Password in Linux

Index

Introduction

Linux usa due file per gestire utenti e le relative password

- `/etc/passwd`
- `/etc/shadow`
  Entrambi sono normali file di testo con una sintassi similare, ma hanno funzioni e permessi diversi.

Originariamente esisteva soltanto il file `passwd`, che includeva le password dell'utente in plaintext; nell'implementazione attuale, per ogni riga (utente) in `passwd`, esiste una corrispondente riga in `shadow` che indica la sua password

---

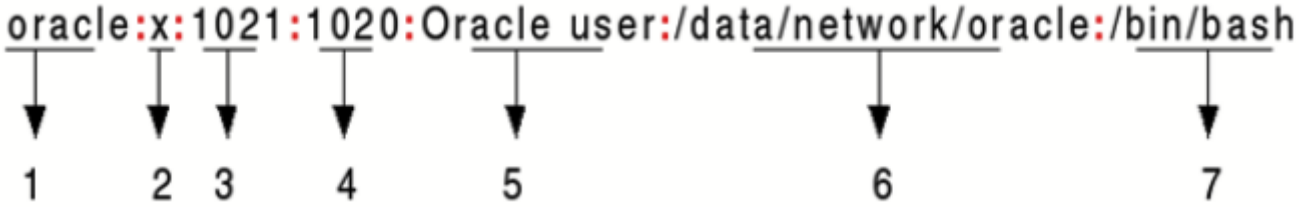
`/etc/passwd`

E' un plaintext file contenente l'intera lista di utenti (account) presenti nel sistema (include non solo gli utenti "normali", ma anche utenti standard di sistema e utenti speciali)

Di default ha i seguenti permessi:

```
-rw-r--r-- 1 root root 2659 Dec 22 12:21 /etc/passwd
```

Ciascuna riga del file `passwd` indica informazioni fondamentali su un utente del sistema e ha il seguente formato



1. username → nome dell'utente usato per il login (alfanumerico max. 32 caratteri)
2. password → inutilizzato oggi (il carattere `x` indica che l'hash della password è nel file `shadow`)
3. user id (uid) → ogni utente nel sistema ha uno user id numerico assegnato (`0` indica sempre l'utente root mentre `1–999` sono riservati per account predefiniti e di sistema)
4. group id (gid) → una volta creato ogni utente è assegnato ad un *primary group* ovvero il gruppo che il sistema operativo assegna ai file creati dall'utente (la descrizione del gruppo e i suoi dettagli sono contenuti in `/etc/group`)
5. GECOS → campo descrittivo che contiene informazioni generali sull'utente
6. home directory → path assoluto alla home directory dell'utente
7. shell → path assoluto della command shell usata dall'utente (eseguibile)

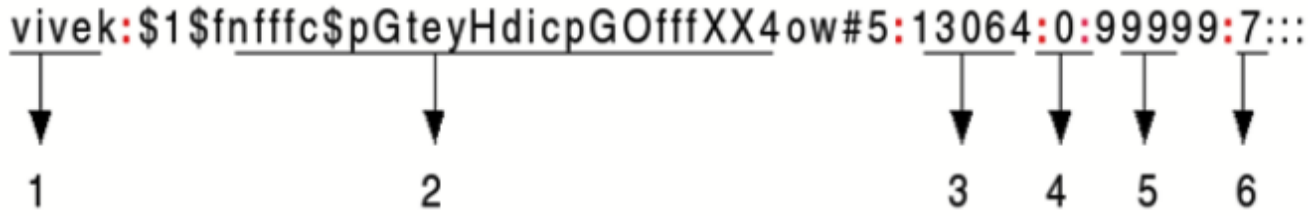
`/etc/shadow`

E' un plaintext file contenente, per ciascun utente del sistema, l'hash della sua password ed altre informazioni aggiuntive. Data la criticità delle informazioni contenute, sottrarre e decifrare lo shadow file spesso è uno degli obiettivi principali di un attaccante; conseguentemente ha permessi molto più restrittivi di `passwd`

Di default, ha i seguenti permessi

```
-rw-r----- 1 root root 2659 Dec 22 12:21 /etc/shadow
```

Ciascuna riga del file shadow contiene informazioni sulla password del rispettivo utente:



1. username → nome dell'utente a cui la password appartiene (definito in `passwd`)
2. password → password dell'utente salvata usando il **Modular Crypt Format**
3. last changed → data dell'ultimo cambiamento della password, espresso in giorni trascorsi da Unix Epoch (01/01/1970)
4. min age → minimo numero di giorni dall'ultimo cambio prima che la password possa essere nuovamente cambiata
5. max age → massimo numero di giorni dopo i quali è necessario cambiare la password
6. warn → quantigiorni prima della scadenza della password va avvisato l'utente

---

Modular Crypt Format

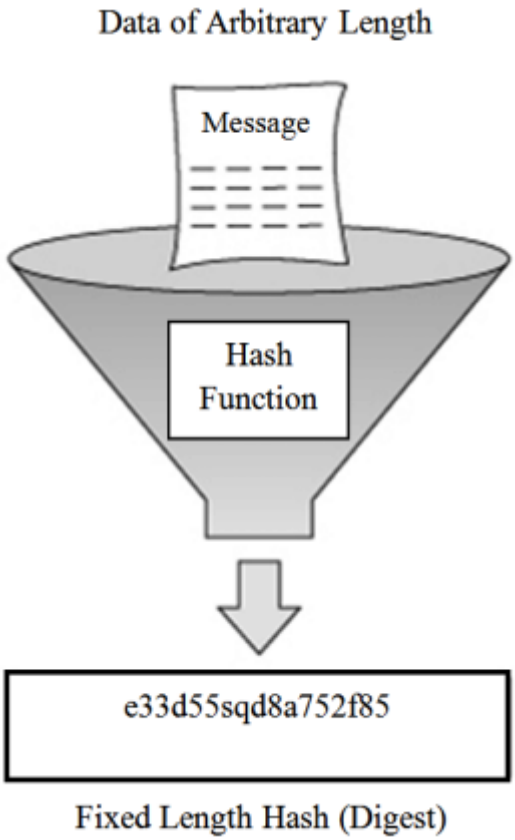
Il Modular Crypt Format è il formato usato usato nello shadow file per salvare gli hash delle password ed è strutturato così

```
$ID$salt$hash
```

- `ID` → algoritmo di hashing usato per questa password (MD5, blowfish, SHA265, ...)
- `salt` → salt usato nel processo di hashing; si tratta di una stringa randomica utilizzata nel processo di hashing per poter garantire unicità dell'hash (facendo l'hash di due password identiche si devono avere due hash diversi)
- `hash` → hash della password, calcolato con l'algoritmo ID e salt

---

Hash functions



Le funzioni di hash sono delle funzioni matematiche che permettono di trasformare sequenze di dati di lunghezza variabile in output di lunghezza fissa in maniera deterministica (ovvero data la stessa stringa in input, darà sempre lo stesso output)

Inoltre una funzione di hash è detta **crittografica** se è computazionalmente difficile:

- calcolare l'inverso della funzione hash
- dato un input x ed il suo hash d, trovare un altro input x_1 che abbia lo stesso hash d
- trovare due input diversi di lunghezza arbitraria x_1 e x_2 che abbiano lo stesso hash d

Password hashing o cirfratura

Date le caratteristiche elencate, capiamo perché le funzioni di hash sono usate per le password. Ma perché non cifrare direttamente la password? In entrambi i casi, le password non sono leggibili da un attaccante
Se si usa cifratura ed un attaccante ottiene la chiave privata, potrebbe decifrare ed ottenere tutte le password in plaintext. Inoltre le funzioni hash sono *one-way* (una volta fatto l'hash non si torna più indietro). Dunque se un attaccante ottiene l'hash non potrà mai scoprire la password che l'ha generato (salvo eccezioni), ma rimane comunque semplice verificare se una password corrisponde a quella salvata in formato hash,

basta infatti fare l’hash della password e verificarne l’equivalenza (hashing è deterministico)