**S10-L2 Permessi di Linux**

**Esercizio di oggi:**

Gestione dei Permessi di Lettura, Scrittura ed Esecuzione in Linux 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.

**Relazione: Gestione dei Permessi di Lettura, Scrittura ed Esecuzione in Linux**

Capita spesso che un computer sia utilizzato da più utenti e quindi il sistema operativo deve mettere a disposizione un meccanismo per mantenere separati i dati di più utenti e fare in modo che i file privati di un utente non siano visibili agli altri. Linux identifica gli utenti come «numeri»: ciascun utente ha un proprio numero univoco detto «uid» e associa ad ogni uid una home directory. Configura poi i permessi (le autorizzazioni) sulle home directory relative allo specifico uid in modo tale che uid diversi da quello specificato non siano in grado di accedere.

Le informazioni riguardanti un utente sono contenute nel file /etc/passwd, che elenca tutti i nomi degli utenti validi con ulteriori informazioni, quali:

● ID dei gruppi di appartenenza

● ID dellʼutente

● Shell di login

● Password: in forma cifrata, utilizzando un meccanismo di hashing per camuffare la password in una stringa di testo incomprensibile.

Una caratteristica fondamentale del file system Linux sono i permessi, le autorizzazioni che vengono assegnate ai file. I file in Linux riportano le autorizzazioni per lʼutente corrente «u», il gruppo «g» e gli altri utenti «o» come nella figura a fianco.

In Linux ogni file ha 3 tipi di permessi:

● Permesso di lettura.

● Permesso di scrittura.

● Permesso di esecuzione.

Nei sistemi Linux i permessi sono riportati come flag dei file. Con il comando «ls –l» è possibile controllare i permessi sui file sia per lʼutente attuale, sia per il gruppo, sia per gli altri utenti, come riportato anche nella figura a destra. Il «-» è utilizzato in mancanza di quel particolare permesso. Per esempio, nella figura a destra, il gruppo e gli altri utenti non hanno il permesso «w» (che come vedremo a breve è il permesso per scrivere).

Il **Permesso di lettura** è indicato con la lettera «r», che sta per read. Eʼ il permesso che consente agli utenti di vedere il contenuto del file o di una directory.

Il **Permesso di scrittura** è indicato con la lettera «w», che sta per write. Eʼ il permesso che consente agli utenti di scrivere o modificare il contenuto di un file. Il permesso write consente anche di eliminare un file.

Il **Permesso di esecuzione** è indicato con la lettera «x», che sta per execute. Eʼ il permesso che consente agli utenti eseguire un file – in questo caso si parla di file eseguibili. Non si troverà mai un permesso «x» su un file di testo.

Può essere necessario talvolta cambiare i privilegi di un file, sia per lʼutente corrente che per il gruppo o gli altri utenti. Per farlo si può utilizzare il comando «chmod» da linea di comando con delle opzioni. Immaginiamo di avere un file con i permessi mostrati in figura a destra, dove abbiamo permessi di lettura per lʼutente corrente, lettura e scrittura per il gruppo e lettura per gli altri utenti. Obiettivo è rendere il file eseguibile (x) per lʼutente (u).

Il comando chmod da eseguire dovrà quindi includere le seguenti info:

● utente al quale assegnare i privilegi.

● tipo di privilegi aggiungere.

chmod u+x «nome\_file»

Dove «u» specifica utente corrente, e «x» il privilegio da aggiungere (+).

Allo stesso modo si utilizza il segno «-» per rimuovere il permesso ad una categoria tra utente (u), gruppo (g) ed altri utenti (o).

Immaginiamo di avere un file con i permessi mostrati in figura in alto a destra. Stavolta lʼobiettivo è rimuovere il permesso di scrittura dal gruppo.

Il comando chmod da eseguire sarà:

chmod g-w «nome\_file»,

in quanto vogliamo eliminare (-) il permesso write (w) dal gruppo (g).

Così come Linux permette di cambiare i permessi associati ad un determinato file, dà anche la possibilità di cambiarne il proprietario, sia per quanto riguarda lʼattributo utente, sia per lʼattributo gruppo. I comandi che vengono impiegati da riga di comando sono:

● Chown, change owner, permette di cambiare il proprietario del file (sia utente che gruppo).

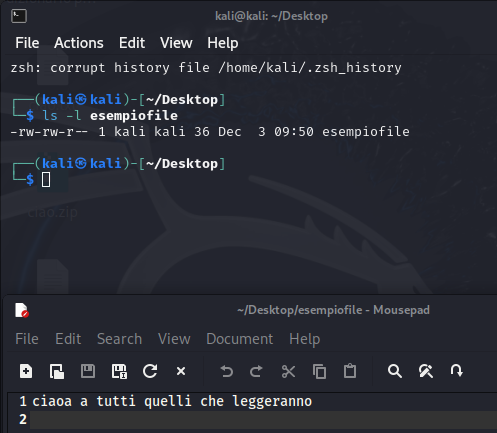
● Chmod, è un comando utilizzato in sistemi operativi di tipo Unix, inclusi Linux, per cambiare i permessi di accesso ai file o alle directory.

● Chgrp, change group, permette di cambiare il proprietario del file (solo gruppo).

**ESERCITAZIONE PRATICA**

1SCREENSHOT

Permessi della cartella prima della modifiche



SREENSHOT 2

Può essere necessario talvolta cambiare i privilegi di un file, sia per lʼutente corrente che per il gruppo o gli altri utenti. Per farlo si può utilizzare il comando «chmod» da linea di comando con delle opzioni. Immaginiamo di avere un file con i permessi mostrati in figura a destra, dove abbiamo permessi di lettura per lʼutente corrente, lettura e scrittura per il gruppo e lettura per gli altri utenti.

Obiettivo è rendere il file eseguibile (x) per lʼutente (u). Il comando chmod da eseguire dovrà quindi includere le seguenti info:

● utente al quale assegnare i privilegi.

● tipo di privilegi aggiungere. chmod u+x «nome\_file»

Dove «u» specifica utente corrente, e «x» il privilegio da aggiungere (+).



SREENSHOT 3

Con il comando

Ls -L nomefile

Possiamo vedere lo stato attuale e possiamo vedere sono stati modificati

