

Permessi Linux

Lo scopo dell'esercitazione odierna era quello di prendere confidenza con i comandi Linux inerenti ai permessi dei file sul sistema.

Ogni file presente sul sistema Linux ha dei permessi, che a seconda dei privilegi utente (o se si è un superutente) che si possiedono, consentiranno di effettuare varie operazioni sui file stessi.

Principalmente esistono 3 tipi di permessi inerenti ai file nei sistemi Linux, ovvero:

- **Permessi di lettura; *r***
- **Permessi di scrittura; *w***
- **Permessi di esecuzione; *x***

Questi permessi sono visibili mediante il comando `ls -l`, oppure `ls -la` se si vogliono includere anche i file nascosti, e ci forniscono informazioni sulle operazioni consentite su un determinato file o directory.

Questi permessi sono divisi in **terzetti**, nello specifico in questo modo:



u g o
d r w x r w x

Ogni terzetto rappresenta una tipologia di utente; nello specifico, da sinistra verso destra avremo la **d**, che ci indicherà che si tratti di una **directory**, e subito dopo avremo i permessi effettivi.

- **Il primo terzetto** è quello relativo agli **User**, come l'utente del sistema può interagire con il file.
- **Il secondo terzetto** è inerente al **Gruppo**, le operazioni che utenti appartenenti ad un medesimo gruppo possono effettuare su quel file
- **Il terzo terzetto**, è invece relativo ad **Others**, come ad esempio i Guest.

Se quindi ad esempio avremo una situazione come **-rw-rw-r**—ciò indicherà anzitutto che si sta parlando di un **file**, poiché la **d** che indica la **directory non è presente**, mentre dai terzetti si evince che l'**User** e il **Gruppo** hanno permessi di **lettura** e **scrittura**, mentre gli **Others** avranno solo permessi di **lettura**.

Ogni permesso inoltre è identificato da un **numero**, cosa molto utile nel momento in cui si parla di cambio dei permessi, questo perché fungono da shortcut e ci consentono di cambiarli in modo più rapido ed efficace.

Per essere specifici, i numeri corrispondenti saranno:

r = 4 w = 2 x = 1

Per cui nel momento in cui utilizzeremo un comando **chmod +777 <file>** vuol dire che **attribuiremo tutti** i permessi (7 è la somma dei 3 valori) a quello specifico file.

Viceversa **chmod -777 <file>** verranno invece **rimossi tutti**.

Esempi di shortcut

Alcuni esempi possono essere:

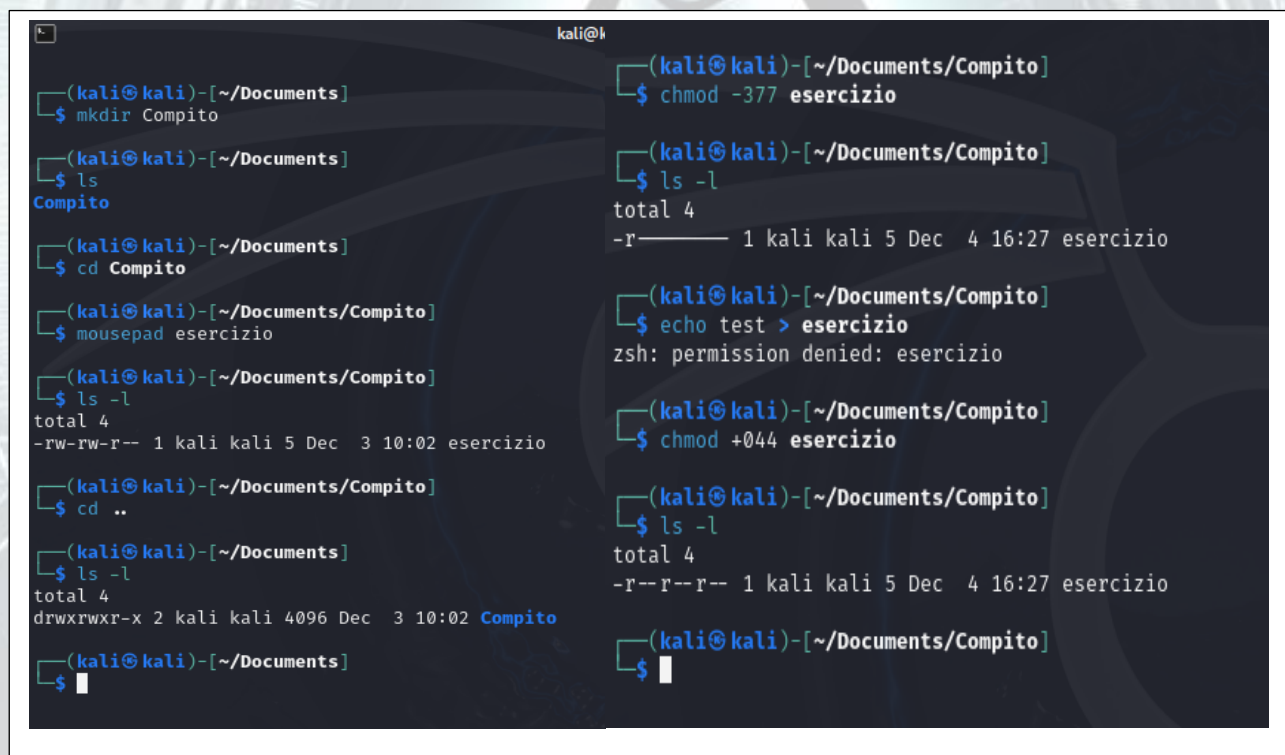
chmod -377 in cui rimuoviamo **wx** da User e **tutti gli altri permessi** a **Groups** e **Others**

chmod +644 in cui diamo i permessi di **rw** a **User** e soltanto i permessi di **r** a **G** e **O**

chmod +600 in cui si da **rw** a **User** e **nessun permesso** a **G** e **O**

Svolgimento dell'esercizio

Allo scopo di prendere confidenza con ciò che si è appreso riguardo i principali permessi **Linux**, è stata creata una directory con all'interno un file, del quale ne sono stati modificati i permessi.



```

kali@kali
(kali@kali)-[~/Documents]
$ mkdir Compito
(kali@kali)-[~/Documents]
$ ls
Compito
(kali@kali)-[~/Documents]
$ cd Compito
(kali@kali)-[~/Documents/Compito]
$ mousepad esercizio
(kali@kali)-[~/Documents/Compito]
$ ls -l
total 4
-rw-rw-r-- 1 kali kali 5 Dec  3 10:02 esercizio
(kali@kali)-[~/Documents/Compito]
$ cd ..
(kali@kali)-[~/Documents]
$ ls -l
total 4
drwxrwxr-x 2 kali kali 4096 Dec  3 10:02 Compito
(kali@kali)-[~/Documents/Compito]
$ chmod -377 esercizio
(kali@kali)-[~/Documents/Compito]
$ ls -l
total 4
-r----- 1 kali kali 5 Dec  4 16:27 esercizio
(kali@kali)-[~/Documents/Compito]
$ echo test > esercizio
zsh: permission denied: esercizio
(kali@kali)-[~/Documents/Compito]
$ chmod +044 esercizio
(kali@kali)-[~/Documents/Compito]
$ ls -l
total 4
-r--r--r-- 1 kali kali 5 Dec  4 16:27 esercizio
(kali@kali)-[~/Documents/Compito]
$
  
```

Come si evince dalle immagini è stato creato un file di testo avente permessi di **-rw** per **Utente** e **Gruppi** e soltanto di lettura **r**—per **Other**.

Allo scopo di esprimere il concetto al meglio, sono stati rimossi tutti i permessi al file, fuorché quello di lettura per l'utente. Dopo di che tramite il comando **echo**, ho provato a inserire testo all'interno di esso, ma ovviamente, ciò non è stato possibile, **permission denied**, poiché non possedevo i permessi di **scrittura**.

In seguito mediante il comando **chmod +044**, sono stati mantenuti invariati i permessi **Utente**, ma sono stati forniti i permessi di lettura anche a **G** ed **O**.

Considerazioni

È molto importante comprendere l'importanza ed il funzionamento per quanto riguarda i **permessi Linux**, questo perché a seconda dei dati che si andranno a maneggiare (file sensibili, directory etc.) ciò potrebbe essere **veramente cruciale**.

Esempi tangibili possono essere file importanti come nelle directory **/etc/** o **/bin/**.

Il file **shadow** ad esempio, nel quale vengono riportate le password inerenti ai vari gruppi e i vari utenti, **non può essere modificato in alcun modo da nessuna delle 3 utenze**, ma può essere letto.

Ciò perché un file di questo calibro, se modificato, potrebbe causare danni veramente impattanti sul sistema.

È quindi nota bene dare **permessi specifici a seconda dello scopo** che il file in questione avrà; se esso sarà un file con il quale il Gruppo dovrà interagire per poter effettuare operazioni, o comunque svolgere un lavoro, bisognerà fornire permessi come **rw-**, e in alcuni casi includere **x**, mentre forniremo soltanto i permessi **r** ad Others ad esempio.

L'unico che avrà il completo controllo e potere nel sistema sarà soltanto il **super admin, il root**, il quale avrà potere decisionale indiscusso su di essi.

LINUX

EQUILIBRY PRESENTATION OF RELATION