Linux Shadow Password HOWTO

Michael H. Jackson `mhjack@tscnet.com`

v1.3, 3 Aprile 1996

Questo documento si propone di descrivere come ottenere, installare e configurare la Linux Password *Shadow Suite*. Esso spiega anche come ottenere e reinstallare altri software e demoni di rete che richiedono accesso alle password degli utenti. Questi ultimi software in realtà non fanno parte della *Shadow Suite*. Questo documento contiene inoltre un esempio di programmazione per aggiungere il supporto shadow ad un programma. Risposte ad alcune delle domande più frequenti sono incluse verso la fine di questo documento. Traduzione a cura di *Isabella Ruocco <isacher@nettaxi.com>* , ultima revisione 25 Maggio 1999.

Indice

1 Introduzione

Questo è il Linux Shadow Password HOWTO. Questo documento descrive perché e come aggiungere il supporto shadow password su un sistema Linux. Sono inclusi anche alcuni esempi su come usare alcune delle caratteristiche della *Shadow Suite*.

Quando si installa la *Shadow Suite* e quando si usano molti dei programmi di utilità, occorre essere collegati come *root*. Quando si installa la *Shadow Suite* verranno effettuati dei cambiamenti al software di sistema, ed è fortemente consigliato fare copie di backup dei programmi come indicato. Consiglio anche di leggere e comprendere tutte le istruzioni prima di iniziare.

1.1 Cambiamenti dalla versione precedente

```
Aggiunte:
   Aggiunta una sotto-sezione sul perche potreste non voler installare le shadow
   Aggiunta una sotto-sezione sull'aggiornamento del programma xdm
   Aggiunta una sezione su come far funzionare le caratteristiche della Shadow Suite
   Aggiunta una sezione contenente le domande piu frequenti (FAQ)

Correzioni/Aggiornamenti:
   Correzione dei riferimenti html sul sunsite
   Correzione della sezione su wu-ftp in modo che tenga conto dell'aggiunta di -lshadow al Makefile
   Correzione di errori secondari di ortografia
   Cambiamento della sezione su wu-ftp in modo da supportare ELF
   Aggiornamenti sui problemi di sicurezza in diversi programmi di login
   Aggiornamenti sulla raccomandazione di Marek Michalkiewicz sulla Linux Shadow Suite
```

1.2 Nuove versioni di questo documento

La più recente versione di questo documento si può anche ottenere via FTP anonimo da:

sunsite.unc.edu

`/pub/Linux/docs/HOWTO/Shadow-Password-HOWTO`

oppure:

`/pub/Linux/docs/HOWTO/other-formats/Shadow-Password-HOWTO{-html.tar,ps,dvi}.gz`

o tramite il World Wide Web dal *Linux Documentation Project Web Server* `http://sunsite.unc.edu/mdw/linux.html` , alla pagina:

Shadow-Password-HOWTO `http://sunsite.unc.edu/linux/HOWTO/Shadow-Password-HOWTO.html`

o direttamente da me, <`mhjack@tscnet.com`>. Sarà anche inviata al newsgroup: `comp.os.linux.answers`

Questo documento è ora impacchettato con i pacchetti Shadow-AAMMGG.

1.3 Commenti e critiche

Per favore inviate qualunque commento, aggiornamento, o suggerimenti a me: *Michael H. Jackson <mhjack@tscnet.com>* . Prima ricevo feedback, prima posso aggiornare e correggere questo documento. Se avete dei problemi, per favore mandate una e-mail direttamente a me, dato che molto raramente rimango aggiornato con i newsgroup.

2 Perché oscurare il vostro file passwd?

Come impostazione predefinita, la maggior parte delle attuali distribuzioni Linux non contengono la *Shadow Suite* installata. Questo riguarda la Slackware 2.3, Slackware 3.0 ed altre famose distribuzioni. Una delle ragioni di questo è che le informazioni sul copyright nella *Shadow Suite* originale non spiegavano chiaramente se si dovesse versare una somma per la ridistribuzione. Linux usa un Copyright GNU (a cui talvolta si fa riferimento come Copyleft) che permette alle persone di impacchettarlo in un supporto conveniente (come una distribuzione su CD-ROM) e di venderlo.

L'attuale manutentore della *Shadow Suite*, *Marek Michalkiewicz <marekm@i17linuxb.ists.pwr.wroc.pl>* ha ricevuto il codice sorgente dall'autore originale con un copyright tipo BSD che permette la ridistribuzione. Ora che i problemi di copyright sono risolti, ci si aspetta che le future distribuzioni conterranno le shadow password come opzione predefinita. Fino ad allora, dovrete installarvele voi.

Se avete installato la vostra distribuzione da un CD-ROM, potrebbe succedere che, anche se la distribuzione non aveva la *Shadow Suite* installata, alcuni dei file che vi occorrono per installare la *Shadow Suite* siano sul CD-ROM.

Comunque, le versioni 3.3.1, 3.3.1-2 della Shadow Suite e la shadow-mk potrebbero avere problemi di sicurezza con il loro programma di login e molti altri programmi SUID root che si trovano insieme ad esse, e non dovrebbero essere più usate.

Tutti i file necessari si possono ottenere via FTP anonimo o tramite il World Wide Web.

Su un sistema Linux senza la *Shadow Suite* installata, le informazioni sugli utenti, comprese la password, sono contenute nel file `/etc/passwd`. La password viene conservata in un formato *criptato*. Se chiedete ad un esperto crittografo, comunque, lui o lei vi diranno che la password è in realtà in un formato *codificato* piuttosto che *criptato*, perché quando viene usato crypt(3), viene preso un testo vuoto e la password è usata come chiave. Perciò, da qui in poi, in questo documento, userò il termine *codificato*.

Tecnicamente ci si riferisce all'algoritmo usato per codificare il campo password come ad una *funzione hash monodirezionale*. Questa è un algoritmo che è facile eseguire in una direzione, ma molto difficile eseguire nella direzione opposta. Altre informazioni sull'algoritmo usato si possono trovare nel paragrafo 2.4 o nella vostra pagina di manuale per crypt(3).

Quando un utente sceglie o gli viene assegnata una password, questa viene codificata con un valore generato casualmente detto *seme* (salt). Questo significa che una certa password può essere memorizzata in 4096 modi diversi. Il valore del *seme* viene memorizzato insieme alla password codificata.

Quando un utente si collega e fornisce una password, prima viene prelevato il *seme* dalla password codificata in memoria. Poi la password digitata viene *codificata* con tale valore del *seme* e quindi confrontata con la password *codificata*. Se c'è corrispondenza l'utente viene autenticato.

È computazionalmente difficile (ma non impossibile), ricostruire la password originale dalla password *codificata* casualmente. Comunque, su ogni sistema con più di qualche utente, almeno alcune delle password saranno parole comuni (o semplici variazioni di parole comuni).

Gli scassinatori di sistemi informatici sono a conoscenza di questo, e semplicemente critteranno un vocabolario di parole e password comuni usando tutti i 4096 possibili valori di *seme*. Quindi confronteranno le password codificate nel vostro file `/etc/passwd` con il loro database. Una volta trovata una corrispondenza, avranno la password per un altro account. Questo viene chiamato *attacco a vocabolario* ed è uno dei metodi più comuni per ottenere o diffondere accessi non autorizzati ad un sistema.

Se ci pensate, una password di 8 caratteri codifica fino a 4096*13 stringhe di caratteri. Perciò un vocabolario di, diciamo, 400.000 parole comuni, nomi, password e semplici variazioni starà facilmente su un disco fisso da 4GB. Lo scassinatore dovrà solo ordinarle e cercare le corrispondenze. Poiché un disco fisso da 4GB si può avere a meno di $1000.00, è ampiamente nelle possibilità di molti scassinatori di sistemi informatici.

Inoltre, se uno scassinatore ottiene prima il file `/etc/passwd`, avrà bisogno solo di codificare il vocabolario con i valori del `seme` effettivamente contenuti nel vostro file `/etc/passwd`. Questo metodo può essere usato dall'adolescente medio con un paio di Megabyte liberi e un computer 486.

Anche senza molto spazio su disco, utility come crack(1) possono di solito corrompere almeno un paio di password su un sistema con un discreto numero di utenti (assumendo che gli utenti del sistema possano scegliersi le loro password).

Il file `/etc/passwd` contiene anche informazioni tipo gli user ID e i group ID che sono usati da molti programmi di sistema. Perciò il file `/etc/passwd` *deve* rimanere accessibile a tutti. Se voi cambiaste il file `/etc/passwd` in modo che nessuno possa leggerlo, la prima cosa che notereste sarebbe che il comando `ls -s` ora mostrerebbe gli user ID invece dei nomi!

La *Shadow Suite* risolve il problema spostando le password in un altro file (di solito `/etc/shadow`). Il file `/etc/shadow` viene impostato in modo che quasi nessuno possa leggerlo. Solo *root* potrà leggere e scrivere il file `/etc/shadow`. Alcuni programmi (come xlock) non devono poter cambiare le password, occorre solo che le possano verificare. Questi programmi possono essere eseguiti *SUID root* oppure si può creare un gruppo *shadow* a cui è permesso l'accesso solo in lettura al file `/etc/shadow`. Quindi i programmi possono essere eseguiti *SGID shadow*.

Spostando le password nel file `/etc/shadow`, stiamo effettivamente impedendo allo scassinatore di avere accesso alle password codificate con cui eseguire l'*attacco a vocabolario*.

Inoltre, la *Shadow Suite* aggiunge molte altre caratteristiche interessanti:

- un file di configurazione per impostare le caratteristiche predefinite di login (`/etc/login.defs`)
- utility per aggiungere, modificare e cancellare account di utenti e gruppi
- invecchiamento e scadenza delle password
- scadenza e blocco degli account
- password di gruppo oscurate (opzionale)
- password di doppia lunghezza (password da 16 caratteri) [SCONSIGLIATO]
- migliore controllo sulla scelta delle password degli utenti
- dial-up password
- programmi di autenticazione secondaria [SCONSIGLIATO]

Installare la *Shadow Suite* contribuisce alla sicurezza del sistema ma ci sono anche molte altre cose che si possono fare per migliorare la sicurezza di un sistema Linux, e ci saranno alcuni Linux Security HOWTO che discutono altre misure di sicurezza ed aspetti correlati.

Per informazioni aggiornate su altri aspetti della sicurezza in Linux, tra cui avvertimenti sulle debolezze note guardate la *Linux Security home page.* `http://bach.cis.temple.edu/linux/linux-security/`

2.1 Perchè potreste NON voler oscurare il vostro file passwd

Ci sono alcune circostanze e configurazioni in cui installare la *Shadow Suite NON* sarebbe una buona idea:

- la macchina non contiene account di utenti
- la vostra macchina funziona su una LAN e usa NIS (Network Information Services) per ottenere e fornire nomi e password degli utenti alle altre macchine sulla rete (in realtà si potrebbe fare, ma è oltre lo scopo di questo documento, e in realtà non aumenterebbe comunque molto la sicurezza)

- la vostra macchina viene usata dai server dei terminali per verificare gli utenti con NFS (Network File System), NIS, o qualche altro metodo
- la vostra macchina esegue altro codice per validare gli utenti, e non c'è nessuna versione shadow disponibile, e non avete il codice sorgente.

2.2 Formato del file /etc/passwd

Un file `/etc/passwd` non oscurato ha il seguente formato:

```
nomeutente:passwd:UID:GID:nome_completo:directory:shell
```

Dove:

`nomeutente`

Il nome (di login) dell'utente

`passwd`

La password codificata

`UID`

Identificativo numerico dell'utente

`GID`

Identificativo numerico predefinito del gruppo

`nome_completo`

Il nome completo dell'utente - in realtà questo campo viene chiamato campo GECOS (General Electric Comprehensive Operating System) e può contenere altre informazioni invece del solo nome completo. I comandi Shadow e le pagine di manuale si riferiscono a questo campo come al campo commento.

`directory`

Home directory dell'utente (percorso completo)

`shell`

Shell di login dell'utente (percorso completo)

Ad esempio:

```
nomeutente:Npge08pfz4wuk:503:100:Nome Completo:/home/nomeutente:/bin/sh
```

Dove `Np` è il seme e `ge08pfz4wuk` è la password *codificata*. La coppia seme/password codificata avrebbe anche potuto essere `kbeMVnZM0oL7I` e queste due sono esattamente la stessa password. Ci sono 4096 possibili codifiche per la stessa password (la password usata in questo esempio è 'password', una *pessima* password).

Una volta che la Shadow Suite è installata, il file `/etc/passwd` invece conterrà:

```
nomeutente:x:503:100:Nome Completo:/home/nomeutente:/bin/sh
```

La `x` nel secondo campo in questo caso è ora soltanto un segnaposto. Il formato del file `/etc/passwd` non è di fatto cambiato, solo che non contiene più le password *codificate*. Questo significa che qualunque programma che legge il file `/etc/passwd`, ma in realtà non ha bisogno di verificare le password, funzionerà ancora correttamente.

Le password sono ora situate nel file shadow (di solito il file `/etc/shadow`).

2.3 Formato del file shadow

Il file `/etc/shadow` contiene le seguenti informazioni:

```
nomeutente:passwd:ult:puo:deve:avv:scad:disab:riservato
```

Dove:

`nomeutente`

Il nome dell'utente

`passwd`

La password codificata

`ult`

Giorni dal 1 Gennaio 1970 fino all'ultima modifica della password

`può`

Giorni prima che la password possa essere cambiata

`deve`

Giorni dopo i quali la password deve essere cambiata

`avv`

Giorni prima della scadenza della password in cui l'utente viene avvisato

`scad`

Giorni dopo la scadenza della password in cui l'account viene disabilitato

`disab`

Giorni a partire dal 1 Gennaio 1970 dopo cui l'account verrà disabilitato

`riservato`

Campo riservato

Il precedente esempio potrebbe allora essere:

```
nomeutente:Npge08pfz4wuk:9479:0:10000::::
```

2.4 Uno sguardo a crypt(3)

Dalla pagina di manuale di crypt(3):

"*crypt* è la funzione di crittaggio delle password. Si basa sull'algoritmo *Data Encryption Standard* con variazioni aventi lo scopo (tra le altre cose) di scoraggiare l'uso di implementazioni hardware per la ricerca della chiave.

[La] chiave è la password digitata dall'utente. [La stringa codificata è tutta vuota.]

[Il] *seme* è una stringa di due caratteri scelta nell'insieme [a-z A-Z 0-9./]. Questa stringa viene usata per perturbare l'algoritmo in uno tra 4096 modi diversi.

Prendendo i 7 bit meno significativi di ogni carattere della chiave, si ottiene una chiave di 56 bit. Questa chiave di 56 bit viene usata per crittare ripetutamente una stringa costante (di solito una stringa costituita di zeri). Il valore restituito punta alla password crittata, un insieme di 13 caratteri ASCII stampabili (i primi due caratteri sono il seme stesso). Il valore di ritorno punta a dati statici il cui contenuto viene sovrascritto da ogni chiamata.

Attenzione: lo spazio chiave consiste di 2**56 cioè 7.2e16 possibili valori. **È possibile** effettuare ricerche esaustive di questo spazio chiave usando computer massivamente paralleli. È disponibile del software, come `crack(1)`, che cercherà la porzione di questo spazio chiave che viene generalmente usata dagli umani per le password. Perciò la scelta delle password dovrebbe, come minimo, evitare parole e nomi comuni. Si raccomanda l'uso di un programma `passwd(1)` che, durante il processo di selezione, controlla se le password sono vulnerabili a manomissioni.

Lo stesso algoritmo DES ha alcune arguzie che rendono l'uso dell'interfaccia `crypt(3)` una scelta inefficace per qualunque altra cosa che non sia l'autenticazione di password. Se state pensando di usare l'interfaccia `crypt(3)` per un progetto di crittografia non lo fate: prendete un buon libro sulla crittografia e una delle librerie DES ampiamente disponibili."

Molte *Shadow Suite* contengono codice per raddoppiare la lunghezza della password a 16 caratteri. Esperti in `DES` sconsigliano questo, dato che la codifica viene semplicemente applicata prima alla metà di sinistra e poi alla metà di destra della password allungata. A causa del modo in cui funziona `crypt`, la password codificata di lunghezza doppia potrebbe risultare addirittura *meno* sicura. Inoltre, è meno facile che un utente riesca a ricordare una password da 16 caratteri.

È in via di sviluppo un lavoro che permetterebbe all'algoritmo di autenticazione di essere sostituito con qualcosa di più sicuro, che supporti password più lunghe (in particolare l'algoritmo MD5) e mantenga compatibilità con il metodo `crypt`.

Se state cercando un buon libro sulla crittografia, vi consiglio:

```
"Applied Cryptography: Protocols, Algorithms, and Source Code in C"
di Bruce Schneier <schneier@chinet.com>
ISBN: 0-471-59756-2
```

3 Ottenere la Shadow Suite

3.1 Storia della Shadow Suite per Linux

NON USATE I PACCHETTI DI QUESTO CAPITOLO, HANNO PROBLEMI DI SICUREZZA

La *Shadow Suite* originale è stata scritta da `John F. Haugh II`.

Esistono diverse versioni che sono state usate su sistemi Linux:

- `shadow-3.3.1` è l'originale;
- `shadow-3.3.1-2` è la patch specifica per Linux fatta da *Florian La Roche <flla@stud.uni-sb.de>* e contiene alcuni ulteriori miglioramenti;
- `shadow-mk` è stata specificamente impacchettata per Linux.

Il pacchetto `shadow-mk` contiene il pacchetto `shadow-3.3.1` distribuito da `John F. Haugh II` con la `patch` `shadow-3.3.1-2` installata, alcune correzioni fatte da *Mohan Kokal <magnus@texas.net>*

che semplificano molto l'installazione, una patch di `Joseph R.M. Zbiciak` per `login1.c` (login.secure) che elimina i bachi di sicurezza -f, -h in /bin/login, e alcune altre patch di vario tipo.

Il pacchetto `shadow.mk` era il pacchetto *precedentemente* raccomandato, ma dovrebbe essere sostituito a causa di *problemi di sicurezza* con il programma di `login`.

Ci sono problemi di *sicurezza* con le versioni 3.3.1, 3.3.1-2 di Shadow, e con shadow-mk che coinvolgono il programma di `login`. Questo baco di `login` riguarda il mancato controllo di un nome di login. Questo provoca un overflow nel buffer, con conseguente crash o peggio. Si è diffusa la voce che questo overflow del buffer possa permettere a qualcuno con un account sul sistema di usare questo baco e le librerie condivise per ottenere l'accesso come *root*. Non discuterò esattamente come questo sia possibile, perché ci sono molti sistemi Linux che ne sono affetti, ma sistemi con queste *Shadow Suite* installate e la maggior parte delle distribuzioni pre-ELF *senza* la *Shadow Suite* sono vulnerabili!

Per avere maggiori informazioni su questo e altri aspetti della sicurezza su Linux, guardate la:

Linux Security home page (Shared Libraries and login Program Vulnerability) `http://bach.cis.temple.edu/linux/linux-security/Linux-Security-FAQ/Linux-telnetd.html`

3.2 Dove prendere la Shadow Suite

L'unica *Shadow Suite* raccomandata è ancora in beta testing, comunque le ultime versioni sono sicure in un ambiente di produzione e non contengono un programma di `login` vulnerabile.

Il pacchetto usa la seguente convenzione di denominazione:

```
shadow-AAMMGG.tar.gz
```

dove `AAMMGG` è la data di rilascio della Suite.

Questa versione alla fine diventerà la *Versione 3.3.3* quando verrà rilasciata dal beta testing ed è mantenuta da *Marek Michalkiewicz <marekm@i17linuxb.ists.pwr.wroc.pl>* . È disponibile come:

shadow-current.tar.gz `ftp://i17linuxb.ists.pwr.wroc.pl/pub/linux/shadow/shadow-current.tar.gz` .

Sono stati anche organizzati i seguenti siti mirror:

- *ftp://ftp.icm.edu.pl/pub/Linux/shadow/shadow-current.tar.gz*
- *ftp://iguana.hut.fi/pub/linux/shadow/shadow-current.tar.gz*
- *ftp://ftp.cin.net/usr/ggallag/shadow/shadow-current.tar.gz*
- *ftp://ftp.netural.com/pub/linux/shadow/shadow-current.tar.gz*

Dovreste usare la versione attualmente disponibile.

NON dovreste usare una versione *precedente* alla `shadow-960129` perché anche quelle hanno il problema di sicurezza di `login` discusso sopra.

Quando questo documento fa riferimento alla *Shadow Suite* mi riferisco a questo pacchetto. Si assume che sia questo il pacchetto che state usando.

Per riferimento, io ho usato `shadow-960129` per fare queste istruzioni di installazione.

Se stavate usando `shadow-mk`, dovreste fare l'aggiornamento a questa versione e ricompilare tutto ciò che avevate originariamente compilato.

3.3 Cosa è incluso nella Shadow Suite

La *Shadow Suite* contiene programmi sostitutivi per:

`su, login, passwd, newgrp, chfn, chsh, e id`

Il pacchetto contiene anche i nuovi programmi:

`chage, newusers, dpasswd, gpasswd, useradd, userdel, usermod, groupadd, groupdel, groupmod, groups, pwck, grpck, lastlog, pwconv, e pwunconv`

Inoltre è compresa la libreria: `libshadow.a` per scrivere e/o compilare programmi che necessitino di accedere alle password degli utenti.

Sono anche comprese pagine di manuale per i programmi.

C'è anche un file di configurazione per il programma di login che sarà installato come `/etc/login.defs`.

4 Compilare i programmi

4.1 Spacchettare l'archivio

Il primo passo dopo aver ottenuto il pacchetto è spacchettarlo. Il pacchetto è nel formato tar (tape archive) e compresso usando gzip, perciò prima spostatelo in `/usr/src`, poi digitate:

```
tar -xzvf shadow-current.tar.gz
```

Questo lo spacchetterà nella directory: `/usr/src/shadow-AAMMGG`

4.2 Configurare con il file config.h

La prima cosa che avete bisogno di fare è sovrascrivere il `Makefile` e il file `config.h`:

```
cd /usr/src/shadow-AAMMGG
cp Makefile.linux Makefile
cp config.h.linux config.h
```

Dovreste poi dare un'occhiata al file `config.h`. Questo file contiene definizioni per alcune delle opzioni di configurazione. Se state usando il pacchetto *consigliato*, vi consiglio, almeno per la prima volta, di disabilitare il supporto per il gruppo shadow.

Come opzione predefinita, sono abilitate le password di gruppo oscurate. Per disabilitarle, editate il file `config.h`, e cambiate il `#define SHADOWGRP` in `#undef SHADOWGRP`. Consiglio di disabilitarle per iniziare, e poi se volete davvero le password di gruppo e gli amministratori di gruppo, li abiliterete in seguito e ricompilerete. Se le lasciate abilitate, *dovete* creare il file `/etc/gshadow`.

Abilitare l'opzione per le password lunghe NON è raccomandato, come discusso sopra.

NON cambiate l'impostazione: `#undef AUTOSHADOW`

L'opzione `AUTOSHADOW` era stata in origine progettata in modo che i programmi che ignoravano la presenza delle shadow password avrebbero continuato a funzionare. Questo in teoria suona bene, ma non funziona correttamente. Se abilitate questa opzione, e il programma viene eseguito da root, potrebbe chiamare `getpwnam()` da root, e in seguito riscrivere il campo modificato nel file `/etc/passwd` (con *la password non più oscurata*). Fanno parte di tali programmi chfn e chsh (non potete aggirare questo problema semplicemente scambiando l'identificativo utente reale con quello effettivo prima di chiamare `getpwnam()` perché anche root potrebbe usare chfn e chsh).

Lo stesso avvertimento vale anche se state compilando libc, che ha un'opzione `SHADOW_COMPAT` che fa la stessa cosa. *NON dovrebbe* essere usata. Se cominciate a rimettere le password codificate nel vostro file `/etc/passwd`, questo è il problema.

Se state usando una versione di `libc` precedente alla 4.6.27, avrete bisogno di fare un paio di modifiche al `config.h` e al `Makefile`. Per il `config.h` editate e cambiate:

```
#define HAVE_BASENAME
```

in:

```
#undef HAVE_BASENAME
```

Poi, nel `Makefile`, cambiate:

```
SOBJS = smain.o env.o entry.o susetup.o shell.o sub.o mail.o motd.o sulog.o age.o tz.o hushed.o
SSRCS = smain.c env.c entry.c setup.c shell.c pwent.c sub.c mail.c motd.c sulog.c shadow.c age.c pwpack.c rad6
```

in:

```
SOBJS = smain.o env.o entry.o susetup.o shell.o sub.o mail.o motd.o sulog.o age.o tz.o hushed.o basename.o
SSRCS = smain.c env.c entry.c setup.c shell.c pwent.c sub.c mail.c motd.c sulog.c shadow.c age.c pwpack.c rad6
```

Questi cambiamenti aggiungono il codice contenuto in `basename.c` che è contenuto in `libc 4.6.27` e successive.

4.3 Fare copie di backup dei vostri programmi originali

Sarebbe anche una buona idea rintracciare e fare copie di backup dei programmi che la Shadow Suite sostituirà. Su un sistema Slackware 3.0 questi sono:

- /bin/su
- /bin/login
- /usr/bin/passwd
- /usr/bin/newgrp
- /usr/bin/chfn
- /usr/bin/chsh
- /usr/bin/id

Il pacchetto BETA ha una destinazione di *salvataggio* nel Makefile, ma è commentata perché distribuzioni diverse mettono i programmi in posti diversi.

Dovreste anche fare una copia di backup del vostro file `/etc/passwd`, ma state attenti a rinominarlo se lo mettete nella stessa directory così non sovrascriverete il comando `passwd`.

4.4 Eseguire il make

È necessario che siate collegati come root per fare la maggior parte dell'installazione.

Eseguite make per compilare gli eseguibili nel pacchetto:

```
make all
```

Potreste vedere l'avvertimento: `rcsid defined but not used`. È tutto a posto, succede solo perché l'autore sta usando un pacchetto con il controllo di versione.

5 Installazione

5.1 Tenete a portata di mano un disco di boot nel caso faceste qualche danno

Se qualcosa va terribilmente male, sarebbe utile avere un disco di boot. Se avete la combinazione boot/root dalla vostra installazione, questa funzionerà, altrimenti leggete il *Bootdisk-HOWTO* `http://sunsite.unc.edu/mdw/HOWTO/Bootdisk-HOWTO.html` , che descrive come fare un disco di boot.

5.2 Rimuovere le pagine di manuale duplicate

Dovreste anche spostare le pagine di manuale che stanno per essere sostituite. Anche se siete abbastanza coraggiosi da installare la Shadow Suite senza fare backup, comunque vorrete togliere le vecchie pagine di manuale. Le nuove pagine di manuale normalmente non sovrascriveranno quelle vecchie perché quelle vecchie sono probabilmente compresse.

Potete usare la combinazione del `comando man -aW` e del `comando locate` per spostare le pagine man che devono essere (ri)mosse. È generalmente più facile trovare quali sono le vecchie pagine man prima di eseguire `make install`.

Se state usando la distribuzione Slackware 3.0, allora le pagine man che volete rimuovere sono:

- /usr/man/man1/chfn.1.gz
- /usr/man/man1/chsh.1.gz
- /usr/man/man1/id.1.gz
- /usr/man/man1/login.1.gz
- /usr/man/man1/passwd.1.gz
- /usr/man/man1/su.1.gz
- /usr/man/man5/passwd.5.gz

Ci potrebbero anche essere delle pagine man con lo stesso nome nelle sottodirectory `/var/man/cat[1-9]` che dovrebbero anch'esse essere cancellate.

5.3 Eseguire make install

Siete ora pronti a digitare (fatelo come root)

```
make install
```

Questo installerà i programmi nuovi e quelli sostitutivi e sistemerà i permessi dei file. Installerà anche le pagine man.

Questo si occupa anche di installare i file include della Shadow Suite nelle posizioni corrette in `/usr/include/shadow`.

Usando il pacchetto BETA dovete copiare manualmente il file `login.defs` nella sottodirectory `/etc` ed essere sicuri che solo *root* possa cambiarlo.

```
cp login.defs /etc
chmod 700 /etc/login.defs
```

Questo file è il file di configurazione per il programma di *login*. Dovreste controllare e apportare i cambiamenti opportuni a questo file per il vostro particolare sistema. Qui è dove potete decidere da quale tty root può collegarsi e stabilire altre opzioni di strategia di sicurezza (come predefinire la scadenza delle password).

5.4 Eseguire pwconv

Il passo successivo è eseguire `pwconv`. Anche questo deve essere fatto da *root*, ed è meglio che venga fatto dalla sottodirectory `/etc`:

```
cd /etc
/usr/sbin/pwconv
```

`pwconv` prende il vostro file `/etc/passwd` e ne estrae i campi allo scopo di creare due file: `/etc/npasswd` e `/etc/nshadow`.

Viene anche fornito un programma `pwunconv` qualora aveste bisogno di ricostruire il file originale `/etc/passwd` dalla combinazione `/etc/passwd` e `/etc/shadow`.

5.5 Rinominare npasswd e nshadow

Ora che avete eseguito `pwconv` avete creato i file `/etc/npasswd` e `/etc/nshadow`. Questi file devono essere copiati in `/etc/passwd` e `/etc/shadow`. Vogliamo anche fare una copia di backup del file originale `/etc/passwd` ed essere sicuri che solo root possa leggerlo. Metteremo la copia di backup nella home directory di root:

```
cd /etc
cp passwd ~passwd
chmod 600 ~passwd
mv npasswd passwd
mv nshadow shadow
```

Dovreste anche assicurarvi che la proprietà e i permessi dei file siano corretti. Se state per usare *X-Windows*, i programmi `xlock` e `xdm` devono poter leggere il file `shadow` (ma non scriverlo).

Ci sono due modi per fare questo. Potete impostare `xlock` come SUID root (`xdm` di solito viene comunque eseguito da root). Oppure potete fare in modo che il proprietario del file `shadow` sia `root` con un gruppo `shadow`, ma prima che lo facciate, siate sicuri di avere un gruppo shadow (guardate in `/etc/group`). Nessuno degli utenti del sistema dovrebbe in realtà stare nel gruppo shadow.

```
chown root.root passwd
chown root.shadow shadow
chmod 0644 passwd
chmod 0640 shadow
```

Il vostro sistema ha ora il file password oscurato. *Dovreste* ora andare su un altro terminale virtuale e verificare che possiate collegarvi.

Davvero, fatelo adesso!

Se non potete, allora c'è qualcosa di sbagliato! Per ritornare a un stato non oscurato, fate ciò che segue:

```
cd /etc
cp ~passwd passwd
chmod 644 passwd
```

Dovreste poi ripristinare nelle loro corrette posizioni i file che avevate salvato prima.

6 Altri programmi a cui potreste dover fare un aggiornamento o applicare una patch

Anche se la shadow suite contiene programmi sostitutivi per la maggior parte dei programmi che hanno bisogno di accedere alle password, ci sono alcuni altri programmi su molti sistemi che richiedono accesso alle password.

Se state usando una *Distribuzione Debian* (o anche se non la usate), potete ottenere i sorgenti Debian per i programmi che devono essere ricompilati da: ftp://ftp.debian.org/debian/stable/source/

Il resto di questa sezione si occupa di come aggiornare `adduser`, `wu_ftpd`, `ftpd`, `pop3d`, `xlock`, `xdm` e `sudo` in modo che supportino la shadow suite.

Guardate il capitolo 8 (Aggiungere il supporto Shadow ad un programma C) per una discussione su come aggiungere il supporto shadow a qualunque altro programma che ne abbia bisogno (anche se il programma deve allora essere eseguito SUID root o SGID shadow per poter veramente accedere al file shadow).

6.1 Il programma adduser della Slackware

Le distribuzioni Slackware (e forse anche altre) contengono un programma interattivo per aggiungere utenti chiamato `/sbin/adduser`. Una versione shadow di questo programma si può ottenere da *ftp://sunsite.unc.edu/pub/Linux/system/Admin/accounts/adduser.shadow-1.4.tar.gz* .

Vi incoraggio ad usare i programmi che vengono forniti con la *Shadow Suite* (`useradd`, `usermod`, e `userdel`) invece del programma Slackware `adduser`. Imparare ad usarli richiede poco tempo, ma vale la pena fare lo sforzo perché avete molto più controllo ed essi eseguono un appropriato lock dei file `/etc/passwd` e `/etc/shadow` (`adduser` non lo fa).

Guardate il capitolo su 7 (Mettere al lavoro la Shadow Suite) per maggiori informazioni.

Ma se dovete proprio usarlo (adduser N.d.T.), ecco cosa dovete fare:

```
tar -xzvf adduser.shadow-1.4.tar.gz
cd adduser
make clean
make adduser
chmod 700 adduser
cp adduser /sbin
```

6.2 Il Server wu_ftpd

La maggior parte dei sistemi Linux contengono il server `wu_ftpd`. Se la vostra distribuzione non ha la shadow installata, allora il vostro `wu_ftpd` non sarà compilato per la shadow. `wu_ftpd` viene lanciato da `inetd/tcpd` come un processo di *root*. Se state eseguendo un vecchio demone `wu_ftpd`, vorrete aggiornarlo comunque perché quelli più vecchi hanno un baco che permetterebbe che l'account *root* venisse compromesso (per maggiori informazione guardate la *Linux security home page* `http://bach.cis.temple.edu/linux/linux-security/Linux-Security-FAQ/Linux-wu.ftpd-2.4-Update.html`).

Fortunatamente, avete solo bisogno di ottenere il codice sorgente e di ricompilarlo con le shadow abilitate.

Se non state usando un sistema ELF, il server `wu_ftp` può essere trovato su Sunsite come *wu-ftp-2.4-fixed.tar.gz* `ftp://sunsite.unc.edu/pub/Linux/system/Network/file-transfer/wu-ftpd-2.4-fixed.tar.gz`

Una volta ottenuto il server, mettetelo in `/usr/src`, quindi digitate:

```
cd /usr/src
tar -xzvf wu-ftpd-2.4-fixed.tar.gz
cd wu-ftpd-2.4-fixed
cp ./src/config/config.lnx.shadow ./src/config/config.lnx
```

Quindi editate `./src/makefiles/Makefile.lnx`, e cambiate la riga:

```
LIBES    = -lbsd -support
```

in:

```
LIBES    = -lbsd -support -lshadow
```

Ora siete pronti ad eseguire lo script build e all'installazione:

```
cd /usr/src/wu-ftpd-2.4-fixed
/usr/src/wu-ftp-2.4.fixed/build lnx
cp /usr/sbin/wu.ftpd /usr/sbin/wu.ftpd.old
cp ./bin/ftpd /usr/sbin/wu.ftpd
```

Questo usa il file di configurazione delle shadow di Linux, compila ed installa il server.

Sul mio sistema Slackware 2.3 devo fare anche le seguenti cose prima di eseguire il `build`:

```
cd /usr/include/netinet
ln -s in_systm.h in_system.h
cd -
```

Sono stati riscontrati dei problemi nel compilare questo pacchetto sotto sistemi ELF, ma la versione Beta della prossima release funziona bene. Si può trovare come

wu-ftp-2.4.2-beta-10.tar.gz `ftp://tscnet.com/pub/linux/network/ftp/wu-ftpd-2.4.2-beta-10.tar.gz`

Una volta ottenuto il server, mettetelo in `/usr/src`, quindi digitate:

```
cd /usr/src
tar -xzvf wu-ftpd-2.4.2-beta-9.tar.gz
cd wu-ftpd-beta-9
cd ./src/config
```

Poi editate `config.lnx`, e cambiate:

```
#undef SHADOW.PASSWORD
```

in:

```
#define SHADOW.PASSWORD
```

Poi,

```
cd ../Makefiles
```

ed editate il file `Makefile.lnx` e cambiate:

```
LIBES = -lsupport -lbsd # -lshadow
```

in:

```
LIBES = -lsupport -lbsd -lshadow
```

Poi eseguite build ed installate:

```
cd ..
build lnx
cp /usr/sbin/wu.ftpd /usr/sbin/wu.ftpd.old
cp ./bin/ftpd /usr/sbin/wu.ftpd
```

Notate che dovreste controllare il vostro file `/etc/inetd.conf` per essere sicuri che è qui che viene realmente realmente il vostro server wu.ftpd. È stato riscontrato che alcune distribuzioni mettono i server dei demoni in posti diversi, e quindi wu.ftpd in particolare potrebbe essere chiamato in qualche altro modo.

6.3 Ftpd standard

Se state usando il server `ftpd` standard, vi consiglio di aggiornarlo al server `wu_ftpd`. A parte il baco conosciuto discusso sopra, generalmente è considerato più sicuro.

Se insistete ad usare quello standard, o avete bisogno di supporto *NIS*, Sunsite ha *ftpd-shadow-nis.tgz* `ftp://sunsite.unc.edu/pub/Linux/system/Network/file-transfer/ftpd-shadow-nis.tgz`

6.4 pop3d (Post Office Protocol 3)

Se avete bisogno di supportare il *Post Office Protocol 3 (POP3)*, avete bisogno di ricompilare un programma `pop3d`. `pop3d` è normalmente eseguito da `inetd/tcpd` come `root`.

Ci sono due versioni disponibili da:

pop3d-1.00.4.linux.shadow.tar.gz `ftp://sunsite.unc.edu/pub/Linux/system/Mail/pop/pop3d-1.00.4.linux.shadow.tar.gz`

e *pop3d+shadow+elf.tar.gz* `ftp://sunsite.unc.edu/pub/Linux/system/Mail/pop/pop3d+shadow+elf.tar.gz`

Entrambi questi sono abbastanza semplici da installare.

6.5 xlock

Se installate la Shadow Suite e poi eseguite *X Windows System* e bloccate (lock) lo schermo senza aggiornare il vostro `xlock`, dovrete usare `CNTL-ALT-Fx` per passare ad un'altra *tty*, collegarvi, e uccidere il processo `xlock` (o usare `CNTL-ALT-BS` per uccidere il server X). Fortunatamente è abbastanza facile aggiornare il vostro programma `xlock`.

Se state usando le Versioni 3.x.x di XFree86, probabilmente state usando `xlockmore` (che è un grande screensaver in aggiunta a lock). Questo pacchetto supporta le *shadow* con una ricompilazione. Se avete un `xlock` precedente, vi consiglio di aggiornarlo a questo.

`xlockmore-3.5.tgz` è disponibile su: `ftp://sunsite.unc.edu/pub/Linux/X11/xutils/screensavers/xlockmore-3.7.tgz`

Fondamentalmente, questo è quello che avete bisogno di fare:

Ottenere il file `xlockmore-3.7.tgz` e metterlo in `/usr/src`, spacchettarlo:

```
tar -xzvf xlockmore-3.7.tgz
```

Editare il file: `/usr/X11R6/lib/X11/config/linux.cf`, e cambiare la riga:

```
#define HasShadowPasswd    NO
in
#define HasShadowPasswd    YES
```

Quindi compilate gli eseguibili:

```
cd /usr/src/xlockmore
xmkmf
make depend
make
```

Quindi spostare tutto al suo posto e aggiornare i proprietari ed i permessi dei file:

```
cp xlock /usr/X11R6/bin/
cp XLock /var/X11R6/lib/app-defaults/
chown root.shadow /usr/X11R6/bin/xlock
chmod 2755 /usr/X11R6/bin/xlock
chown root.shadow /etc/shadow
chmod 640 /etc/shadow
```

Il vostro xlock ora funzionerà correttamente.

6.6 xdm

`xdm` è un programma che presenta uno schermo di login per X-Windows. Alcuni sistemi avviano `xdm` quando viene detto al sistema di andare ad uno specifico livello di esecuzione (vedere `/etc/inittab`).

Con la *Shadow Suite* installata, `xdm` avrà bisogno di essere aggiornato. Fortunatamente è abbastanza facile aggiornare il vostro programma `xdm`.

`xdm.tar.gz` è disponibile su: `ftp://sunsite.unc.edu/pub/Linux/X11/xutils/xdm.tar.gz`

Prendete il file `xdm.tar.gz` e mettetelo in `/usr/src`, quindi per spacchettarlo:

```
tar -xzvf xdm.tar.gz
```

Editate il file: `/usr/X11R6/lib/X11/config/linux.cf`, e cambiate la riga:

```
#define HasShadowPasswd    NO
in
#define HasShadowPasswd    YES
```

Quindi compilate gli eseguibili:

```
cd /usr/src/xdm
xmkmf
make depend
make
```

Poi mettete tutto al suo posto:

```
cp xdm /usr/X11R6/bin/
```

`xdm` è eseguito da *root* perciò non avete bisogno di cambiare i permessi del file.

6.7 sudo

Il programma `sudo` permette ad un amministratore di sistema di lasciare che gli utenti eseguano programmi che normalmente richiederebbero accesso da root. Questo è comodo perché lascia limitato l'accesso di amministratore all'account root stesso, mentre permette agli utenti di fare cose tipo il mount dei dispositivi.

`sudo` necessita di leggere le password perché verifica la password dell'utente quando viene invocato. `sudo` già viene eseguito SUID root, perciò accedere al file `/etc/shadow` non è un problema.

`sudo` per la shadow suite, è disponibile su:

`ftp://sunsite.unc.edu/pub/Linux/system/Admin/sudo-1.2-shadow.tgz`

Attenzione: Quando installate `sudo` il vostro file `/etc/sudoers` sarà sostituito con uno predefinito, perciò avrete bisogno di farne una copia di backup se avete aggiunto qualcosa a quello predefinito (potreste anche editare il Makefile e rimuovere la riga che copia il file predefinito in `/etc`).

Il pacchetto è già predisposto per le shadow, perciò tutto quello che è richiesto è ricompilare il pacchetto (mettetelo in `/usr/src`):

```
cd /usr/src
tar -xzvf sudo-1.2-shadow.tgz
cd sudo-1.2-shadow
make all
make install
```

6.8 imapd (pacchetto Pine E-Mail)

`imapd` è un server e-mail simile a `pop3d`. `imapd` è incluso nel pacchetto *Pine E-mail*. La documentazione inclusa nel pacchetto afferma che nei sistemi Linux è predefinita l'opzione di includere il supporto shadow. Comunque, ho trovato che questo non è vero. Inoltre, la combinazione script di build/Makefile su questo pacchetto rende molto difficile aggiungere la libreria `libshadow.a` in tempo di compilazione, perciò non sono riuscito ad aggiungere il supporto shadow per `imapd`.

Se qualcuno è riuscito a farlo, per favore mi mandi una e-mail, ed io includerò qui la soluzione.

6.9 pppd (Point-to-Point Protocol Server)

Il server pppd può essere impostato in modo che usi diversi tipi di autenticazione: *Password Authentication Protocol* (PAP) e *Cryptographic Handshake Authentication Protocol* (CHAP). Il server pppd di solito legge le stringhe contenenti le password che usa da `/etc/ppp/chap-secrets` e/o `/etc/ppp/pap-secrets`. Se state usando questo comportamento predefinito di pppd, non è necessario reinstallare pppd.

pppd vi permette anche di usare il parametro *login* (o su linea di comando, o nella configurazione del file `options`). Se viene data l'opzione *login*, il pppd userà il file `/etc/passwd` per il nome utente e la password per il *PAP*. Questo, ovviamente, non funzionerà più ora che il nostro file shadow è oscurato. Per quanto riguarda pppd-1.2.1d questo richiede aggiunta di codice per il supporto shadow.

L'esempio dato nel prossimo capitolo consiste nell'aggiunta di supporto shadow a `pppd-1.2.1d` (una vecchia versione di pppd).

`pppd-2.2.0` contiene già il supporto shadow.

7 Mettere al lavoro la Shadow Suite

Questo capitolo tratta alcune cose che dovete sapere ora che avete la *Shadow Suite* installata sul vostro sistema. Ulteriori informazioni sono contenute nelle pagine di manuale per ogni comando.

7.1 Aggiungere, Modificare e Cancellare utenti

La *Shadow Suite* ha aggiunto i seguenti comandi orientati a linea di comando per aggiungere, modificare, e cancellare utenti. Potreste anche aver installato il programma `adduser`.

7.1.1 useradd

Il comando `useradd` può essere usato per aggiungere utenti al sistema. Potete anche invocare questo comando per cambiare le impostazioni predefinite.

La prima cosa che dovreste fare è esaminare le impostazioni predefinite e apportare cambiamenti specifici per il vostro sistema:

```
useradd -D
```

```
GROUP=1
HOME=/home
INACTIVE=0
EXPIRE=0
SHELL=
SKEL=/etc/skel
```

Le impostazioni predefinite probabilmente non sono quelle che volete, perciò se cominciaste ad aggiungere utenti adesso dovreste specificare tutte le informazioni per ciascun utente. Comunque, possiamo e dovremmo cambiare i valori predefiniti.

Sul mio sistema:

- Voglio che il gruppo predefinito sia 100
- Voglio che le password scadano ogni 60 giorni
- Non voglio bloccare un account se la password è scaduta
- Voglio che la shell predefinita sia `/bin/bash`

Per fare questi cambiamenti userei:

```
useradd -D -g100 -e60 -f0 -s/bin/bash
```

Ora eseguendo `useradd -D` darà:

```
GROUP=100
HOME=/home
INACTIVE=0
EXPIRE=60
SHELL=/bin/bash
SKEL=/etc/skel
```

Solo nel caso voleste saperlo, questi valori predefiniti sono contenuti nel file `/etc/default/useradd`.

Ora potete usare `useradd` per aggiungere utenti al sistema. Per esempio, per aggiungere l'utente `fred`, usando i valori predefiniti, dovreste fare come segue:

```
useradd -m -c "Fred Flintstone" fred
```

Questo creerà la voce seguente nel file `/etc/passwd`:

```
fred:*:505:100:Fred Flintstone:/home/fred:/bin/bash
```

E la voce seguente nel file `/etc/shadow`:

```
fred:!:0:0:60:0:0:0:0
```

Verrà creata la home directory di `fred` e il contenuto di `/etc/skel` sarà copiato là grazie all'opzione `-m`.

Inoltre, dato che non abbiamo specificato un UID, è stato usato il primo disponibile.

L'account di `fred` è stato creato, ma `fred` non sarà ancora in grado di collegarsi fino a quando sbloccheremo l'account. Facciamo questo cambiando la password.

```
passwd fred
```

```
Changing password for fred
Enter the new password (minimum of 5 characters)
Please use a combination of upper and lower case letters and numbers.
New Password: *******
Re-enter new password: *******
```

Che, in italiano, sarebbe qualcosa del genere:

```
Cambio la password di fred
Inserire la nuova password (minimo 5 caratteri)
Per favore, utilizzare una combinazione di maiuscole, minuscole e cifre.
Nuova Password: *******
Reinserire la nuova password: *******
```

Ora `/etc/shadow` conterrà:

```
fred:J0C.WDR1amIt6:9559:0:60:0:0:0:0
```

E `fred` potrà ora collegarsi ed usare il sistema. La cosa bella di `useradd` e degli altri programmi che vengono forniti con la *Shadow Suite* è che fanno cambiamenti ai file `/etc/passwd` e `/etc/shadow` in modo non interrompibile. Perciò , se state aggiungendo un utente, e contemporaneamente un altro utente sta cambiando la sua password, entrambe le operazioni verranno eseguite correttamente.

Dovreste usare i comandi forniti anziché editare direttamente `/etc/passwd` e `/etc/shadow`. Se voi editaste il file `/etc/shadow`, e un utente cambiasse la sua password mentre voi state editando, e poi voi salvaste il file che stavate editando, il cambiamento della password dell'utente andrebbe perso.

Qui c'è un piccolo script interattivo che aggiunge utenti usando `useradd` e `passwd`:

```
#!/bin/bash
#
# /sbin/newuser - Uno script per aggiungere utenti al sistema usando i
#                 comandi useradd e passwd della Shadow Suite.
#
# Scritto da Mike Jackson <mhjack@tscnet.com> come esempio per il
# Linux Shadow Password Howto. Viene esplicitamente concesso il
# permesso di usarlo e modificarlo.
#
# Questo potrebbe essere modificato per mostrare i valori predefiniti
# e permettere modifiche simili al programma Slackware
# adduser. Potrebbe essere modificato per non permettere voci stupide
# (i.e. miglior controllo degli errori).
#
##
# Valori predefiniti per il comando useradd
##
GROUP=100        # Gruppo predefinito
HOME=/home       # Collocazione della home directory (/home/nomeutente)
SKEL=/etc/skel   # Struttura tipica di una nuova directory home.
INACTIVE=0       # Giorni tra la scadenza della password e la
                 # disabilitazione dell'account (0 = mai)
EXPIRE=60        # Durata della password in giorni
SHELL=/bin/bash  # Shell predefinita (intero percorso)
##
#  Valori predefiniti per il comando passwd
##
PASSMIN=0        # Giorni tra i cambiamenti della password
PASSWARN=14      # Giorni prima che scada la password in cui viene
                 # dato un avviso
##
#  Assicurarsi che sia root ad eseguire lo script.
##
WHOAMI=`/usr/bin/whoami`
if [ $WHOAMI != "root" ]; then
        echo "Devi essere root per aggiungere nuovi utenti!"
        exit 1
fi
##
#  Chiedere il nome utente e il nome completo.
##
echo ""
echo -n "Nome utente: "
read USERNAME

echo -n "Nome completo: "
read FULLNAME

#
echo "Aggiunta dell'utente: $USERNAME."
```

```
#
# Notate che le "" intorno a $FULLNAME sono richieste perche
# questo campo quasi sempre conterra almeno uno spazio, e senza
# le " il comando useradd, quando raggiunge il carattere SPAZIO,
# penserebbe che vi stiate spostando sul prossimo parametro.
#
/usr/sbin/useradd -c"$FULLNAME" -d$HOME/$USERNAME -e$EXPIRE \
        -f$INACTIVE -g$GROUP -m -k$SKEL -s$SHELL $USERNAME

##
#  Impostare i valori predefiniti per le password
##
/bin/passwd -n $PASSMIN -w $PASSWARN $USERNAME >/dev/null 2>&1
##
#  Lascia che il comando passwd chieda la password (due volte)
##
/bin/passwd $USERNAME

##
#  Mostra cio che e stato fatto.
##
echo ""
echo "Voce di /etc/passwd:"
echo -n "   "
grep "$USERNAME:" /etc/passwd
echo "Voce di /etc/shadow:"
echo -n "   "
grep "$USERNAME:" /etc/shadow
echo "Riassunto dei risultati del comando passwd:"
echo -n "   "
passwd -S $USERNAME

echo ""
```

Usare uno script per aggiungere utenti è davvero molto più preferibile che editare direttamente i file `/etc/passwd` o `/etc/shadow` o usare un programma come il programma Slackware `adduser`. Sentitevi liberi di usare e modificare questo script per il vostro particolare sistema.

Per maggiori informazioni su `useradd` vedere la pagina di manuale in linea.

7.1.2 usermod

Il programma `usermod` viene usato per modificare le informazioni su un utente. Le opzioni sono simili a quelle del programma `useradd`.

Diciamo che volete cambiare la shell di `fred`, fareste ciò che segue:

```
usermod -s /bin/tcsh fred
```

Ora la voce di `fred`nel file `/etc/passwd` sarebbe diventata questa:

```
fred:*:505:100:Fred Flintstone:/home/fred:/bin/tcsh
```

Facciamo in modo che l'account di `fred` scada il 09/15/97:

```
usermod -e 09/15/97 fred
```

Ora la voce di `fred` in `/etc/shadow` diventa:

```
fred:J0C.WDR1amIt6:9559:0:60:0:0:10119:0
```

Per maggiori informazioni sul comando `usermod` vedere la pagina di manuale in linea.

7.1.3 userdel

`userdel` fa proprio quello che vi aspettate, cancella l'account dell'utente. Semplicemente usate:

```
userdel -r nomeutente
```

Il `-r` fa sì che tutti i file nella home directory dell'utente vengano cancellati insieme alla home directory stessa. I file collocati in altri file system dovranno essere cercati e cancellati manualmente.

Se volete semplicemente bloccare l'account invece che cancellarlo, usate piuttosto il comando `passwd`.

7.2 Il comando passwd e l'invecchiamento delle password

Il comando `passwd` ha l'ovvio uso di cambiare le password. Inoltre, viene usato dall'utente *root* per:

- Bloccare (lock) e sbloccare (unlock) gli account (`-l` e `-u`)
- Impostare il massimo numero di giorni per cui una password rimane valida (`-x`)
- Impostare il minimo numero di giorni tra cambiamenti della password (`-n`)
- Impostare il numero di giorni di avviso che una password sta per scadere (`-w`)
- Impostare il numero di giorni dopo la scadenza della password prima che l'account venga bloccato (`-i`)
- Permettere la visualizzazione delle informazioni di account in un formato più chiaro `-S`)

Per esempio, diamo ancora un'occhiata a `fred`

```
passwd -S fred
fred P 03/04/96 0 60 0 0
```

Questo significa che la password di `fred` è valida, che è stata cambiata l'ultima volta il 03/04/96, che può essere cambiata in qualunque momento, scade dopo 60 giorni, fred non sarà avvertito e l'account non verrà disabilitato quando la password scadrà.

Questo semplicemente significa che se `fred` si collega dopo che la password scade, al collegamento gli verrà richiesta una nuova password.

Se decidiamo che vogliamo avvertire `fred` 14 giorni prima che la sua password scada e inattivare il suo account 14 giorni dopo che lui la lascia scadere, dovremo fare quanto segue:

```
passwd -w14 -i14 fred
```

Ora `fred` è diventato:

```
fred P 03/04/96 0 60 14 14
```

Per ulteriori informazioni sul comando `passwd` vedere le pagine di manuale in linea.

7.3 Il file login.defs

Il file `/etc/login` è il file di configurazione per il programma `login` e anche per l'intera *Shadow Suite*.

`/etc/login` contiene impostazioni che riguardano dall'aspetto del prompt fino alla scadenza predefinita quando un utente cambia la sua password.

Il file `/etc/login.defs` è abbastanza ben documentato dai commenti contenuti al suo interno. Comunque, ci sono alcune cose da notare:

- Contiene alcuni flag che possono essere attivati o disattivati che determinano il numero di collegamenti che vengono effettuati.
- Contiene puntatori ad altri file di configurazione.
- Contiene valori predefiniti per cose tipo l'invecchiamento delle password.

Dalla lista sopra potete vedere che questo è un file abbastanza importante, e dovreste essere sicuri che sia presente, e che le impostazioni siano quelle che desiderate per il vostro sistema.

7.4 Password di gruppo

Il file `/etc/groups` può contenere password che permettono ad un utente di diventare membro di un particolare gruppo. Questa funzione è abilitata se definite la costante `SHADOWGRP` nel file `/usr/src/shadow-AAMMGG/config.h`.

Se definite questa costante e poi compilate, dovete creare un file `/etc/gshadow` che contenga le password del gruppo e le informazioni di amministrazione del gruppo.

Quando avete creato `/etc/shadow`, avete usato un programma chiamato `pwconv`, non c'è nessun programma equivalente per creare il file `/etc/gshadow`, ma in realtà non importa, se ne occupa lui stesso.

Per creare il file iniziale `/etc/gshadow` fate come segue:

```
touch /etc/gshadow
chown root.root /etc/gshadow
chmod 700 /etc/gshadow
```

Una volta che create nuovi gruppi, questi verranno aggiunti ai file `/etc/group` e `/etc/gshadow`. Se voi modificate un gruppo aggiungendo o togliendo utenti o cambiando la password del gruppo, il file `/etc/gshadow` verrà modificato.

I programmi `groups`, `groupadd`, `groupmod`, e `groupdel` sono forniti come parte della *Shadow Suite* per modificare i gruppi.

Il formato del file `/etc/group` è quello che segue:

```
nomegruppo:!:GID:membro,membro,...
```

Dove:

`nomegruppo`

Il nome del gruppo

`!`

Il campo che normalmente contiene la password, che ora è però situata nel file `/etc/gshadow`

`GID`

L'identificativo numerico del gruppo

`membro`

Elenco dei membri del gruppo

Il formato del file `/etc/gshadow` è quello che segue:

```
nomegruppo:password:ammin,ammin,...:membro,membro,...
```

Dove:

`nomegruppo`

Il nome del gruppo

`password`

La password del gruppo codificata

`ammin`

Elenco degli amministratori del gruppo

`membro`

Elenco dei membri del gruppo

Il comando `gpasswd` è usato solo per aggiungere o togliere amministratori e membri a o da un gruppo. Solo `root` o qualcuno appartenente all'elenco degli amministratori può aggiungere o togliere membri del gruppo.

La password del gruppo può essere modificata con il comando `passwd` da *root* o chiunque appartenga alla lista degli amministratori del gruppo.

Nonostante il fatto che attualmente non ci sia una pagina di manuale per `gpasswd`, digitando `gpasswd` senza alcun parametro si ottiene un elenco di opzioni. È abbastanza semplice capire come funziona il tutto una volta che avete capito i formati dei file e i concetti.

7.5 Programmi per il controllo della consistenza

7.5.1 pwck

Il programma `pwck` viene fornito per offrire un controllo di consistenza sui file `/etc/passwd` e `/etc/shadow`. Esso controllerà ogni nome utente e verificherà che abbia quanto segue:

- il corretto numero di campi
- un nome utente univoco
- un valido identificatore di utente e di gruppo
- un valido gruppo primario
- una valida home directory
- una valida shell di login

Darà anche un avvertimento per ogni account privo di password.

È una buona idea eseguire `pwck` dopo aver installato la *Shadow Suite*. È anche una buona idea eseguirlo periodicamente, magari una volta alla settimana o al mese. Se usate l'opzione `-r`, potete usare `cron` per eseguirlo con una cadenza regolare e riceverne per posta il rapporto.

7.5.2 grpck

`grpck` è il programma per il controllo della consistenza per i file `/etc/group` and `/etc/gshadow`. Esso esegue i seguenti controlli:

- il corretto numero di campi
- un nome del gruppo univoco
- elenco valido di membri ed amministratori

Dispone anche dell'opzione `-r` per rapporti automatizzati.

7.6 Password di dial-up

Le password di dial-up sono un altro strumento opzionale di difesa per i sistemi che permettono l'accesso tramite una linea telefonica commutata. Se avete un sistema che permette a molte persone di connettersi localmente o tramite una rete, ma volete porre dei limiti su chi possa accedere per telefono e connettersi, allora le password di dial-up fanno al caso vostro. Per abilitare le password di dial-up, dovete editare il file `/etc/login.defs` ed assicurarvi che `DIALUPS_CHECK_ENAB` sia impostato a `yes`.

Due sono i file che contengono informazioni di dial-up: `/etc/dialups` che contiene le tty (una per riga, senza la parte iniziale /dev). Se una tty compare nella lista, allora vengono effettuati i controlli di dial-up.

Il secondo file è `/etc/d_passwd`. Questo file contiene il percorso completo di una shell, seguito da una password opzionale.

Se un utente si collega attraverso una tty elencata in `/etc/dialups`, e la sua shell è presente nel file `/etc/d_passwd` gli sarà permesso l'accesso solo se fornirà la corretta password.

Un altro utile scopo per usare password di dial-up potrebbe essere quello di impostare una linea che permetta solo un certo tipo di connessione (come una connessione PPP o UUCP). Se un utente cerca di ottenere un altro tipo di connessione (i.e. un elenco di shell), deve conoscere una password per usare la linea.

Prima che possiate usare la caratteristica del dial-up, dovete creare i file.

Viene fornito il comando `dpasswd` per assegnare password per le shell nel file `/etc/d_passwd`. Vedere la pagina di manuale per ulteriori informazioni.

8 Aggiungere il supporto shadow ad un programma C

Aggiungere il supporto shadow ad un programma è in realtà abbastanza semplice. L'unico problema è che il programma deve essere eseguito da root (o SUID root) in modo che il programma possa accedere al file `/etc/shadow`.

Questo presenta un grande problema: occorre seguire una condotta di programmazione molto attenta quando si creano programmi SUID. Per esempio, se un programma ha un comando che invoca una shell, questa non deve essere eseguita con i diritti di root anche se il programma è SUID root.

Per aggiungere il supporto shadow ad un programma in modo che possa controllare le password, ma per il resto non necessita di essere eseguito da root, è molto più sicuro eseguire il programma SGID shadow. Il programma `xlock` ne è un esempio.

Nell'esempio fatto prima, `pppd-1.2.1d` già viene eseguito SUID root, perciò aggiungere il supporto shadow non dovrebbe rendere il programma più vulnerabile.

8.1 File di intestazione (header)

I file di intestazione (header) dovrebbero stare in `/usr/include/shadow`. Ci dovrebbe anche essere un `/usr/include/shadow.h`, ma sarebbe un link simbolico a `/usr/include/shadow/shadow.h`.

Per aggiungere il supporto shadow ad un programma, dovete includere i file di intestazione:

```
#include <shadow/shadow.h>
#include <shadow/pwauth.h>
```

Potrebbe essere una buona idea usare le direttive del compilatore in modo da condizionare la compilazione del codice shadow (io lo faccio nell'esempio che segue).

8.2 La libreria libshadow.a

Quando avete installato la *Shadow Suite* il file `libshadow.a` è stato creato ed installato in `/usr/lib`.

Quando si compila il supporto shadow in un programma, bisogna dire al linker di includere la libreria `libshadow.a`.

Questo viene fatto da:

```
gcc program.c -o program -lshadow
```

Comunque, come vedremo nell'esempio che segue, la maggior parte dei programmi di grandi dimensioni usa un `Makefile`, che di solito ha una variabile chiamata `LIBS=...` che noi modificheremo.

8.3 La struttura Shadow

La libreria `libshadow.a` usa una struttura chiamata `spwd` per le informazioni che preleva dal file `/etc/shadow`. Questa è la definizione della struttura `spwd` dal file di intestazione `/usr/include/shadow/shadow.h`:

```
struct spwd
{
  char *sp_namp;              /* nome di login  */
  char *sp_pwdp;              /* password codificata */
  sptime sp_lstchg;           /* data dell'ultimo cambiamento */
  sptime sp_min;              /* minimo numero di giorni tra cambiamenti */
  sptime sp_max;              /* massimo numero di giorni tra cambiamenti */
  sptime sp_warn;             /* numero di giorni di avvertimento prima
                                 che scada la password */
  sptime sp_inact;            /* numero di giorni dopo la scadenza della
                                 password prima che l'account venga
                                 disabilitato */
  sptime sp_expire;           /* giorni dal 1/1/70 fino alla scadenza
                                 dell'account */
  unsigned long sp_flag;      /* riservato per uso futuro */
};
```

La *Shadow Suite* può mettere altre cose nel campo `sp_pwdp` proprio a fianco della password codificata. Il campo della password potrebbe contenere:

```
nomeutente:Npge08pfz4wuk;@/sbin/extra:9479:0:10000::::
```

Questo significa che, oltre alla password, dovrebbe essere chiamato il programma `/sbin/extra` per ulteriori autenticazioni. Il programma chiamato riceverà il nome utente e un'opzione che indica perché viene chiamato. Vedere il file `/usr/include/shadow/pwauth.h` e il codice sorgente di `pwauth.c` per ulteriori informazioni.

Ciò che voglio dire è che dovremmo usare la funzione `pwauth` per eseguire la vera autenticazione, dato che si occuperà anche dell'autenticazione secondaria. L'esempio sotto fa proprio questo.

L'autore della *Shadow Suite* fa presente che poiché molti dei programmi esistenti non la usano potrebbe essere rimossa o cambiata dalle future versioni della *Shadow Suite*.

8.4 Funzioni Shadow

Il file `shadow.h` contiene anche i prototipi delle funzioni contenute nella libreria `libshadow.a`:

```
extern void setspent __P ((void));
extern void endspent __P ((void));
extern struct spwd *sgetspent __P ((__const char *__string));
extern struct spwd *fgetspent __P ((FILE *__fp));
extern struct spwd *getspent __P ((void));
extern struct spwd *getspnam __P ((__const char *__name));
extern int putspent __P ((__const struct spwd *__sp, FILE *__fp));
```

La funzione che useremo nell'esempio è: `getspnam` che ritorna una struttura `spwd` per il nome passato per argomento.

8.5 Esempio

Questo è un esempio di aggiunta del supporto shadow ad un programma che ne ha bisogno, ma non lo possiede.

Questo esempio usa il *Point-to-Point Protocol Server* (pppd-1.2.1d), che ha una modalità in cui esegue l'autenticazione *PAP* usando i nomi e le password degli utenti dal file `/etc/passwd` anziché dai file *PAP* o *CHAP*. Non dovreste aver bisogno di aggiungere questo codice a `pppd-2.2.0` perché c'è già.

Questa caratteristica del pppd probabilmente non è molto usata, ma se avete installato la *Shadow Suite*, non funzionerà comunque perché le password non si trovano più in `/etc/passwd`.

Il codice per l'autenticazione degli utenti sotto `pppd-1.2.1d` si trova nel file `/usr/src/pppd-1.2.1d/pppd/auth.c`.

Il seguente codice deve essere aggiunto all'inizio del file dove si trovano tutte le altre direttive `#include`. Abbiamo racchiuso gli `#include` tra direttive condizionali (i.e. vengono presi in considerazione solo se stiamo compilando per il supporto shadow).

```
#ifdef HAS_SHADOW
#include <shadow.h>
#include <shadow/pwauth.h>
#endif
```

Il passo successivo consiste nel modificare il codice vero e proprio. Stiamo ancora apportando cambiamenti al file `auth.c`.

Funzione `auth.c` prima delle modifiche:

```
/*
 * login - Controlla il nome e la password dell’utente nel database delle
 * password di sistema, e permette il login se l’utente e OK.
 *
 * restituisce:
 *      UPAP_AUTHNAK: Login fallito.
 *      UPAP_AUTHACK: Login riuscito.
 * In entrambi i casi, msg punta al messaggio appropriato.
 */
static int
login(user, passwd, msg, msglen)
    char *user;
    char *passwd;
    char **msg;
    int *msglen;
{
    struct passwd *pw;
    char *epasswd;
    char *tty;

    if ((pw = getpwnam(user)) == NULL) {
        return (UPAP_AUTHNAK);
    }
     /*
     * XXX Se non c’e nessuna password, li lascia collegare senza.
     */
    if (pw->pw_passwd == ’\0’) {
        return (UPAP_AUTHACK);
    }

    epasswd = crypt(passwd, pw->pw_passwd);
    if (strcmp(epasswd, pw->pw_passwd)) {
        return (UPAP_AUTHNAK);
    }

    syslog(LOG_INFO, "user %s logged in", user);

    /*
     * Scrive una voce wtmp per questo utente.
     */
    tty = strrchr(devname, ’/’);
    if (tty == NULL)
        tty = devname;
    else
        tty++;
    logwtmp(tty, user, "");         /* Aggiunge una voce di login al wtmp */
    logged_in = TRUE;

    return (UPAP_AUTHACK);
}
```

La password dell’utente viene messa in `pw->pw_passwd`, così tutto quello che dobbiamo fare in realtà è aggiungere la funzione `getspnam`. Questa metterà la password in `spwd->sp_pwdp`.

Aggiungeremo la funzione `pwauth` per eseguire l’autenticazione vera e propria. Questa eseguirà automaticamente l’autenticazione secondaria se il file shadow è impostato per farlo.

Funzione `auth.c` dopo le modifiche per il supporto shadow:

```
/*
 * login - Controlla il nome e la password dell’utente nel database delle
 * password di sistema, e permette il login se l’utente e OK.
 *
 * Questa funzione e stata modificata in modo da supportare la
 * Linux Shadow Password Suite se USE_SHADOW e definito.
 *
 * restituisce:
 *      UPAP_AUTHNAK: Login fallito.
 *      UPAP_AUTHACK: Login riuscito.
 * In entrambi i casi, msg punta al messaggio appropriato.
 */
static int
login(user, passwd, msg, msglen)
    char *user;
    char *passwd;
    char **msg;
    int *msglen;
{
    struct passwd *pw;
    char *epasswd;
    char *tty;

#ifdef USE_SHADOW
    struct spwd *spwd;
    struct spwd *getspnam();
#endif

    if ((pw = getpwnam(user)) == NULL) {
        return (UPAP_AUTHNAK);
    }

#ifdef USE_SHADOW
        spwd = getspnam(user);
        if (spwd)
                pw->pw_passwd = spwd->sp-pwdp;
#endif

     /*
     * XXX Se non c’e nessuna password, NON li lascia collegare senza.
     */
    if (pw->pw_passwd == ’\0’) {
        return (UPAP_AUTHNAK);
    }
```

```
#ifdef HAS_SHADOW
    if ((pw->pw_passwd && pw->pw_passwd[0] == '@'
         && pw_auth (pw->pw_passwd+1, pw->pw_name, PW_LOGIN, NULL))
        || !valid (passwd, pw)) {
        return (UPAP_AUTHNAK);
    }
#else
    epasswd = crypt(passwd, pw->pw_passwd);
    if (strcmp(epasswd, pw->pw_passwd)) {
        return (UPAP_AUTHNAK);
    }
#endif

    syslog(LOG_INFO, "user %s logged in", user);

    /*
     * Scrive una voce wtmp per questo utente.
     */
    tty = strrchr(devname, '/');
    if (tty == NULL)
        tty = devname;
    else
        tty++;
    logwtmp(tty, user, "");        /* Aggiunge una voce di login al wtmp */
    logged_in = TRUE;

    return (UPAP_AUTHACK);
}
```

Un attento esame rivelerà che abbiamo fatto un'altra modifica. La versione originale permetteva l'accesso (restituiva `UPAP_AUTHACK`) se non c'era NESSUNA password nel file `/etc/passwd`. Questo *non* è una buona cosa, perché un uso comune di questa caratteristica di login è quello di usare un account che permetta l'accesso al processo ppp e quindi confrontare il nome utente e la password forniti da PAP con il nome utente nel file `/etc/passwd` e la password nel file `/etc/shadow`.

Perciò se abbiamo impostato la versione originale in modo da eseguire, al posto della shell per un utente, ad esempio `ppp`, allora chiunque potrebbe ottenere una connessione ppp impostando la sua PAP con utente `ppp` e senza password.

Abbiamo risolto questo anche restituendo `UPAP_AUTHNAK` invece che `UPAP_AUTHACK` nel caso in cui il campo password fosse vuoto.

È abbastanza interessante il fatto che `pppd-2.2.0` abbia lo stesso problema.

Poi abbiamo bisogno di modificare il Makefile in modo che avvengano due cose: `USE_SHADOW` deve essere definita, e `libshadow.a` deve essere aggiunta al processo di link.

Editate il Makefile, e aggiungete:

```
LIBS = -lshadow
```

Quindi troviamo la riga:

```
COMPILE_FLAGS = -I.. -D_linux_=1 -DGIDSET_TYPE=gid_t
```

E la cambiamo in:

```
COMPILE_FLAGS = -I.. -D_linux_=1 -DGIDSET_TYPE=gid_t -DUSE_SHADOW
```

Ora eseguite il make ed installate.

9 Domande poste frequentemente (FAQ)

D: Ero abituato a controllare con quale tty *root* potesse collegarsi usando il file `/etc/securettys`, ma sembra non funzionare più, cosa è successo?

R: Il file `/etc/securettys` non fa assolutamente nulla ore che la *Shadow Suite* è installata. Le tty che *root* può usare sono ora situate nel file di configurazione di login `/etc/login.defs`. La voce in questo file potrebbe puntare ad un altro file.

D: Ho installato la *Shadow Suite*, ma ora non posso collegarmi, cosa ho dimenticato?

R: Probabilmente hai installato i programmi Shadow, ma non hai eseguito `pwconv` o hai dimenticato di copiare `/etc/npasswd` in `/etc/passwd` e `/etc/nshadow` in `/etc/shadow`. Inoltre, potresti aver bisogno di copiare `login.defs` in `/etc`.

D: Nella sezione su xlock, si dice di cambiare il gruppo proprietario del file `/etc/shadow` e di farlo diventare `shadow`. Non ho un gruppo `shadow`, cosa faccio?

R: Puoi aggiungerne uno. Semplicemente edita il file `/etc/group`, e inserisci una riga per il gruppo shadow. Devi assicurarti che il numero del gruppo non sia usato da un altro gruppo, e devi inserirlo prima della voce `nogroup`. Oppure puoi semplicemente impostare SUID root `xlock`.

D: Esiste una mailing list per la Linux Shadow Password Suite?

R: Sì, ma è per lo sviluppo e il beta testing della prossima Shadow Suite per Linux. Puoi iscriverti alla lista mandando una e-mail a: `shadow-list-request@neptune.cin.net` avente per subject: `subscribe`. La lista si occupa in realtà delle release Linux `shadow-AAMMGG`. Dovresti iscriverti se vuoi essere coinvolto in ulteriori sviluppi o se installi la Suite sul tuo sistema e vuoi avere informazioni sulle più recenti release.

D: Ho installato la *Shadow Suite*, ma quando uso il comando `userdel`, ottengo: userdel: cannot open shadow group file, (in italiano: userdel: non posso aprire il file shadow group); cosa ho sbagliato?

R: Hai compilato la *Shadow Suite* con l'opzione `SHADOWGRP` abilitata, ma non hai un file `/etc/gshadow`. Devi o editare il file `config.h` e ricompilare, oppure creare un file `/etc/group`. Vedere la sezione sui gruppi shadow.

D: Ho installato la *Shadow Suite* ma ho di nuovo le password codificate nel mio file `/etc/passwd`, cosa c'è che non va?

R: O hai abilitato l'opzione `AUTOSHADOW` nel file Shadow `config.h`, oppure il tuo `libc` è stato compilato con l'opzione `SAHDOW_COMPAT`. Devi capire qual è il problema, e ricompilare.

10 Messaggio di copyright

The Linux Shadow Password HOWTO is Copyright (c) 1996 Michael H. Jackson.

Permission is granted to make and distribute verbatim copies of this document provided the copyright notice and this permission notice are preserved on all copies.

Permission is granted to copy and distribute modified versions of this document under the conditions for verbatim copies above, provided a notice clearly stating that the document is a modified version is also included in the modified document.

Permission is granted to copy and distribute translations of this document into another language, under the conditions specified above for modified versions.

Permission is granted to convert this document into another media under the conditions specified above for modified versions provided the requirement to acknowledge the source document is fulfilled by inclusion of an obvious reference to the source document in the new media. Where there is any doubt as to what defines 'obvious' the copyright owner reserves the right to decide.

Ovvero (ni noti che l'unica licenza valida è quella in lingua originale):

Il Linux Shadow Password HOWTO è Copyright (c) 1996 di Michael H. Jackson.

È concesso il permesso di fare e distribuire copie testuali di questo documento, a patto che la nota sul copyright e questa nota di permesso siano mantenute su tutte le copie.

È concesso il permesso di copiare e distribuire versioni modificate di questo documento sotto le condizioni delle copie testuali dette sopra, a condizione che venga inclusa nel documento modificato una nota che dica chiaramente che il documento è una versione modificata.

È concesso il permesso si copiare e distribuire traduzioni di questo documento in un'altra lingua, sotto le condizioni specificate sopra per le versioni modificate.

È concesso il permesso di convertire questo documento in altri mezzi sotto le condizioni specificate sopra per le versioni modificate, a condizione che il nuovo mezzo contenga il riconoscimento del documento sorgente tramite un ovvio riferimento al documento sorgente stesso. Dove ci sia un qualunque dubbio sul significato di 'ovvio' il proprietario del copyright si riserva il diritto di decidere.

11 Varie e Riconoscimenti

Il codice di esempio per `auth.c` è tratto da pppd-1.2.1d e ppp-2.1.0e, Copyright (c) 1993 The Australian National University e Copyright (c) 1989 Carnegie Mellon University.

Grazie a Marek Michalkiewicz <marekm@i17linuxb.ists.pwr.wroc.pl> per aver scritto e mantenuto la *Shadow Suite* per Linux, e per la sua revisione e i suoi commenti a questo documento.

Grazie a Ron Tidd <rtidd@tscnet.com> per la sua utile revisione e il suo collaudo.

Grazie a tutti coloro che mi hanno mandato feedback per contribuire a migliorare questo documento.

Per favore, se avete commenti o suggerimenti, mandatemeli per posta.

saluti

Michael H. Jackson <mhjack@tscnet.com>