Home criptografado no Slackware e Slamd64

Silvio Rhatto

3 de janeiro de 2009

Resumo

Tratarei aqui de como deixar seu diretório home num arquivo criptografado que é montado/desmontado quando você loga ou desloga. Existe o módulo pam_mount do PAM que faz algo semelhante, mas por questão de segurança o Slack não suporta o PAM. Usaremos o dm-crypt, substituto do cryptoloop incluído na série 2.6 do kernel.

1 Introdução

O procedimento a seguir foi efetuado em máquinas rodando Slackware e Slamd64. Efetue todos os passos a seguir como root e certifique-se que seu usuário normal esteja deslogado. Antes de começar, dê uma lida no site do *dm-crypt* e faça um backup das suas coisas.

Se você usa Debian ou outro sistema que suporte o PAM, experimente primeiro usar o pam_mount, conforme descrito no Debian Grimoire.

2 Instalação de pacotes

Você precisará dos programas device-mapper, cryptsetup-luks e as respectivas dependências. O device-mapper já vem no */testing* desde o slack 10.1.0. Para o cryptsetup-luks e dependências, você terá de compilar ou então utilizar os pacotes existentes em http://slack.sarava.org/packages ou em outro repositório.

A montagem/desmontagem do volume criptografado no home do usuário quando esse se loga/desloga é feita através de dois scripts, *cryptcreate* e *homecrypt*, ambos presentes na suíte homecrypt e cujo pacote se encontra disponível em http://slack.sarava.org/packages/noarch.

Para prosseguir, instale todos esses pacotes utilizando seu método preferido (installpkg, createpkg, simplaret, etc).

3 Kernel com dm-crypt e Crypto API

Se você não usa um kernel padrão da sua distribuição Slackware, certifique-se de que ele esteja compilado com as seguintes opções:

```
CONFIG_BLK_DEV_DM=m
CONFIG_DM_CRYPT=m
```

Certifique-se também que a opção *CONFIG_BLK_DEV_LOOP* (loop device) esteja como módulo ou built-in. Para que o módulo seja carregado ao iniciar o sistem, adicione em seu */etc/rc.d/rc.modules*, caso não exista, a linha

```
/sbin/modprobe dm_crypt
```

4 Crie um dispositivo de loop

O kernel 2.6 suporta até 256 loop devices. Para prevenir muitos problemas e simplificar nossos scripts, criaremos um loop device específico para cada usuário que usar home criptografado.

```
ls -l /dev/loop*

brw-rw----  1 root disk 7,  0 1996-06-03 22:47 loop0
brw-rw----  1 root disk 7,  1 1996-06-03 22:47 loop1
brw-rw----  1 root disk 7, 10 2002-03-19 22:13 loop10
brw-rw----  1 root disk 7, 11 2002-03-19 22:13 loop11
brw-rw----  1 root disk 7, 12 2002-03-19 22:13 loop12
brw-rw----  1 root disk 7, 13 2002-03-19 22:13 loop13
brw-rw----  1 root disk 7, 14 2002-03-19 22:13 loop14
brw-rw----  1 root disk 7, 15 2002-03-19 22:13 loop15
brw-rw----  1 root disk 7,  2 1996-06-03 22:47 loop2
brw-rw----  1 root disk 7,  3 1996-06-03 22:47 loop3
brw-rw----  1 root disk 7,  4 1996-06-03 22:47 loop4
brw-rw----  1 root disk 7,  5 1996-06-03 22:47 loop5
brw-rw----  1 root disk 7,  6 1996-06-03 22:47 loop6
brw-rw----  1 root disk 7,  7 1996-06-03 22:47 loop7
brw-rw----  1 root disk 7,  8 1996-06-03 22:48 loop8
brw-rw----  1 root disk 7,  9 2002-03-19 22:13 loop9
```

O próximo dispositivo de loop usaria o minor number 16, então vamos criar o 17o loop-device:

```
export USUARIO="nome-do-seu-usuario"
mknod /dev/loop-$USUARIO b 7 16
chown root.disk /dev/loop-$USUARIO
chmod 660 /dev/loop-$USUARIO
```

Se você estiver usando udev, talvez seja uma boa idéia colocar esses dispositivos nas suas regras ou então adicionar sua criação no rc.local.

Substitua usuario pelo nome do seu usuário. Se você usa o losetup como módulo, adicione no seu */etc/modprobe.conf*

```
options loop max_loop=17
```

Reinicie seu sistema pra ver se está tudo ok. É importante que exista o dispositivo */dev/mapper/control* depois da reinicialização.

5 Crie um arquivo com o tamanho igual ou maior que o seu home

```
export USUARIO="nome-do-seu-usuario"
dd if=/dev/urandom of=$USUARIO.img bs=100M count=5
```

6 Agora vamos criar um sistema criptografado nesse arquivo

```
losetup /dev/loop-$USUARIO usuario.img
cryptsetup --verbose --verify-passphrase luksFormat /dev/loop-$USUARIO
cryptsetup luksOpen /dev/loop-$USUARIO $USUARIO
```

Nesse ponto você deve escolher qual será a senha para acessar sua "partição". Agora crie um sistema de arquivos:

```
mkfs.ext2 /dev/mapper/$USUARIO
```

Não crie sistemas com journaling no topo de um loopfile, vide este comentário.

7 Mova seu home pra home.old, monte o novo sistema e copie seus arquivos

```
mv /home/$USUARIO /home/$USUARIO.old
mkdir -p /mnt/crypt/home/$USUARIO
ln -s /mnt/crypt/home/$USUARIO /home/$USUARIO
chown $USUARIO.users /home/$USUARIO
chmod 700 /home/$USUARIO
mount /dev/mapper/$USUARIO /home/$USUARIO
rsync -Cav /home/$USUARIO.old/ /home/$USUARIO/
umount /home/$USUARIO
cryptsetup luksClose $USUARIO
losetup -d /dev/loop0
```

Futuramente, quando você já estiver acostumado/a com sua pasta pessoal criptografada, apague a pasta */home/$USUARIO.old* usando algum método de remoção segura.

8 Sudoers

É necessário ainda que o usuário possa rodar o cryptsetup para seu dispositivo com permissão de root e sem a necessidade de senha. Pra isso, adicione o seguinte no seu *sudoers*:

```
usuario  ALL=NOPASSWD: /usr/bin/homecrypt
```

onde *usuario* é o nome do seu usuário.

9 Login/logout gráfico

Se você usa KDM como login gráfico, adicione no arquivo */etc/kde/kdm/Xstartup* a linha

```
/usr/bin/cryptcreate $USER
```

Se você usa GDM, adicione a linha anterior em */etc/X11/gdm/PreSession/Default*. Também adicione no arquivo */etc/kde/kdm/Xreset* (se você usa KDM) a linha

```
/usr/bin/homecrypt off $USER
```

No caso do GDM, você deve adicionar a linha anterior no arquivo */etc/X11/gdm/PostSession/Default*.

10 Logout via shell

Adicionar no seu .profile as linhas

```
alias logout="sudo /usr/bin/homecrypt off $(whoami); logout"
alias exit="sudo /usr/bin/homecrypt off $(whoami); exit"
```

11 Adicione a seguinte linha no */etc/fstab*

```
/dev/mapper/usuario /mnt/crypt/home/usuario ext2 user,noauto,exec 0 0
```

12 Pronto

Agora experimente logar com seu usuário. Após digitar a senha do arquivo criptografado, seu home será automaticamente montado. Quando você se deslogar pela última vez, seu home será desmontado. Funciona inclusive com logins via ssh, mas para que funcione sempre procure sempre se deslogar usando "logout"ao invés de Ctrl-D. No caso do X11, evite encerrar sua sessão usando Ctrl+Alt+Backspace. Nesses casos é possível que você seja deslogado mas sem que o script de desmontagem do home criptografado sja chamado.

13 Finalizando processos

Antes de desmontar um volume criptografado, o *homecrypt* pode ainda matar alguns processos do/a usuário cujos nomes dos programas estiverem listados no arquivo */home/usuario/logout.kill*, como por exemplo

```
gpg-agent ssh-agent
```

Se esse arquivo existir, então o *homecrypt* o executará automaticamente toda vez que seu usuário de deslogar.

14 Conclusões

Esse esquema garante apenas que quando você não estiver logado ninguém conseguirá acessar seu home. Quando ele estiver montado, o superusuário pode tranquilamente acessá-lo (experimente).

Ao invés de usar um arquivo, você pode utilizar uma partição inteira, algo que aumenta muito a performance e elimina o uso do losetup. A grande vantagem de usar um arquivo é a facilidade para gravar backups ou manter seu home em dispositivos como canivetes usb ou cd-rw para backups.

A desvantagem de usar o dm_crypt dessa forma em relação ao PAM é que você precisará de duas senhas para se logar. Isso também pode ser encarado como uma vantagem, já que um ataque no shadow (mesmo atualmente ainda sendo algo bem remoto) do seu sistema possibilitaria o acesso irrestrito aos seus dados.

Além disso, é interessante criptografar também sua swap. Existe um ótimo tutorial a respeito aqui.

15 Sobre

Este mini-howto foi escrito por Rhatto (rhatto at riseup.net) na esperança de que seja útil. Ele pode ser distribuído de acordo com a http://slack.sarava.org/copyleft.