Linux Shadow Password HOWTO, Version française

Michael H. Jackson, mhjack@tscnet.com
Traduction française : Igor Genibel (Igor.Genibel@emi.u-bordeaux.fr) . v1.3, 3 Avril 1996

Ce document décrit comment obtenir, installer et configurer l'ensemble *Shadow Password* sous Linux. Il prend aussi en compte l'adaptation des autres logiciels et démons réseau accédant habituellement aux mots de passe des utilisateurs. Bien qu'ils ne fassent pas parti du paquetage, ces programmes ne fonctionnerons plus correctement après son installation. Ce document contient en plus un exemple de code pour mettre à jour vos applications pour qu'elles supportent le système de mots de passe cachés. Des réponses à des questions fréquemment posées sont fournies en fin de document.

Contents

1 Introduction

Ceci est le Shadow password HOWTO pour linux. Ce document décrit comment configurer votre système linux pour utiliser le système des mots de passe cachés. Quelques exemples d'utilisation concernant certaines caractéristiques de la *suite Shadow* (Note: ce terme n'est volontairement pas traduit afin de ne pas alourdir la compréhension du document) sont aussi inclus.

Lorsque vous installerez le paquetage des mots de passe cachés et lorsque vous vous servirez des utilitaires, vous devrez être logé root. Lorsque vous installerez le paquetage, vous devrez effectuer certains changements sur des utlitaires systèmes, il est donc très fortement recommendé de faire des copies de sauvegarde de programmes indiqués. Je vous recommande aussi de lire et de comprendre toutes les instructions avant de commencer.

1.1 Note du traducteur

Tout au long du document vous constaterez très certainement que je n'ai pas traduit le terme "Shadow Password". La raison est simple: le traduction de ce terme n'est pas élégante. Je pense qu'une simple explication (voir l'introduction) de celui-ci vaut beaucoup mieux qu'une "françisation" douteuse.

1.2 Changements par rapport aux versions précédentes

Ajouts:

- `Ajout de la sous-section Pourquoi vous ne devriez pas installer le système Shadow Password`
- `Ajout de la sous-section Mise à jour du programme xdm`
- `Ajout de la section Comment faire que les caractéristiques de la suite Shadow fonctionnent`
- `Ajout de la section contenant les questions les plus fréquemment posées`

Corrections/Mise à jour:

- `Correction des références html sur Sunsite`
- `Correction dans la section sur wu-ftp pour ajouter -lshadow dans le Makefile`
- `Correction de quelques fautes de frappes et de langage`
- `Chagement dans la section sur wu-ftp pour supporter le format ELF`
- `Mise à jour pour mettre en évidence les problèmes de sécurite de certains programmes de login`
- `Mise à jour pour recommander la Suite Linux Shadow par Marek Michalkiewicz`

1.3 Nouvelles versions de ce document:

La dernière version de ce document pourra toujours être récupérée via FTP anonyme sur:

- Version Originale: **sunsite.unc.edu**

/pub/Linux/docs/HOWTO/SHADOW-HOWTO

ou:

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

- Version Francaise: **ftp.lip6.fr**

  /pub/linux/french/docs/HOWTO/Shadow-Password-HOWTO*

Ou bien par le World Wide Web: *Linux Documentation Project Web Server* http://sunsite.unc.edu/mdw/linux.html , à la page:

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

sinon directement par moi, <mhjack@tscnet.com>. Il sera aussi posté dans le newsgroup: comp.os.linux.answers

Ce document est maintenant stocké en portant le nom Shadow-AAJJMM.

1.4 Commentaires ou suggestions.

Merci de m'envoyer tout commentaire, mise à jour ou suggestion: *Michael H. Jackson <mhjack@tscnet.com>* . Plus vite je reçois vos commentaires, plus vite je mets à jour ce document. Si vous avez des problèmes, merci de m'écrire directement, je suis très rarement à jour avec les newsgroups.

2 Pourquoi utiliser le système de mots de passe Shadow?

La plupart des distributions Linux actuelles ne contiennent pas le support des mots de passe shadow (La Slackware 2.3, Slackware 3.0 et d'autres distributions assez populaires...). Une des raisons est que le copyright concernant la suite Shadow n'était pas clair sur les droits de distribution. **Linux** utilise la licence GNU (quelques fois référencée sous le nom de *Copyleft*) qui permet à quiconque de le stocker sur n'importe quel support (comme un CD-ROM par exemple) et est responsable des droits pour cela.

Le mainteneur actuel de la *Suite Shadow*, *Marek Michalkiewicz <marekm@il7linuxb.ists.pwr.wroc.pl>* a reçu les sources de l'auteur originel sour un copyright style BSD permettant la redistribution. Maintenant que le problème de la distribution est résolu, il est probable que les prochaines distributions contiendront les shadow password par défaut. En attendant vous devrez l'installer vous-même.

Si vous avez installé votre distribution depuis un CD-ROM, vous pouvez trouver que, dans la mesure où la distribution n'a pas la *suite shadow* d'installée, quelques fichiers dont vous avez besoin pour installer la *suite shadow* peuvent se situer sur le CD-ROM.

Cependant, la suite shadow 3.3.1, 3.3.1-2 et shadow-mk possèdent toutes un problème de sécurité avec leur programme de login ainsi qu'avec d'autres programmes possédant le droit setuid root. En conséquence, elle ne doivent pas être utilisés plus longtemps.

Tous les fichiers nécessaires peuvent être récupérés via ftp anonyme ou via le World Wide Web.

Sur un système linux sans l'ensemble Shadow installé, les informations sur l'utilisateur, et en particulier le mot de passe sont stockées dans le fichier /etc/passwd. Le mot de passe est enregistré dans un format *encrypté*. Si vous demandez à un expert en cryptographie, il (ou elle) vous répondra que le mot de passe n'est pas *encrypté* mais *encodé*. En fait, lors de l'utilisation de crypt(3), le mot de passe est considéré comme la clé pour encoder un texte de valeur nulle. C'est la raison pour laquelle à partir de maintenant, j'utiliserai le terme *encodé*.

L'algorithme utilisé pour encoder le mot de passe fonctionne à sens unique, c'est-à-dire qu'il est très difficile à partir du mot de passe encodé de retrouver l'original. Vous trouverez plus d'informations à propos de l'algorithme utilisé dans la section 2.4 () les pages de manuel de crypt(3).

Lorsqu'un utilisateur définit un mot de passe, il est encodé avec une valeur aléatoire appelée *sel* (Note : *salt* en Anglais). C'est-à-dire qu'un même mot de passe pourrait être enregistré de 4096 façons différentes. La valeur du *sel* est alors enregistrée avec le mot de passe désormais encodé.

Lorsqu'un utilisateur se connecte et saisit son mot de passe, le *sel* est tout d'abord retrouvé à partir du mot de passe encodé. Alors, le mot de passe entré est encodé avec le *sel* précédemment retrouvé.

Il est, avec des moyens informatiques, difficile (mais pas impossible) de prendre un mot de passe *encodé* aléatoirement et de retrouver le mot de passe original. Quoi qu'il en soit, sur un système ayant de nombreux utilisateurs, il est probable que certains mots de passes soient évidents: un simple mot, un nom, ou une combinaison de mots simples.

Mais le pirate de système sait tout cela, et va simplement encoder un dictionnaire de mots de passe usuels en utilisant les 4096 possibilités de *sel*. Il va alors comparer les mots de passe encodés dans le fichier `/etc/passwd` par sa propre base de donnée. Quand il aura trouvé une équivalence, il aura le mot de passe d'un compte. Cela s'appelle une *attaque au dictionnaire*, et c'est une des méthodes les plus courantes pour accéder à un système sans autorisation.

En y réfléchissant, à un seul mot de passe de 8 caractères correspond 4096 mots de passes encodés de 13 caractères (c'est à dire 4096x13 octets). Donc un dictionnaire de 400 000 mots simples, noms, mots de passes, ou simple variations, tiendrait facilement sur un disque dur de 4Go. Le pirate a juste besoin de les trier et de les comparer.

Même sans avoir beaucoup d'espace disque, des utilitaires comme crack(1) peuvent en général casser pas mal de mots de passe sur un système contenant suffisamment d'utilisateurs. (En considérant que les utilisateurs du système sont autorisés à lire leur propre mot de passe).

Le fichier `/etc/passwd` contient aussi des informations comme l'identificateur de l'utilisateur (UID) et l'identificateur de groupe (GID) qui sont utilisés par de nombreux programmes. C'est pour cela que le fichier passwd *doit* être lisible par tout le monde. Si vous changiez les permissions de `/etc/passwd` de telle sorte que plus personne ne puisse le lire, la première chose que vous pourriez constater, c'est que la commande `ls -l` affiche désormais le user ID au lieu du nom !

Le kit Shadow résout ce problème en déplaçant les mots de passe encodés vers un autre fichier (en général `/etc/shadow`). Il n'y a que le root qui a les permissions de lecture et d'écriture sur le fichier *shadow*. Quelques programmes (comme xlock) nécessitent que le groupe *shadow* puisse lire et écrire dans le fichier `/etc/shadow`. Il est préférable que les programmes qui ont juste besoin de lire et vérifier le mot de passe soient lancés SGID *shadow* plutôt que SGID root.

En déplaçant les mots de passe vers le fichier `/etc/shadow`, nous écartons effectivement au pirate la possibilité d'avoir accès aux mots de passe encodés avec lesquels ils auraient pu faire une *attaque au dictionnaire*.

De plus, le *kit shadow* propose de nouvelles possibilités intéressantes:

- Un fichier de configuration pour configurer les options de login (`/etc/login.defs`),
- Des utilitaires pour ajouter, modifier et effacer des comptes utilisateurs,
- Gestion de la durée des mots de passe,
- Gestion de la durée d'un compte,
- Groupes shadow (optionnels),
- Mots de passe de double longueur (16 caractères),

- Meilleur contrôle sur la sélection du mot de passe d'un utilisateur,
- Mots de passe Dial-up,
- Programmes d'authentification secondaire.

Installer l'*ensemble shadow*, c'est contribuer à la sécurité de votre système, mais il y a bien d'autre choses à faire pour sécuriser votre système. Il y aura probablement une série de HOWTO discutant de la sécurité et des méthodes de sécurisation.

Pour le moment, pour avoir des informations sur la sécurité et linux, incluant les vulnérabilités connues du système, allez voir la:

Linux Security home page. `http://bach.cis.temple.edu/linux/linux-security/`

2.1 Pourquoi ne devriez-vous pas installer le système Shadow Password

Il y a quelques circonstances et quelques configurations qui font qu'installer la *Suite Shadow* n'est pas une bonne idée:

- La machine ne possède pas de comptes utilisateurs,
- Votre machine fonctionne sur un réseau local et utilise NIS (Network Information Service) pour récupérer ou fournir des noms d'utilisateurs et des mots de passe à d'autres machines sur le réseau (Cela peut actuellement être fait et n'améliore pas la sécurité pour autant),
- Votre machine est utilisée par des serveurs de terminaux afin de vérifier des utilisateurs via NFS (Network File Sytem), NIS, ou quelqu'autre méthode,
- Votre machine utilise d'autres logiciels pour valider les utilisateurs, et il n'y a pas de version disponible, et vous n'avez pas les sources.

2.2 Format du fichier /etc/passwd

Sur un système ne possédant pas la suite Shadow, voici le format du fichier `/etc/passwd`

```
username:passwd:UID:GID:full_name:directory:shell
```

En détail:

`username`

Le nom de l'utilisateur (login)

`passwd`

Le mot de passe encodé

`UID`

Identificatuer de l'utilisateur: user ID

`GID`

Identificatuer du groupe: group ID

`full_name`

Le nom complet de l'utilisateur (Prénom Nom) - Ce champ est appelé le champ GECOS (General Electric Comprehensive Operating System) et peut éventuellement contenir d'autres informations `directory`

Répertoire personnel de l'utilisateur

`shell`

Shell par défaut de l'utilisateur

Par exemple:

```
username:Npge08pfz4wuk:503:100:Full Name:/home/username:/bin/sh
```

`Np` est le *sel* et `ge08pfz4wuk` est le mot de passe *encodé*. Pour le même mot de passe, son équivalent encodé aurait tout aussi bien pu être `kbeMVnZM0oL7I`. Il y a 4096 possibilités d'encodage pour le même mot de passe. (Le mot de passe de cet exemple est 'password', un très *mauvais* mot de passe).

Une fois l'ensemble shadow installé, voilà à quoi ressemblera votre fichier `/etc/passwd`:

```
username:x:503:100:Full Name:/home/username:/bin/sh
```

Un `x` est venu remplacer le mot de passe encodé. Mis à part ça, le format du fichier `/etc/passwd` reste en fait inchangé. Ceci permet à tous les programmes qui lisent le fichier `/etc/passwd` sans avoir besoin d'accéder aux mots de passe de fonctionner correctement.

Les mots de passes encodés sont désormais dans le fichier `/etc/shadow`.

2.3 Format du fichier /etc/shadow

Le fichier `/etc/shadow` contient les informations suivantes:

```
username:passwd:last:may:must:warn:expire:disable:reserved
```

En détail:

`username`

Le Nom de l'Utilisateur

`passwd`

Le mot de passe encodé

`last`

Date de la dernière modification (en nombre de jours depuis le 1er janvier 1970).

`may`

Nombre de jours avant que le mot de passe puisse être modifié

`must`

Nombre de jours avant que le mot de passe doive être modifié

`warn`

Nombre de jours durant lesquels l'utilisateur est prévenu de l'expiration de son mot de passe.

`expire`

Nombre de jours entre l'expiration du mot de passe et la fermeture du compte.

`disable`

Date de la fermeture du compte (en nombre de jours depuis le 1er janvier 1970).

`reserved`

Champ réservé

Donc, l'exemple précedent devrait être:

```
username:Npge08pfz4wuk:9479:0:10000::::
```

2.4 Apercu de la fonction crypt(3)

extrait de la page de manuel de crypt(3)

"*crypt* est la fonction de cryptage du mot de passe. Elle est basée sur l'algorithme du DES (*Data Encryption Standard*) avec quelques modifications pour éviter les recherches matérielles de la clé.

La clé est le mot de passe de l'utilisateur

Le *sel* est composé de deux caractères choisis dans l'ensemble [a-zA-Z0-9./]. cette chaine de caractère est utilisée pour perturber l'algorithme de 4096 différentes façons.

En prenant les 7 derniers bits de chaque caractère du mot de passe, on obtient une clé de 56 bits. Cette clé est utilisée pour crypter une chaîne de caractère constante (généralement constituée de zéro). La valeur retournée pointe sur le mot de passe crypté: une série de 13 caractères ASCII imprimables (les deux premiers caractères correspondent au sel). La valeur retournée pointe sur une donnée statique dont le contenu est modifié à chaque appel.

Attention: Une clé de 56 bits correspond à: 2^56 donc 7.3e16 valeurs possibles. Les recherches exhaustives **sont possibles** en utilisant des ordinateurs à architecture massivement parallèle. Des logiciels comme `crack(1)` travaillent avec des clés qui sont généralement utilisées par les humains. C'est-à-dire que la sélection de mots de passe testés sont des mots simples, mots de passe fréquemment utilisés et des noms. L'utilisation d'un programme `passwd(1)` qui recherche des mots de passe trop simple est recommandé.

L'algorithme DES lui-même est très limité, ce qui fait qu'envisager l'utilisation de `crypt(3)` pour autre chose que de l'authentification de mots de passe n'est pas une bonne idée. Si vous envisagez d'utiliser `crypt(3)` pour un projet de cryptographie, ne le faites pas, procurez-vous plutôt un bon livre sur le cryptage de données et une des nombreuses bibliothèques DES."

Si vous recherchez un bon livre sur le cryptage de données, je vous recommande:

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

3 Se procurer le paquetage Shadow.

3.1 Historique du paquetage Shadow pour Linux?

N'UTILISEZ PAS LES PAQUETAGES DECRITS DANS CETTE SECTION, ILS CONTIENNENT DES PROBLEMES DE SECURITE

Le paquetage Shadow original a été écrit par `John F. Haugh II`.

De nombreuses versions peuvent être utilisées sur un système Linux:

- `shadow-3.3.1` est l'original
- `shadow-3.3.1-2` est le patch spécifique à Linux fait par *Florian La Roche (flla@stud.uni-sb.de)* et contient quelques améliorations.
- *shadow-mk* est le paquetage spécifique à Linux.

Le paquetage `shadow-mk` est en fait constitué du paquetage `shadow-3.3.1` distribué par `John F. Haugh II` patché avec `shadow-3.3.1-2` avec en plus:

- des corrections par *Mohan Kokal* <*magnus@texas.net*> rendant l'installation bien plus évidente,
- un patch par `Joseph R.M. Zbiciak` pour `login1.c` (login.secure) qui élimine les trous de sécurité -f, -h de /bin/login et quelques autres patches divers.

Le paquetage `shadow-mk` était précédemment recommandé, mais il doit être remplacé à cause d'un trou de sécurité du programme `login`.

Il y a des *trous de sécurité* dans les versions 3.3.1, 3.3.1-2 et shadow-mk qui sont dûs au programme `login`. Le bogue `login` implique de ne pas vérifier la longueur du nom de login. Cela entraîne un surpassement de la zone tampon qui provoque un crash ou pire encore. Il est dit que ce surpassement de zone tampon pourrait permettre à quiconque ayant un compte sur le système d'utiliser ce bogue ainsi que des bibliothèques partagées pour gagner un accès root. Je ne pourrais pas vous dire exactement comment cela est possible mais de nombreux systèmes Linux sont affectés. Mais les systèmes possédants ces *paquetages Shadow*, ainsi que la plupart des distributions pre-ELF *sans* le *paquetage Shadow* sont vulnérables !

Pour de plus amples informations sur cette publication ainsi que d'autres publications concernant les problèmes de sécurité de Linux, consultez la *Linux Security Home Page (Shared Libraries and login Program Vulnerability)* à <*http://bach.cis.temple.edu/linux/linux-security/Linux-Security-Faq/Linux-telnetd.html*>

3.2 Où trouver la Suite Shadow

La seule suite recommandée est en béta test, donc les dernières versions sont utilisables en environnement de production et ne contiennent pas de programme login vulnérable.

Le paquetage utilise la convention de notation suivante :

```
shadow-AAMMJJ.tar.gz
```

où `AAMMJJ` est la date de publication de la suite.

Cette version sera éventuellement la version 3.3.3 lorsqu'elle sera publiée après le béta test; et est maintenue par *Marek Michalkiewicz* <*marekm@il7linuxb.ists.pwr.wroc.pl*> . Elle est disponible sous la forme : *shadow-current.tar.gz* à l'adresse <*ftp://il7linuxb.ists.pwr.wroc.pl/pub/linux/shadow/shadow-current.tar.gz*> .

Les miroirs suivants sont aussi disponibles :

- ftp://ftp.icm.edu.pl/pub/Linux/shadow/shdow-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

Vous devez utiliser la version actuelle disponible.

Vous NE devez PAS utiliser une version plus ancienne que la version `shadow-960129` du fait qu'elles possèdent le problème de sécurité décrit plus avant.

Lorsque ce document fait référence à la *Suite Shadow*, je ferais référence à ce paquetage. Il est donc supposé que vous utilisez ce paquetage.

Pour information, j'ai utilisé le paquetage `shadow-960129` pour faire les instructions d'installation.

Si vous utilisiez précédemment le paquetage `shadow-mk`, vous devriez mettre à jour cette version et reconstruire tout ce que vous avez originellement compilé.

3.3 Ce qui est inclus dans le paquetage Shadow

La paquetage shadow contient les programmes de remplacement pour:

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

Mais il contient aussi des nouveaux programmes:

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

De plus, la bibliothèque: `libshadow.a` est incluse pour permettre de compiler les programmes nécessitant un accès en lecture/écritures aux mots de passe.

Les pages de manuel sont aussi incluses.

Il y a aussi un programme de configuration pour le program `login` intallé sous le nom de `/etc/login.defs`.

4 Compiler les programmes.

4.1 Extraire l'archive

La première étape après avoir récupéré les paquetage est l'extraction de l'archive. C'est une archive de format tar et compressée avec gzip. Donc tout d'abord déplacez la vers `/usr/src`, et tapez:

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

Ceci extraira l'ensemble dans le répertoire: `/usr/src/shadow-AAMMJJ`

4.2 Configurer le fichier config.h

La première chose à faire est d'écraser les fichiers `Makefile` et `config.h`:

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

Jetez un coup d'oeil au fichier `config.h`. Ce fichier contient les définitions pour quelques options de configuration. Si vous utilisez le paquetage shadow *recommandé*, je vous recommande de dévalider le support de groupes `shadow`; pour la première fois.

Par défaut les mots de passe pour les groupes caché sont validés. Pour les dévalider, éditez le fichier `config.h`, et remplacez `#define SHADOWGRP` en `#undef SHADOWGRP`. Je recommande de commencer sans les groupes cachés, mais si vous souhaitez réellement des mots de passe pour les groupes ainsi que des administrateurs de groupes vous pourrez ultérieurement valider l'option et recompiler le tout. Si vous la laissez validée, vous devez créer le fichier `/etc/gshadow`.

Valider l'option gérant les longs mots de passe n'est pas recommandée.

Ne PAS changer le champ: `#undef AUTOSHADOW`

L'option `AUTOSHADOW` était prévue pour que les programmes non adaptés aux mots de passe shadow puissent toujours fonctionner. Cela paraît intéressant en théorie mais ne fonctionne pas correctement. Si vous validez cette option, et qu'un programme focntionne avec les droits de `root`, il se peut qu'il utilise la fonction `getpwnam()` avec les droits `root`, et, plus tard, qu'il écrive la donnée modifiée dans le fichier `/etc/passwd`. Il ne possèdera donc plus les propriétés *shadow passwords*. *chfn* et *chsh* sont de tels programmes (Vous pouvez passer outre en échangeant l'uid réel et effectif avant d'appeler `getpwnam()` car `root` peut utiliser *chfn* et *chsh* aussi).

Le même avertissement est aussi valable si vous compilez la `libc`. Il y a une option `SHADOW_COMPAT` qui fait la même chose. Elle ne *doit PAS être* utilisée! Si vous commencez à remettre des mots de passe encodés dans le fichier `/etc/passwd`, cela pose un problème.

Si vous utilisez une `libc` de versions inférieure à 4.6.27, vous devriez faire un ou deux changements dans le fichier `config.h` ainsi que dans le `Makefile`. En ce qui concerne le fichier `config.h`, éditez le et remplacez:

```
#define HAVE_BASENAME
```

par

```
#undef HAVE_BASENAME
```

Dans le fichier `Makefile`, remplacez:

```
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 enc.c entry.c setup.c shell.c \
        pwent.c sub.c mail.c motd.c sulog.c shadow.c age.c pwpack.c rad64.c \
        tz.c hushed.c
```

par

```
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 enc.c entry.c setup.c shell.c \
        pwent.c sub.c mail.c motd.c sulog.c shadow.c age.c pwpack.c rad64.c \
        tz.c hushed.c basename.c
```

Ce changement ajoute le code contenu dans `basename.c` qui est contenu dans la `libc 4.6.27` ou plus.

4.3 Faire une copie de sauvegarde de vos programmes originaux.

Faites une copie de sauvegarde des fichiers qui vont être remplacés par le kit shadow. Sur un système Slackware 3.0:

- /bin/su

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

Le paquetage béta possède une cible *save* dans le `Makefile`, mais elle est commentée car les différentes distributions placent ces programmes à différents endroits.

Vous devriez aussi faire une copie de sauvegarde du fichier `/etc/passwd`, mais faites attention à le nommer différemment de façon à ne pas écraser l'original.

4.4 Lancer make

Vous avez besoin d'être `root` *pour la plupart de l'installation*

Lancez make pour compiler les exécutables du paquetage.

```
make all
```

Vous pourrez voir les avertissements suivants: `rcsid defined but not used`. Ce n'est rien, il survient seulement parce que l'auteur utilise un paquetage de contrôle de version.

5 Installer

5.1 Ayez une disquette de boot à portée de main

Si la mise à jour se déroulait mal, il serait utile d'avoir une disquette de boot. Si vous avez l'ensemble des deux disquettes boot/root que vous avez utilisées lors de l'installation de votre système, elles feront probablement l'affaire. Dans le cas contraire, vous devrez en générer: jetez un coup d'oeil au *Bootdisk-HOWTO* `http://sunsite.unc.edu/mdw/HOWTO/Bootdisk-HOWTO.html` (version francaise: *Bootdisk-HOWTO* `ftp://ftp.ibp.fr/pub/linux/french/docs/HOWTO/Bootdisk-HOWTO`) qui vous décrira la marche à suivre.

5.2 Supprimer les pages de manuel en double

Vous devriez aussi déplacer les pages de manuels qui vont être remplacées. Même si vous avez le courage d'installer le kit Shadow sans procéder à un quelconque sauvegarde, vous aurez à supprimer vos anciennes pages: elles ne sont pas écrasées car - dans la plupart des cas - elles sont enregistrées dans un format compressé.

Vous pouvez utiliser une combinaison de la commande `man -aW` et de la commande `locate` pour localiser les pages à effacer. Il est généralement plus aisé de retrouver les anciennes pages avant de lancer `make intall`.

Si vous utilisez la distribution Slackware 3.0, alors les pages de manuel que vous devez supprimer sont:

- /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

Regardez dans les sous répertoires `/var/man/cat[1-9]` Il est possible qu'il y ait des pages de manuel du même nom qui devront être effacées.

5.3 Lancer make install

C'est désormais le moment de taper: (faites ceci en tant que root).

```
make install
```

Ceci installera les nouveaux programmes, remplacera les anciens, définira les permissions de fichiers, et installera les pages de manuel.

Make install prend en compte l'installation des fichiers `include` pour les mettre au bon endroit dans `/usr/include/shadow`.

Si vous utilisez le paquetage béta, vous devez copier à la main le fichier `login.defs` dans les répertoires `/etc/` et être sûr que seul `root` peut le modifier.

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

Ce fichier est le fichier de configuration pour le programme *login*. Vous devriez regarder et faire des changements dans ce fichier pour votre propre système. C'est là que vous décidez sur quel `terminal` `root` peut se connecter, ainsi que d'autres paramètres de sécurité (comme l'expiration par défaut des mots de passe).

5.4 Lancer pwconv

La prochaine étape consiste à lancer `pwconv`. Ceci doit être fait en tant que `root`, et à partir du répertoire `/etc` :

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

`pwconv` lit les données du fichier `/etc/passwd` et les sépare en deux fichiers: `/etc/npasswd` et `/etc/nshadow`.

Un programme `pwunconv` permet de faire la démarche inverse: à partir du fichier `/etc/passwd` et `/etc/shadow`, il génère un unique `/etc/passwd`.

5.5 Renommer npasswd et nshadow

Après avoir lancé `pwconv`, vous avez normalement créé deux fichiers: `/etc/npasswd` et `/etc/nshadow`. Ses fichiers doivent être respectivement renommés en `/etc/passwd` et `/etc/shadow`. Faites aussi une copie de votre fichier `/etc/passwd` original, mais faites attention que seul le `root` puisse y avoir l'accès. Nous le déplacerons dans le répertoire personnel de root:

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

Vérifiez aussi que les permissions et les propriétaires des fichiers soient corrects. Si vous utilisez *X-windows*, le programme `xlock` doit pouvoir lire directement le fichier `/etc/shadow` (mais pas y écrire). La meilleure solution consiste à configurer le fichier `shadow` en utilisateur `root` et groupe `shadow`. Avant toute chose vérifiez que le groupe `shadow` existe bien (regardez dans le fichier `/etc/group`). Actuellement, il ne devrait y avoir aucun utilisateur appartenant à ce groupe.

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

Votre système est désormais équipé de mots de passe shadow. Déplacez-vous vers une autre console virtuelle et vérifier si vous pouvez vous loguer.

Si vous ne pouvez pas vous loguer c'est que la mise à jour s'est mal déroulée ! Pour revenir à un système de mot de passes non shadow, entrez ce qui suit:

```
cd /etc
cp ~passwd passwd
chmod 644 passwd
cd /usr/src/shadow-mk
make restore
```

Ceci restaurera le fichier `passwd` original, et restaurera tous les fichiers précédemment sauvegardés.

6 Les autres programmes à mettre à jour

Le paquetage shadow contient la plupart des programmes de remplacement aux programmes accédant aux mots de passe. Mais toutefois, il reste quelques programmes présents en général sur la plupart des systèmes qui nécessitent une mise à jour pour fonctionner correctement.

Si vous utilisez une *Distribution Debian* (et même si vous n'en utilisez pas), vous pouvez obtenir les sources des programmes que vous avez besoin de recompiler à : *<ftp://ftp.debian/org/debian/stable/source/>*

La but de cette section concerne la mise à jour des programmes: `adduser`, `wu_ftpd`, `ftpd`, `pop3d`, `xlock` `xdm` et `sudo`

Reportez vous à la section 8 () pour vous aider à mettre à jour les programmes qui nécessitent l'accès aux mots de passes (sans que le programme soit SUID root ou SGID shadow).

6.1 le programme Adduser (Slackware)

Les distributions Slackware (et probablement d'autres) contiennent un programme interactif `/bin/adduser` permettant d'ajouter facilement des utilisateurs. Une version Shadow de ce programme peut être trouvée sur: *ftp://sunsite.unc.edu/pub/Linux/ system/Admin/accounts/adduser.shadow-1.4.tar.gz* .

Je vous encourage à utiliser les programmes qui sont fournis par le paquetage shadow (`useradd, usermod` et `userdel`) à la place de `adduser`. Ils nécessitent un peu de temps pour savoir s'en servir, mais l'effort est d'autant plus grand que ces programmes effectuent un blocage de fichier sur `/etc/passwd` et `/etc/shadow`, ce que ne fait pas `adduser`

Lisez la section 7 () pour de plus amples informations.

L'installation est aisée:

```
tar -xzvf adduser.shadow-1.4.tar.gz
cd adduser.shadow.1.4
make adduser
chmod 700 adduser
make install
```

6.2 Le serveur wu_ftpd

La plupart des distributions Linux incluent le serveur `wu_ftpd`. Si votre distribution n'est pas "native shadow", votre serveur `wu_ftpd` n'est pas compilé pour le support shadow. `wu_ftpd` est lancé à partir de `inetd/tcpd` en tant que processus `root`. Si vous utilisez un ancien démon `wu_ftpd`, vous devrez de toute façon le mettre à jour car les vieilles versions ont un bug, le compte root pouvait être compromis (pour plus d'informations, consultez la page web: `http://bach.cis.temple.edu/linux/linux-security/Linux-Security-FAQ/Linux-wu.ftpd-2.4-Update.html`).

Heureusement, la seule démarche à faire est de récupérer les sources et de les compiler avec l'option shadow.

Le serveur `wu_ftpd` peut être récupéré sur Sunsite: *wu-ftp-2.4-fixed.tar.gz* `ftp://sunsite.unc.edu/pub/Linux/system/Network/file-transfer/wu-ftpd-2.4-fixed.tar.gz`

Une fois l'archive récupérée, placez là dans `/usr/src` et tapez:

```
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
```

éditer alors le fichier `./src/makefiles/Makefile.lnx` et changez la ligne:

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

par

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

Maintenant vous êtes près à lancer le script de compilation installer le résultat:

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

Sur mon système basé sur une Slackware 3.0 j'ai du faire ces modifications avant de lancer build: `build`:

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

Des problèmes ont été rapportés lors de la compilation de ce paquetage sous des systèmes ELF, mais la béta version de la prochaine publication fonctionne bien. Elle peut être trouvé à: *wu-ftp-2.4.2-beta-10.tar.gz* *<ftp://tscnet.com/pub/linux/network/ftp/wu-ftpd-2.4.2-beta-20.tar.gz>*

Une fois que vous avez récupéré le serveur, placez-le dans le répertoire `/usr/src/` et tapez:

```
cd /usr/src
tar -xzvf wu-ftpd-2.4.2-beta-10.tar.gz
cd wu-ftpd-beta-10
cd ./src/config
```

éditez alors le fichier config.lnx et remplacez:

```
#undef SHADOW_PASSWORD
```

par

```
#define SHADOW_PASSWORD
```

Allez alors dans le répertoire des `Makefiles`

```
cd ../Makefiles
```

et éditez le fichier `Makefile.lnx`. Modifiez alors:

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

par

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

Il ne reste plus qu'à compiler le programme et l'installer:

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

Notez que vous devrez contrôler le fichier `/etc/inetd.conf` afin d'être sûr que votre serveur `wu-ftpd` soit réellement présent. Il a été rapporté que certaines distributions placent les serveurs démons à d'autres endroits, et donc, wu-ftpd en particulier pourrait être nommé différemment.

6.3 ftpd standard

Si vous utilisez le serveur `ftpd standard`, tout d'abord, je vous recommande de passer au serveur `wu_ftpd`, mis à part les bugs cités précédemment, il est considéré comme plus sécurisé.

Si vous insistez et voulez garder la version standard - ou bien vous avez besoin du support NIS - le fichier est sur Sunsite:

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)

Si vous utilisez le 3eme *Post Office Protocol* (POP3), vous devrez recompiler le programme `pop3d`. `pop3d` est normalement lancé par `inet2/tcpd` dans un process `root`.

Il y a deux versions disponibles sur Sunsite:

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`

et

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

Les deux versions sont très simples à installer.

6.5 xlock

Si vous utilisez `X-window` et que vous ne mettez pas à jour `xlock`, vous devrez utiliser `CTRL-ALT-Fx` pour vous déplacer sur un autre terminal, vous loguer et tuer le process `xlock` (ou utiliser CTRL-ALT-BS pour tuer le serveur X). Mais par chance, la mise à jour d' `xlock` n'est vraiment pas compliquée.

Si vous utilisez XFree86 Versions 3.x.x, c'est probablement `xlockmore` qui est installé (c'est un superbe économiseur d'écran et un système de lock). Ce paquetage supporte *shadow* après recompilation. Si vous utilisez une vieille version `xlock`, je vous recommande celle-ci.

`xlockmore-3.7.tgz` disponible sur Sunsite:

`ftp://sunsite.unc.edu/pub/Linux/X11/xutils/screen-savers/xlockmore-3.7.tgz`

En gros, voilà comment procéder:

Récupérez `xlockmore-3.7.tgz` et copiez-le dans `/usr/src`, décompressez-le:

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

Editez le fichier: `/usr/X11R6/lib/X11/config/linux.cf`, et changez la ligne:

```
#define HasShadowPasswd    NO
```

en

```
#define HasShadowPasswd    YES
```

Alors, construisez les exécutables:

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

Maintenant, déplacez le tout vers le bon endroit, et mettez-à-jour les propriétaires et les permissions de fichier:

```
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
```

Votre xlock fonctionnera désormais correctement.

6.6 xdm

`xdm` est un programme qui présente un écran de *login* pour *W-Window*. Quelques sytèmes démarrent `xdm` lorsqu'il se situe dans un niveau spécifique (voir `/etc/inittab`).

Avec le *kit Shadow* installé, `xdm` doit être mis à jour. Heureusement, il est relativement facile de mettre à jour votre programme `xdm`.

`xdm.tar.gz` est disponible à: *<ftp://sunsite.unc.edu/pub/Linux/X11/xutils/xdm.tar.gz>;*

Récupérez `xdm.tar.gz` et placez-le dans le répertoire `/usr/src`, et décompressez-le:

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

éditez le fichier `/usr/X11R6/lib/X11/config/linux.cf`, et changez la ligne:

```
#define HasShadowPassword          NO
```

en

```
#define HasShadowPassword          YES
```

Vous pouvez alors compiler les exécutables;:

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

Copier alors l'exécutable:

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

`xdm` est exécuté en tant que `root` donc vous n'avez pas à changer les permissions.

6.7 sudo

Le programme `sudo` permet à l'administrateur système de laisser des utilisateurs lancer des programmes qui normalement nécessiteraient les permissions root. C'est intéressant car ça permet à l'administrateur de se limiter lui même l'accès root pendant qu'il permet aux utilisateurs de faire des opérations comme monter un disque.

`sudo` a besoin d'accéder aux mots de passe car il vérifie le mot de passe des utilisateurs quand il est invoqué.`sudo` fonctionne déjà SUID root, donc accéder au fichier `/etc/shadow` n'est pas un problème.

la mise à jour `sudo` pour shadow est disponible:

Cette version a été prévue pour fonctionner avec des mots de passe shadow, donc la seule chose à faire est de recompiler le tout (mettez-le dans `/usr/src`):

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

6.8 imapd (paquetage Email pine)

imapd est un serveur e-mail tout comme pop3d. imapd est compris dans l'ensemble Email pine. La documentation qui est fournie avec le paquetage prétend que la configuration par défaut pour un système linux fonctionne avec shadow. Or j'ai constaté que ce n'est pas vrai. De plus, je n'ai pas encore compris comment fonctionne la combinaison Makefile / Script Build et je n'ai pas réussi à le modifier pour qu'il supporte le format shadow.

Si quelqu'un arrive à faire cette mise à jour, merci de m'envoyer un email, je l'incluerai ici.

6.9 pppd (Serveur Point-to-Point protocol)

Le serveur pppd peut être configuré selon de nombreuses méthodes d'authentification: *Password Authentifocation Protocol* (PAP) et *Cryptographic Authentification Protocol* (CHAP). Le serveur pppd utilise en général les mots de passe stockés dans le fichier `/etc/ppp/chap-secrets` et `/etc/ppp/pap-secret`. Si vous utilisez cette méthode, ce n'est pas la peine de faire de mise-à-jour.

pppd vous permet aussi d'utiliser le paramètre *login* (soit en ligne de commande, soit dans le fichier de configuration). Si l'option `login` est utilisée, alors pppd utilisera pour le *PAP* le fichier `/etc/passwd` pour le nom d'utilisateur et le mot de passe. Bien sur ça ne fonctionnera plus sur un système mots de passe shadow. Pour pppd-1.2.1d, un ajout de code est nécesssaire.

L'exemple donné dans la prochaine section est la modification du code de `pppd-1.2.1d` (une vieille version de pppd).

`pppd-2.2.0` contient déjà le support shadow.

7 Faire en sorte que la suite shadow fonctionne

Cette section explique quelques éléments que vous souhaitez savoir depuis que vous avez la suite shadow sur votre système. De plus amples informations sont disponibles dans les pages de manuels.

7.1 Ajouter, modifier, et supprimer des utilisateurs.

La *Suite Shadow* a ajouté les commandes suivantes qui sont orientées "ligne de commande", pour ajouter, modifier et supprimer des utilisateurs. Vous avez aussi sûrement dû installer le programme `adduser`.

7.1.1 useradd

la commande `useradd` peut être utilisée pour ajouter des utilisateurs à votre système. Vous appelez aussi cette commande pour changer les paramètres par défaut.

La première chose à faire est d'examiner les paramètres par défaut et effectuer des changements pour votre propre système.

```
useradd -D
```

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

Les paramètres par défauts ne sont probablements pas ceux que vous souhaitez, donc, si vous commencez à ajouter des utilisateurs maintenant, vous aurez à spécifier toutes les informations pour chacun d'entre eux. C'est pour cela que nous pouvons et devons changer les valeurs par défaut.

Sur mon sytème:

- Je veux que le groupe par défaut soit 100,
- Je veux que les mots de passe expirent au bout de 60 jours,
- Je ne veux pas que le compte soit bloqué lors de l'expiration du mot de passe,
- Je veux que le shell par défaut soit `/bin/bash`

Pour effectuer ces changements, j'ai dû utiliser:

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

Maintenant, lancer `useradd -D` donne:

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

Ces valeurs par défaut sont stockées dans le fichier `/etc/defaults/useradd`

Maintenant vous pouvez utiliser `useradd` pour ajouter des utilisateurs à votre système. Par exemple, pour ajouter l'utilisateur `fred`, en utilisant les valeurs par défaut, vous devez utiliser ce qui suit:

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

Cela créera une entrée dans le fichier `/etc/passwd`:

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

Ainsi que cette entrée dans le fichier `/etc/shadow`:

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

Le répertoire d'accueil de `fred`sera créé et le contenu de `/etc/skel` sera copié à cet endroit grâce à l'option `-m`

De plus, lorsque l'on ne spécifie pas l'IUD, le prochain disponible est utilisé.

Le compte de `fred` est maintenant créé, mais `fred` ne peut pas se loger tant que nous ne dévérouillons pas le compte. Nous effectuons cela en changeant le mot de passe.

```
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: *******
```

Maintenant `/etc/shadow` contient:

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

`fred` est maintenant capable de se connecter et d'utiliser le système. La chose intéressante à propos de `useradd` et des autres programmes provenant de la Suite Shadow, est qu'ils effectuent les changements dans les fichiers `/etc/passwd` et `/etc/shadow` automatiquement. Donc si vous ajoutez un utilisateur, et qu'un autre utilisateur change son mot de passe au même moment, les deux opérations sont effectuées correctement.

Vous devriez plutôt utiliser les commandes fournies qu'éditer directement les fichier `/etc/passwd` et `/etc/shadow`. Si vous éditez le fichier `/etc/shadow` et qu'un utilisateur change son mot de passe au même moment, ce que vous sauverez, sera bien dans le fichier mais, le nouveau mot de passe de l'utilisateur sera perdu.

Voici un petit script intéractif permettant d'ajouter des utilisateurs en utilisant les commandes `useradd` et `passwd`:

```
#!/bin/bash
#
# /sbin/newuser - A script to add users to the system using the Shadow
#                 Suite's useradd and passwd commands.
#
# Written my Mike Jackson <mhjack@tscnet.com> as an example for the Linux
# Shadow Password Howto.  Permission to use and modify is expressly granted.
#
# This could be modified to show the defaults and allow modification similar
# to the Slackware Adduser program.  It could also be modified to disallow
# stupid entries.  (i.e. better error checking).
#
##
#  Defaults for the useradd command
##
GROUP=100        # Default Group
HOME=/home       # Home directory location (/home/username)
SKEL=/etc/skel   # Skeleton Directory
INACTIVE=0       # Days after password expires to disable account (0=never)
EXPIRE=60        # Days that a passwords lasts
SHELL=/bin/bash  # Default Shell (full path)
##
#  Defaults for the passwd command
##
PASSMIN=0        # Days between password changes
PASSWARN=14      # Days before password expires that a warning is given
```

```
##
#  Ensure that root is running the script.
##
WHOAMI=`/usr/bin/whoami`
if [ $WHOAMI != "root" ]; then
        echo "You must be root to add news users!"
        exit 1
fi
##
#  Ask for username and fullname.
##
echo ""
echo -n "Username: "
read USERNAME
echo -n "Full name: "
read FULLNAME
#
echo "Adding user: $USERNAME."
#
# Note that the "" around $FULLNAME is required because this field is
# almost always going to contain at least on space, and without the
# the useradd command would think that you we moving on to the next
# parameter when it reached the SPACE character.
#
/usr/sbin/useradd -c"$FULLNAME" -d$HOME/$USERNAME -e$EXPIRE \
        -f$INACTIVE -g$GROUP -m -k$SKEL -s$SHELL $USERNAME
##
#  Set password defaults
##
/bin/passwd -n $PASSMIN -w $PASSWARN $USERNAME >/dev/null 2>&1
##
#  Let the passwd command actually ask for password (twice)
##
/bin/passwd $USERNAME
##
#  Show what was done.
##
echo ""
echo "Entry from /etc/passwd:"
echo -n "    "
grep "$USERNAME:" /etc/passwd
echo "Entry from /etc/shadow:"
echo -n "    "
grep "$USERNAME:" /etc/shadow
echo "Summary output of the passwd command:"
echo -n "    "
passwd -S $USERNAME
echo ""
```

Utiliser un script pour ajouter des utilisateurs est préférable à l'édition des fichiers `/etc/passwd` et `/etc/shadow` ainsi qu'à l'utilisation de programmes comme `adduser` de la distribution Slackware.

Vous êtes libre d'utiliser et de modifier le script à en fonction de votre système.

Pour plus d'informations sur `useradd`, consultez le manuel en ligne.

7.1.2 usermod

Le programme `usermod` est utilisé pour modifier les informations relatives à un utilisateur. Les options sont les mêmes que pour `useradd`.

Disons que l'on souhaite changer le shell de `fred`. Vous devrez faire la chose suivante:

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

Maintenant l'entrée concernant `fred` dans le fichier `/etc/passwd` est devenue:

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

On change la date d'expiration du compte au 15/09/97:

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

Maintenant l'entrée concernant `fred` dans le fichier `/etc/shadow` est devenue:

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

Pour plus d'informations concernant `usermod`, consultez la page de manuel en ligne.

7.1.3 userdel

`userdel` fait exactement ce que vous voulez, il efface le compte d'un utilisateur. Utilisez simplement:

```
userdel -r username
```

L'option `-r` implique que tous les fichiers du répertoire d'accueil d'un utilisateur seront effacés. Les fichiers situés en dehors du répertoire d'accueil devront être cherchés et effacés manuellement.

Si vous souhaitez simplement vérouiller un compte au lieu de l'effacer, utilisez la commande `passwd`.

7.2 La commande passwd et la durée du mot de passe.

La commande `passwd` a pour but de changer les mots de passe. De plus, elle est utilisée par l'utilisateur `root` pour:

- Vérouiller et dévérouiller des comptes (-l et -u),
- Définir le nombre de jours de validité d'un mot de passe (-x),
- Définir le nombre de jours minimums pour le changement de mot de passe (-n),
- Définir le nombre de jours d'alerte concernant l'expiration d'un mot de passe(-w),
- Définir le nombre de jours après que le mot de passe soit expiré pour vérouiller le compte(-i),

- Permettre de voir les informations concernant un utilisateur dans un format clair(-S).

Par exemple, jetons un coup d'oeil à `fred`:

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

Cela signifie que le mot de passe de `fred` est valide, qu'il a été changé pour la dernière fois le 04/03/96, qu'il peut être changé à n'importe quel moment, qu'il expire au bout de 60 jours, que `fred` ne sera pas averti, et que le compte ne sera pas vérouillé lors de l'expiration du mot de passe.

Cela veut simplement dire que si `fred` se loge après l'expiration de son mot de passe, il lui sera demandé de taper un nouveau mot de passe.

Si nous souhaitons prévenir `fred` 14 jours avant l'expiration de son mot de passe, et vérouiller son compte 14 jours après l'avoir laissé expiré, nous devrions faire la chose suivante:

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

Maintenant les informations concernant `fred` sont changées en:

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

Pour de plus amples informations concernant `passwd`, se référer au manuel en ligne.

7.3 Le fichier login.defs

Le fichier `/etc/login.defs` est le fichier de configuration du programme `login` ainsi que celui de toute le *Suite Shadow*.

Le fichier `/etc/login.defs` contient les paramètres allant de l'apparance de l'invite à l'expiration par défaut concernant les mots de passe utilisateurs.

Le fichier `/etc/login.defs` est assez bien documenté de part ses propres commentaires. De plus, il y a quelques points a noter:

- Il contient des drapeaux qui peuvent être activé ou désactivé concernant la taille de journalisation,
- Il contient des pointeurs sur d'autres fichiers de configuration,
- Il contient les valeurs par défaut comme la durée d'un mot de passe.

Des informations précédantes on peut en déduire que c'est un fichier important, vous devez être sûr qu'il existe et que les valeurs sont celles que vous désirez pour votre système.

7.4 Les mots de passe pour les groupes

Le fichier `/etc/groups` peut contenir des mots de passe permettant à un utilisateur de devenir un membre d'un groupe particulier. Cette fonction est validée si vous validez la constante `SHADOWGRP` dans le fichier `/usr/src/shadow-AAMMJJ/config.h`.

Si vous définissez cette constante et que vous compilez, vous devez créer un fichier `/etc/gshadow` pour stocker les mots de passe pour les groupes, ainsi que les informations concernant l'administration du groupe.

Lorsque vous avez créé le fichier `/etc/shadow`, vous avez utilisé un programme appelé `pwconv`, il n'y a pas d'équivalent pour créer le fichier `/etc/gshadow`, mais ce n'est pas grave.

Pour créer le fichier `/etc/gshadow` initial, faites la chose suivante:

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

Une fois que vous créez un nouveau groupe, il sera ajouté dans le fichier `/etc/group` ainsi que dans le fichier `/etc/gshadow`. Si vous modifiez un groupe en ajoutant, retirant, ou en changeant le mot de passe du groupe, le fichier `/etc/gshadow` sera changé.

Les programmes `groups`, `groupadd`, `groupmod`, et `groupdel` sont fournis dans la *Suite Shadow* pour modifier les groupes.

Le format du fichier `/etc/group` est:

```
groupname:!:GID:member,member,...
```

où:

`groupname`
: Le nom du groupe,

`!`
: Le champs contenant normalement le mot de passe qui est maintenant stocké dans le fichier `/etc/gshadow`,

`GID`
: L'identificateur numérique du groupe,

`member`
: La liste des membres du groupe.

Le format du fichier `/etc/gshadow` est:

```
groupname:password:admin,admin,...:member,member,...
```

où:

`groupname`
: Le nom du groupe,

`password`
: Le mot de passe encodé,

`admin`
: La liste des administrateurs de groupe,

`member`
: La liste des membres du groupe.

La commande `gpasswd` est utilisée pour ajouter, retirer des administrateurs et des membres d'un groupe. `root` ou un administrateur du groupe peut ajouter ou retirer des membres du groupe.

Le mot de passe du groupe peut être changé en utilisant le programme `passwd` par `root` ou un administrateur du groupe.

En dépit du fait qu'il n'y ait pas encore de page de manuel pour `gpasswd`, tapez `gpasswd` sans paramètres pour obtenir la liste des options. C'est relativement facile de comprendre comment tout marche un fois que vous avez compris le format du fichier et les concepts.

7.5 Programmes de vérification de la structure

7.5.1 pwck

Le programme `pwck` est fourni pour vérifier la cohérence des fichiers `/etc/passwd` et `/etc/shadow`. Il vérifie chaque nom d'utilisateur ainsi que les points suivants:

- Le nombre correct de champs,
- Nom unique,
- Nom et groupe valide,
- Groupe primaire valide,
- Répertoire d'accueil valide,
- Shell valide.

Il prévient aussi lorsqu'un compte ne possède pas de mot de passe.

C'est une bonne idée de lancer `pwck` après avoir installé la *Suite Shadow*. C'est aussi une bonne idée de le lancer périodiquement, une fois par semaine ou par mois. Si vous utilisez l'option `-r`, vous pouvez utiliser `cron` pour lancer une analyse régulière et avoir un rapport sous forme de courrier.

7.5.2 grpck

`grpck` est le programme de vérification de la cohérance des fichiers `/etc/group` et `/etc/gshadow`. Il effectue les vérifications suivantes:

- Le nombre correct de champs,
- Unicité du nom de groupe,
- Validité de la liste des membres et des administrateurs.

Il possède aussi l'option `-r` pour des rapports automatiques.

7.6 Les mots de passe "Accès à distance"

Les mots de passe "Accès à distance" sont une autre ligne de défense des systèmes permettant la connexion à distance. Si vous avez un système qui permet à des utilisateurs de se connecter localement ou par l'intermédiaire d'un réseau, mais vous voulez contrôler qui peut appeler et se connecter, les mots de passe "Accès à distance" sont pour vous. Pour valider les mots de passe "Accès à distance", vous devez éditer le fichier `/etc/login.defs` et vous assurer que `DIALUPS_CHECK_ENAB` est positionnée à `yes`.

Les deux fichiers contenant les informations d'accès à distance sont `/etc/dialups` et `/etc/d_passwd`. Le fichier `/etc/dialups` contient les terminaux (un par ligne, avec l'entête "`/dev/`" supprimé). Si un terminal est listé alors, la vérification d'accès à distance est effectuée;.

Le second fichier `/etc/d_passwd` contient le chemin complet d'un shell, suivit d'un mot de passe optionnel.

Si un utilisateur se connecte à un terminal décrit dans une ligne du fichier `/etc/dialup` et que son shell est listé dans le fichier `/etc/d_passwd` alors l'accès lui est autorisé en fournissant le mot de passe correct.

Une autre possibilité utile des mots de passe "Accès à distance" est de spécifier un ligne qui ne permet qu'un certain type de connexion (PPP, ou UUCP par exemple). Si un utilisateur essaye d'avoir un autre type de connexion, (ie. une liste de shell), il doit connaitre un mot de passe pour l'utiliser.

Avant de pouvoir utiliser les possibilités de l'accès à distance, vous devez créer les fichiers.

La commande `dpaswd` est fournie pour assigner un mot de passe à un shell dans le fichier `d_passwd`. Lisez le manuel en ligne pour de plus amples informations.

8 Ajouter le support shadow à un programme en C.

Ajouter le support shadow à un programme C est assez facile. Le seul problème est que le programme doit être lancé par root (ou SUID root) pour qu'il puisse accéder au fichier `/etc/shadow`.

Ceci présente un réel problème, il faut faire très attention lors de la création de programmes SUID. Par exemple, il ne faut pas qu'un programme SUID root puisse permettre un accès au shell.

La meilleure solution pour qu'un programme puisse accéder aux mots de passe encodés sans être SUID root, est de lancer ce programme SUID shadow à la place. C'est le cas par exemple du programme `xlock`.

Dans l'exemple donné précédemment, `pppd-1.2.1d` fonctionne déjà SUID root, donc ajouter le support shadow ne le rendra pas plus vulnérable.

8.1 Les fichiers d'en-tête

Les fichiers d'en-tête doivent être stockés dans le répertoire `/usr/include/shadow`. Le fichier `/usr/include/shadow.h`, doit être un lien symbolique vers `/usr/include/shadow/shadow.h`.

Pour ajouter le support shadow à un programme, vous devez inclure les fichiers de header:

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

La meilleure solution est d'utiliser des directives de compilation pour compiler conditionnellement le code shadow (Il y aura un exemple par la suite).

8.2 La bibliothèque libshadow.a

Quand vous avez installé *l'ensemble shadow*, le fichier `libshadow.a` a été créé et installé dans le répertoire `/usr/lib`.

Lorsque vous compilez un programme avec le support shadow, vous devez préciser à l'éditeur de liens d'inclure la bibliothèque `libshadow.a` dans le lien:

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

Ceci dit, et vous le verrez par la suite dans notre exemple, la plupart des programmes plus ou moins gros utilisent un fichier `Makefile`, qui en général, utilise une variable appelée LIBS=... que vous pourrez modifier.

8.3 La structure shadow

La bibliothèque `libshadow.a` utilise une structure appelée `spwd` pour récupérer les informations contenues dans le fichier `/etc/shadow`. Voici la définition de la structure `spwd` provenant de `/usr/include/shadow/shadow.h`:

```
struct spwd
{
  char *sp_namp;          /* nom de login */
  char *sp_pwdp;          /* mot de passe encode */
  sptime sp_lstchg;       /* date de la derniere modification */
  sptime sp_min;          /* nombre de jours minimum entre les modifs */
  sptime spmax;           /* nombre de jours maximum entre les modifs*/
  sptime sp_warn;         /* nombre de jours de warning avant l'expiration
                             du mot de passe */
  sptime sp_inact;        /* nombre de jours d'utilisation du compte
                             apres l'expiration. */
  sptime sp_expire;       /* nombre de jours a partir du 01/01/70 jusqu'a
                                l'expiration du compte */
  unsigned long sp_flag; /* reserve pour une utilisation future */
};
```

L'*ensemble shadow* peut placer des données dans le champ `sp_pwdp` juste après le mot de passe encodé, le champ password pourrait contenir:

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

Cela signifie qu'en plus du mot de passe, le programme `/sbin/extra` sera appelé pour procéder à une authentification supplémentaire. Le programme appelé recevra comme argument, le nom d'utilisateur et un *switch* qui indiquera pourquoi il est appelé. Regardez le fichier `/usr/include/shadow/pwauth.h` et le code source de `pwauth.c` pour plus d'informations.

La fonction d'authentification `pwauth` est toujours utilisée avant la deuxième authentification..

8.4 Les fonctions Shadow.

Le fichier `shadow.h` contient aussi la déclaration des fonctions contenues dans la bibliothèque `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 fonction que nous allons étudier est `getspnam`, elle récupère une structure `spwd` à partir d'un nom donné.

8.5 Exemple

Voici un exemple d'ajout du support shadow à un programme qui en nécessite mais pour qui ce support n'existe pas par défaut.

Nous allons nous baser sur l'exemple du serveur `pppd-1.2.1d` (*Serveur Point-to-Point protocol*) configuré avec l'option *login*: il va chercher les mots de passe pour son authentification PAP dans le fichier `/etc/passwd` au lieu des fichiers PAP ou CHAP. Vous n'avez pas besoin d'ajouter ce code à `pppd-2.2.0`, c'est déjà fait.

Bien que cette possibilité de pppd ne soit pas très utilisée, elle ne fonctionnera plus dès lors que vous aurez installé l'ensemble shadow: les mots de passe ne sont plus stockés dans `/etc/passwd`.

La partie du code source d'authentification des utilisateurs avec `pppd-1.2.1d` se trouve dans le fichier `/usr/src/pppd-1.2.1d/pppd/auth.c`.

Le code qui suit doit être ajouté au début du fichier, là où sont toutes les autres directives `#include`.

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

Maintenant, il faut modifier le code actuel. Nous sommes toujours avec le fichier `auth.c`.

La fonction `auth.c` avant les modifications:

```
/*
 * login - Controle le nom d'utilisateur et le mot de passe par rapport
 * a ceux stockes sur le systeme.
 * Accepte la connection si l'utilisateur est OK.
 *
 * retourne:
 *      UPAP_AUTHNAK: Connection refusee.
 *      UPAP_AUTHACK: Connection Acceptee.
 * Dans un cas comme dans l'autre, msg pointe sur le message approprie.
 */
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 Si il n'y a pas de mots de passe, accepte la connection.
      */
    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);

    /*
     * Ecris une entree wtmp pour cet utilisateur.
     */
    tty = strrchr(devname, '/');
    if (tty == NULL)
        tty = devname;
    else
        tty++;
    logwtmp(tty, user, "");    /* Ajoute une entree wtmp de connection */
    logged_in = TRUE;

    return (UPAP_AUTHACK);
}
```

Le mot de passe de l'utilisateur est placé dans `pw->pw_passwd`, donc, nous devons ajouter la fonction `getspnam` qui placera le mot de passe dans `spwd->sp_pwdp`.

Nous rajouterons la fonction `pwauth` pour l'authentification actuelle. Une seconde authentification sera effectuée si le fichier shadow est configuré pour.

Voici la fonction auth.c apres les modifications pour le support de shadow:

```
/*
 * login - Controle le nom d'utilisateur et le mot de passe par rapport
 * a ceux stockes sur le systeme.
 * Accepte la connection si l'utilisateur est OK.
 *
 * Cette fonction a ete modifiee pour etre compatible avec les mots de
 * passe Shadow Linux si USE_SHADOW a ete defini
 *
 * retourne:
 *      UPAP_AUTHNAK: Connection refusee.
 *      UPAP_AUTHACK: Connection Acceptee.
 * Dans un cas comme dans l'autre, msg pointe sur le message approprie.
 */

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
    if ((spwd = getspnam(user)) == NULL) {
           pw->pw_passwd = "";
    } else {
    pw->pw_passwd = spwd->sp_pwdp;
    }
#endif

     /*
     *  XXX Si il n'y a pas de mots de passe, accepte la connection.
     */
    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);

    /*
     * Ecris une entree wtmp pour cet utilisateur.
     */
    tty = strrchr(devname, '/');
    if (tty == NULL)
        tty = devname;
    else
        tty++;
    logwtmp(tty, user, "");     /* Ajoute une entree wtmp de connection  */
    logged_in = TRUE;

    return (UPAP_AUTHACK);
}
```

En examinant précisément le code, vous verrez que d'autres modifications ont été effectuées. La version originale autorisait l'accès (en retournant `UPAP_AUTHACK`) quand il n'y avait pas de mots de passe dans le fichier `passwd`. Il ne fallait *pas* laisser ceci car utilisé avec l'option `login`, pppd utilise le nom d'utilisateur dans `/etc/passwd` et le mot de passe dans `/etc/shadow` pour son authentification PAP.

Donc si nous avions gardé la version originale, n'importe qui aurait pu établir une connexion ppp avec un mot de passe vide.

Nous avons arrangé ça en retournant `UPAP_AUTHNAK` à la place de `UPAP_AUTHACK` dans le cap ou le champ mot de passe est vide.

A savoir que pppd-2.2.0 possède le même problème.

Nous devons modifier le Makefile pour que deux choses soient prises en compte: `USE_SHADOW` doit être défini, et `libshadow.a` doit être ajouté au processus d'édition de liens.

Editez le Makefile, et ajoutez:

```
LIBS = -shadow
```

Alors, trouvez la ligne:

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

et replacez-la par:

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

Maintenant, lancez make et installez.

9 Foire Aux Questions

Q: J'essaye de contrôler sur quel terminal `root` peut se connecter en utilisant `/etc/securettys`, mais il semble que cela ne marcher plus. Qu'arrive-t-il ?

R: Les fichier `/etc/securettys` ne fait absolument rien lorsque la *Suite Shadow* est installée. Le terminal à partir duquel `root` peut se connecter et maintenant situé dans le fichier `/etc/login.defs`. L'entrée dans ce fichier peut pointer sur un autre fichier.

Q: J'ai installé la *Suite Shadow*, mais je ne peux plus me connecter, qu'ai-je oublié ?

R: Vous avez probablement installé les programmes, mais vous n'avez très certainement pas exécuté `pwconv` ou bien vous avez oublié de copier le fichier `/etc/npasswd` vers le fichier `/etc/passwd` ainsi que le fichier `/etc/nshadow` vers le fichier `/etc/shadow`. De plus vous aurez besoin de placer le fichier `login.defs` dans le répertoire `/etc/`.

Q: Dans la section sur xlock, il est dit de positionner le groupe propiétaire du fichier `/etc/shadow` à `shadow`. Je n'ai pas de groupe `shadow`, comment je fais ?

R: Vous pouvez en ajouter un. Editez simplement le fichier `/etc/group` et insérez une ligne pour le groupe `shadow`. Vous devez vous assurer que le numéro du groupe n'est pas déjà utilisé par un autre groupe, de plus vous devez l'insérer avant l'entrée `nogroup`. Ou bien vous pouvez positionner le bit `suid` du programme `xlock` à `root`.

Q: Y-a-t'il une liste de diffusion pour la suite Shadow password de Linux ?

R: Oui, mais c'est pour le développement et les tests des béta versions de la prochaine Suite Shadow pour Linux. Vous pouvez vous y inscrire en envoyant un courrier à: `shadow-list-request@neptune.cin.net` avec pour sujet `subscribe`. La liste est actuellement en cours de discussion à propos des parutions des series `shadow-AAMMJJ`. Vous devriez vous inscrire si vous souhaitez vous investir dans le développement ou bien si vous venez d'installer la suite sur votre système et que vous souhaitez vous tenir informé des nouvelles parutions.

Q: J'ai installé la *Suite Shadow*, mais lorsque j'utilise la commande `userdel`, j'obtiens, "*userdel: cannot open shadow group file*", qu'est-ce que j'ai fait de travers ?

R: Vous avez compilé la suite avec l'option `SHADOWGRP` de validé, mais vous ne possédez pas de fichier `/etc/gshadow`. Vous avez besoin d'éditer le fichier `config.h` et de recompiler la suite. Allez voir la section sur les groupes shadow.

Q: J'ai installé la *Suite Shadow* mais je retrouve des mot de passe encodés dans mon fichier `/etc/passwd`, qu'est-ce qui ne va pas ?

R: Soit vous avez compilé la suite avec l'option `AUTOSHADOW` du fichier `config.h`, soit votre `libc` a été compilée avec l'option `SHADOW_COMPAT`. Vous devez déterminer quel est votre problème et recompilez.

10 Copyright.

11 Divers et Remerciements

Les exemples de code pour `auth.c` proviennent de `pppd-1.2.1d` et `ppp-2.1.0e`, Copyright (c) 1993 de l'Université Nationale D'Australie et Copyright (c) 1989 de Université Carnegie Mellon.

Merci à *Marek Michalkiewicz <marekm@il7linuxb.ists.pwr.wroc.pl>*

pour écrire et maintenir la *Suite Shadow pour Linux* ainsi que pour ses commentaires sur ce document.

Merci à *Ron Tidd <rtidd@tscnet.com>* pour sa précieuse aide et ses tests.

Merci à tous ceux qui m'ont envoyé des commentaires qui ont permis d'améliorer ce document.

S'il vous plait, si vous avez des suggestions ou des commentaires, envoyez-moi un courrier.

amitiés

Mickael H. Jackson <mhjack@tscnet.com>