1. Discretionary Access Control (DAC)

DAC est le modèle par défaut dans la plupart des systèmes UNIX/Linux, où les propriétaires des fichiers peuvent définir les permissions d'accès pour leurs fichiers.

Exemple avec DAC

1. **Création d'un fichier** :

```
touch monfichier.txt
```

2. **Vérification des permissions actuelles** :

```
ls -l monfichier.txt
```

Vous verrez quelque chose comme :

```
-rw-r--r-- 1 user user 0 Jun 20 12:34 monfichier.txt
```

3. **Modification des permissions** :

Pour changer les permissions et rendre le fichier exécutable uniquement par le propriétaire :

```
chmod 744 monfichier.txt
```

4. **Vérification des nouvelles permissions** :

```
ls -l monfichier.txt
```

Vous verrez :

```
-rwxr--r-- 1 user user 0 Jun 20 12:34 monfichier.txt
```

2. Mandatory Access Control (MAC)

Pour MAC, nous utiliserons SELinux (Security-Enhanced Linux) qui est une implémentation courante de MAC sur les systèmes Linux.

Exemple avec SELinux

1. **Vérification du statut de SELinux** :

```
sestatus
```

2. **Activation de SELinux (si nécessaire)** :

```
sudo setenforce 1
```

3. **Création d'un fichier dans un répertoire avec des politiques spécifiques** :

```
sudo touch /var/www/html/securefile
```

4. **Application d'un contexte de sécurité** :

```
sudo chcon -t httpd_sys_content_t /var/www/html/securefile
```

5. **Vérification du contexte de sécurité** :

```
ls -Z /var/www/html/securefile
```

Vous verrez quelque chose comme :

```
-rw-r--r--. root root unconfined_u:object_r:httpd_sys_content_t:s0
/var/www/html/securefile
```

3. Role-Based Access Control (RBAC)

Pour RBAC, nous utiliserons `sudo` pour attribuer des rôles spécifiques (permissions) à différents utilisateurs.

Exemple avec RBAC

1. **Édition du fichier sudoers** :

```
sudo visudo
```

2. **Ajout d'un rôle pour un utilisateur** :

Supposons que nous voulons donner à l'utilisateur `alice` la capacité d'exécuter toutes les commandes administratives :

```
alice ALL=(ALL) ALL
```

Pour donner un accès limité, par exemple, seulement pour redémarrer le service apache :

```
alice ALL= /bin/systemctl restart apache2
```

3. **Vérification du rôle** :

L'utilisateur `alice` peut maintenant utiliser `sudo` pour exécuter des commandes spécifiques :

```
sudo systemctl restart apache2
```

Conclusion

- **DAC** : Contrôle par le propriétaire des fichiers, permettant une gestion flexible mais parfois complexe à grande échelle.
- **MAC** : Contrôle centralisé et strict basé sur des politiques de sécurité prédéfinies, souvent utilisé dans des environnements nécessitant une sécurité élevée.
- **RBAC** : Contrôle basé sur les rôles attribués aux utilisateurs, simplifiant la gestion des permissions dans les grandes organisations.

Chacun de ces modèles a ses propres cas d'utilisation et avantages spécifiques, et le choix dépendra des exigences de sécurité et de gestion de votre organisation.