如何取得，安裝，設定 shadow 密碼

作者: Michael H. Jackson, *mhjack@tscnet.com*
譯者: Sung Min-Ju, *songmj@ms1.hinet.net*
v1.3, 3 April 1996 翻譯日期: 15 MAY 2000

這份文件主要描述如何取得，安裝和設定 *Shadow Suite* 密碼。它也描述取得和安裝需要存取使用者密碼之其他軟體和網路監控程式(network daemons)。這些其他軟體不是 Shadow Suit 的眞實部分，但是這些程式將需要被重新編譯用以支援*Shadow Suite*。這份文件亦包括一個程式範例：對程式加入 shadow 支援。本文結尾部分爲常問問題及答案。

目錄

1 簡介

這篇文章爲 Linux Shadow-Password-HOWTO。 主要是在描述爲何及如何於 Linux 系統加入 shadow 密碼支援。 其亦包括如何使用 *Shadow Suite's* 某些範例。

當進行 *Shadow Suite* 安裝及使用許多的 utility 程式時‧你必須以 *root* 權限簽入。 且進行 *Shadow Suite* 安裝時，系統軟體將被改變，因此強烈的建議 您照說明檔備分程式。在此強調在您開始作業前需先閱讀及了解所有的說明書。

1.1 更改上一版本部分

```
新增：
        新增一個子節：爲何您不要安裝 shadow
        新增一個子節：修改 xdm 程式
        新增一節：爲何您不要安裝 shadow
        新增一節：如何於未來運作 Shadow Suite
        新增一節：常問問題及答案


更正/修改：
        更正在 Sunsite 的 html 參考資料
        更正在 wu-ftp 節在 Makefile 新增 -lshadow
        更正較不嚴重拼音和措詞錯誤
        更改 wu-ftp 節用以支援 ELF
        修改在不同簽入(login)程式安全問題
        修改 Linux Shadow Suite 建議爲 Marek Michalkiewicz
```

1.2 新版文件

您可利用匿名檔進入該 FTP 站下載最新版文件： **sunsite.unc.edu**

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

or:

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

或透過網站： *Linux Documentation Project Web Server* `<http://sunsite.unc.edu/mdw/linux.html>`，網頁： *Shadow-Password-HOWTO* `<http://sunsite.unc.edu/linux/HOWTO/Shadow-Password-HOWTO.html>` 或與我聯絡： <mhjack@tscnet.com>. 亦可透過新聞群組張貼： `comp.os.linux.answers`

這些文件現在已經包於 Shadow-YYDDMM 套件中。

1.3 回覆

請將任何評語、修改或建議寄至： *Michael H. Jackson <mhjack@tscnet.com>* 我會盡快回覆並更正該文件。 如果你發現任何問題，請直接 email 給我，我會將此最新技術張貼於新聞群組。

2 爲何 shadow 你的 passwd 檔?

大部分目前 Linux 發行版本預設值並未包含*Shadow Suite* 安裝。 這些版本包括 Slackware 2.3, Slackware 3.0 和其他受歡迎的發行版本。 主要原因之一是在原始*Shadow Suite*版權聲明中並未很清楚的描述該軟體是 否需使用者付費。 Linux 使用 GNU 版權通常允續使用者可免費且任意使用相關套件。

*Shadow Suite*現在維護人員*Marek Michalkiewicz <marekm@i17linuxb.ists.pwr.wroc.pl>* 已經可以從在 BSD 樣式允需再使用版權之原始作者那接收原始碼。 目前版權發行已 解決，因此可以預期在未來版本之預設值將包括 password shadowing。 即使如此， 你仍需要自行安裝。

如果你的版本是從 CD-ROM 安裝。你可能發現即使目前版本並未有*Shadow Suite* 安裝，但你仍然可以在該片 CD-ROM 找到你需要安裝的*Shadow Suite*。

然而，所有*Shadow Suite 3.3.1, 3.3.1-2* 版本和 *shadow-mk* 伴隨簽入*(login)*程式和 其他*suid root*程式均有安全上問題，因此不應該使用太久。

所有必要檔案均可以透過匿名 FTP 站或網站找到。

在沒有安裝*Shadow Suite*的 Linux 系統，包括密碼的帳號使用者資訊通常儲存在 `/etc/passwd` 檔。 儲存的密碼爲 加密*(encrypted)*格式。 然而，如果你問一個密碼專家，他/她將 告訴你眞正的密碼檔只是編碼*(encoded)*而不是加密格式，因爲當使用 crypt(3) 時，文字檔倍設爲 null 且密碼是鍵值(key)。 因此，接下來我將在這篇文件中使用 編碼。

使用在密碼欄位編碼的演算法於技術上是使用*one way hash function*。這是一個 在單一方向計算簡單但逆向計算非常困難的演算法。關於這正確的演算法可以在 2.4 節或 於 crypt(3) 操作手冊找到。

當使用者挑選或指定一個密碼，系統將隨機產生一個值，叫做*salt*，將密碼進行 編碼。 這表示任何特定的密碼可以 4096 中不同方法儲存。 *salt* 值將儲存被編 碼的密碼。

當使用者簽入或提供一個密碼， *salt* 首先接收欲儲存編碼密碼。然後這提供密碼 會和 *salt* 值一起編碼，且比較已經編碼密碼。如果有 match，該使用者通過權限檢查。

隨機編碼和復原原始密碼是有計算複雜度的(但不是不可能的)。然而，在某些系統 很多使用者的密碼都設爲一般文字(或是一般文字的簡單變化)

系統駭客知道這件事，且將簡單的加密文字和一般使用4096 *salt* 值密碼之目錄。然 後他們將比較在資料庫之 `/etc/passwd` 檔之編碼密碼，只要他們找到一個比對，他們 可以找到一個帳號之密碼。這和*dictionary attack*有關，且用於未經許可存取系統取 得和展開常用的方法之一。

如果你想過一個 8 碼的密碼編碼成 4096 * 13 碼的字串，那麼一個用在描述 400,000 一般文字、名字、密碼和簡單變化的字典將需要 4GB 硬碟存取空間。 駭客需要做的只是分類字串跟檢查 比對結果。 自從 4GB 硬碟可以以美金 1000.00 以下買到後，對大多數系統駭客的意義可想而 知。

如果駭客首先發現你的 `/etc/passwd` 檔，他們只需要將眞正包含在 `/etc/passwd` 檔 `salt` 值的字典編碼，這方法可由擁有486級電腦或幾個有 幾百 MB 硬碟空間的十來歲的小孩適用。

即使沒有很大的磁碟空間，像 crack(1) 的工具程式通常可以在足夠使用者系統中至少破解一對 密碼（假設系統使用者倍允許挑選他們想要的密碼）。

`/etc/passwd` 檔也包括一些相關資訊，像使用者 ID 和群組 ID所使用的系統程式；因此 `/etc/passwd` 檔 必須 保持全世界可讀性. 如果你改變 `/etc/passwd` 檔以致於沒有人可以讀取，你將發現的第一件事是 `ls -l` 命令將取代名稱而顯示使用者 ID。

Shadow Suite 透過將密碼存至另一個檔（通常是 `/etc/shadow` 檔）來解決該 問題。 `/etc/shadow` 檔對任何人設定可讀檔，只有 *root* 權限可以讀寫 `/etc/shadow` 檔。某些程式（像 xlock）不需要改變密碼，只需要確認密碼即可。這些程式 可以以 *suid root* 執行或者你可以設定一個群組 *shadow* 用來唯讀 `/etc/shadow` 檔，然後這些程式可以執行 *sgid shadow*。

透過移動密碼至 `/etc/shadow` 檔，我們可以有力的備有駭客從存取編碼密碼到執行 *dictionary attack* 的證據。

除此之外， *Shadow Suite* 新增許多特色：

- 設定檔於設定爲簽入時預設(`/etc/login.defs`)
- 新增、修改和刪除使用者帳號群組之工具程式
- 密碼壽命計算及到期日
- 帳號到期日跟鎖死
- 隱藏群組密碼 (可選擇的)
- 兩倍長度密碼 (16 字元密碼) [不建議使用]
- 針對使用者密碼選擇有較好的控制
- 可撥接密碼
- 備用有效權限程式 [不建議使用]

安裝 *Shadow Suite* 貢獻爲有更安全系統，但是還有其他方法可以改善 Linux 系統的安全，且最終將有一系列的 Linux 安全 HOWTO's 將討論其他安全基準和相關文件版本.

針對目前其他 Linux 安全文件資訊，請參照網址：*Linux Security home page.* `<http://bach.cis.temple.edu/linux/linux-security/>`

2.1 爲何您不要 shadow 你的 passwd 檔

有一些狀況跟設定運用在安裝 *Shadow Suite* 將 不是 好主意： There are a few circumstances and configurations in which installing the *Shadow Suite* would *NOT* be a good idea:

- 主機沒有包含使用者帳號。
- 主機是在 LAN 上跑且使用網路資訊服務(Network Information Services, NIS)得到或供應使用者名稱和密給網路上的其他機器使用(事實上這還是 可以執行，但是實際上並不能增加任何安全)。
- 機器是使用終端主機來驗證使用者經由 NFS(Network File System), NIS 或 某些其他方法。
- 機器跑其他軟體驗證使用者且沒有任何 shadow 版本或原始碼可獲得。

2.2 格式化 /etc/passwd 檔

一個 non-shadowed /etc/passwd 檔格式如下所示：

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

其中

username
: 使用者(簽入)名稱

passwd
: 編碼密碼

UID
: 使用者編號

GID
: 預設存組編號

full_name
: 使用者全名 - 事實上這個欄位稱作 GECOS (General Electric Comprehensive Operating System) 欄位且可以儲存全名外的資訊。Shadow commands and manual pages refer to this field as the comment field.

directory
: 使用者根目錄 (絕對路徑)

shell
: 使用者簽入的環境 (絕對路徑)

舉例說明：

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

中，Np 是 salt 且 ge08pfz4wuk 是編碼密碼。 已編碼的 salt/password 就像 kbeMVnZM0oL7I 且這兩個字串是一樣的密碼。對相同 密碼可能有 4096 種可能的編碼。(本範例的密碼是 "passwaor"，這是比較不好 的密碼)。

只要 shadow suite 已經安裝， /etc/passwd 檔將被替代成：

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

本範例之第二欄位 x 現在只是一個 place holder。 /etc/passwd檔的格式 並未真的改變，只是不再包含 編碼密碼。這表示任何程式可以讀取 /etc/passwd 檔，但並不真的需要確認密碼是不是正確地運作。

這些密碼現在被重新放在 shadow 檔(通常是在 /etc/shadow 檔)。

2.3 格式化 shadow 檔

`/etc/shadow` 檔包括下列資訊：

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

其中

`username`
: 使用者名稱

`passwd`
: 編碼密碼

`last`
: 密碼上次更動日期，以從1970年1月1日算起的天數代表

`may`
: 密碼改變前天數

`must`
: 密碼最常使用天數

`warn`
: 代表期限前幾天就事先警告使用者

`expire`
: 超過密碼過期天數後，就關閉該帳號

`disable`
: 帳號關閉，以從1970年1月1日算起的天數代表

`reserved`
: 預備欄位

依照之前範例將變成：

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

2.4 回顧 crypt(3).

從 crypt(3) 使用者文件得到：

"*crypt* 是密碼加密方程式。 It is based on the *Data Encryption Standard* algorithm with variations intended (among other things) to discourage use of hardware implementations of a key search.

[The] key 是使用者輸入的密碼。 [編碼字串全是 NULLs]

[The] *salt* 是從 [a-zA-Z0-9./] 集合中選出的兩個位元字串。 該字串是用於擾亂在 4096 種不同方法之一個演算法。

透過得到 key 的每個位元的最低 7 bit[s]，可以建立 56-bit key。 這56-bit key 是用在重複加密一個常數字串(通常是包含所有靈的字串)。 傳回值指到加密密碼是一連串 可印出之13個 ASCII 字元(最前面的一兩個字元表示 salt 本身)。 透過每次的呼叫可將 傳回直指到靜態資料。

告訊息： Key space 包括 2**56 即等於 7.2e16 可能的值。與用重大的平行電腦 將 Key space 極盡的搜尋是可能。像 `crack(1)` 軟體用來搜尋部分的由人 產生之密碼的 key space 是可獲得的。因此，密碼的選擇至少應該避免使用一般字及名字。 一個用來檢查解開密碼選擇的 `passwd(1)` 程式使用是值得推薦的。

DES 演算法本身有一些使 `crypt(3)` 在任何其他密碼權限區隔介面很差選擇的使用 方面劇變。如果你計畫使用 `crypt(3)` 介面來加密計畫，以下描述千萬別這麼作： 得要一本加密好書和廣泛獲得 DES 程式庫 ." 一起使用。

大多 *Shadow Suites* 包括 16 位元的密碼長度兩倍之原始碼。 在 des 專 家建議避免使用當開始對較常密碼先簡單的編碼左半然後右半。由於 `crypt` 運作方法，這將造成 較差的 安全編碼密碼。除此之外，使用者有可能記住 16 位元密碼是 一件比較差的事情。

目前有允許權限理論用以取代某些更安全和支援較長密碼(例如 MD5 演算法)且保有和 `crypt` 方法相容的發展工作正在進行。

如果你正在尋找一本加密的好書，建議如下：

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

3 取得 Shadow Suite.

3.1 Shadow Suite for Linux 的歷史(暫不翻譯)

3.2 History of the Shadow Suite for Linux

DO NOT USE THE PACKAGES IN THIS SECTION, THEY HAVE SECURITY PROBLEMS

The original *Shadow Suite* was written by `John F. Haugh II`.

There are several versions that have been used on Linux systems:

- `shadow-3.3.1` is the original.
- `shadow-3.3.1-2` is Linux specific patch made by *Florian La Roche <flla@stud.uni-sb.de>* and contains some further enhancements.
- `shadow-mk` was specifically packaged for Linux.

The `shadow-mk` package contains the `shadow-3.3.1` package distributed by `John F. Haugh II` with the `shadow-3.3.1-2 patch` installed, a few fixes made by *Mohan Kokal <magnus@texas.net>* that make installation a lot easier, a patch by `Joseph R.M. Zbiciak` for `login1.c` (login.secure) that eliminates the -f, -h security holes in /bin/login, and some other miscellaneous patches.

The `shadow.mk` package was the *previously* recommended package, but should be replaced due to a *security problem* with the `login` program.

There are *security problems* with Shadow versions 3.3.1, 3.3.1-2, and shadow-mk involving the `login` program. This `login` bug involves not checking the length of a login name. This causes the buffer to overflow causing crashes or worse. It has been rumored that this buffer overflow can allow someone with an account on the system to use this bug and the shared libraries to gain *root* access. I won't discuss exactly how this is possible because there are a lot of Linux systems that are affected, but systems with these *Shadow Suites* installed, and most pre-ELF distributions *without* the *Shadow Suite* are vulnerable!

For more information on this and other Linux security issues, see the *Linux Security home page (Shared Libraries and login Program Vulnerability)* <http://bach.cis.temple.edu/linux/linux-security/Linux-Security-FAQ/Linux-telnetd.html>

3.3 如何取得 Shadow Suite？

目前建議 *Shadow Suite* 版本目前還是 BETA 測試版，然後，最近版本在生產環境 是安全的且沒有包含易受攻擊的 簽入(`login`) 程式。

該套件(package)使用慣例命名爲：

```
shadow-YYMMDD.tar.gz
```

其中 `YYMMDD` 是Suite 的發行日期。

目前 BETA 測試版本是 *Version 3.3.3*，且由 *Marek Michalkiewicz* <*marekm@i17linuxb.ists.pwr.wroc.pl*> 維護。

還可以從該處得到： *shadow-current.tar.gz* <ftp://i17linuxb.ists.pwr.wroc.pl/pub/linux/shadow/shadow-current.tar.gz>.

下列網站也可以找到相關資訊：

- *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*

你應該可以獲得目前最新的版本。

你應該不要是用比 `shadow-960129` 更舊版本，因爲它們有 簽入 的安全問題。

於參考資料方面，我用 `shadow-960129` 檔進行安裝介紹。

如果你之前使用 `shadow-mk`，你應該更信這個版本且重建編譯。

3.4 Shadow Suite包含什麼？

Shadow Suite 包括對下列功能之替代程式：

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

該套件還包括新程式：

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

除此之外，函式庫： libshadow.a 也包括需要存取使用者密碼之寫和編譯程式。

式之操作手冊也包含在其中。

也有對簽入程式的 configuration file，它將被安裝在 /etc/login.defs 檔。

4 編譯程式

4.1 解壓縮

在接收套件後第一個步驟就是 unpacking。該套件是 tar 檔案格式使用 gzip 壓縮，所以首先將該檔案移到 /usr/src，然後輸入：

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

這將會 unpack 到一個目錄：/usr/src/shadow-YYMMDD

4.2 設定 config.h 檔

第一件事是你需要複製 Makefile 和 config.h 檔：

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

然後你應該留意 config.h 檔。 該檔案包括某些設定選項的定義。如果你使用 建議 套件，我建議你在第一次設定關掉 group shadow support。

shadowed group passwords 預設值是開啓的。 在 config.h 檔關到這個設定， 且改變 #define SHADOWGRP 變成 #undef SHADOWGRP。我建議您一開始 關掉它們，然後如果你真的需要 group passwords 和 group administrators 時你在 開啓它們和重新編譯。 如果你開啓它， 你 必須 建立 /etc/gshadow 檔。

開啓長密碼的選項也不建議使用。

不要 改變 #undef AUTOSHADOW 的設定。

AUTOSHADOW 選項初始設計是用以讓 shadow 可以像 function 一樣執行。理論上 聽起來不錯，但是沒辦法正確的運作。 如果你開啓這個選項，且這個程式以 root 權限在執 行， 它會像 root 權限執行一項呼叫 getpwnam()，然後變更 /etc/passwd 檔 (有 *no-longer-shadowed* 密碼)。 這類程式包括 chfn 和 chsh。(如果 root 在呼叫 getpwnam() 之前使用 chfn 和 chsh，使用者帳號將 沒有辦法眞實且有效交易。)

如果你要建立 libc，同樣的警告也有效，它有個 SHADOW_COMPAT 作相同的事。 它 不應該被使用！如果你開始從你的 /etc/passwd 檔擷取編碼密碼， 這會是個問題。

如果你正使用比 4.6.27 還高的 libc 版本，你將需要在 config.h 和 Makefile兩個檔做很多改變。 在 config.h 檔的編輯和改變： 從：

```
#define HAVE_BASENAME
```

到：

```
#undef HAVE_BASENAME
```

然後在 Makefile 檔的改變：

```
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 rad64.c \
        tz.c hushed.c

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 rad64.c \
        tz.c hushed.c basename.c
```

這些包含在 basename.c 程式碼的改變均需倍包括在 libc 4.6.27 內。

4.3 備份原始程式

在 shadow suite 要更新時，製作程式備份將是一個很好的點子。在 Slackware 3.0 系統中， 這些檔案是：

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

這 BETA 套件已經有個 儲存 在 Makefile 的目的檔，但是 因為不同的版本通常將程式放在不同的地方，因此常被人評論。

你應該備份你的 /etc/passwd 檔，但是你要很小心地命名，不然如我你將它放在 相同目錄，你將無法重寫 passwd 命令。

4.4 執行 make

你需要以 *root* 權限簽入以執行安裝程序.

執行 make 來編譯套件中的執行檔：

```
make all
```

你可能會看到警告語： `rcsid defined but not used`. 這沒關係， 因爲作者使用版本控制套件才會發生。

5 安裝

5.1 手邊準備一個開機片

如果眞的發生嚴重錯誤，有個開機磁片之有用的。如果你要 boot/root 合併的安裝，你可以 參考 *Bootdisk-HOWTO* `<http://sunsite.unc.edu/mdw/HOWTO/Bootdisk-HOWTO.html>` 以製作 root 開機之開機片。

5.2 移除複製的 man pages

你也應該將操作手冊更新，即使你很厲害到足以不用備份安裝 Shadow Suite，你仍然要將 將舊的操作手冊移除，因爲新版的操作手冊無法正常的覆蓋舊版本。

你可以使用一個組合： `man -aW` 命令 和 `locate` 命令 尋找需備移除 的操作手冊。在你執行 `make install` 前找出較舊版本比較容易。

如果你使用 Slackware 3.0 版本，然後你要移除的操作手冊是：

- /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

在 `/var/man/cat[1-9]` 次目錄也有相同的名字需要被刪除。

5.3 執行 make install

現在你已經準備要輸入： (以 root 權限執行)

```
make install
```

這將安裝最新和更新程式且修復檔案許可權。它也會安裝操作手冊。

這也將在安裝時考慮將 Shadow Suite 包含檔案放在正確的位置 /usr/include/shadow。

用 BETA 套件，你需要手動複製 login.defs 這個檔到 /etc 這個目錄下， 而且要確認只有 *root* 權限可以改變它。

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

這個檔案是 簽入 程式的 configuration file。 你應該檢查跟確定這個檔的改變狀況。 這是你決定哪個 tty 的 root 可以從哪裡簽入和設定其他安全方針的地方(像預設密碼的到期日)。

5.4 執行 pwconv

接下來的步驟是執行 pwconv。 這也需以 *root*執行且最好在 /etc 目錄下執行：

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

pwconv 擷取你的 /etc/passwd 檔且刪除某些欄位爲了建立兩個檔案： /etc/npasswd 和 /etc/nshadow.

一個 pwunconv 也提供你建立一個 /etc/passwd 和 /etc/shadow 組合的正常 /etc/passwd 檔案。

5.5 重新命名 npasswd 和 nshadow

現在你已經執行 pwconv，而且你已經建立 /etc/npasswd 和 /etc/nshadow 檔案。這需要複製到 /etc/passwd 和 /etc/shadow 檔。 我們也需要複製原始 /etc/passwd 檔，而且確定只有 root 可以讀它。 我們將該檔案放在 root 得根目錄：

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

你也應該確定檔案的擁有者跟存取權限是正確的。 如果你將要使用 *X-Windows*， xlock 和 xdm 可能需要讀取 shadow 檔(但需不要寫入該檔)。

有兩個方法可以做。 你可以設定 xlock 的 suid 是 root (xdm 通常以 root 權限執行)。 或者你可以使 root 成爲 shadow 的群組可以擁有 shadow 檔，但是在作這之前，要確定你已經有個 shadow group (可以在 /etc/group 檔看到)。 不應改有任何使用者眞的在 shadow group。

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

你的系統現在有 shadow 的密碼檔囉。 你現在 應該 重開一個終端機視窗和確認你 可以簽入(login)。 請馬上確實的作這件事！

如過你不能，可能某些事情有錯誤囉！ 爲了要回到 to a non-shadowed 狀態，請作下列步驟：

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

你將重新儲存這些檔案到先前你儲存它們的正確位置。

6 其他你需要升級(upgrade)或補強(patch)程式

即使 shadow suite 對大部分需要存取密碼檔的程式可以包含更新程式，但是仍有一些需要 存取密碼檔的額外程式在系統中。

如果你正執行 *Debian* 版本 (或者即使你不是)，你可以找到 Debian 需要 rebuild 的原始碼：ftp://ftp.debian.org/debian/stable/source/

這節剩餘的部分在討論更新 `adduser`, `wu_ftpd`, `ftpd`, `pop3d`, `xlock`,`xdm` 和 `sudo` 程式以便於讓這些程式支援 shadow suite。

請看 8 (Adding Shadow Support to a C program) 這節，主要是在 討論如何將 shadow 支援放到其他需要程式(雖然這些程式需要以 SUID root 執行或 SGID shadow 需正確存取 shadow 檔)。

6.1 Slackware adduser 程式

Slackware 版本包含一個新增使用者的交互程式叫做 `/sbin/adduser`。該程式的 shadow 版本可以在 *ftp://sunsite.unc.edu/pub/Linux/system/ Admin/accounts/adduser.shadow-1.4.tar.gz*找到。

我很鼓勵你使用 *Shadow Suite* 供應的程式(例如`useradd`, `usermod`, 和 `userdel`) 來取代 slackware 的 `adduser` 程式。 它們只需花一點時間學習，但是它將值得的，因爲你可以看到更多控制和在正確展現鎖住 `/etc/passwd` 和 `/etc/shadow` 的檔案(`adduser` 就沒有辦法囉)。

參考7 (Putting the Shadow Suite to use) 該節得到更多相關資訊。

但是如果你已經擁有囉，接下來是你要做的部分：

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

6.2 wu_ftpd Server

大部分的 Linux 系統都有 `wu_ftpd` server。 如果你的版本沒有附帶 shadow 安裝， 那麼你的 `wu_ftpd` 就沒有辦法對 shadow 編譯。 `wu_ftpd` 是從 `inetd/tcpd` 開始且以 *root* 權限執行的程式。 如果你正在

跑一個舊版的 wu_ftpd daemon， 無論如何你將要更新它因爲較舊版本有 bug 將危及 *root* 權限。(參考 *Linux security home page* <http://bach.cis.temple.edu/linux/linux-security/Linux-Security-FAQ/Linux-wu.ftpd-2.4-Update.html> 得到更多相關資訊)。

幸運的是你只需要取的有嵌入 shaow 的原始碼和重新編譯就可以囉！

如果你不是正在執行 ELF 系統， wu_ftp server 可以從 Sunsite 網址 *wu-ftp-2.4-fixed.tar.gz* <ftp://sunsite.unc.edu/pub/Linux/system/Network/file-transfer/wu-ftpd-2.4-fixed.tar.gz> 找到。

當你獲得這個 server，把它放在 /usr/src目錄，然後輸入：

```
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
```

然後編輯 ./src/makefiles/Makefile.lnx，和改變

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

這一行到：

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

現在你已經準備好執行 script 建立跟安裝：

```
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
```

這是用在 Linux shadow configuration file、編譯和安裝 server。

在我的 Slackware 2.3 系統，我也需要在執行前作下列步驟：

build:

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

在 ELF 系統下會有編譯問題的報告，但是下一版的 Beta 版則可以正確的執行。 可以從 *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> 找到。

當你獲得這個 server，把它放在 /usr/src目錄，然後輸入：

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

然後編輯 config.lnx，和改變：

```
#undef SHADOW.PASSWORD
```

這一行到：

```
#define SHADOW.PASSWORD
```

然後，

```
cd ../Makefiles
```

且編輯 `Makefile.lnx` 檔和改變

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

這一行到：

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

然後建立(build)和安裝(install)：

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

註：你應該檢查你的 `/etc/inetd.conf` 檔來確認你的 wu.ftpd server 是不是真的活著。 有些版本可能將 server daemons 放在不同的地方或用不同的名字表示。

6.3 標準 ftpd

如果你正在執行標準的 `ftpd` server，我將建議你更新 `wu_ftpd` server。 離開上述的 bug，系統會比較安全。

如果你堅持在標準模式，或者你需要 *NIS* 支援，在 Sunsite *ftpd-shadow-nis.tgz* `<ftp://sunsite.unc.edu/pub/Linux/system/Network/file-transfer/ftpd-shadow-nis.tgz>` 有參考資料。

6.4 pop3d (Post Office Protocol 3)

如果你需要支援第三版 *Post Office Protocol (POP3)*，你將需要重新編譯 `pop3d` 程式。 `pop3d` 可以透過 `inetd/tcpd` 以 `root` 權限正常的執行。

從 Sunsite 有兩個版本可以獲得： *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>` 和 *pop3d+shadow+elf.tar.gz* `<ftp://sunsite.unc.edu/pub/Linux/system/Mail/pop/pop3d+shadow+elf.tar.gz>`

這兩個都很簡單可以安裝。

6.5 xlock

如果你安裝 shadow suite，然後執行 *X Windows System* 和 lock 螢幕沒以更新你的 xlock 檔， 你將必須使用 CNTL-ALT-Fx 去切換 另一個 *tty*，簽入(login)和殺掉(kill) xlock process (或使用 CNTL-ALT-BS 殺掉 X server)。 很幸運的這也很容易可以更新你的 xlock 程式。

如果你正執行 XFree86 Versions 3.x.x，且正正確使用 xlockmore (是一個很棒的螢幕保護程式). 這個套件支援 *shadow*，只要重新編譯即可。如果你有任何較老的 xlock 版本，我建議你更新下列版本：

xlockmore-3.5.tgz 可以從 <ftp://sunsite.unc.edu/pub/Linux/X11/xutils/screensavers/xlockmore-3.7.tgz> 網站獲得。

基本上這是你所要的。

擷取 xlockmore-3.7.tgz，並將它放在 /usr/src 目錄並解壓縮：

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

編輯這個檔： /usr/X11R6/lib/X11/config/linux.cf, 和改變

```
#define HasShadowPasswd    NO
```

這一行到：

```
#define HasShadowPasswd    YES
```

然後建立可執行檔：

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

然後搬移所以檔案到正確目錄且更新檔案擁有者及執行權限：

```
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
```

你的 xlock 將可以正確的運作囉！

6.6 xdm

xdm 是一個可以表示在 X-Windows 簽入畫面的程式。某些系統開始 xdm 當系統被告知道一個特定的執行水準(參考 /etc/inittab)。

伴隨著 *Shadow Suite* 安裝， xdm 需要被更新。 很幸運的這也很容易可以更新你的 xdm 程式。

xdm.tar.gz 可以從下列網址獲得： <ftp://sunsite.unc.edu/pub/Linux/X11/xutils/xdm.tar.gz>

擷取 xdm.tar.gz 檔並將它放在 /usr/src目錄然後解壓縮：

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

編輯這個檔： /usr/X11R6/lib/X11/config/linux.cf，且改變

```
#define HasShadowPasswd    NO
```

這一行到：

```
#define HasShadowPasswd    YES
```

然後建立可執行檔：

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

然後搬移所有檔案正確目錄：

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

xdm 以 *root* 權限在執行，所以你不需要改變檔案存取權限。

6.7 sudo

sudo 程式允許系統管理員讓使用者可以以 root 權限正常的執行程式。 這是非常方便的因爲它可以限制管理者執行 root 帳號本身權限，還可以允許使用者作 像 mounte drives 的事情。

sudo 需要讀取密碼因爲在執行時需確認使用者密碼。 sudo 已經執行 SUID root， 所以存取 /etc/shadow 黨不是問題。

sudo 支援 shadow suite 可在下列網址取得： <ftp://sunsite.unc.edu/pub/Linux/system/Admin/sudo-1.2-shadow.tgz>

警告：當你安裝 sudo 你的 /etc/sudoers 檔將取代預設值，所以你 需要備份原始程式。如果你有家任何設定在預設程式，你可能要編輯 Makefile 且移除複製該檔到 /etc 的這行。

該套件已經對 shadow 進行設定，所以只要重新編譯該套件即可 (把它放在 /usr/src 目錄)：

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

6.8 imapd (E-Mail [pine package])

imapd 是一個像 pop3d 的 email server。 imapd 隨著 *Pine E-mail* 套件發生。 其操作手冊在介紹該套件時即包含 shadow 支援。然而，我發現這不全然正確。 再加上在編譯時加上 libshadow.a 函式庫 和該套件結合 build script / Makefile 是非常不容易的。所以對 imapd 加入 shadow 支援是不太可能的。

如果有任何答案，可以 Email 給我，我會將該解答放到這裡。

6.9 pppd (Point-to-Point Protocol Server)

pppd server 可以使用幾種權限設定： *Password Authentication Protocol* (PAP) 和 *Cryptographic Handshake Authentication Protocol* (CHAP)。 pppd server 從 `/etc/ppp/chap-secrets` 和/或 `/etc/ppp/pap-secrets` 檔讀取密碼字串。 如果你正使用預設 pppd 的執行，就沒有必要再重新安裝 pppd。

pppd 以允許你使用 *login* 參數。 如果 *login* 選項被選取， pppd 將使用 `/etc/passwd` 檔的帳號密碼給 *PAP*. 當然在不久密碼檔會是 shadowedm。pppd-1.2.1d 檔已經加入對 shadow 的支援。

下一節加入支援 shadow 的範例是針對 `pppd-1.2.1d` (一個較老版本的 pppd).

`pppd-2.2.0` 版就已經包括 shadow 支援囉。

7 將 Shadow Suite 放進來使用。

這節描述你需要知道有些程式在安裝時就已經有 *Shadow Suite*。 大部分的資訊在操作手冊可以找到。

7.1 新增、修改和刪除使用者

Shadow Suite 新增下列指令用來新增、修改和刪除使用者。 這也是可以安裝 `adduser` 程式。

7.1.1 useradd

`useradd` 使令可用在系統中新增使用者。 你也可以採用此指令來改變預設字串。

你應該做的第一件事是檢查預設值設定和針對你的系統進行改變：

```
useradd -D
```

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

預設值不全是你要的，所以如果你開始新增使用者，你必須詳閱每個使用者資訊。 而且，我們可能和應該改變設定值。

在我的系統上：

- 我要預設群組是 100
- 我要密碼每到 60 天就到期
- 我不要鎖住帳號因爲密碼會到期
- 我要預設 shell 是 `/bin/bash`

爲了這些改變，我要使用：

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

現在執行 useradd -D 將得到：

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

儘管依照你需要修改，預設值將存在 /etc/default/useradd.

先在你可以使用 useradd 來新增系統使用者。舉例說明，新增一使用者 fred 使用預設值方式如下：

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

這將在 /etc/passwd 檔中的一行建立如下：

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

且在 /etc/shadow 檔中的一行建立如下；

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

fred的根目錄將被建立且 /etc/skel 的內容將被複製因爲指令句中有 -m 設定。

因爲我們並未詳述 UID，系統會直接尋找下一個可獲得的編號。

fred的帳號被建立囉，但是 fred 仍然不能簽入直到我們不再鎖住(unlock)這個帳號。 透過更改密碼完成 unlock 帳號，方法如下：

```
passwd fred
```

```
Changing password for fred蕡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: *******
```

現在 /etc/shadow 檔將包含：

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

且 fred 將可以簽入和使用該系統。 useradd 和其他附帶 *Shadow Suite* 比較好的地方是可以自動改變 /etc/passwd 和 /etc/shadow。 所以如果你正在新增一個使用者， 且另一個使用者正在更改密碼，這兩個操作都可以正確的執行。

你使用提供的指令比直接存取 /etc/passwd 和 /etc/shadow 檔還好。 如果你正編輯 /etc/shadow 檔，且有個使用者在你編輯時要改變他的密碼， 然後你儲存編輯結果，這個使用者的密碼將會遺失掉。

這裡是使用 useradd 和 passwd 新增使用者的一些 interactive script：

```
#!/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 "'s
```

```
# 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 ""
```

新增使用者是用 script 比直接編輯 /etc/passwd / /etc/shadow 檔或使用 像 Slackware 的 adduser 程式還要好。

需要更多 useradd 資訊請參照線上操作手冊。

7.1.2 usermod

usermod 程式是用在修改使用者資訊。 它的參數使用和 useradd 程式類似。

如果你要更新 fred 的 shell，你要作下列步驟：

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

現在 fred 的 /etc/passwd 檔將變成：

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

如果要使 fred 的帳號到期日為 09/15/97：

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

現在 `fred` 在 `/etc/shadow` 的欄位變成：

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

需要更多 `usermod` 資訊請參照線上操作手冊。

7.1.3 userdel

`userdel` 用在刪除使用者，使用方法爲：

```
userdel -r username
```

`-r` 參數可以將該使用者根目錄全部移除。位在期待目錄的檔案則需手動移除。

如果你只是要簡單的鎖住帳號而沒有要刪除它，建議你使用 `passwd` 指令。

7.2 passwd 指令和 passwd 老化

`passwd` 指令很明顯使用在改變密碼，除此之外， 可由 *root* 使用在：

- Lock 和 unlock 帳號 (-l and -u)
- 設定密碼合法的最大天數 (-x)
- 設定密碼改變間的最小天數 (-n)
- 設定密碼到期的警告天數 (-w)
- 設定在帳號未被鎖死密碼到期後的警告天數 (-i)
- 允許查詢帳號資訊 (-S)

舉例說明，如果要鎖死 `fred` 帳號：

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

這表示 `fred` 的密碼是有效的，它在 03/04/96 被修改且任何時間都可被修改， fred 將不會收到警告且帳號將不會因密碼到期而關閉。

這表示如果 `fred` 在密碼到期後簽入，它將被要求用一個新密碼簽入。

如果我們決定要警告 `fred` 在密碼過期前 14 天，且讓它的帳號在到期後14天警告， 我們需要作下列步驟：

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

現在 `fred` 改變爲：

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

需要更多 `passwd` 資訊請參照線上操作手冊。

7.3 login.defs 檔

/etc/login 檔是對 login 程式的 configuration file 且對 *Shadow Suite*。

/etc/login 包含從預設值密碼改變的驅動設定。

/etc/login.defs 檔是一個很好的文件檔，然而仍有些事情要注意：

- It contains flags that can be turned on or off that determine the amount of logging that takes place.
- It contains pointers to other configuration files.
- It contains defaults assignments for things like password aging.

跟去上述你可以發現這是一個重要檔，且你應該確認目前設定及你將對你系統的設定內容。

7.4 群組密碼

/etc/groups 檔包括允許是用者存取群組之密碼。 如果你定義 SHADOWGRP 在 /usr/src/shadow-YYMMDD/config.h 檔將開啓該功能。

如果你定義該常數且編譯它，你需建立一個 /etc/gshadow 檔來保存群組密碼 和群組管理者資訊。

當你建立 /etc/shadow。你使用一個呼叫程式叫做 pwconv， 該程式不會建立 /etc/gshadow 檔，但是這沒關係，只要你自行建立即可。

爲了建立起始 /etc/gshadow 檔要執行下列步驟：

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

每次你建立一個新群組，它們會被加到 /etc/group 和 /etc/gshadow 檔。 如果你透過新增或移除使用者來修改群組或改變群組密碼，/etc/gshadow 檔都將被改變。

groups, groupadd, groupmod, 和 groupdel 程式是用來供應 *Shadow Suite* 部分可以變更群組。

/etc/group 檔格式如下：

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

其中：

groupname

The name of the group

!

The field that normally holds the password, but that is now relocated to the /etc/gshadow file.

GID

The numerical group ID number

`member`

List of group members

`/etc/gshadow` 檔格式如下：

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

其中：

`groupname`

The name of the group

`password`

The encoded group password.

`admin`

List of group administrators

`member`

List of group members

`gpasswd` 指令是用在新增或移除管理者和群組成員。 `root` 或其他在 群組管理者人員可新增或移除群組成員。

群組密碼可以透過 `passwd` 指令改變，需透過 *root* 或在該群組管理者有權限的帳號方可修改。

Despite the fact that there is not currently a manual page for `gpasswd`, typing `gpasswd` without any parameters gives a listing of options. It's fairly easy to grasp how it all works once you understand the file formats and the concepts.

7.5 檢查程式一致性

7.5.1 pwck

`pwck` 程式提供在 `/etc/passwd` 和 `/etc/shadow` 檔的一致性檢查。 它將檢查每個使用者名稱且依照下列步驟確認：

- the correct number of fields
- unique user name
- valid user and group identifier
- valid primary group
- valid home directory
- valid login shell

它也會警告沒有密碼的帳號。

在安裝 *Shadow Suite* 後執行 `pwck` 是一個很好的點子。 它也可以每週或每月週期性的執行。 如果你使用 `-r` 參數，你可以用 `cron` 來執行且收到電子郵件報告

7.5.2 grpck

`grpck` 檢查 `/etc/group` 和 `/etc/gshadow` 檔一致性的程式。 它作下列檢查：

- the correct number of fields
- unique group name
- valid list of members and administrators

它也有 `-r` 參數自動產生報表。

7.6 Dial-up 密碼

Dial-up 密碼是另一個對系統防禦的選項列，該系統允許撥接存取。 如果你有一個系統允許許多人區域網路連結，但是你想限制撥接的權限，那你需使用 dial-up 密碼。 爲了要開啓 dial-up 密碼，你必須編輯 `/etc/login.defs` 檔且確定將 `DIALUPS_CHECK_ENAB` 設定爲 `yes`.

有兩個檔案包括 dial-up 資訊， `/etc/dialups` 包括 ttys (one per line, with the leading "/dev/" removed)。 如果 tty 有被列出， dial-up 表示已經被檢查。

第二個檔是 `/etc/d_passwd`。 這個檔包括 shell 全部合法路徑名稱。

如果以個使用者簽入一條列在 `/etc/dialups` 的線(line)，且他的 shell 被列在 `/etc/d_passwd` 檔，他將被允許存取透過提供正確的密碼。

另一個使用 dial-up 密碼的目的是設定指允許某些形式連結的線(可能是PPP 或 UUCP 連結)。 如果一個使用者試著得到另一種形式連結 (i.e. a list of shells)，他必須知道使用這條線的密碼。

在你可以在未來使用 dial-up 前，你密需建立一些檔案。

`dpasswd` 指令提供對在 `/etc/d_passwd` 檔的 shells 指派密碼。 可以看操作手冊的到更多資訊。

8 加入 shadow 支援 C 語言

新增支援 shadow 程式事實上是很直接的。 唯一的問題是程式需要以 root (或 SUID root) 權限執行，這樣才可以存取 `/etc/shadow` 檔。

這顯示一個大問題： 當建立 SUID 程式時需要很小心依照程式運作。舉例說明： 如果以個 程式有 shell escape，如果程式本身是 SUID root 將不需要以 root 方式呈現。

對程式新增支援 shadow 而言，它可以檢查密碼，但不需以 root 權限執行，而是以 SUID shadow 取代執行比較安全。 `xlock` 程式就是一個例子。

接下來範例介紹， `pppd-1.2.1d` 已經以 SUID as root 方式執行，所以新增 shadow 支援應該不會使程式產生任何影響。

8.1 標頭檔(Header files)

標頭檔應存在 `/usr/include/shadow`。 應該有一個 `/usr/include/shadow.h`檔， 但是它將 symbolic link 到 `/usr/include/shadow/shadow.h`。

爲了新增支援 shadow 程式，你需要 include 標頭檔：

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

8.2 libshadow.a 函式庫(library)

當你安裝 *Shadow Suite*， `libshadow.a` 檔被建立和安裝在 `/usr/lib` 目錄。

當編譯一個 shadow support 程式，linker 需要包括 `libshadow.a` 函式庫進入鏈結。

執行如下：

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

然而，就像我們接下來要看的例子，大部分大程式使用 `Makefile` 且 通常有變數呼叫 `LIBS=...` 需要被修改。

8.3 Shadow 結構(Structure)

`libshadow.a` 函式庫對它從 `/etc/shadow` 檔接收資訊使用結構化呼叫。 這是從 `/usr/include/shadow/shadow.h` 標頭檔的 `spwd` 結構定義：

```
struct spwd
{
  char *sp_namp;               /* login name */
  char *sp_pwdp;               /* encrypted password */
  sptime sp_lstchg;            /* date of last change */
  sptime sp_min;               /* minimum number of days between changes */
  sptime sp_max;               /* maximum number of days between changes */
  sptime sp_warn;              /* number of days of warning before password
                                  expires */
  sptime sp_inact;             /* number of days after password expires
                                  until the account becomes unusable. */
  sptime sp_expire;            /* days since 1/1/70 until account expires
*/
  unsigned long sp_flag;       /* reserved for future use */
};
```

Shadow Suite 可以放除了編碼密碼之外的資料到 `sp_pwdp` 欄位。 密碼欄位可包括：

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

這表示一個額外的密碼， `/sbin/extra` 程式應該被更多的權限呼叫。 程式的呼叫需取得使用者名稱和指出爲何需被呼叫的 switch才可通過。 查看 `/usr/include/shadow/pwauth.h` 和原始碼 `pwauth.c` 獲得更多資訊。

爲何我們應使用 `pwauth` 去表示眞正的權限，這是什麼意思，它將使第二組權限也 跑得很好。

Shadow Suite 作者指出因爲大部分存在的程式都不這麼作囉，所以 *Shadow Suite*未來的版本將移除。

8.4 Shadow 函式(Functions)

shadow.h 包含 libshadow.a 函式庫：

```
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));
```

我們將使用的範例程式是： getspnam 將對供應名稱恢復對我們 spwd 結構。

8.5 範例

這是一個範例描述新增 shadow 支援程式，但預設值並沒有。

本範例使用 *Point-to-Point Protocol Server* (pppd-1.2.1d)，它有個模式是表示 從 /etc/passwd 檔取代 *PAP* 或 *CHAP* 檔使用帳號密碼的 *PAP* 權限，你將不需要在 pppd-2.2.0 加這些程式碼，因爲它已經存在囉。

pppd 的未來大致上不會被使用很多，但是如果你安裝 *Shadow Suite*，儲存在 /etc/passwd 檔的密碼將無法運作。

在 pppd-1.2.1d 權限使用的程式碼是位在 /usr/src/pppd-1.2.1d/pppd/auth.c 檔。

接下來程式碼需要被加在所有其他 #include 指令檔案的最上頭，我們將注意有環境指令的 #includes 。

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

接下來要做的事情是變更實際碼， 我們將變更 auth.c 檔。

變更前 auth.c 檔 function 爲：

```
/*
 * login - Check the user name and password against the system
 * password database, and login the user if OK.
 *
 * returns:
 *      UPAP_AUTHNAK: Login failed.
 *      UPAP_AUTHACK: Login succeeded.
 * In either case, msg points to an appropriate message.
 */
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 If no passwd, let them login without one.
     */
    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);

    /*
     * Write a wtmp entry for this user.
     */
    tty = strrchr(devname, '/');
    if (tty == NULL)
        tty = devname;
    else
        tty++;
    logwtmp(tty, user, "");             /* Add wtmp login entry */
    logged_in = TRUE;

    return (UPAP_AUTHACK);
}
```

使用者的密碼被放在 pw->pw_passwd，所以我們需新增 getspnam function，這將會把密碼放到 spwd->sp_pwdp。

我們將新增 pwauth function 來表示眞正的權限。 這將在 shadow 檔設定時 自動產生第二個權限。

變更爲可以支援 shadow 後的 auth.c function：

```
/*
 * login - Check the user name and password against the system
 * password database, and login the user if OK.
 *
 * This function has been modified to support the Linux Shadow Password
 * Suite if USE_SHADOW is defined.
 *
 * returns:
 *      UPAP_AUTHNAK: Login failed.
 *      UPAP_AUTHACK: Login succeeded.
 * In either case, msg points to an appropriate message.
 */
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 If no passwd, let NOT them login without one.
     */
    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);

    /*
     * Write a wtmp entry for this user.
     */
    tty = strrchr(devname, '/');
    if (tty == NULL)
        tty = devname;
    else
        tty++;
    logwtmp(tty, user, "");             /* Add wtmp login entry */
    logged_in = TRUE;

    return (UPAP_AUTHACK);
}
```

嚴謹的範例將啓發我們在作其他改變的幫助。 原始的版本如果在 /etc/passwd 檔 沒有任何密碼，可允許存取傳回的 UPAP_AUTHACK。這是不好的，因爲 簽入的使用是使用一個允許存取 PPP process的帳號，然後檢查帳號密碼，該帳號密碼是由 RAP、在 /etc/passwd 檔的帳號和 /etc/shadow 檔的密碼供應。

所以如果我們設定原本版本對每個使用者，如 ppp 可以在 shell 執行，然後任何人可以 獲得 ppp 鏈結透過設定他們對使用者 ppp 的 PAP 和 null 的密碼。

我們修正 UPAP_AUTHNAK 取代 UPAP_AUTHACK 如果密碼欄位是空的。

有趣的是 pppd-2.2.0 有相同的問題。

接下來我們需要變更 Makefile 以便讓兩件事發生：

USE_SHADOW 必須被重新定義且libshadow.a 需要被新增到鏈結 process。

編輯 Makefile 且新增：

```
LIBS = -lshadow
```

然後我們找到這一行：

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

然後改變它變成：

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

現在執行 make 跟 install.

9 常問問題及答案

Q: 我用 tty's 用 *root* 權限使用 `/etc/securettys` 檔，但是都沒有 任何運作，該如何解決？

A: `/etc/securettys` 檔在 *Shadow Suite* 安裝後就不具任何意義囉。

tty's 用 *root* 權限可以使用一個放在 `/etc/login.defs` 簽入設定檔進行修改。 該設定檔也可能指到另一個檔。

Q: 我對 *Shadow Suite* 進行安裝後，現在我沒辦法簽入(login)系統， 我有忽略什麼步驟嗎？

A: 你是真的安裝 Shadow 程式囉，但是並未執行 `pwconv` 或你忘記複製 `/etc/npasswd` 到 `/etc/passwd` 且複製 `/etc/nshadow` 到 `/etc/shadow`囉。 你也需要複製 `login.defs` 到 `/etc`。

Q: 在 xlock 那一章節有提到改變 `/etc/shadow` 檔的群組擁有者到 `shadow`。我沒有 `shadow` 群組，該如何處理？

A: 你可以新增一個。 簡單的編輯 `/etc/group` 檔，然後新增一行給 shadow 群組，你需要確認群組編號並未被其他群組使用，且你需要在 nogroup 紀錄之前插入該設定。 或這你可以簡化對 root權限設定 suid `xlock`。

Q: 有任何對 Linux Shadow Password Suite 的電子郵件列表嗎？

A: 有，但是它的目的是針對 Linux 下一個 Shadow Suite版本發展跟測試，你可以 到 `shadow-list-request@neptune.cin.net` 電子郵件摘要輸入： `subscribe`。 這是在討論 Linux `shadow-YYMMSS`系列版本。你應該參加如果你 想參予未來發展或你安裝 Suite 在你的電腦且想要得到新版本的資訊。

Q: 我安裝 *Shadow Suite*，但是當我使用 `userdel` 命令後， 我得到 "userdel: cannot open shadow group file" 訊息，我有哪裡作錯了嗎？

A: 你編譯 *Shadow Suite* 時有開啓 `SHADOWGRP` 選項，但是 你沒有一個 `/etc/gshadow` 檔案，你需要編輯 `config.h` 檔和重新編 譯或建立一個 `/etc/group` 檔，請看 shadow groups 那一節。

Q: 我安裝 *Shadow Suite* 但是我無法找到編碼密碼在我的 `/etc/passwd` 檔，發生什麼問題？

A: 你可能在 Shadow `config.h` 檔開啓 `AUTOSHADOW` 選項或者 你的 `libc` 被 `SAHDOW_COMPAT` 選項編譯，你需要決定是哪個問題，然後 再重新編譯。

10 版權聲明(暫不翻譯)

11 Miscellaneous and Acknowledgments.

The code examples for `auth.c` are taken from pppd-1.2.1d and ppp-2.1.0e, Copyright (c) 1993 and The Australian National University and Copyright (c) 1989 Carnegie Mellon University.

Thanks to Marek Michalkiewicz <marekm@i17linuxb.ists.pwr.wroc.pl> for writing and maintaining the *Shadow Suite* for Linux, and for his review and comments on this document.

Thanks to Ron Tidd <rtidd@tscnet.com> for his helpful review and testing.

Thanks to everyone who has sent me feedback to help improve this document.

Please, if you have any comments or suggestions then mail them to me.

regards

Michael H. Jackson <mhjack@tscnet.com>