

# Esercizio\_S9\_L2\_Permessi\_Linux

## Consegna

Abbiamo visto come si gestiscono i permessi in Linux.

**Obiettivo:** Configurare e gestire i **permessi di lettura, scrittura ed esecuzione** per file o directory in un sistema Linux.

La scelta dei file o delle directory da configurare spetta allo studente. Infine, lo studente dovrà creare degli **screenshot** che mostrino i passaggi effettuati e scrivere una relazione spiegando le scelte fatte riguardo ai permessi.

Fornire:

- **Screenshot della Creazione del File o della Directory:**
  - *Fornisci uno screenshot che mostri i comandi utilizzati per creare il file o la directory.*
- **Screenshot della Verifica dei Permessi Attuali:**
  - *Fornisci uno screenshot che mostri i comandi `ls -l` e l'output prima della modifica dei permessi.*
- **Screenshot della Modifica dei Permessi:**
  - *Fornisci uno screenshot che mostri i comandi `chmod` utilizzati e l'output successivo con `ls -l`. Screenshot del*
- **Test dei Permessi:**
  - *Fornisci uno screenshot che mostri i tentativi di scrivere nel file o di creare un nuovo file nella directory, insieme ai comandi e agli output.*
- **Relazione:**
  - *Scrivi una relazione spiegando le scelte fatte riguardo ai permessi configurati. La relazione deve includere:*
  - *La motivazione delle scelte fatte per i permessi di lettura, scrittura ed esecuzione.*
  - *Un'analisi dei risultati ottenuti durante i test dei permessi.*

# Svolgimento

Per l'esecuzione di questo esercizio ho deciso di simulare alcune directory presenti all'interno di un'azienda fittizia chiamata **AziendaX** al cui interno vi fosse un reparto **amministrazione** presieduto da **luca**, un reparto **vendite** gestito da **mara** - ed una cartella condivisa tra i due reparti precedenti chiamata **Contacts**.

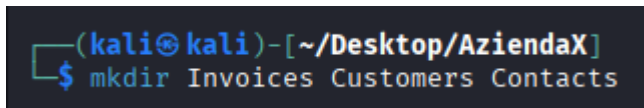
Ho quindi per prima cosa creato una cartella denominata come l'azienda all'interno del Desktop:

```
mkdir ~/Desktop/AziendaX
```

All'interno della cartella appena creata ho poi generato altre tre directories:

- Invoices
- Customers
- Contacts

```
cd AziendaX && mkdir Invoices Customers Contacts
```

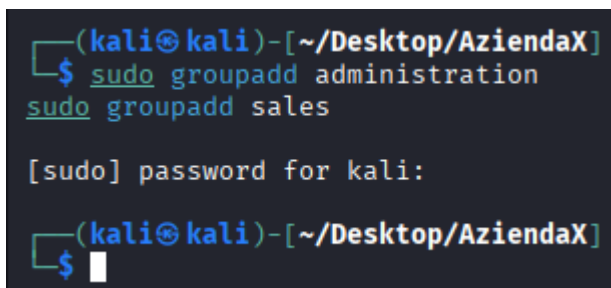


```
(kali㉿kali)-[~/Desktop/AziendaX]  
$ mkdir Invoices Customers Contacts
```

Una volta fatto ciò è stato il momento di creare dei gruppi di appartenenza per le rispettive directories:

- administration
- sales

```
sudo groupadd administration  
sudo groupadd sales
```



```
(kali㉿kali)-[~/Desktop/AziendaX]  
$ sudo groupadd administration  
sudo groupadd sales  
  
[sudo] password for kali:  
  
(kali㉿kali)-[~/Desktop/AziendaX]  
$
```

Ai due gruppi appena creati ho poi assegnato un utente ciascuno:

- **luca** → administration
- **mara** → sales

```
sudo useradd -m -s /bin/bash -G administration luca
sudo useradd -m -s /bin/bash -G sales mara
```

```
(kali㉿kali)-[~/Desktop/AziendaX]
$ sudo useradd -m -s /bin/bash -G administration luca
sudo useradd -m -s /bin/bash -G sales mara
```

*useradd* → crea un nuovo utente.

*-m* → crea la home directory dell'utente (es: /home/luca).

*-s /bin/bash* → imposta la shell predefinita (bash).

*-G <gruppo>* → aggiunge l'utente a un gruppo secondario (nel nostro caso *administration* o *sales*).

*luca / mara* → nome degli utenti.

Ho poi assegnato loro una password:

```
sudo passwd luca
sudo passwd mara
```

```
(kali㉿kali)-[~/Desktop/AziendaX]
$ sudo passwd luca
sudo passwd mara
New password:
Retype new password:
passwd: password updated successfully
New password:
Retype new password:
passwd: password updated successfully
```

Per convenienza e solo a scopo didattico ho impostato la psw uguale al nome utente.

Osserviamo ora i permessi delle cartelle create in precedenza:

```
ls -l
```

```
(kali㉿kali)-[~/Desktop/AziendaX]
$ ls -l
total 12
drwxrwxr-x 2 kali kali 4096 Sep 16 07:42 Contacts
drwxrwxr-x 2 kali kali 4096 Sep 16 07:42 Customers
drwxrwxr-x 2 kali kali 4096 Sep 16 07:42 Invoices
```

Notiamo che le cartelle sono in **possesso dell'utente kali** e che gli altri utenti hanno **solo il permesso di lettura e di apertura** delle varie cartelle ma non di scrittura; vanno dunque cambiati!

Ho poi quindi variato il **proprietario ed il gruppo di appartenenza** delle directories:

```
sudo chown luca:administration Invoices
```

```
sudo chown mara:sales Customers
```

A questo punto, per far sì che la cartella Contacts fosse accessibile da entrambi gli utenti è possibile procedere principalmente in due modi: il primo consiste semplicemente nel creare un gruppo comune al cui interno inserire luca e mara:

```
sudo groupadd shared
```

```
sudo usermod -aG shared luca
```

```
sudo usermod -aG shared mara
```

```
sudo chown root:shared Contacts
```

```
sudo chmod 770 Contacts
```

Il secondo metodo, che ora vedremo, consiste invece nell'utilizzare l'**ACL - Access Control List**.

Procediamo dunque assegnando a **root:administration** la proprietà della cartella **Contacts**:

```
sudo chown root:administration Contacts
```

Una volta fatto ciò possiamo procedere ad assegnare permessi custom sulla cartella ad altri gruppi:

```
sudo setfacl -m g:sales:rwx Contacts
```

*setfacl → serve a impostare le ACL (Access Control List).*

*-m = modify → per aggiungere o modificare una regola di ACL esistente.*

*g: → permessi a un gruppo.*

*sales → nome del gruppo.*

*rwx → i permessi da dare (read, write, execute).*

*Contacts → è la directory/oggetto su cui applichi la regola.*

Andando poi a visionare i permessi notiamo che le modifiche sono avvenute come da programma:

```
ls -l  
getfacl Contacts
```

```
(kali㉿kali)-[~/Desktop/AziendaX]  
$ sudo setfacl -d -m g:administration:rwx,g:sales:rwx,m:rwx Contacts  
  
(kali㉿kali)-[~/Desktop/AziendaX]  
$ getfacl Contacts  
# file: Contacts  
# owner: root  
# group: administration  
# flags: -s-  
user::rwx  
group::rwx  
group:sales:rwx  
mask::rwx  
other::—  
default:user::rwx  
default:group::rwx  
default:group:administration:rwx  
default:group:sales:rwx  
default:mask::rwx  
default:other::—
```

Giunti a questo punto potremmo decidere di escludere l'accesso alle varie cartelle da parte di chiunque non sia parte del gruppo proprietario:

```
chmod 770 Invoices Customers Contacts
```

```
(kali㉿kali)-[~/Desktop/AziendaX]  
$ sudo chmod 770 Contacts Customers Invoices  
[sudo] password for kali:  
  
(kali㉿kali)-[~/Desktop/AziendaX]  
$ ls -l  
total 12  
drwxrwx—+ 2 root administration 4096 Sep 16 07:42 Contacts  
drwxrwx— 2 mara sales 4096 Sep 16 07:42 Customers  
drwxrwx— 2 luca administration 4096 Sep 16 07:42 Invoices  
  
(kali㉿kali)-[~/Desktop/AziendaX]  
$
```

Per evitare che i file creati all'interno della cartella Contacts ricevano i permessi di gruppo dello user che li crea è necessario assicurarsi che essi mantengano quelli della cartella madre.

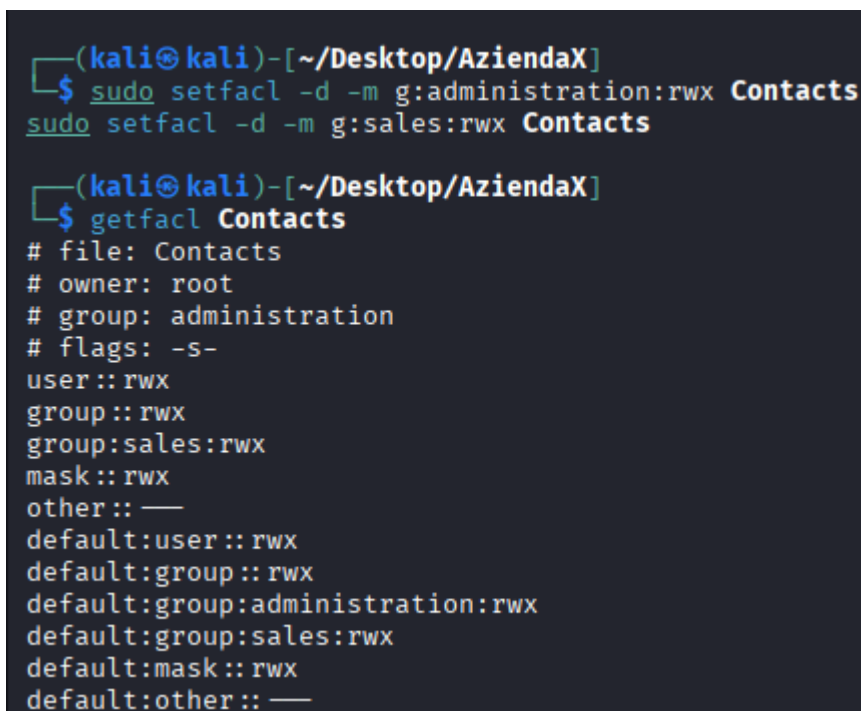
Per fare ciò ho utilizzato il comando:

```
sudo chmod g+s Contacts  
sudo chmod g+s Customers  
sudo chmod g+s Invoices
```

*g+s → “setgid bit” sulle directory: i nuovi file/cartelle creati dentro avranno come gruppo quello della directory (qui administration).*

Possiamo infine impostare i permessi **predefiniti** per nuovi file/cartelle creati all'interno

```
sudo setfacl -d -m g:administration:rwx Contacts  
sudo setfacl -d -m g:sales:rwx Contacts
```



```
(kali㉿kali)-[~/Desktop/AziendaX]  
$ sudo setfacl -d -m g:administration:rwx Contacts  
sudo setfacl -d -m g:sales:rwx Contacts  
  
(kali㉿kali)-[~/Desktop/AziendaX]  
$ getfacl Contacts  
# file: Contacts  
# owner: root  
# group: administration  
# flags: -s-  
user::rwx  
group::rwx  
group:sales:rwx  
mask::rwx  
other::—  
default:user::rwx  
default:group::rwx  
default:group:administration:rwx  
default:group:sales:rwx  
default:mask::rwx  
default:other::—
```

Possiamo ora notare che gli users dei gruppi sales ed administration hanno i permessi rwx anche di default sui file creati all'interno di **Contacts** mentre tutti gli altri utenti non hanno alcun accesso alla cartella.

Facciamo lo stesso anche per le altre due cartelle:

```
sudo setfacl -d -m g:administration:rwx,m:rwx Invoices  
sudo setfacl -d -m g:sales:rwx,m:rwx Customers
```

# Testing

Provando ora a creare un file all'interno della cartella **administration** tramite user luca noteremo che il file verrà correttamente creato ed erediterà i permessi ACL settati per il gruppo **administration**:

```
sudo -u luca touch Invoices/invoice1.txt && sudo -u luca ls -l Invoices
```

```
(kali㉿kali)-[~/Desktop/AziendaX]
$ sudo -u luca touch Invoices/invoice1.txt && sudo -u luca ls -l Invoices
total 0
-rw-rw----+ 1 luca administration 0 Sep 16 09:51 invoice1.txt
```

Al contrario, se provassimo a creare tramite luca, un file all'interno della cartella **Customers** non riusciremmo ad effettuare l'operazione in quanto la cartella è di proprietà di **sales** e luca non è parte di quel gruppo:

```
sudo -u luca touch Customers/list1.txt
```

```
(kali㉿kali)-[~/Desktop/AziendaX]
$ sudo -u luca touch Customers/list1.txt
touch: cannot touch 'Customers/list1.txt': Permission denied
```

Le stesse regole valgono ovviamente anche per **mara** ed i suoi rispettivi permessi:

```
(kali㉿kali)-[~/Desktop/AziendaX]
$ sudo -u mara touch Customers/list1.txt

(kali㉿kali)-[~/Desktop/AziendaX]
$ sudo -u mara touch Invoices/invoice2.txt
touch: cannot touch 'Invoices/invoice2.txt': Permission denied
```

Provando invece a scrivere un file all'interno di **Contacts** con luca e mara dovremmo essere in grado di operare all'interno con entrambi:

```
sudo -u luca touch Contacts/test1.txt && sudo -u mara touch Contacts/test2.txt
&& sudo -u mara ls -l Contacts
```

```
(kali㉿kali)-[~/Desktop/AziendaX]
$ sudo -u luca touch Contacts/test1.txt && sudo -u mara touch Contacts/test2.txt && sudo -u mara ls -l Contacts
total 0
-rw-rw----+ 1 luca administration 0 Sep 16 09:58 test1.txt
-rw-rw----+ 1 mara administration 0 Sep 16 09:58 test2.txt
```

Sia il file creato da luca che il file creato da mara sono presenti all'interno della cartella ed hanno i permessi che ci siamo prefissati di attribuirgli.

# Conclusioni

In questo esercizio ho modellato uno scenario aziendale realistico separando i dati per reparto (*Invoices* per *administration*, *Customers* per *sales*) e predisponendo un'area condivisa (*Contacts*) con regole chiare e sicure.

- **Isolamento per reparto:** tramite `chown` e `chmod 770` ho garantito che solo i membri del gruppo proprietario possano entrare e lavorare nelle cartelle di competenza. Questo rispetta il **principio del privilegio minimo**, riducendo il rischio di accessi o modifiche non autorizzate.
- **Condivisione controllata:** su *Contacts* ho usato le ACL per concedere diritti completi sia a *administration* che a *sales* senza aprire permessi agli *others*. Le ACL permettono di estendere l'accesso a più gruppi/utenti senza cambiare la proprietà della directory, risultando più flessibili rispetto alla sola gestione a gruppi.
- **Ereditarietà coerente:** con il bit SGID (`chmod g+s`) su tutte le directory e le ACL di default ho fatto sì che i nuovi file e sottocartelle ereditino automaticamente il gruppo della directory e i permessi attesi. Questo evita inconsistenze nel tempo e riduce interventi manuali.
- **Coerenza dei permessi effettivi:** mantenendo la mask ACL adeguata (dove necessario) ho impedito che i permessi concessi a gruppi/ACL venissero "tagliati". In questo modo i diritti visibili sono anche quelli realmente applicati.
- **Verifica pratica (PoC):** i test con `sudo -u <utente>` hanno confermato il comportamento desiderato:
  - **luca** può operare in *Invoices* ma riceve *Permission denied* in *Customers*;
  - **mara** può operare in *Customers* ma non in *Invoices*;
  - **entrambi** possono leggere/scrivere in *Contacts* grazie alle ACL.

Nel complesso, la combinazione di permessi POSIX (`chmod/chown`), ACL e SGID fornisce un controllo granulare e sostenibile nel tempo: le aree private restano protette, l'area condivisa funziona senza "scorciatoie" insicure e la manutenzione futura è semplificata grazie all'ereditarietà automatica.