

Access Control

Lorenzo Colombi e Lucia Ferrari

Dipartimento di Ingegneria
Università di Ferrara

28 febbraio 2025

1 Utenti e gruppi

2 File

3 Processi

1 Utenti e gruppi

2 File

3 Processi

Lista degli utenti nel file /etc/passwd

```
$ cat /etc/passwd | grep fglmtt  
fglmtt:x:15875:15875:Fogli Mattia:/home/fglmtt:/bin/bash
```

```
$ cat /etc/passwd | grep fglmtt  
fglmtt:x:15875:15875:Fogli Mattia:/home/fglmtt:/bin/bash
```

/etc/passwd non è l'unico posto in cui vengono definiti gli utenti. Per avere una lista completa di tutti gli utenti conosciuti da un certo host

```
$ getent passwd
```

```
$ cat /etc/passwd | grep fglmtt
```

```
fglmtt:x:15875:15875:Fogli Mattia:/home/fglmtt:/bin/bash
```

- username
- password
- user id
- group id
- comment field
- working directory (home)
- initial shell

La "x" indica che la password si trova nel file /etc/shadow. Altrimenti la password criptata sarebbe visibile.

```
$ cat /etc/passwd | grep fglmtt  
fglmtt:x:15875:15875:Fogli Mattia:/home/fglmtt:/bin/bash
```

I sistemi operativi moderni (UNIX-like) hanno spostato le password cifrate in /etc/shadow per evitare che chiunque possa vedere l'hash delle password degli utenti. Nota che

```
$ ls -l /etc/passwd  
-rw-r--r-- 1 root root 2250 Dec 19 21:44 /etc/passwd  
  
$ ls -l /etc/shadow  
-rw-r----- 1 root shadow 1476 Dec 19 21:44 /etc/shadow
```

```
$ cat /etc/passwd | grep fglmtt  
fglmtt:x:15875:15875:Fogli Mattia:/home/fglmtt:/bin/bash
```

Per informazione su user/group id e gruppi senza dover consultare /etc/passwd

```
$ id  
uid=15875(fglmtt) gid=15875 groups=15875
```


Testo

- 1 Eseguire in modalità interattiva un container con Ubuntu
- 2 Verificare il proprio user/group id
- 3 Creare due nuovi utenti `alice` e `bob` con relativa home directory
- 4 Assegnare una password a ciascun utente
- 5 Passare da un utente all'altro

Esercizio 1

Suggerimenti

❶ `$ docker run -dit --name myFirstUbuntu ubuntu bash`

Esercizio 1

Suggerimenti

- 1 `$ docker run -dit --name myFirstUbuntu ubuntu bash`
- 2 `$ docker attach myFirstUbuntu ubuntu`

Suggerimenti

- 1 `$ docker run -dit --name myFirstUbuntu ubuntu bash`
- 2 `$ docker attach myFirstUbuntu ubuntu`
- 3 `$ id`

Esercizio 1

Suggerimenti

- ❶ `$ docker run -dit --name myFirstUbuntu ubuntu bash`
- ❷ `$ docker attach myFirstUbuntu ubuntu`
- ❸ `$ id`
- ❹ `$ useradd --create-home <username>`

Esercizio 1

Suggerimenti

- 1 `$ docker run -dit --name myFirstUbuntu ubuntu bash`
- 2 `$ docker attach myFirstUbuntu ubuntu`
- 3 `$ id`
- 4 `$ useradd --create-home <username>`
- 5 `$ passwd <username>`

Esercizio 1

Suggerimenti

- 1 \$ docker run -dit --name myFirstUbuntu ubuntu bash
- 2 \$ docker attach myFirstUbuntu ubuntu
- 3 \$ id
- 4 \$ useradd --create-home <username>
- 5 \$ passwd <username>
- 6 \$ su - <username>

```
$ cat /etc/passwd | grep -e alice -e bob  
alice:x:1000:1000::/home/alice:/bin/sh  
bob:x:1001:1001::/home/bob:/bin/sh
```



```
$ cat /etc/shadow | grep -e alice -e bob  
alice:$y$j9T$yevkBsuUAIfxFHZ7oQF8G1$HJmqWKjivjzy8Q7/otlCF6ccuzd7tCPtLICyyYd9Rj3:19111:0:99999:7:::  
bob:$y$j9T$3hGSsU9QYXrx.3IgsKN7S/$TZLP01mvSvdKM79UC7w6Bz4Mow4Ab.Je/xjLF7awDc2:19111:0:99999:7:::
```

```
$ cat /etc/group | grep -e alice -e bob  
alice:x:1000:  
bob:x:1001:
```

```
$ ls -l /home  
drwxr-x--- 2 alice alice 4096 Apr 29 07:38 alice  
drwxr-x--- 2 bob bob 4096 Apr 29 07:38 bob
```

```
$ su - alice  
$ cd /home/bob  
-sh: 2: cd: can't cd to /home/bob
```

Testo

- 1 Impostare una password per il gruppo bob
- 2 Impersonare l'utente alice
- 3 Impostare bob come group id
- 4 Verificare il proprio user/group id
- 5 Entrare nella home directory dell'utente bob (/home/bob)

Suggerimenti

① \$ gpasswd <groupname>

Suggerimenti

- 1 `$ gpasswd <groupname>`
- 2 `$ su - <username>`

Suggerimenti

- 1 `$ gpasswd <groupname>`
- 2 `$ su - <username>`
- 3 `$ sg <groupname>`

Suggerimenti

- 1 `$ gpasswd <groupname>`
- 2 `$ su - <username>`
- 3 `$ sg <groupname>`
- 4 `$ id`

Suggerimenti

- 1 `$ gpasswd <groupname>`
- 2 `$ su - <username>`
- 3 `$ sg <groupname>`
- 4 `$ id`
- 5 `$ cd <path/to/dir>`

1 Utenti e gruppi

2 File

3 Processi

Testo

- 1 Impersonare l'utente bob
- 2 Creare la directory `mydir` in `/home/bob/`
- 3 Creare il file `only-bob-can-delete-this.txt` in `/home/bob/mydir`
- 4 Togliere tutti i permessi al gruppo per `only-bob-can-delete-this.txt`
- 5 Impersonare l'utente alice
- 6 Impostare bob come group id
- 7 alice può cancellare `/home/bob/mydir/only-bob-can-delete-this.txt`?
- 8 se sì, limitare i permessi del gruppo bob di conseguenza

Suggerimenti

① \$ su - <username>

Esercizio 3

Suggerimenti

- 1 \$ su - <username>
- 2 \$ mkdir <path/to/dir>

Esercizio 3

Suggerimenti

- 1 \$ su - <username>
- 2 \$ mkdir <path/to/dir>
- 3 \$ touch <path/to/file>

Esercizio 3

Suggerimenti

- 1 \$ su - <username>
- 2 \$ mkdir <path/to/dir>
- 3 \$ touch <path/to/file>
- 4 \$ chmod g-rwx <path/to/file>

Suggerimenti

- 1 `$ su - <username>`
- 2 `$ mkdir <path/to/dir>`
- 3 `$ touch <path/to/file>`
- 4 `$ chmod g-rwx <path/to/file>`
- 5 `$ su - <username>`

Suggerimenti

- 1 `$ su - <username>`
- 2 `$ mkdir <path/to/dir>`
- 3 `$ touch <path/to/file>`
- 4 `$ chmod g-rwx <path/to/file>`
- 5 `$ su - <username>`
- 6 `$ sg <group>`
- 7 `$ rm <path/to/file>`

Suggerimenti

- ❶ `$ su - <username>`
- ❷ `$ mkdir <path/to/dir>`
- ❸ `$ touch <path/to/file>`
- ❹ `$ chmod g-rwx <path/to/file>`
- ❺ `$ su - <username>`
- ❻ `$ sg <group>`
- ❼ `$ rm <path/to/file>`
- ❽ **Per cancellare un file c'è bisogno dei permessi di scrittura ed esecuzione per la directory che contiene quel file, ma non c'è bisogno dei permessi di lettura e scrittura per il file stesso**

Suggerimenti

- ❶ `$ su - <username>`
- ❷ `$ mkdir <path/to/dir>`
- ❸ `$ touch <path/to/file>`
- ❹ `$ chmod g-rwx <path/to/file>`
- ❺ `$ su - <username>`
- ❻ `$ sg <group>`
- ❼ `$ rm <path/to/file>`
- ❽ **Per cancellare un file c'è bisogno dei permessi di scrittura ed esecuzione per la directory che contiene quel file, ma non c'è bisogno dei permessi di lettura e scrittura per il file stesso**
- ❾ `$ chmod g-wx <path/to/dir>`

Testo

- 1 Impersonare l'utente bob
- 2 Verificare i permessi di `/var/tmp`
- 3 Creare il file `bob.log` in `/var/tmp`
- 4 Impersonare l'utente alice
- 5 Cancellare il file `/var/tmp/bob.log`
- 6 Perchè alice non può cancellare `/var/tmp/bob.log` nonostante other abbia sia i permessi di scrittura che di esecuzione per `/var/tmp`?

Esercizio 4

Suggerimenti

① \$ su - <username>

Esercizio 4

Suggerimenti

- 1 `$ su - <username>`
- 2 `$ ls -l <path/to/dir>`

Esercizio 4

Suggerimenti

- 1 `$ su - <username>`
- 2 `$ ls -l <path/to/dir>`
- 3 `$ touch <path/to/file>`

Suggerimenti

- 1 `$ su - <username>`
- 2 `$ ls -l <path/to/dir>`
- 3 `$ touch <path/to/file>`
- 4 `$ su - <username>`

Suggerimenti

- 1 `$ su - <username>`
- 2 `$ ls -l <path/to/dir>`
- 3 `$ touch <path/to/file>`
- 4 `$ su - <username>`
- 5 `$ rm <path/to/file>`

Suggerimenti

- 1 `$ su - <username>`
- 2 `$ ls -l <path/to/dir>`
- 3 `$ touch <path/to/file>`
- 4 `$ su - <username>`
- 5 `$ rm <path/to/file>`
- 6 Perchè è impostato lo **sticky bit**. Quando lo sticky bit è impostato per una directory i file in quella directory possono essere rimossi o rinominati solo se lo user ha i permessi di scrittura e o possiede il file o possiede la directory o è superuser

1 Utenti e gruppi

2 File

3 Processi

Testo

- ① Impersonare l'utente alice
- ② Creare il file `/home/alice/i-am-good.sh` come segue

```
#!/bin/sh  
/usr/bin/env echo "I'am good"
```

- ③ Creare il file `/home/alice/echo` come segue

```
#!/bin/sh  
/usr/bin/echo "I'm malicious!"
```

- ④ Fare in modo che `/home/alice/i-am-good.sh` esegua `/home/alice/echo` anziché la versione "vera" di echo

Esercizio 5

Suggerimenti

```
❶ $ su - <username>
```

ATTENZIONE

Potrebbe essere necessario installare Vim o Nano!

Esercizio 5

Suggerimenti

- 1 `$ su - <username>`
- 2 `vim/nano AND $ chmod +x <filename>`

ATTENZIONE

Potrebbe essere necessario installare Vim o Nano!

Suggerimenti

- ❶ `$ su - <username>`
- ❷ `vim/nano AND $ chmod +x <filename>`
- ❸ `vim/nano AND $ chmod +x <filename>`

ATTENZIONE

Potrebbe essere necessario installare Vim o Nano!

Suggerimenti

- 1 `$ su - <username>`
- 2 `vim/nano AND $ chmod +x <filename>`
- 3 `vim/nano AND $ chmod +x <filename>`
- 4 `$ export PATH=<path/to/malicious-echo>:$PATH`

ATTENZIONE

Potrebbe essere necessario installare Vim o Nano!

Esercizio 6 + challenge sulla piattaforma

Testo

Le ultime 3 challenge sono gentilmente prese in prestito dalla Ca' Foscari:

<https://secgroup.dais.unive.it/teaching/system-security/lab-unix-linux-access-control/>