Loopback Encrypted Filesystem HOWTO

Copyright by Ryan T. Rhea, *zzrhear@pobox.winthrop.edu* <mailto:zzrhear@pobox.winthrop.edu> v1.0, 22 September 1999

이 문서는, 사용자가 마운트할 때, 그 내용을 동적으로 평이하게 암호화하는 파일시스템를 설정하고 사용하는 방법을 설명한다. 파일시스템은 정규 파일에 저장되는데, 숨겨지거나 눈에 잘 띄지 않게 이름이 붙여질 수 있다. 이것은 고수준의 안전한 데이터 저장을 위해 허용된다.

목 차

제 1 절 시작하기 전에

이 과정은 커널 소스 코드, 이 코드 컴파일링 지식 뿐만 아니라 상당한 인내를 요구한다. 부트 디스크를 준비해 둘 것을 매우 권장한다. 또한, 암호화된 파일시스템 상에 당신의 중요한 데이터를 영구적으로 저장하기 전에 백업을 해 두어야 한다. - 이것 역시 어떠한 다른 파일시스템처럼 손상될 수 있다.

최소한으로, 여러분은 리눅스 커널의 적어도 2.2.9 버전에 패치를 해야만 할 것이다. 패치를 적용하는 데 있어 더 자세한 지시사항은 이 문서에 나중의 "세부사항" 절에 있다. 커널 소스는 다음에서 구할 수 있다: <*ftp.kerneli.org/* <ftp://ftp.kerneli.org/>>

커널을 재컴파일하는 과정에 관한 하우투는 다음에 있다: <*metalab.unc.edu/LDP/HOWTO/* <http://metalab.unc.edu/LDP/HOWTO/>> 이 문서는 다음과 같은 조건을 붙여, 무료로, 전체로 혹은 부분적으로 재생산되어도 좋다.

? 모든 완전하거나 부분적인 복사본에 저작권 통지문과 이 허가 통지문을 그대로 보존하여야 한다.

? 어떠한 번역이나 파생되는 작품도 배포 전에 서면으로 저자의 허락을 얻어야 한다.

? 만일 당신이 이 작품을 부분적으로 배포하려고 한다면, 이 매뉴얼의 완전한 버전을 구하기 위한 지시사항을 반드시 포함시켜야 하고, 완전한 버전을 구하기 위한 수단도 제공되어야 한다.

? 이 문서에 포함된 모든 소스 코드는, 다음에서 익명 FTP를 통해 얻을 수 있는 GNU 일반 공개 인증(General Public License) 하에 있다. <*prep.ai.mit.edu/pub/gnu/COPYING/* <ftp://prep.ai.mit.edu/pub/gnu/COPYING/>>

제 2 절 소개

본 과정은 루프백 파일시스템을 마운트하기 위해 '/dev/loop*' (여기서 *는 대부분의 설치에서 0-7이 될 수

있다) 장치를 사용한다. 동일한 과정이 암호화 없이도 비(非)리눅스 파티션에서 리눅스 파일시스템을 저장하기 위해 사용될 수 있다. 이에 관한 하우투는 앞서 언급한 LDP 사이트에 있다.

암호화의 다양한 유형이 쓰일 수 있는데, XOR, DES, twofish, blowfish, cast128, serpent, MARS, RC6, DFC, IDEA를 포함한다. 프로그램 'losetup' (loopback setup)은 당신의 암호화된 파일을 파일시스템과 암호 유형으로 연결하는 것이다. kerneli.org와 국제적인 crypto 패치를 운영하는 Alexander Kjeldaas에 의하면, DES와 losetup은 현재 호환성이 없다. 이것은 이 둘의 패리티 비트를 처리하는 방식에서의 차이에 기인한다. 다른 암호들보다 매우 훨씬 덜 안전하기 때문에 DES를 지원하는 계획은 없다.

Twofish, blowfish, cast128, serpent는 모두 어떤 용도로 사용하든간에 무료 사용이 허가된다. 다른 것들은 사용 허가권에 대한 제한을 가지기도 하고, 그렇지 않은 것도 있다. 그것들 중에 다수는 AES 표준을 위한 후보이다. 최종적인 것은 전세계적으로 그것들의 암호를 저작권 사용료 없이 사용토록 될 것이다.

이 문서는 serpent 알고리듬을 사용한다. 아직 놀랄만큼 빠르고, GPL 하에 무료로 배포 가능하기 때문이다. 그것의 문서에 의하면, serpent는 Ross Anderson, Eli Biham, Lars Knudsenserpent이 고안한 128 비트 블럭 암호를 사용한다. 그것은 사용자한테 최고의 실제적인 수준의 보장을 제공하여 어떠한 지름길 공략도 발견되지 않도록 할 것이다. 소스 코드 뿐만 아니라 serpent에 관한 문서를 다음에서 구할 수 있다.

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

제 3 절 요약

많은 단계들이 본 과정에 포함된다. 다음 절에서 이 단계들을 위해 "세부사항"을 제공할 것이다. 참고 자료를 제공하기 위해 첫 번째로 요약을 제공하는 것이 좋을 것이라고 생각한다(당신이 유닉스/리눅스에 익숙하다면 아마도 세부사항을 전혀 필요치 않을 것이다). 다음과 같이 요약할 수 있다.

1. 가장 최신의 국제적인 crypto 패치를 내려받는다 <*ftp.kerneli.org/pub/kerneli/* `<http://ftp.kerneli.org/pub/kerneli/>`>
2. 커널을 패치한다
3. 'config' (또는 'menuconfig' 또는 'xconfig')를 실행한다.

   무엇보다도, 여러분이 어떤 다른 선택사항을 보기 전에 반드시 'Code Maturity level options' 아래에서 'Prompt for development and/or incomplete code/drivers'를 가능케 해야 한다. 'Crypto options' 아래에서 'crypto ciphers'를 가능케 하라. DES는 2.2.10.4까지 호환성이 없다 - 전혀 지원되지 않을 수도 있다는 것을 기억하라. 'Block Devices' 아래에서 선택해야 하는 몇 가지 중요한 선택사항이 있는데, 'Loopback device support', 'Use relative block numbers as basis for transfer functions (RECOMMENDED)', 'General encryption support'이다. 'cast 128' 또는 가지각색의 네트워크 범주 아래의 선택사항들을 선택하지 말라. 커널의 구성에 대해 더 이상 다루지 않을 것이다. 그것은 이 문서의 범위를 벗어난 것으로 LDP 사이트에서 구할 수 있다.
4. 새 커널을 컴파일한다.
5. '/etc/lilo.conf'를 편집하여 새 커널 이미지를 추가한다. 'lilo -v'를 실행하여 부트 로더(boot loader)에 커널을 추가한다.
6. 최신의 'util-linux'를 위한 소스를 내려받는다 <*ftp.kernel.org/pub/linux/utils/util-linux/* `<ftp://ftp.kernel.org/pub/linux/utils/util-linux/>`>
7. 'util-linux' 소스를 압축을 푼다
8. 여러분 것에서 찾은 일치하는 패치를 적용한다
9. 'INSTALL' 파일을 주의깊게 읽는다. 이 패키지는 많은 시스템 종속된 파일들(당신의 시스템이 매우 혼란스러워질 것이기 때문에 파일과 같은 중요한 도구들은 소스를 컴파일하기 전에 부트 디스크 및

무기를 준비한 다)을 위한 소스를 포함한다. 기본적으로, 중요한 인증 도구들이 컴파일되지 않고 중첩 작성되지 않도록, 여러분은 거의 모든 'HAVE_*' 필드를 yes로 설정하기를 원한다. 새 암호화 계획을 수용하디 위해 다시 신축(rebuilt)하기 원하는 도구는 'mount'와 'losetup'이다. 여러분이 이 단계를 위해 아래의 "" 절을 참고할 것을 제안한다.

10. 'util-linux' 소스를 컴파일하고 설치한다
11. 새 커널로 장비를 재부팅한다
12. '/etc/fstab'을 편집하여 다음과 같이 마운트 지점을 위한 기입사항을 추가한다

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

13. 여러분의 파일시스템을 간직하게 될 디렉토리를 만든다
14. 사용자로서 다음과 같이 암호화된 파일을 만든다 dd if=/dev/zero of=/etc/cryptfile bs=1k count=1000
15. 다음과 같이 losetup을 실행한다

    losetup -e serpent /dev/loop0 /etc/cryptfile 암호를 입력하는 한 번의 기회를 가질 뿐이라서 주의해야 한다. 만일 여러분이 암호를 이중 점검하기를 원하면, 다음의 명령어를 사용할 수 있다.

    losetup -d /dev/loop0 이것은 loop 장치를 활성화시키지 않을 것이다. 다음에, 여러분의 암호를 검사하기 위해서, 다음과 같이, 다시 losetup을 실행하게 될 것이다.

    losetup -e serpent /dev/loop0 /etc/cryptfile
16. 다음과 같이 여러분의 ext2 파일시스템을 만든다 mkfs -t ext2 /dev/loop0 100000
17. 이제 암호화된 파일시스템을 마운트할 수 있다 mount -t ext2 /dev/loop0 /mnt/crypt
18. 끝났다면, 다음과 같이 파일시스템을 언마운트하고 보호한다. umount /dev/loop0 losetup -d /dev/loop0'

제 4 절 세부 사항

커널 패치: '2.2.x' 릴리즈로부터 패치에 의해 업그레이드 할 수 있다. '2.2.x'를 위해 릴리즈된 각 패치는 버그 수정을 포함한다. 새로운 특징들은 리눅스 '2.3.x' 개발 커널에 추가될 것이다. 패치를 해서 설치하려면, 모든 새 패치 파일들을 구하고, 다음과 같이 한다.

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

현재의 소스 트리의 버전보다 더 큰 모든 버전에 대해, 차례대로, xx를 반복한다. 커널 소스를 위한 기본적인 디렉토리는 '/usr/src/linux'이다. 여러분의 소스를 어떤 다른 곳에 설치한다면, '/usr/src/linux'로부터 심볼릭 링크를 사용할 것을 제안하는 바이다.

'util-linux' 패키지 컴파일을 위해 'MCONFIG'를 편집한다. 다음 내용은 'util-linux' 패키지를 컴파일하는데 사용된 'MCONFIG'로부터 발췌한 것이다. 이것은 레드햇 5.2를 기반으로 하는, 나만의 설정을 위한 매우 특별한 것을 알린다. 요점은 'login', 'getty', 또는 'passwd'와 같은 어떠한 중요 시스템 도구들을 덮어쓰지 않도록 해야 한다는 것이다. 어쨌든 중요한 내용들은 다음과 같다.

```
  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
```

제안: '/dev/loop0'부터 '/dev/loop7'까지 8개의 loopback 장치 중 어떤 것이든지 사용 가능하다. 마운트 지점에 대해 눈에 띄지 않는 디렉토리를 사용하라. home 폴더 내에서 700 퍼미션을 가지는 폴더를 만들 것을 제안하겠다. 동일한 사항이 데이터를 간직한 파일에 대해서도 적용된다. 나는 '/etc' 폴더 내에서 'sysfile' 또는 'config.data'와 같은 파일명을 사용한다. 이것은 대개 간과될 것이다.

나는 하나의 명령어로 파일시스템을 마운트하거나 언마운트하는 매우 간단한 펄 스크립트를 만들었다. 이대로 작성하고, 실행 가능하도록 만들어(chmod u+x) 여러분의 경로에 넣어 두면 된다.

```
#!/usr/bin/perl -w
#
#simple utility to setup loopback encryption filesystem
#Copyright 1999 by Ryan T. Rhea
$cmd1 = `losetup -e serpent /dev/loop0 /etc/cryptfile`;
$cmd2 = `mount /mnt/crypt`;
print $cmd1;
print $cmd2;
```

위의 스크립트를 'loop'라고 이름 붙인 다음, 한 번의 명령('loop')과 비밀번호로 여러분의 방식대로 될 것이다.

```
#!/usr/bin/perl -w
#
#simple utility to deactivate loopback encryption filesystem
#Copyright 1999 by Ryan T. Rhea
$cmd1 = `umount /mount/crypt`;
$cmd2 = `losetup -d /dev/loop0`;
print $cmd1;
print $cmd2;
```

두 번째 것을 'unloop'이라고 이름 붙이고, 'unloop'을 입력하면 당신의 파일시스템이 즉시 작동하지 않을 것이다.