1. Access Control List (ACL) Configuration

Objective:

To enforce policies on network traffic by controlling access to specific network resources based on IP address, subnet, or protocol, thereby restricting unauthorized access to systems and services.

ACL Configuration:

- **Location:** Building #1 Network
- **Access Control Measure:** ACL to limit access to critical servers for specific groups (e.g., developers).
- **Purpose:** Ensure that developers have read-only access to designated servers, while other groups (e.g., IT) have full administrative access as necessary.

ACL Configuration Details:

- The ACL will be used to allow or deny network access based on IP addresses of different groups.
- **Example ACL for Developers:**
    - **Rule:** Allow developers to access certain servers with read-only permissions.
    - **Action:** Deny all access for non-developers.

Example ACL Configuration (simplified):

bash

Copy

```
access-list 100 permit ip 192.168.10.0 0.0.0.255 host 192.168.1.10 (Server A)
access-list 100 permit ip 192.168.10.0 0.0.0.255 host 192.168.1.20 (Server B)
access-list 100 deny ip any any
```

Explanation:

- The first two rules allow access from the **developers group** (subnet 192.168.10.0/24) to specific servers (Server A and Server B).
- The final rule denies access from all other networks to these servers.

Effect:

- The ACL ensures that only devices within the developer subnet (192.168.10.0/24) can access the servers with read-only permissions. Any other attempts are blocked by the deny rule.

Example Detected Event:

- **Event:** A device outside the developer subnet tries to access Server A.
- **Detection:** The ACL blocks the access attempt, and the firewall logs the event.
- **Response:** The unauthorized device is denied access, preventing potential security breaches.

2. Access Control Model: Mandatory Access Control (MAC)

Objective:

To implement a strict access control policy where users and systems are granted access based on a predefined security policy, rather than individual user permissions.

Access Control Model Chosen: MAC (Mandatory Access Control)

- **Location:** Building #1 Network
- **Policy:** Access permissions are controlled by the system, based on security labels assigned to both users and resources (e.g., files, servers).
- **Purpose:** The system uses security labels (e.g., sensitivity levels such as Confidential, Secret, Top Secret) to control access to resources, ensuring that access decisions are based on policy and not on user discretion.

MAC Configuration:

- **Security Labels:**
  - **Developers:** Assigned to a "Confidential" label.
  - **IT:** Assigned to a "Top Secret" label (for higher privileges and access).
- **Access Control Mechanism:** The MAC model enforces restrictions such that users can only access resources for which they have the appropriate clearance level.

Example MAC Configuration:

bash

Copy

```
setfattr -n user.security_level -v Confidential /home/developers
setfattr -n user.security_level -v TopSecret /home/it
```

Explanation:

- The developers' directory is labeled as "Confidential" and accessible only by users with that clearance level.
- IT staff have a higher clearance level ("Top Secret") and can access all resources, including those labeled as "Confidential."

Effect:

- The MAC model ensures that access is granted based on the security classification of the resources and the user’s clearance level. Unauthorized users will be denied access to sensitive information.

Example Detected Event:

- **Event:** A developer attempts to access an IT resource labeled as "Top Secret."
- **Detection:** The system denies access based on the MAC policy since the developer has a lower clearance level ("Confidential").
- **Response:** The access attempt is logged, and the developer is informed that they do not have sufficient clearance.

3. User Access Level Configuration

Objective:

To assign specific access levels to different user groups, ensuring that users have the minimum necessary permissions to perform their tasks while preventing unauthorized access to sensitive resources.

User Access Levels:

- **Developers Group:**
  - **Access Level:** Read-only access by default.
  - **Resources:** Developers can access specific servers (e.g., Server A and Server B), but cannot modify any files or configurations.
  - **Security:** Users in this group are assigned read-only access to ensure that they can view resources, but cannot change them.
- **IT Group:**
  - **Access Level:** Full administrative access.
  - **Resources:** IT members can access all servers and resources, including the ability to make changes to configurations, install software, and perform other administrative tasks.
  - **Security:** IT users are required to authenticate using two factors: a password and a physical key (token-based authentication).

Access Level Example (Developers):

bash

Copy

```
chmod 444 /home/developers/*
```

- This command gives developers read-only access to files within the "developers" directory.

Access Level Example (IT):

- IT users have full access:

bash

Copy

```
chmod 777 /home/it/*
```

- This allows IT users full access to the "IT" directory.

Two-Factor Authentication for IT Group:

- **Password:** The IT user must first authenticate using a password.
- **Physical Key:** A physical key (e.g., USB token) is required for additional authentication.

Example User Access Level Configuration (IT):

bash

Copy

```
authconfig --enabletwofactor --enablepam
```

Effect:

- Developers can only view files and data, reducing the risk of accidental or intentional tampering.

- IT staff can modify configurations and perform necessary maintenance with full administrative rights.
- Two-factor authentication for IT staff increases security by ensuring that only authorized personnel can access critical systems.

Example Detected Event:

- **Event:** A developer tries to modify a configuration file on Server A.
- **Detection:** The file system denies write access to the file, as the developer has read-only access.
- **Response:** The action is logged, and the developer is informed that they do not have permission to modify the file.