ENPM665 - Classwork Lab 3: Investigating an AWS Breach with CloudTrail & IAM

Name: Raghava Gatadi

UMD ID: raghavag / 121941190

Course & Section: ENPM665, 0101

Executive Summary

This report details the findings of a forensic investigation into suspicious activity within the Terrapin CloudWorks AWS account. The investigation confirms that a security breach occurred on August 26, 2023. An external attacker utilized a compromised IAM access key belonging to the user temp-user to gain initial access. The attacker then escalated privileges by assuming the high-privilege AdminRole, which was used to access an S3 bucket and exfiltrate a sensitive file named emergency.txt.

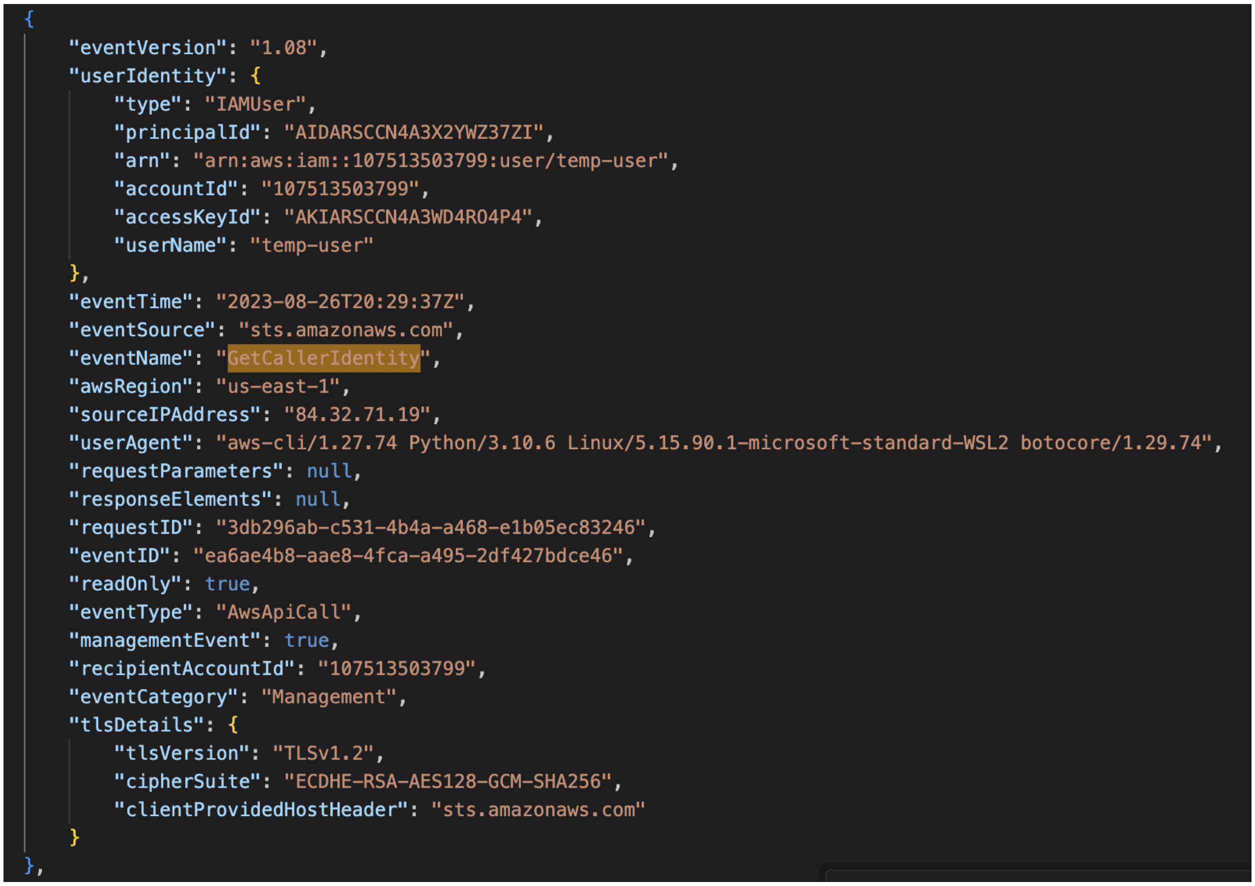
1. Attack Timeline

The attack followed a clear, linear progression from initial access to data exfiltration over approximately 48 minutes. The suspected entry point was the programmatic use of a compromised access key (**AKIARSCCN4A3WD4RO4P4**) belonging to the IAM user **temp-user**.

- **20:29:37Z: Initial Access & Reconnaissance**

  The attacker makes their first API call, GetCallerIdentity, from the IP address 84.32.71.19 to confirm the identity of the compromised credentials.

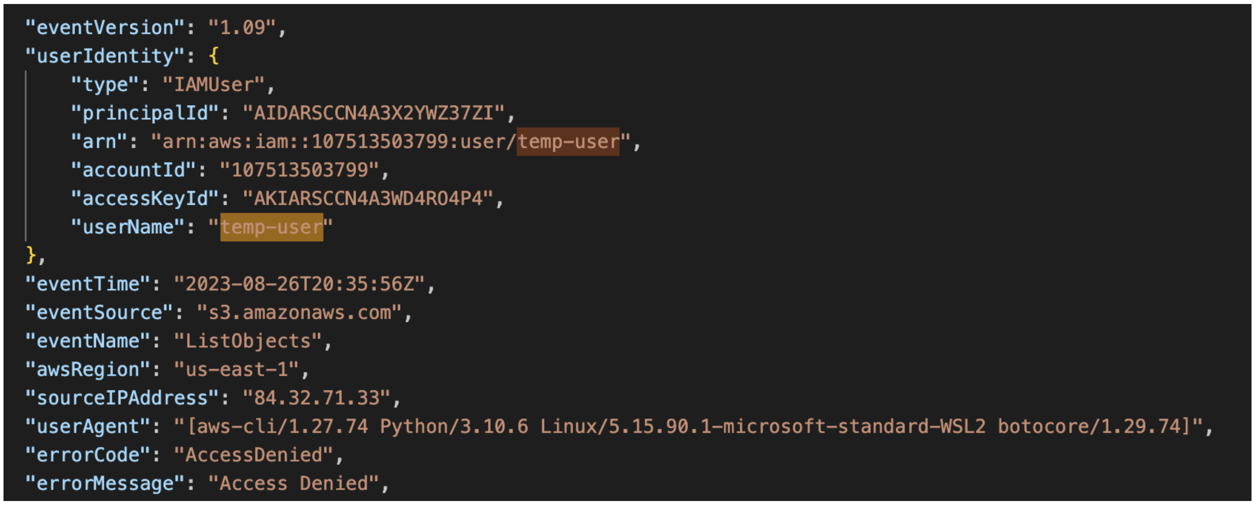
  IAM Identity: temp-user



```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "AIDARSCCN4A3X2YWZ37ZI",
        "arn": "arn:aws:iam::107513503799:user/temp-user",
        "accountId": "107513503799",
        "accessKeyId": "AKIARSCCN4A3WD4RO4P4",
        "userName": "temp-user"
    },
    "eventTime": "2023-08-26T20:29:37Z",
    "eventSource": "sts.amazonaws.com",
    "eventName": "GetCallerIdentity",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "84.32.71.19",
    "userAgent": "aws-cli/1.27.74 Python/3.10.6 Linux/5.15.90.1-microsoft-standard-WSL2 botocore/1.29.74",
    "requestParameters": null,
    "responseElements": null,
    "requestID": "3db296ab-c531-4b4a-a468-e1b05ec83246",
    "eventID": "ea6ae4b8-aae8-4fca-a495-2df427bdce46",
    "readOnly": true,
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "107513503799",
    "eventCategory": "Management",
    "tlsDetails": {
        "tlsVersion": "TLSv1.2",
        "cipherSuite": "ECDHE-RSA-AES128-GCM-SHA256",
        "clientProvidedHostHeader": "sts.amazonaws.com"
    }
},
```

- **20:35:56Z - 20:47:31Z: Failed Enumeration**

  The attacker makes numerous attempts to discover resources (e.g., ListBuckets, DescribeInstances) as temp-user but is blocked by AccessDenied errors. This indicates the user had limited permissions, prompting the attacker to seek escalation.



```
"eventVersion": "1.09",
"userIdentity": {
    "type": "IAMUser",
    "principalId": "AIDARSCCN4A3X2YWZ37ZI",
    "arn": "arn:aws:iam::107513503799:user/temp-user",
    "accountId": "107513503799",
    "accessKeyId": "AKIARSCCN4A3WD4RO4P4",
    "userName": "temp-user"
},
"eventTime": "2023-08-26T20:35:56Z",
"eventSource": "s3.amazonaws.com",
"eventName": "ListObjects",
"awsRegion": "us-east-1",
"sourceIPAddress": "84.32.71.33",
"userAgent": "[aws-cli/1.27.74 Python/3.10.6 Linux/5.15.90.1-microsoft-standard-WSL2 botocore/1.29.74]",
"errorCode": "AccessDenied",
"errorMessage": "Access Denied",
```

- **20:54:28Z: Privilege Escalation**

  The attacker successfully uses the sts:AssumeRole permission on temp-user to escalate privileges, assuming the AdminRole.

  - **IAM Identities Involved**: temp-user (initiator), AdminRole (target)

- **21:17:10Z: Target Discovery**

  Now operating as AdminRole, the attacker lists the contents of the emergency-data-recovery S3 bucket to identify target files.

- **21:17:16Z: Data Exfiltration**

  The attacker downloads the file emergency.txt from the S3 bucket, completing their objective.

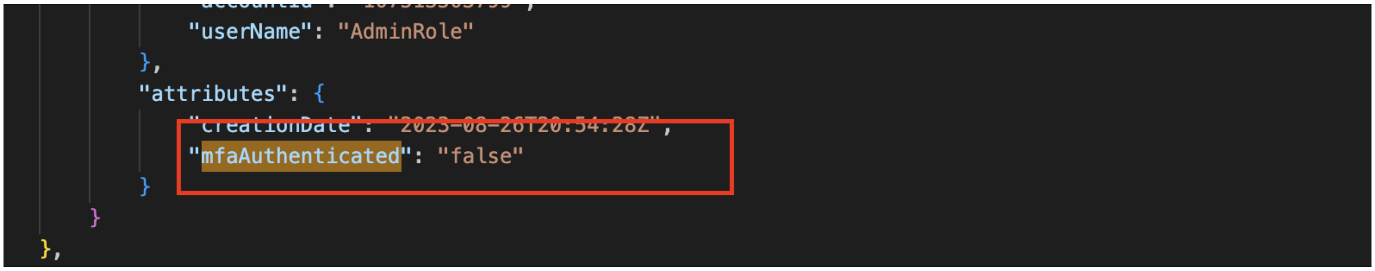
  IAM Identity: AdminRole (assumed)

This log entry marks the start of the incident, showing the first command executed by the attacker using the compromised temp-user credentials.

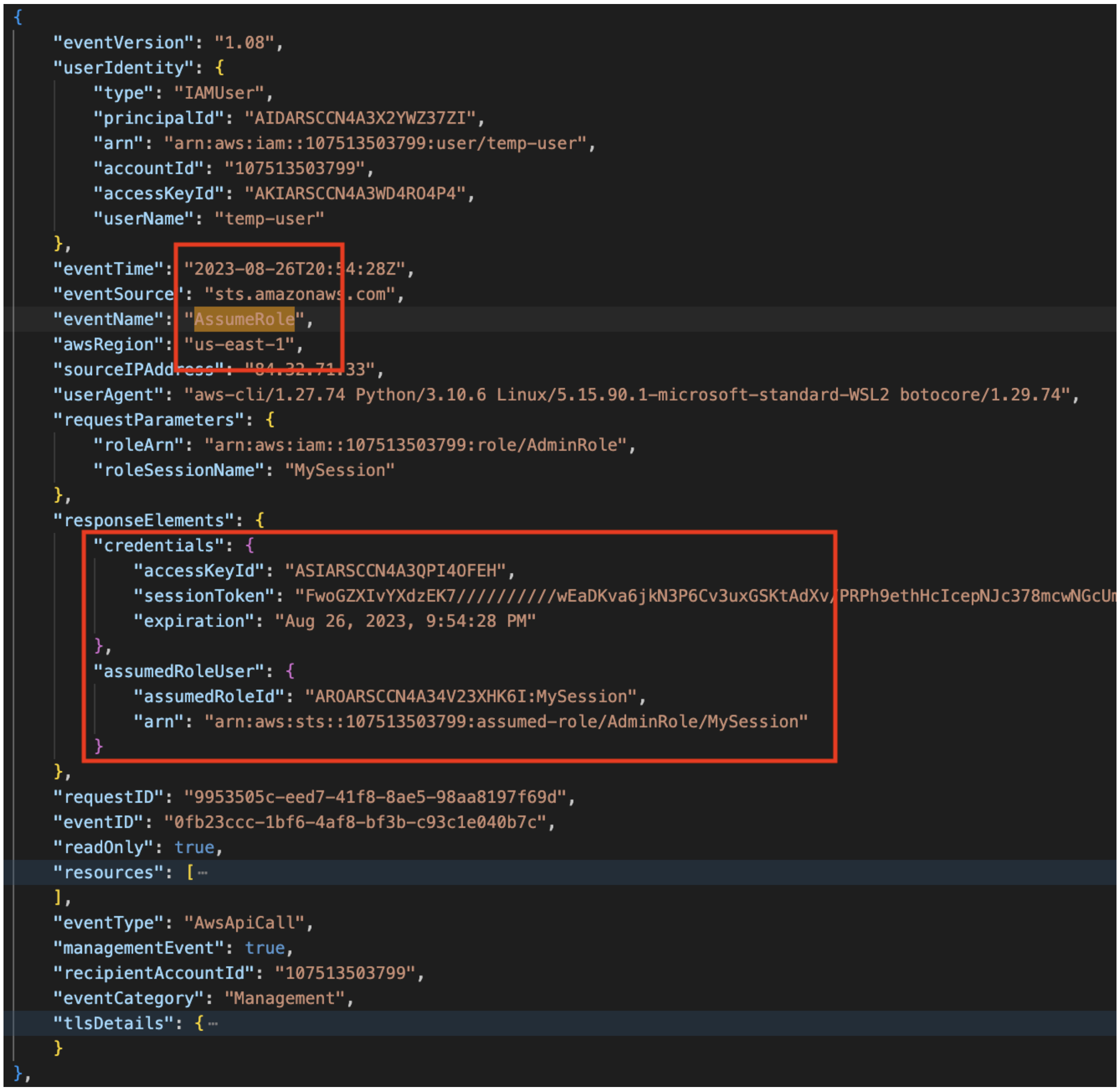
2. Privilege Escalation or Role Abuse

The pivotal moment of the attack was the successful privilege escalation from a low-permission user to an administrative role. This was achieved via the AssumeRole API call.

- **Identity Context Before**: The attacker operated as the IAM user temp-user, whose permissions were insufficient, resulting in numerous AccessDenied errors.
- **Identity Context After**: The attacker assumed the AdminRole, gaining elevated privileges to access and exfiltrate data from sensitive S3 buckets. The log confirms this switch was not authenticated with MFA ("mfaAuthenticated": "false").



```
                "accountId": "107513303795",
                "userName": "AdminRole"
            },
            "attributes": {
                "creationDate": "2025-08-26T20:54:28Z",
                "mfaAuthenticated": "false"
            }
        }
    },
```



```
{
    "eventVersion": "1.08",
    "userIdentity": {
        "type": "IAMUser",
        "principalId": "AIDARSCCN4A3X2YWZ37ZI",
        "arn": "arn:aws:iam::107513503799:user/temp-user",
        "accountId": "107513503799",
        "accessKeyId": "AKIARSCCN4A3WD4RO4P4",
        "userName": "temp-user"
    },
    "eventTime": "2023-08-26T20:54:28Z",
    "eventSource": "sts.amazonaws.com",
    "eventName": "AssumeRole",
    "awsRegion": "us-east-1",
    "sourceIPAddress": "84.32.71.33",
    "userAgent": "aws-cli/1.27.74 Python/3.10.6 Linux/5.15.90.1-microsoft-standard-WSL2 botocore/1.29.74",
    "requestParameters": {
        "roleArn": "arn:aws:iam::107513503799:role/AdminRole",
        "roleSessionName": "MySession"
    },
    "responseElements": {
        "credentials": {
            "accessKeyId": "ASIARSCCN4A3QPI4OFEH",
            "sessionToken": "FwoGZXIvYXdzEK7//////////wEaDKva6jkN3P6Cv3uxGSKtAdXv/PRPh9ethHcIcepNJc378mcwNGcUm
            "expiration": "Aug 26, 2023, 9:54:28 PM"
        },
        "assumedRoleUser": {
            "assumedRoleId": "AROARSCCN4A34V23XHK6I:MySession",
            "arn": "arn:aws:sts::107513503799:assumed-role/AdminRole/MySession"
        }
    },
    "requestID": "9953505c-eed7-41f8-8ae5-98aa8197f69d",
    "eventID": "0fb23ccc-1bf6-4af8-bf3b-c93c1e040b7c",
    "readOnly": true,
    "resources": […
    ],
    "eventType": "AwsApiCall",
    "managementEvent": true,
    "recipientAccountId": "107513503799",
    "eventCategory": "Management",
    "tlsDetails": {…
    }
},
```

This log is the smoking gun for privilege escalation. It explicitly shows temp-user successfully requesting and receiving temporary credentials for AdminRole, fundamentally changing their access level.

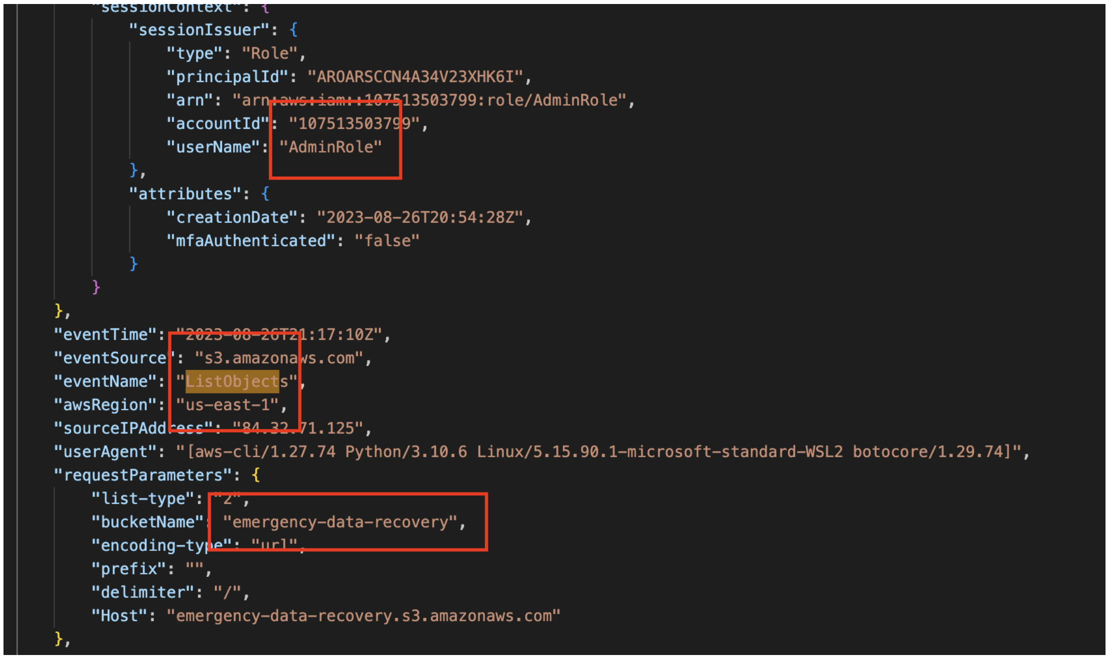
3. Resource Access & Data Handling

The attacker's activity was focused and goal-oriented. After escalating privileges, they immediately targeted data stored in Amazon S3.

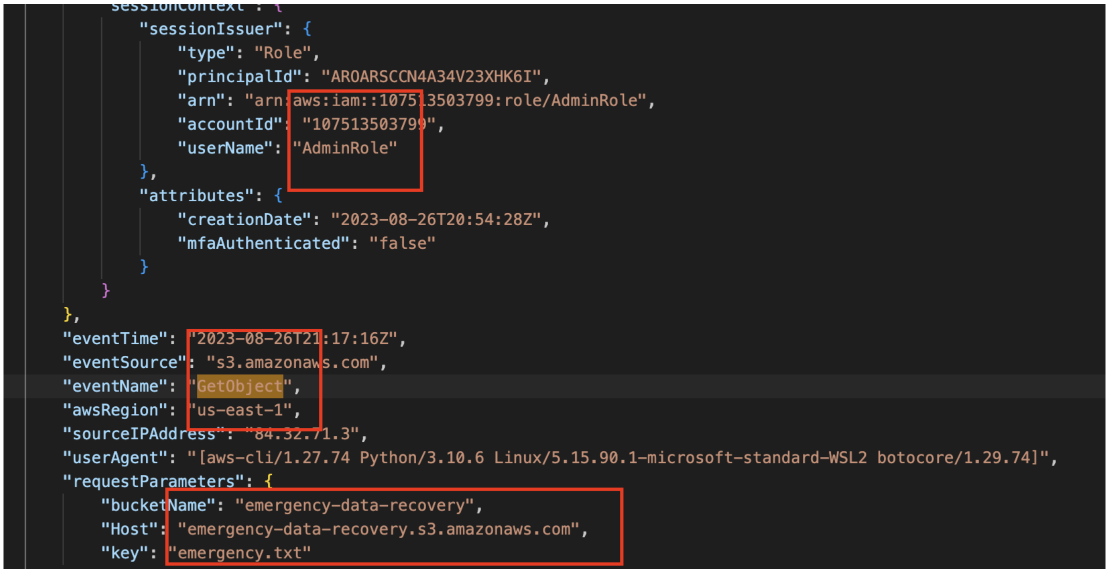
- **AWS Services Accessed**:
    - **AWS STS**: Used for identity confirmation (GetCallerIdentity) and privilege escalation (AssumeRole).
    - **Amazon S3**: Used for reconnaissance (ListObjects) and data exfiltration (GetObject).

- ○ **Amazon IAM/EC2** (Attempted): The attacker made numerous failed attempts to list resources, indicating an interest in other services.

- **Suspicious Access & Downloads**:
  The attacker's S3 actions demonstrate clear intent. At 21:17:10Z, they listed the contents of the emergency-data-recovery bucket from IP 84.32.71.125. Just six seconds later, at 21:17:16Z, they downloaded the file emergency.txt from a different IP (84.32.71.3), confirming targeted data exfiltration.



```
            "sessionContext": {
                "sessionIssuer": {
                    "type": "Role",
                    "principalId": "AROARSCCN4A34V23XHK6I",
                    "arn": "arn:aws:iam::107513503799:role/AdminRole",
                    "accountId": "107513503799",
                    "userName": "AdminRole"
                },
                "attributes": {
                    "creationDate": "2023-08-26T20:54:28Z",
                    "mfaAuthenticated": "false"
                }
            }
        },
        "eventTime": "2023-08-26T21:17:10Z",
        "eventSource": "s3.amazonaws.com",
        "eventName": "ListObjects",
        "awsRegion": "us-east-1",
        "sourceIPAddress": "84.32.71.125",
        "userAgent": "[aws-cli/1.27.74 Python/3.10.6 Linux/5.15.90.1-microsoft-standard-WSL2 botocore/1.29.74]",
        "requestParameters": {
            "list-type": "2",
            "bucketName": "emergency-data-recovery",
            "encoding-type": "url",
            "prefix": "",
            "delimiter": "/",
            "Host": "emergency-data-recovery.s3.amazonaws.com"
        },
```



```
            "sessionContext": {
                "sessionIssuer": {
                    "type": "Role",
                    "principalId": "AROARSCCN4A34V23XHK6I",
                    "arn": "arn:aws:iam::107513503799:role/AdminRole",
                    "accountId": "107513503799",
                    "userName": "AdminRole"
                },
                "attributes": {
                    "creationDate": "2023-08-26T20:54:28Z",
                    "mfaAuthenticated": "false"
                }
            }
        },
        "eventTime": "2023-08-26T21:17:16Z",
        "eventSource": "s3.amazonaws.com",
        "eventName": "GetObject",
        "awsRegion": "us-east-1",
        "sourceIPAddress": "84.32.71.3",
        "userAgent": "[aws-cli/1.27.74 Python/3.10.6 Linux/5.15.90.1-microsoft-standard-WSL2 botocore/1.29.74]",
        "requestParameters": {
            "bucketName": "emergency-data-recovery",
            "Host": "emergency-data-recovery.s3.amazonaws.com",
            "key": "emergency.txt"
```

4. Indicators of Compromise (IoCs)

Several technical and behavioral indicators were identified during the investigation, which can be used for future threat hunting and alerting.

- **Attacker IP Addresses**:
    - *84.32.71.3*, *84.32.71.5*, *84.32.71.19*, *84.32.71.33*, *84.32.71.48*, *84.32.71.125*
- **Attacker User Agents**:
    - aws-cli/1.27.74 Python/3.10.6 Linux/5.15.90.1-microsoft-standard-WSL2 botocore/1.29.74
    - aws-sdk-go-v2/1.3.2
- **IAM Principals Involved**:
    - temp-user (Compromised IAM User)
    - AdminRole (Abused IAM Role)
- **Behavioral Patterns**:
    - High Volume of A sudden spike of failed API calls from a single user (temp-user) was a clear indicator of malicious reconnaissance.
    - After Recon Failure: The successful privilege escalation immediately followed the period of failed discovery attempts.

5. Prevention Recommendations & Reflection

Based on the attack chain, the following cloud-specific security improvements are recommended to prevent similar incidents.

- **Prevention Recommendations**:
    1. **Enforce Principle of Least Privilege**: The temp-user should not have had the sts:AssumeRole permission for a highly privileged target like AdminRole. IAM permissions must be reviewed and scoped down to the minimum required for a user's function.
    2. **Require MFA for Sensitive Actions**: The trust policy for AdminRole should be modified to require multi-factor authentication (MFA) for assumption. This would have blocked the attacker at the privilege escalation stage, as they only possessed a single factor (the access key).

3. **Implement Access Key Rotation and Auditing**: The initial point of failure was a compromised access key. A strict key rotation policy (e.g., every 90 days) should be enforced. Furthermore, unused or old access keys must be disabled and removed.
4. **Establish Threat Detection and Alerting**: Create automated alerts in Amazon CloudWatch or a SIEM for high-risk behavioral patterns. An alert for a high count of AccessDenied errors from one user, followed by a successful AssumeRole call, would have detected this incident in near real-time.
5. **Utilize IP-Based Controls**: Where feasible, use IAM Condition Keys in policies to restrict API calls to known corporate IP ranges, preventing access from unauthorized locations.

- **Reflection**:
  This investigation highlights how a single compromised credential can lead to a significant data breach if defense-in-depth is not implemented. It demonstrates the critical importance of monitoring for behavioral anomalies, such as failed API calls, as they often precede more severe actions like privilege escalation. The ability to connect these seemingly disparate events in the CloudTrail logs was essential to reconstructing the full attack narrative and understanding the attacker's motives.