Loopback Encrypted Filesystem HOWTO

作者：Ryan T. Rhea, zzrhear@pobox.winthrop.edu
譯者：趙平望 tchao@worldnet.att.net
v1.1, 29 November 1999 翻譯日期：2000年1月15日

本文件解釋如何安裝和使用一種使用者加載後可動態和無需採取特別步驟就能加密文檔內容的文檔系統。這種文檔系統存放在通常文件內，它可作爲隱藏文件隱藏，也可使用極可能會被忽略的一般文檔名稱存放，以此提高數據資料儲存的安全。

目錄

1 前言

建立這個文檔系統需要有內核的源代碼、編輯代碼的能力和耐心，同時強烈建議手邊隨時有啓動盤可用。 把重要資料長期存放在加密的文檔系統之前，應首先製作備份保存， 因爲任何存放在電腦的資料都有遭到損壞不可取用的可能。

要完成建立這個文檔系統，起碼要修補 Linux 內核 2.2.9 版。關於修補的細節， 見下文4 ()一節內的內容。

內核源代碼可從下列網址下載：

<ftp://ftp.kerneli.org/>

關於重編內核的程序可查閱相關 HOWTO 文件，網址如下：

<http://metalab.unc.edu/LDP/HOWTO/>

這份文件可全文或部分轉載，不收費用，但需符合下列條件：

- 在全文或部分轉載中應全文載列版權説明和這份轉載同意書。
- 任何翻譯或由這份文件生成的文本在散發前均應取得作者的書面同意。
- 如果只散發本文部分內容，則應在散發的文本內列入取得本文全文的詳細方法和途徑。
- 文內所有源代碼均受到 GNU 一般公共許可証的保護。許可証內容可通過匿名 FTP 網址下載：

<ftp://prep.ai.mit.edu/pub/gnu/COPYING/>

2 導言

這項過程使用“/dev/loop*”（在多數安裝系統中* 爲 0-7 ）加載 loopback 文檔系統。 採用同種方法可將 Linux 的文檔系統不加密地存放在非 Linux 分割區。在前述 LDP 網址上存有關于這方面的 HOWTO 。

文檔加密的方法很多，包括 XOR, DES, twofish, blowfish, cast128, serpent, MARS, RC6, DFC 和 IDEA 。 “losetup”程式的工作便是將加密文檔和文檔系統及其密碼聯系在一起。根據管理 kerneli.org 和國際加密修補軟件（international crypto patches）的 Alexander Kjeldaas 先生的看法， DES 和 losetup 目前並不兼容。 這是由于這兩種軟件處理 parity bit 的方法不同的緣故。目前 Linux 系統並無支持 DES 的計畫，因爲 DES 密碼的加密程度比較不夠嚴密。

Twofish, blowfish, cast128 和 serpent 密碼可任意使用，沒有任何許可証的限制。 其他密碼可能有一些許可証方面的規定。有些密碼入選作爲 AES 標準。 最後選定的密碼將作爲全世界免費使用的密碼。

本文使用 serpent 加密法加密，因爲這種加密法保密性強，運行極快，同時根據 GPL 的規定可免費散發。 在 serpent 有關的文件中指出，serpent 軟件使用 Ross Anderson, Eli Biham 和 Lars Knudsen 設計的長度 128-bit 的密碼組。這對使用者的保密要求提供了最高保證， 因爲到目前爲止，並無解碼的簡單辦法。有關 serpent 的文件及其源代碼可從下列網址下載：

<http://www.cl.cam.ac.uk/~rja14/serpent.html>

這份文件假設使用者將密碼直接編入內核。不過，密碼也可作爲模組編入， 但在該文件中並未對這種方法加以討論。不過方法也很簡單，只需編輯“/etc/conf.module”; 詳情見前面提到的關於編輯內核的 HOWTO 文件。

3 摘要

這項過程涉及許多步驟。在下節4 ()中對這些步驟有詳細說明。 先對這些步驟作出摘要說明也許是不錯的主意，因爲 Unix 和 Linux 的高手也許不需要詳細步驟。 這些步驟如下：

1. 下載最新的國際加密修補軟件 (編寫本文時的最新版本爲“patch-int-2.2.10.4”)：

   <http://ftp.kerneli.org/pub/kerneli/>

2. 修補內核

3. 運行 'config' (或 'menuconfig' 或 'xconfig')，爲新內核設置 'MakeFile'。 設定加密的各個選項並不集中在一起。首先，要設定任何選項都得選用 'Code Maturity level options' 項下的 'Prompt for development and/or incomplete code/drivers'。在 'Crypto options' 項下選用 'crypto ciphers' 和 'serpent' 兩項。在此再次假設使用 serpent 加密，不過也可試用其他的加密辦法。 在此必需指出，DES 到 2.2.10.4 版都還與系統不兼容 - 估計未來也不會兼容。在 'Block Devices' 下有几個重要選項必需選定。這包括 'Loopback device support', 'Use relative block numbers as basis for transfer functions (RECOMMENDED)' 和 'General encryption support' 等各項。此處不要選 'cast 128' 或 'twofish' 加密。此外在各種網路中也不需選取任何加密選項。 任何有關內核的設置方法均可參閱 LDP 文件，不在此處再次贅述。

4. 編輯新內核

5. 編輯 '/etc/lilo.conf'，以便在設置文檔內增加新內核。運行 'lilo -v' 將內核加到 boot loader 內。

6. 從下列網址下載最新的 'util-linux' 源代碼 (此處使用 'util-linux-2.9v' 版)：

   `<ftp://ftp.kernel.org/pub/linux/utils/util-linux/>`

7. 解壓 'util-linux' 源代碼。

8. 利用在 '/usr/src/linux/Documentation/crypto/' 目錄內的相應修補軟件。

9. 仔細閱讀 'INSTALL'。這套軟件內有許多與系統有關的文檔的源代碼 （重要的工具如'login', 'passwd'和'init'等）。如果在編輯這些源代碼之前 不仔細地編輯 MCONFIG，最好手頭隨時有啓動片可用，因爲系統隨時都會錯亂。 基本上，將所有 'HAVE_*' 都設成“yes”，使所有重要的系統軟件都不會被更改。 需要重建的工具是 'mount' 和 'losetup'，以適合新的加密需要。 細節不妨參看下文4 ()。

10. 編輯和安裝 'util-linux'。

11. 用新內核重新啓動電腦。

12. 編輯 '/etc/fstab'，增加加載點，步驟如下：

```
/dev/loop0  /mnt/crypt  ext2  user,noauto,rw,loop 0 0
```

13. 如上文 '/mnt/crypt' 的方式，建立能收存文檔系統的目錄。

14. 作爲使用者，收存加密的文檔如下：

```
dd if=/dev/urandom of=/etc/cryptfile bs=1M count=10
```

15. 運行 losetup 如下：

```
losetup -e serpent /dev/loop0 /etc/cryptfile
```

   注意：設定使用密碼的機會只有一次。可用下列指令查核使用密碼：

```
losetup -d /dev/loop0
```

   這條指令會使 loop device 停止作用。隨後再啓動 losetup 就可測試使用密碼，方法如下：

```
losetup -e serpent /dev/loop0 /etc/cryptfile
```

16. 設定 ext2 文檔系統如下：

```
mkfs -t ext2 /dev/loop0 100000
```

17. 此時就可加載加密的文檔系統：

```
mount -t ext2 /dev/loop0 /mnt/crypt
```

18. 完成加密之後，可卸載和保護文檔系統如下：

```
umount /dev/loop0
losetup -d /dev/loop0'
```

4 詳細內容

內核修補軟件：

可從“2.2.x”版的內核開始修補內核。爲“2.2.x”版的內核編寫的修補軟件都附帶除錯程序（bugfixes）。新功能都會加入 Linux “2.3.x”版的開發內核。修補內核的辦法是首先取得所有修補軟件，然後以下列指令修補：

```
cd /usr/src
gzip -cd patchXX.gz  patch -p0
```

重復 xx 各版本的修補，從次序低的 xx 往高的方向依次修補。

內核源代碼的默認目錄是 '/usr/src/linux'。如源代碼在其他目錄，可從 '/usr/src/linux' 建立一個符號連接（symbolic link）。

爲 **'util-linux'** 的編輯設定 **'MCONFIG'**：

以下是編輯 'util-linux' 時修改 'MCONFIG' 檔的部分內容。隨著系統版本的不同，修改方式並不完全相同，此處基本上以 RedHat 5.2 爲準。關鍵之處是不要覆蓋重要的系統工具，例 如'login'、'getty'或 'passwd' 等。以下列出一些重要的設定：

```
CPU=$(shell uname -m sed s/I.86/intel/)

LOCALEDIR=/usr/share/locale

HAVE_PAM=no

HAVE_SHADOW=yes

HAVE_PASSWD=yes

REQUIRE_PASSWORD=yes

ONLY_LISTED_SHELLS=yes

HAVE_SYSVINIT=yes

HAVE_SYSVINIT_UTILS=yes

HAVE_GETTY=yes

USE_TTY_GROUP=yes

HAVE_RESET=yes

HAVE_SLN=yes

CC=gcc
```

建議：

從'dev/loop0' 到 '/dev/loop7'，這 8 個 loopback devices 均可用于此處。 利用目錄名稱不太顯眼的目錄作爲加載點。不妨在 home 目錄內建立一個權限爲 700 的加密目錄。 也用不顯眼的目錄存放加密文檔。不妨在 '/etc' 內使用 'sysfile' 或 'config.data' 這類名稱。 一般這類名稱的目錄或文檔都不太受人注意。

下列 Perl 腳本可用于加載和卸載文檔系統。將其抄入系統，改成可運行（chmod u+x）， 然後存放在路徑目錄中。

```
#!/usr/bin/perl -w
#
#minimal utility to setup loopback encryption filesystem
#Copyright 1999 by Ryan T. Rhea
`losetup -e serpent /dev/loop0 /etc/cryptfile`;
`mount /mnt/crypt`;
```

將上述腳本稱爲 'loop'，就可用一個指令（'loop'）和密碼設定 loopback 加密文檔系統。

```
#!/usr/bin/perl -w
#
#minimal utility to deactivate loopback encryption filesystem
#Copyright 1999 by Ryan T. Rhea
`umount /mount/crypt`;
`losetup -d /dev/loop0`;
```

將這個腳本稱爲 'unloop'，以後只要鍵入 'unloop' 就可立即停止這個文檔系統的運作。