

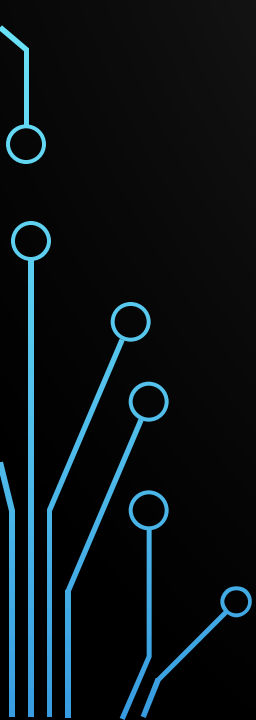
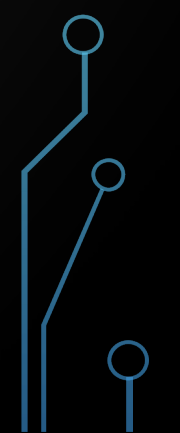
The image features a dark gray background with a central title. In the four corners, there are decorative elements consisting of light blue lines and small circles, resembling a circuit board or a network diagram. These elements are more prominent in the bottom-left and bottom-right corners.

INTEGRATION OF DAC AND RBAC POLICIES



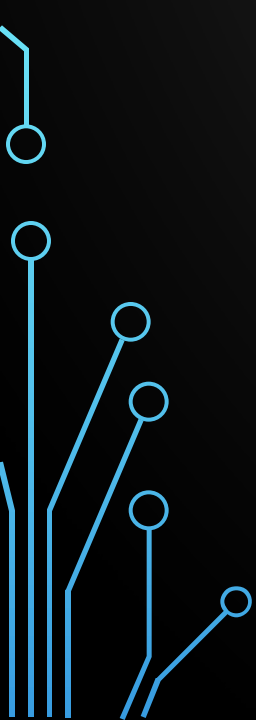

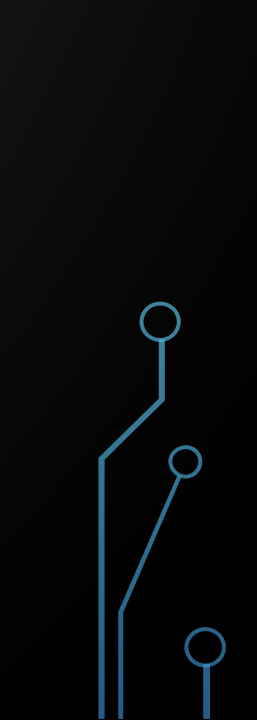
INTRODUCTION



- DAC stands for Discretionary Access Control. It allows the data owner to determine who has access to specific resources and what actions they can perform.
 - The DAC policy specifies the rules and permissions associated with accessing and manipulating data within a database. They are crucial for maintaining the confidentiality and integrity of data within a database.
 - RBAC stands for Role-Based Access Control. It is an access control mechanism that restricts system access to authorized users and assigns permissions based on the roles those users have.
 - A RBAC policy is a set of rules and guidelines that govern the access permissions granted to users or systems based on their roles within an organization or system.
 - It enhances security, helps with compliance, and provides a structured approach to managing access within an organization or system.
- 
- 



METHODOLOGY

- Creating the users, groups and assigning the users to groups using DAC policies.
 - Setting user and group permissions
 - Showing how April created a required document Reqs.pdf
 - Allowing the developers to review Reqs.pdf created by April
 - Implementing RBAC policies on users
 - Deleting the user 'Steven' from developer group by the administrator
- 
- 
- 

STEP 1: CREATING USERS

Creating the users by using 'adduser' command and protecting them with password

List of users: Susan

April

Carla

Steven

```
(kali㉿kali)-[~]
$ sudo adduser susan
info: Adding user `susan' ...
info: Selecting UID/GID from range 1000 to 59999 ...
info: Adding new group `susan' (1001) ...
info: Adding new user `susan' (1001) with group `susan (1001)' ...
info: Creating home directory `/home/susan' ...
info: Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for susan
Enter the new value, or press ENTER for the default
    Full Name []: Susan
    Room Number []: 1
    Work Phone []: 1
    Home Phone []: 1
    Other []: 1
Is the information correct? [Y/n] Y
info: Adding new user `susan' to supplemental / extra groups `users' ...
info: Adding user `susan' to group `users' ...

(kali㉿kali)-[~]
$ sudo adduser april
info: Adding user `april' ...
info: Selecting UID/GID from range 1000 to 59999 ...
info: Adding new group `april' (1002) ...
info: Adding new user `april' (1002) with group `april (1002)' ...
info: Creating home directory `/home/april' ...
info: Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for april
Enter the new value, or press ENTER for the default
    Full Name []: April
    Room Number []: 2
    Work Phone []: 2
    Home Phone []: 2
    Other []: 2
Is the information correct? [Y/n] Y
info: Adding new user `april' to supplemental / extra groups `users' ...
```

```
(kali㉿kali)-[~]
$ sudo adduser carla
info: Adding user `carla' ...
info: Selecting UID/GID from range 1000 to 59999 ...
info: Adding new group `carla' (1003) ...
info: Adding new user `carla' (1003) with group `carla (1003)' ...
info: Creating home directory `/home/carla' ...
info: Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for carla
Enter the new value, or press ENTER for the default
    Full Name []: Carla
    Room Number []: 3
    Work Phone []: 3
    Home Phone []: 3
    Other []: 3
Is the information correct? [Y/n] Y
info: Adding new user `carla' to supplemental / extra groups `users' ...
info: Adding user `carla' to group `users' ...

(kali㉿kali)-[~]
$ sudo adduser steven
info: Adding user `steven' ...
info: Selecting UID/GID from range 1000 to 59999 ...
info: Adding new group `steven' (1004) ...
info: Adding new user `steven' (1004) with group `steven (1004)' ...
info: Creating home directory `/home/steven' ...
info: Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for steven
Enter the new value, or press ENTER for the default
    Full Name []: Steven
    Room Number []: 4
    Work Phone []: 4
    Home Phone []: 4
    Other []: 4
Is the information correct? [Y/n] Y
info: Adding new user `steven' to supplemental / extra groups `users' ...
info: Adding user `steven' to group `users' ...
```


STEP 2: CREATING GROUPS

Creating the groups for different roles by using 'addgroup' command

List of groups: Administrator

Architect

Developer

```
(kali㉿kali)-[~]  
$ sudo addgroup administrator  
info: Selecting GID from range 1000 to 59999 ...  
info: Adding group `administrator' (GID 1005) ...  
  
(kali㉿kali)-[~]  
$ sudo addgroup architect  
info: Selecting GID from range 1000 to 59999 ...  
info: Adding group `architect' (GID 1006) ...  
  
(kali㉿kali)-[~]  
$ sudo addgroup developer  
info: Selecting GID from range 1000 to 59999 ...  
info: Adding group `developer' (GID 1007) ...
```

STEP 3: ASSIGNING USERS TO GROUPS

Assigning users to their respective groups using 'usermod' command.

Users and their groups: Susan: Administrator

April: Architect

Carla: Developer

Steven: Developer

```
(kali㉿kali)-[~]  
$ sudo usermod -aG administrator susan  
  
(kali㉿kali)-[~]  
$ sudo usermod -aG architect april  
  
(kali㉿kali)-[~]  
$ sudo usermod -aG developer carla  
  
(kali㉿kali)-[~]  
$ sudo usermod -aG developer steven
```

STEP 4: VERIFYING USER GROUP MEMBERSHIPS

Checking the group memberships using 'id' command.

```
(kali㉿kali)-[~]  
$ id susan  
uid=1001(susan) gid=1001(susan) groups=1001(susan),100(users),1005(administrator)  
  
(kali㉿kali)-[~]  
$ id april  
uid=1002(april) gid=1002(april) groups=1002(april),100(users),1006(architect)  
  
(kali㉿kali)-[~]  
$ id carla  
uid=1003(carla) gid=1003(carla) groups=1003(carla),100(users),1007(developer)  
  
(kali㉿kali)-[~]  
$ id steven  
uid=1004(steven) gid=1004(steven) groups=1004(steven),100(users),1007(developer)
```


STEP 5: SETTING USER AND GROUP PERMISSIONS

Assigning the architect group to april in its home directory and giving read, write and execute permissions using 'chown' and 'chmod' command respectively.

```
(kali㉿kali)-[~]  
$ sudo chown april:april /home/april  
  
(kali㉿kali)-[~]  
$ sudo chown april:architect /home/april  
  
(kali㉿kali)-[~]  
$ sudo chmod g+rx /home/april  
  
(kali㉿kali)-[~]  
$ ls -ld /home/april  
drwxrwx— 5 april architect 4096 Dec  1 18:22 /home/april
```

STEP 6: CREATING THE REQUIRED FILE

Creating Documents directory in the April user account and creating Reqs.pdf file in the April's document directory.

```
(kali㉿kali)-[~]  
$ sudo chown april:april /home/april  
  
(kali㉿kali)-[~]  
$ sudo chown april:architect /home/april  
  
(kali㉿kali)-[~]  
$ sudo chmod g+rx /home/april  
  
(kali㉿kali)-[~]  
$ ls -ld /home/april  
drwxrwx— 5 april architect 4096 Dec  1 18:22 /home/april
```

As April would like all developers to review the Reqs.pdf file we need to restrict the access permission to other groups

```
(april@kali)-[~]  
$ chown :developer Documents  
  
(april@kali)-[~]  
$ chmod 750 Documents  
  
(april@kali)-[~]  
$ ls  
Documents
```

```
(april@kali)-[~/Documents]  
$ chown :developer Reqs.pdf  
  
(april@kali)-[~/Documents]  
$ chmod 740 Reqs.pdf  
  
(april@kali)-[~/Documents]  
$ ls -l  
total 0  
-rwxr----- 1 april developer 0 Dec  1 19:12 Reqs.pdf
```

```
(april@kali)-[~/Documents]  
$ cat Reqs.pdf  
  
(april@kali)-[~/Documents]  
$ echo "Only developers can review it" >> Reqs.pdf
```

STEP 7: REVIEWING THE REQUIRED FILE BY DEVELOPERS

```
(april@kali)-[~/Documents]
$ su carla
Password:
(carla@kali)-[/home/april/Documents]
$ ls
Reqs.pdf

(carla@kali)-[/home/april/Documents]
$ cat Reqs.pdf
Only developers can review it

(carla@kali)-[/home/april/Documents]
$ echo "I am Carla" >> Reqs.pdf
bash: Reqs.pdf: Permission denied

(carla@kali)-[/home/april/Documents]
$ █
```

```
(april@kali)-[~/Documents]
$ su steven
Password:
(steven@kali)-[/home/april/Documents]
$ ls
Reqs.pdf

(steven@kali)-[/home/april/Documents]
$ cat Reqs.pdf
Only developers can review it

(steven@kali)-[/home/april/Documents]
$ echo "I am Steven" >> Reqs.pdf
bash: Reqs.pdf: Permission denied

(steven@kali)-[/home/april/Documents]
$ █
```

STEP 8: IMPLEMENTING RBAC POLICIES ON USERS

```
#Defaults:%sudo env_keep += "GPG_AGENT_INFO"

# Host alias specification

# User alias specification

# Cmnd alias specification

# User privilege specification
root    ALL=(ALL:ALL) ALL

# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL

# Administrator can run any command as any user
%administrator ALL=(ALL:ALL) ALL

# Architect can read and write to Reqs.pdf
%april ALL=(architect) /bin/cat /home/april/Documents/Reqs.pdf, /bin/bash -c 'echo > /home/april/Documents/Reqs.pdf'

# Developer can read Reqs.pdf
%carla ALL=(developer) /bin/cat /home/april/Documents/Reqs.pdf

# See sudoers(5) for more information on "@include" directives:

@includedir /etc/sudoers.d
```


STEP 9: DELETING THE USER

Deleting the user 'steven' from developer group by the administrator

```
(kali㉿kali)-[~]  
$ su susan  
Password:  
(susan㉿kali)-[/home/kali]  
$ cd  
  
(susan㉿kali)-[~]  
$  
  
(susan㉿kali)-[~]  
$ sudo deluser steven developer  
info: Removing user `steven' from group `developer' ...
```

STEP 10: CHECKING FOR THE ACCOUNT AND ACCESS IN DEVELOPERS

```
(kali㉿kali)-[~]  
$ su steven  
Password:  
(steven㉿kali)-[/home/kali]  
$ groups steven  
steven : steven users
```

```
(april㉿kali)-[~/Documents]  
$ su steven  
Password:  
(steven㉿kali)-[/home/april/Documents]  
$ ls  
ls: cannot open directory '.': Permission denied  
  
(steven㉿kali)-[/home/april/Documents]  
$ cat Reqs.pdf  
cat: Reqs.pdf: Permission denied  
  
(steven㉿kali)-[/home/april/Documents]  
$
```

STEP 11: CHECKING WHETHER NEW USERS ADDED TO DEVELOPERS GROUP CAN REVIEW THE FILE.

```
(kali㉿kali)-[~]
$ sudo adduser hasini
info: Adding user `hasini' ...
info: Selecting UID/GID from range 1000 to 59999 ...
info: Adding new group `hasini' (1008) ...
info: Adding new user `hasini' (1008) with group `hasini (1008)' ...
info: Creating home directory `/home/hasini' ...
info: Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for hasini
Enter the new value, or press ENTER for the default
  Full Name []: Hasini
   Room Number []: 5
  Work Phone []: 5
  Home Phone []: 5
    Other []: 5
Is the information correct? [Y/n] y
info: Adding new user `hasini' to supplemental / extra groups `users' ...
info: Adding user `hasini' to group `users' ...

(kali㉿kali)-[~]
$ sudo usermod -aG developer hasini
```

```
(april㉿kali)-[~/Documents]
$ su hasini
Password:
(hasini㉿kali)-[/home/april/Documents]
$ ls
Reqs.pdf

(hasini㉿kali)-[/home/april/Documents]
$ cat Reqs.pdf
Only developers can review it

(hasini㉿kali)-[/home/april/Documents]
$ echo "This is Hasini" >> Reqs.pdf
bash: Reqs.pdf: Permission denied

(hasini㉿kali)-[/home/april/Documents]
$
```

The image features a dark gray background with the text "Thank You..." centered in a light blue, sans-serif font. The corners of the image are decorated with stylized, light blue circuit-like lines and small circles, resembling a printed circuit board (PCB) layout. These decorative elements are located in the top-left, top-right, bottom-left, and bottom-right corners.

Thank You...