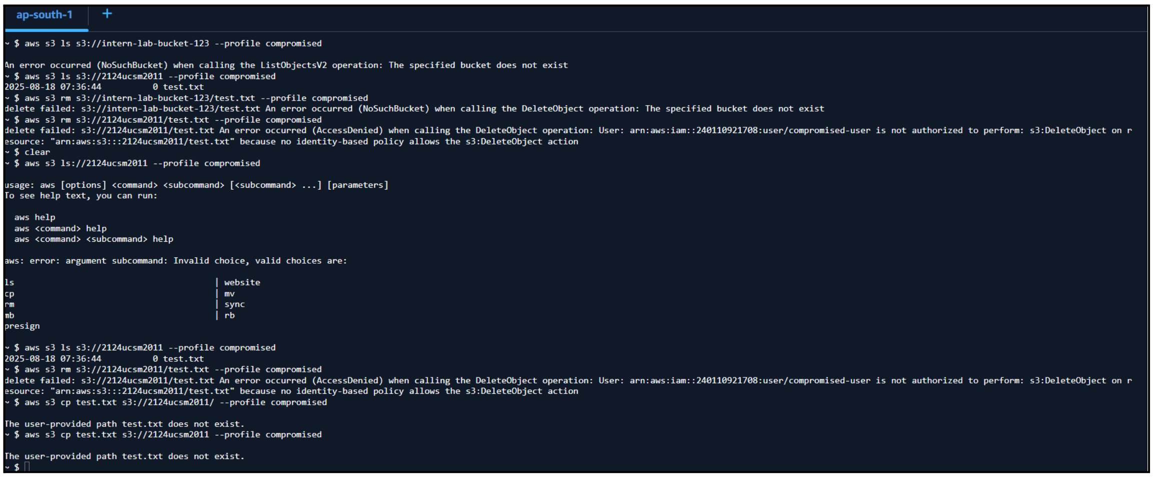
Access key	Secret access key
AKIATPZ5Z47WIAQWLAVJ	*************** Show

Access key best practices

- Never store your access key in plain text, in a code repository, or in code.
- Disable or delete access key when no longer needed.
- Enable least-privilege permissions.
- Rotate access keys regularly.

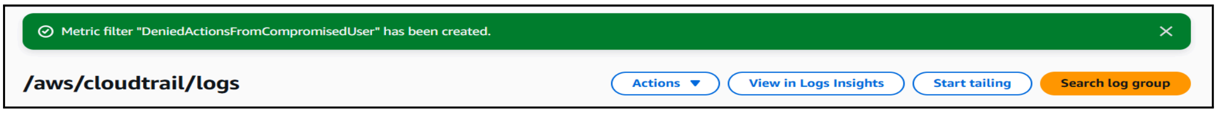
For more details about managing access keys, see the best practices for managing AWS access keys.

4) Share these keys intentionally with a test machine/script that attempts unauthorized AWS CLI actions (e.g., trying to delete S3 objects without permission).



```
~ $ aws s3 ls s3://intern-lab-bucket-123 --profile compromised

An error occurred (NoSuchBucket) when calling the ListObjectsV2 operation: The specified bucket does not exist
~ $ aws s3 ls s3://2124ucsm2011 --profile compromised
2025-08-18 07:36:44          0 test.txt
~ $ aws s3 rm s3://intern-lab-bucket-123/test.txt --profile compromised
delete failed: s3://intern-lab-bucket-123/test.txt An error occurred (NoSuchBucket) when calling the DeleteObject operation: The specified bucket does not exist
~ $ aws s3 rm s3://2124ucsm2011/test.txt --profile compromised
delete failed: s3://2124ucsm2011/test.txt An error occurred (AccessDenied) when calling the DeleteObject operation: User: arn:aws:iam::240110921708:user/compromised-user is not authorized to perform: s3:DeleteObject on r
esource: "arn:aws:s3:::2124ucsm2011/test.txt" because no identity-based policy allows the s3:DeleteObject action
~ $ clear
~ $ aws s3 ls://2124ucsm2011 --profile compromised

usage: aws [options] <command> <subcommand> [<subcommand> ...] [parameters]
To see help text, you can run:

  aws help
  aws <command> help
  aws <command> <subcommand> help

aws: error: argument subcommand: Invalid choice, valid choices are:

ls                                       | website
cp                                       | mv
rm                                       | sync
mb                                       | rb
presign

~ $ aws s3 ls s3://2124ucsm2011 --profile compromised
2025-08-18 07:36:44          0 test.txt
~ $ aws s3 rm s3://2124ucsm2011/test.txt --profile compromised
delete failed: s3://2124ucsm2011/test.txt An error occurred (AccessDenied) when calling the DeleteObject operation: User: arn:aws:iam::240110921708:user/compromised-user is not authorized to perform: s3:DeleteObject on r
esource: "arn:aws:s3:::2124ucsm2011/test.txt" because no identity-based policy allows the s3:DeleteObject action
~ $ aws s3 cp test.txt s3://2124ucsm2011/ --profile compromised

The user-provided path test.txt does not exist.
~ $ aws s3 cp test.txt s3://2124ucsm2011 --profile compromised

The user-provided path test.txt does not exist.
~ $
```

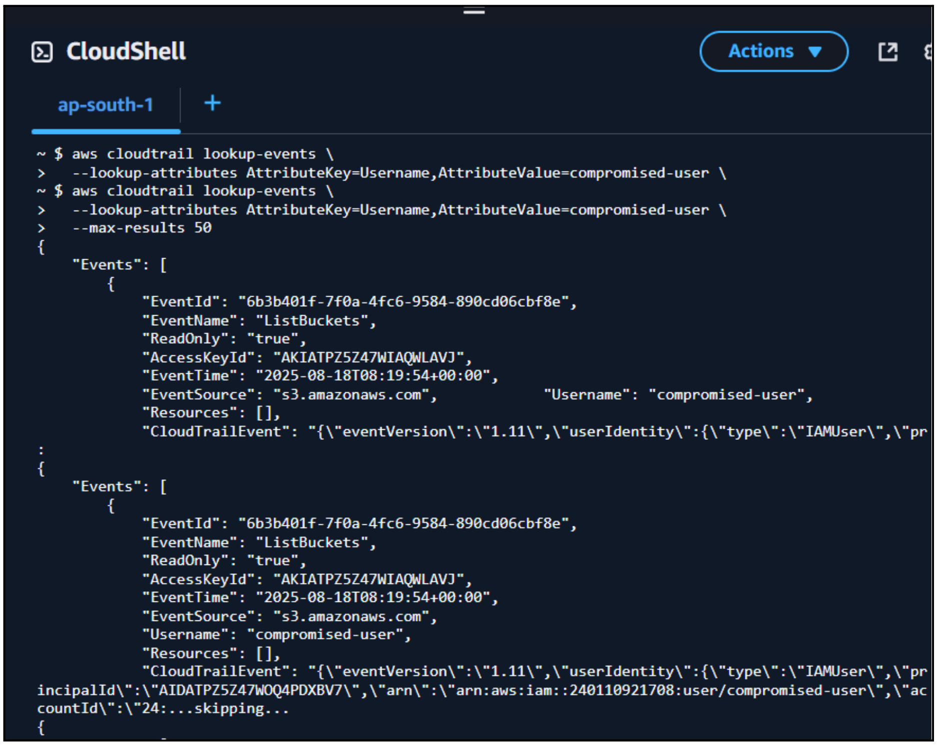
5) Monitor AWS CloudTrail and CloudWatch to detect unusual activity from the user account.


Metric filter "DeniedActionsFromCompromisedUser" has been created.

/aws/cloudtrail/logs

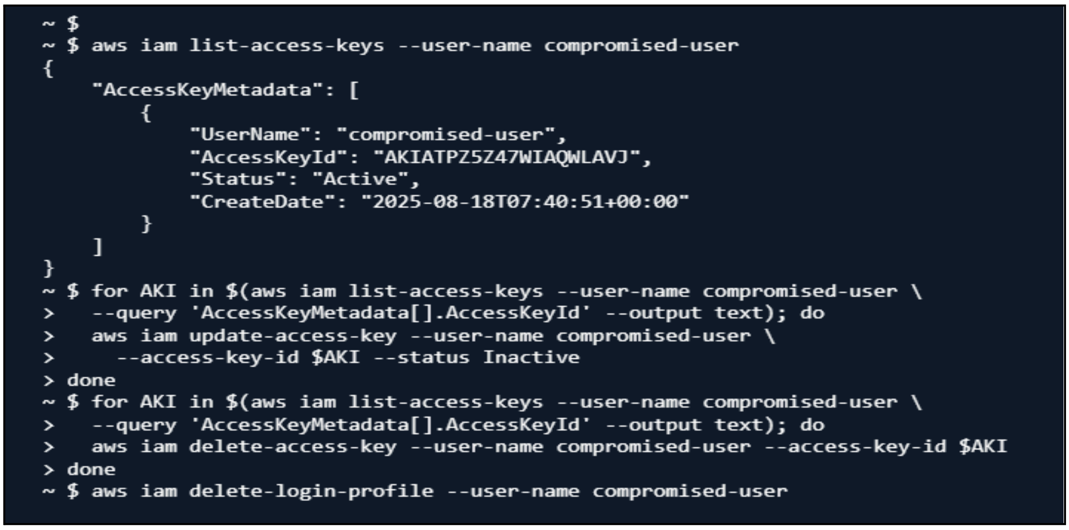
Task 2- Respond to the Incident.

1) Identify the Source: Go to CloudTrail → Event History. Filter by compromised-user and check the activity timeline.



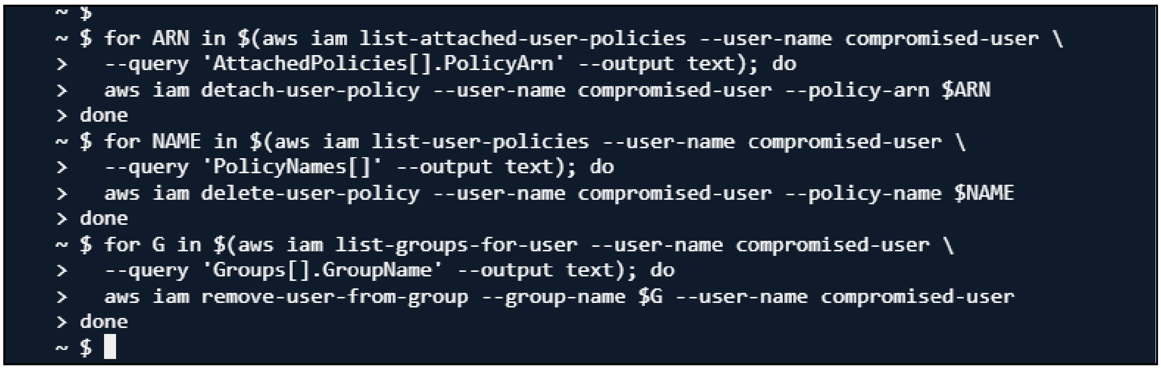
```
~ $ aws cloudtrail lookup-events \
>   --lookup-attributes AttributeKey=Username,AttributeValue=compromised-user \
~ $ aws cloudtrail lookup-events \
>   --lookup-attributes AttributeKey=Username,AttributeValue=compromised-user \
>   --max-results 50
{
    "Events": [
        {
            "EventId": "6b3b401f-7f0a-4fc6-9584-890cd06cbf8e",
            "EventName": "ListBuckets",
            "ReadOnly": "true",
            "AccessKeyId": "AKIATPZ5Z47WIAQWLAVJ",
            "EventTime": "2025-08-18T08:19:54+00:00",
            "EventSource": "s3.amazonaws.com",            "Username": "compromised-user",
            "Resources": [],
            "CloudTrailEvent": "{\"eventVersion\":\"1.11\",\"userIdentity\":{\"type\":\"IAMUser\",\"pr
:
{
    "Events": [
        {
            "EventId": "6b3b401f-7f0a-4fc6-9584-890cd06cbf8e",
            "EventName": "ListBuckets",
            "ReadOnly": "true",
            "AccessKeyId": "AKIATPZ5Z47WIAQWLAVJ",
            "EventTime": "2025-08-18T08:19:54+00:00",
            "EventSource": "s3.amazonaws.com",
            "Username": "compromised-user",
            "Resources": [],
            "CloudTrailEvent": "{\"eventVersion\":\"1.11\",\"userIdentity\":{\"type\":\"IAMUser\",\"pr
incipalId\":\"AIDATPZ5Z47WOQ4PDXBV7\",\"arn\":\"arn:aws:iam::240110921708:user/compromised-user\",\"ac
countId\":\"24:...skipping...
{
```

2) Secure the Account: Deactivate the compromised Access Keys: aws iam update-access-key --access-key-id <ACCESS_KEY> --status Inactive --user-name compromised-user



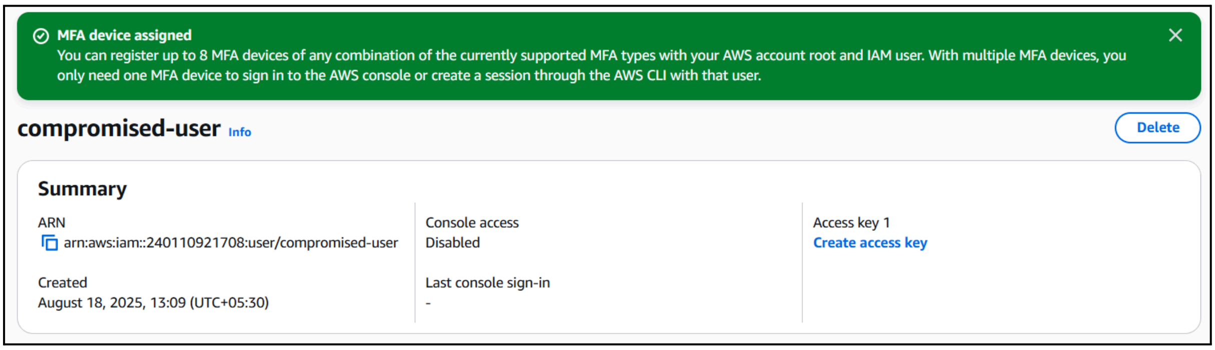
```
~ $
~ $ aws iam list-access-keys --user-name compromised-user
{
    "AccessKeyMetadata": [
        {
            "UserName": "compromised-user",
            "AccessKeyId": "AKIATPZ5Z47WIAQWLAVJ",
            "Status": "Active",
            "CreateDate": "2025-08-18T07:40:51+00:00"
        }
    ]
}
~ $ for AKI in $(aws iam list-access-keys --user-name compromised-user \
>   --query 'AccessKeyMetadata[].AccessKeyId' --output text); do
>   aws iam update-access-key --user-name compromised-user \
>     --access-key-id $AKI --status Inactive
> done
~ $ for AKI in $(aws iam list-access-keys --user-name compromised-user \
>   --query 'AccessKeyMetadata[].AccessKeyId' --output text); do
>   aws iam delete-access-key --user-name compromised-user --access-key-id $AKI
> done
~ $ aws iam delete-login-profile --user-name compromised-user
```

3) Restrict Permissions: Detach all policies from the user. Remove from any IAM group.



```
~ $
~ $ for ARN in $(aws iam list-attached-user-policies --user-name compromised-user \
>   --query 'AttachedPolicies[].PolicyArn' --output text); do
>   aws iam detach-user-policy --user-name compromised-user --policy-arn $ARN
> done
~ $ for NAME in $(aws iam list-user-policies --user-name compromised-user \
>   --query 'PolicyNames[]' --output text); do
>   aws iam delete-user-policy --user-name compromised-user --policy-name $NAME
> done
~ $ for G in $(aws iam list-groups-for-user --user-name compromised-user \
>   --query 'Groups[].GroupName' --output text); do
>   aws iam remove-user-from-group --group-name $G --user-name compromised-user
> done
~ $ █
```

4) Enable MFA (Multi-Factor Authentication) on the account.



MFA device assigned

You can register up to 8 MFA devices of any combination of the currently supported MFA types with your AWS account root and IAM user. With multiple MFA devices, you only need one MFA device to sign in to the AWS console or create a session through the AWS CLI with that user.

compromised-user

Summary

ARN: arn:aws:iam::240110921708:user/compromised-user

Console access: Disabled

Access key 1: Create access key

Created: August 18, 2025, 13:09 (UTC+05:30)

Last console sign-in: -

Task 3- Document the Incident Response Steps.

Incident Summary:

- **Incident Type**: Simulated account compromise.
- **Impact**: Unauthorized S3 deletion attempt blocked.
- **Detection Method**: AWS CloudTrail & CloudWatch alerts.
- **Response Actions**: Access keys deactivated, permissions removed, MFA enabled.