Universidade Federal da Paraíba – UFPB
Departamento de Informática – DI

João Filho Matos Figueiredo
www.joaomatosf.com

Criptografando seu Disco

Um usuário que configura todo seu sistema, ou de sua empresa, seguindo todas as políticas de segurança das quais se recorda, porém deixa de criptografar suas partições, deve poder garantir a segurança física dos discos. Caso contrário, seu parceiro que não possuí privilégios no sistema, poderá usar um simples LiveCD, montar as partições do PC local e acessar todos aqueles arquivos “secretos”, tabelas de bancos de dados, arquivos de configurações do sistema e tudo mais que a sua imaginação criativa o permitir. Apresentar-se-á o uso do DM-Crypo para criptografar os discos/partições e evitar que seus dados possam ser violados, mesmo se os discos forem confiscados ou roubados.

Motivações

A velha abordagem de restringir, pela BIOS, o boot a apenas o disco principal, não se mostra eficiente, uma vez que não é uma tarefa difícil resetar sua senha e reconfigurar para que o boot seja, novamente, efetuado pelo CD/DVD, ou outra media. Pode-se também retirar os discos e analisá-los em outro computador, assim, ainda que dados “importantes” tenham sido apagados, estes poderão ser recuperados e analisados sem muito esforço.

Metodologia

Diferente da criptografia de diretórios, criptografar a raiz inteira de um disco demanda uma atenção extra. Esta tarefa pode ser auxiliada utilizando-se uma media bootavel, porém, a confecção desta media, com todos os pacotes necessários, leva mais tempo que a abordagem aqui adotada, sintetizada nos seguintes passos:

1- Iniciar o seu sistema normalmente
2- Criar as novas partições criptografadas
3- Copiar todo o seu sistema para as novas partições
4- Configurar o boot indicando o novo local do sistema de arquivos.

Ao final destes passos, a partição inicial, não criptografada, poderá ser usada para outros fins, dado que o sistema passará para uma nova partição. Caso se faça necessário que o sistema criptografado permaneça na partição original, os passos adicionais podem ser adotados:

1- Iniciar o seu sistema normalmente
2- Criar uma nova partição temporária
3- Copiar todo o seu sistema para a nova partição temporária
4- Configurar o boot para ser feito pela nova partição temporária
5- Reiniciar pela nova partição
6- Configurar a criptografia na partição original (isto apagará os dados)
7- Copiar de volta o seu sistema da partição temporária para a partição original (agora criptografada)
8- Configurar o boot de volta para a partição original (o /boot deverá ser levado para uma partição não criptografada)

Destaca-se que existe uma gama de outras maneiras de se atingir os mesmo resultados.

Este documento segue o primeiro modo, exposto acima, apenas por considerá-lo mais didático.

É importante ressaltar, também, que uma pequena partição (cerca de no máximo 100M) será necessária para conter o diretório /boot. Esta partição será não criptografada, pois conterá as ferramentas necessárias para descriptografar a raiz e fazer o boot do sistema. Normalmente escolhe-se usar um flash USB ou um CD para tal, criando um tipo de "chave" que deve ser conectada ao PC a fim de iniciar o gerenciador de boot e "arrancar" o sistema.

Preparação do Cenário

Para começar, deve-se verificar se o pacote crytpsetup está instalado:
```
#cryptsetup --version
```

A maioria das distribuições GNU/Linux já o incluem. Se não o encontrar, use o seu gerenciador de pacotes e o instale.

Suponha que as partições originais estão divididas como a seguir:

swap	/dev/sda1	
raiz_sem_criptografia	/dev/sda2	← aqui está o sistema de arquivos não criptografado
raiz_a_ser_criptografada	/dev/sdb1	← aqui será o novo sistema criptografado

Verifica-se quais algoritmos de criptografia são suportados pelo kernel:
```
#cat /proc/crypto
```

Normalmente estão carregados por padrão os módulos para o des*, aes e o md5. Se for necessário um nível melhor de segurança, pode-se carregar o módulo blowfish ou towfish e utilizá-lo mais adiante.

Os seguintes módulos se fazem necessários, os quais podem ser carregados como a seguir, caso ainda não estejam:
```
#modprobe dm-mod
#modprobe dm-crypt
#modprobe aes
#modprobe sha256
#modprobe sha1
```

Se não for especificado um algoritmo durante a criação do cabeçalho de criptografia (luks) na partição, visto mais adiante, o aes será escolhido, usando 128bits, com modo cbc ou essiv. Sugere-se a utilização de 256bits na chave, o que poderá afetar insignificantemente no desempenho, porém, se os discos oferecem acesso realmente lento e não houver grandes necessidades de níveis elevados de segurança, pode-se utilizar 128bits. Ademais, o modo essiv tem a importante característica de garantir o não repudio, uma vez que evita ataques conhecidos como watermark[1].

Criptografando a swap

Como de costume, inicia-se os testes pela swap[pobre cobaia =P]

1. Desativa-se a swap

```
#swapoff -a
```

2. Cria-se o cabeçalho LUKS mapeando para a partição swap. Nesta partição, a criptografia pode ser feita usando uma chave não conhecida, uma vez que os dados nela não são

importantes. Assim, uma vez que o computador é desligado, todos os dados contidos na swap se perderão permanentemente, dado que ela será montada com uma chave diferente a cada boot.

```
#cryptsetup -d /dev/urandom create swap /dev/sda1
```

O parâmetro –d /dev/urandom especifica a escolha de uma senha (chave) randômica a cada montagem da swap.

O comando acima usa algoritmo AES com 128bits por default, caso queira elevar essa segurança, pode-se usar um algoritmo mais forte, ao preço de desempenho, como abaixo:

```
#cryptsetup -c blowfish -s 256 -d /dev/urandom create swap /dev/sda1 (opcional)
```

Ou, ainda mais:

```
#cryptsetup -c twofish-cbc-essiv:sha256 -s 256 –d /dev/urandom create swap /dev/sda1(opcional)
```

Isto deverá criar um dispositivo em /dev/mapper/swap, que será a camada intermediaria de criptografia, conforme ilustra a Figura 1 para a partição raiz sdb1.

Figura 1 – Criptografia via camada mapper

Para verificar o status da partição, use:

```
#cryptsetup status swap
```

3. A nova partição deve ser aberta:

```
#cryptsetup -v luksOpen /dev/sda1 swap
```

4. Seta-se o novo dispositivo como swap:

```
#mkswap /dev/mapper/swap
```

5. Ativa-se a swap:

```
#swapon /dev/mapper/swap
```

Para verificar se a nova swap foi corretamente carregada, use um dos comandos abaixo:

```
#cat /proc/swaps
```

ou

```
#free
```

Caso tudo tenha corrido bem, já se tem uma nova swap criptografada funcionando. Agora se faz necessário configurá-la para ser carregada automaticamente durante o boot do sistema.

6. Em distribuições baseadas em debian, o arquivo **/etc/crypttap** deve ser alterado, adicionando-se a linha:

```
swap /dev/sda1 /dev/urandom swap
```

7. No Suse, distribuição na qual este procedimento foi executado, apenas o arquivo **/etc/fstab** deve ser editado (este arquivo também deverá ser editado em outras distribuições), adicionando-se a linha:

```
/dev/mapper/swap  swap  swap  defaults  0 0
```

Obs: Em algumas situações, o dispositivo mapeado para swap poderá ser montando com outro nome(diferente de /dev/mapper/swap), o qual, normalmente, se referencia por /dev/mapper/cr_<dispositivo>. No Suse Server, por exemplo, o nome do dispositivo ficou /dev/mapper/cr_sda1, e não /dev/mapper/swap. Assim, deve-se alterar o arquivo /etc/fstab e substituir a linha anterior pela correta a seguir:

```
/dev/mapper/cr_sda1    swap          swap      defaults          0 0
```

Para verificar o nome correto assumido pelo dispositivo, lista-se todos os dispositivos mapper com o comando abaixo:

```
#ls /dev/mapper
```

O swap recém criado será listado, podendo ser facilmente identificado. Após realizar as devidas correções, é hora de reiniciar o computador a fim de se verificar a automontagem da nova swap criptografada.

Os comandos abaixo, já mencionados, confirmam o funcionamento (ou não) da swap:

```
#cat /proc/swaps
```

ou

```
#free
```

Criptografando a Raiz

Concluído a swap, o processo com a raiz pode ser iniciado. Mais uma vez é válido lembrar que partição a ser criptografada para a raiz será a /dev/sdb1, a qual passará a ser o novo sistema de arquivos.

1. O seguinte comando cria o cabeçalho de criptografia luks na partição física, com algoritmo AES e 128bits:

```
#cryptsetup -v luksFormat /dev/
```

1.1 Como já mencionado, se for de interesse elevar o nível de segurança, sugere-se a utilização de um algoritmo mais forte, em modo essiv e com uma chave maior. Para tal, é usado o comando abaixo:

```
#cryptsetup -v -c twofish-cbc-essiv:sha256 -s 256 luksFormat /dev/sdb1 (opcional)
```

Argumentos:

-c: especifica o algoritmo. Pode-se, também, utilizar o blowfish como outra alternativa segura.

-s: especifica o tamanho da chave.

Obs: O módulo no kernel do algoritmo escolhido deve ser carregado (modprobe twofish, por exemplo), exceto se foi utilizado o AES, que já é carregado por default.

O comando acima solicita a escolha de uma chave (passphrsa), a qual será sempre necessária para abrir o sistema de arquivos. Em outras palavras, o usuário deverá sempre

fornecer esta senha ao iniciar o sistema, a fim de que o dispositivo mapper possa acessar os arquivos criptografados. Sem a senha, o conteúdo acessado não vai passar de “lixo”.

Em determinadas situações, como em laboratórios públicos, cybercafes e semelhantes, este método pode não ser “confortável”, dada a necessidade de um usuário em prontidão a fim digitar a senha sempre que um computador for reiniciado. O cryptsetup permite adicionar até oito chaves extras, que podem ser usadas por usuários diferentes e, se necessário, serem revogadas posteriormente, mas esta solução também não resolveria o problema de um laboratório freqüentado por muitos usuários.

Uma possível solução está na possibilidade de usar um arquivo como chave, o qual será utilizado para montar as partições automaticamente. Normalmente usa-se este arquivo em uma media segura, como um flash USB ou um CD, uma vez que ao ter acesso a ele, pode-se descriptografar os dados na partição.

A seguir será demonstrado o uso da solução que solicita a passphrsa ao usuário e, também, a solução via arquivo dentro da partição de boot, embora esta última não seja uma abordagem segura. O usuário deve escolher o método que melhor se adapte as suas necessidades.

2. Configuração da partição criptografada, utilizando o método que solicita uma senha digitada:

```
# cryptsetup -v luksFormat /dev/sdb1
```

Ou com mais segurança:

```
#cryptsetup -v -c twofish-cbc-essiv:sha256 -s 256 luksFormat /dev/sdb1 (opcional)
```

2.1 Configuração da partição criptografada, utilizando o método de senha via arquivo:

Deve-se gerar o arquivo com a chave, o que pode ser feito como abaixo:

```
#dd if=/dev/urandom of=/mnt/keyfile bs=1 count=256
```

Este comando criará o arquivo /mnt/keyfile, que pode ser usado como chave para a partição, de acordo com o comando abaixo:

```
# cryptsetup -v luksFormat /dev/sdb1 /mnt/keyfile
```

Ou com mais segurança:

```
#cryptsetup -v -c twofish-cbc-essiv:sha256 -s 256 luksFormat /dev/sdb1 /mnt/keyfile (opcional)
```

Feito o procedimento anterior, mais detalhes da partição podem ser obtidos com o comando abaixo:

```
#cryptsetup luksDump /dev/sdb1
```

3. Agora, abre-se a partição. Será solicitado a passphrase(chave) escolhida anteriormente:

```
#cryptsetup luksOpen /dev/sdb1 raiz
```

3.1 Caso se tenha usado um arquivo como chave, usa-se:

```
#cryptsetup luksOpen /dev/sdb1 raiz --key-file=/mnt/keyfile
```
← Este arquivo não pode ser perdido, caso contrario, toda a partição se perderá junto, uma vez que ele é a chave.

4. O sistema de arquivos deve ser definido na nova partição, o qual precisa concordar com o mesmo tipo do sistema de arquivos da partição original.

```
#mkfs.reiserfs -j /dev/mapper/raiz /dev/mapper/raiz
```
← Se a partição original for raiserfs

ou

```
#mkfs.ext3 -O dir_index,resize_inode /dev/mapper/raiz
```
← Se a partição original for ext3

Neste ponto, a camada intermediaria mapper já criptografa todo dado a ser gravado em /dev/sdb1 e descriptografa os dados a serem lidos, em tempo real. Os arquivos da partição original já podem ser passados para a nova partição criptografada, como a seguir:

5. Monta-se a partição:

```
#mkdir /mnt/raiz
#mount /dev/mapper/raiz /mnt/raiz
```

6. Faz-se a copia da raiz de diretórios para a nova partição critptografada, não sendo necessário copiar os diretórios /dev, /proc e /sys, os quais serão populados durante o boot.

```
#cd /
#cp -avx bin/ boot/ dev/ etc/ home/ lib/ media/ opt/ root/ sbin/ srv/ tmp/ usr/ var/ /mnt/raiz/
#cd /mnt/raiz
#mkdir proc
#mkdir sys
#mkdir mnt
```

7. Alterar arquivo /mnt/raiz/etc/fstab substituindo a linha da raiz anterior pela seguinte:

```
/dev/mapper/raiz     /      reiserfs   acl,user_xattr        1 1
```
← se a partição for raserfs

Finalmente já se pode criar o novo initrd, que gerenciará o boot, sendo o responsável por montar a camada mapper, descriptografando os dados a serem lidos e possibilitando o boot do sistema.

8. Um backup do initrd pode ser feito como abaixo:

```
#cp /boot/initrd-<versao-do-kernel> /boot/initrd.backup
```

9. Criação do novo initrd, que solicitará a senha durante o boot:

```
#mkinitrd -d /dev/mapper/raiz
```

9.1 Para o método que utiliza a senha via arquivo, faz-se:

```
#cp /mnt/keyfile /boot
#mkinitrd -d /dev/mapper/raiz -l /boot/keyfile
```

É necessário verificar se a versão do mkinitrd suporta LUKs. A sua documentação é uma boa fonte:

```
#man mkinitrd
```

Verifica-se a existencia da seguinte linha:

```
 -l keyfile
            Use "keyfile" as LUKS key.
```

Caso ela não se faça presente, talvez o mkinitrd instalado não suporte partições criptografadas. Assim, será necessário atualizar a versão do mkinitrd, usando o gerenciador preferível. No Suse Server 10, a versão corrente no mkinitrd já traz esse suporte, não tendo sido necessário a sua atualização.

Se tudo correu bem, durante a compilação do initrd será informado a inclusão do LUKS, a exemplo da linha em vermelho abaixo.

```
Filesystem modules:     reiserfs
Including:      LUKS initramfs fsck.reiserfs
Bootsplash:     SuSE-SLES (1024x768)
16244 blocks
joao:/boot #
```

← confirma o suporte a LUKS

O gerenciador do boot também deverá sofrer alterações, a fim de refletir a referencia para o sistema de arquivos na nova partição. Edite o arquivo /boot/grub/menu.list e adicione a seguinte entrada:

```
title Linux Cript
      root(hd0,1) #aqui entra o local da partição(não criptografada) que contém o /boot
```

```
                #(hd0,0) → sda1
                #(hd0,1) → sda2 … e assim por diante.
        kernel /boot/vmlinuz-2.6.16.60-0.21-default root=/dev/sdb1 vga=0x317 splash=silent showopts #o arqumento root=/dev/sdb1 informa o local da nova partição física.
        initrd /boot/initrd-2.6.16.60-0.21-default
```

Feito isto, o processo está concluído. Caso o método com chave via arquivo tenha sido o escolhido, o boot será feito automaticamente. Por outro lado, se o método de senha via usuário foi escolhido, será solicitada a chave durante o boot, para que a partição possa ser aberta.

A tabela de partição final é a seguinte:
/dev/sdb1 → raiz (criptografada)
/dev/sda1 → swap (criptografada)
/dev/sda2 → /boot + lixo! (não criptografada)

Como já mencionado, pode-se passar a usar outra media para o diretório boot. Caso deseje continuar usando a sda1, sugere-se um backup do diretório /boot e em seguida destrua o antigo sistema de arquivos que “sobrou” nesta partição. Por fim, deve-se redimensioná-la para que reflita apenas o diretório /boot.

Se o antigo sistema na sda1 não continha dados importantes, pode-se apenas remover todos os diretórios, exceto o /boot, e redimensionar a partição para um tamanho necessário apenas ao boot.

Conclusão

O método de criptografia apresentado eleva significantemente o sigilo das informações armazenadas nos discos rígidos, porém, ainda assim, lembra-se que técnicas demasiadamente sofisticadas podem obter bons resultados na recuperação dos arquivos[5]. Ainda assim, a criptografia de disco continua a ser um dos meios mais seguros desta categoria de proteção, visto que as melhores investidas contra ela, tais como a citada, exigem condições ideais a fim de obterem sucesso.

Referencias:

1. Linux hard disk encryption settings. Disponível em <http://clemens.endorphin.org/LinuxHDEncSettings>
2. DM-Crypt with LUKS. Disponível em <http://en.gentoo-wiki.com/wiki/SECURITY_System_Encryption_DM-Crypt_with_LUKS>
3. The Diceware Passphrase FAQ. Disponível em <http://world.std.com/~reinhold/dicewarefaq.html>
4. LUKS On Gentoo. Disponível em <http://www.saout.de/tikiwiki/tiki-index.php?page=LUKSOnGentoo>
5. Cientistas congelam memória de notebook e quebram criptografia de disco. Disponível em <http://www.inovacaotecnologica.com.br/noticias/noticia.php?artigo=cientistas-congelam-memoria-de-notebook-e-quebram-criptografia-de-disco>