**Project 4: IAM-Controlled S3 Access Script (Linux Permission + IAM Policy)**

**1. Introduction**

This project focuses on securing access to an Amazon S3 bucket by combining **Linux file permissions** and **AWS Identity and Access Management (IAM) policies**. The main goal is to implement the principle of least privilege, ensuring that only authorized users can upload, download, or manage files in S3.

By creating a **Bash script** for controlled access, we integrate Linux system-level controls with AWS IAM roles/policies, providing a two-layer security mechanism.

---

**2. Objectives**

- Implement IAM roles and policies for **fine-grained S3 access control**.
- Use Linux user groups and file permissions to manage **local access** to the script.
- Automate upload, download, and list operations in S3 via the AWS CLI.
- Demonstrate restricted access to unauthorized users at both **OS and IAM levels**.

---

**3. System Architecture**

**Components**

1. **Amazon S3** – Central storage for files.
2. **IAM User/Role** – Defines access permissions to S3.
3. **Linux User Accounts** – Controls which users can execute the script.
4. **Bash Script** – Provides a controlled interface for S3 operations.

**Access Control Layers**

- **Linux Layer**:
    - Script accessible only by users in a specific Linux group (e.g., s3users).
    - File permissions set with chmod and chown.
- **IAM Layer**:
    - IAM policy grants **read/write access** to a specific bucket.
    - Unauthorized IAM users cannot interact with the bucket even if they have local script access.

---

**4. Implementation Steps**

**Step 1: IAM Policy Creation**

Example IAM policy for controlled S3 access:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "s3:PutObject",
        "s3:GetObject",
        "s3:ListBucket"
      ],
      "Resource": [
        "arn:aws:s3:::project4-bucket",
        "arn:aws:s3:::project4-bucket/*"
      ]
    }
  ]
}
```

**Step 2: Assign Policy to IAM User/Role**

- Attach the above policy to a user (project4-user).
- Configure AWS CLI using aws configure.

**Step 3: Linux Permissions**

```
# Create a group for S3 users
sudo groupadd s3users

# Add allowed users to group
sudo usermod -aG s3users alice

# Restrict script permissions
sudo chown root:s3users s3_access.sh
sudo chmod 750 s3_access.sh
```

Only members of s3users can run the script.

**Step 4: Bash Script (s3_access.sh)**

```
#!/bin/bash
# IAM-Controlled S3 Access Script

BUCKET="project4-bucket"

case "$1" in
  upload)
    aws s3 cp "$2" s3://$BUCKET/
    ;;
  download)
    aws s3 cp s3://$BUCKET/"$2" "$3"
    ;;
  list)
    aws s3 ls s3://$BUCKET/
    ;;
  *)
    echo "Usage: $0 {upload <file>|download <s3file> <dest>|list}"
    exit 1
    ;;
esac
```

**Step 5: Testing**

- **Authorized user (Linux + IAM)**: Can upload, download, and list files.
- **Unauthorized Linux user**: Cannot execute script due to chmod 750.
- **Unauthorized IAM user**: Script runs but S3 operations fail with **AccessDenied** error.

---

**5. Security Benefits**

- **Defense in Depth**: Even if IAM credentials are compromised, Linux restrictions prevent local misuse.
- **Least Privilege**: Users are granted only necessary permissions.
- **Auditability**: IAM logs track all S3 operations; Linux system logs track script execution.

---

**6. Challenges and Solutions**

- **IAM Misconfiguration** → Resolved by testing policies with AWS CLI --dry-run.
- **Linux User Management** → Group-based access simplified permissions.
- **Error Handling in Script** → Added usage instructions and exit codes.

---

**7. Conclusion**

This project successfully implemented a **two-layer security model** for accessing Amazon S3. By combining **Linux file permissions** with **IAM policies**, access control was enforced both locally and in the cloud. The approach ensures stronger security for sensitive data in multi-user environments.

---

**8. Future Enhancements**

- Add logging to the script for better auditing.
- Integrate MFA (Multi-Factor Authentication) for IAM users.
- Extend script to support bucket creation and deletion.