Usando o eCryptfs para encriptar arquivos, montagem automatica do home

Versão em vídeo desta seção pode estar disponível no canal do *Guia Foca* no **YouTube**: Criptografia com o eCrypt.

O **eCryptfs** permite fazer criptografia a nível do sistema de arquivos. Ele consiste na criptografia de uma pasta dentro do sistema de arquivo chamada de `Private`. O suporte nativo ao **eCryptFS** foi adicionado oficionalmente a partir do kernel `2.6.19` e para ele funcionar, precisa que o suporte do kernel ao módulo `ecryptfs` esteja incluído na distribuição.

OBS: No `Debian`, o pacote **ecryptfs-utils** foi removido da *Buster* (*Debian 10*) devido a bugs na desmontagem automática do diretório `Private`, mantendo o conteúdo acessível para usuários com privilégios (BUG 765854). Este problema pode ser contornado adicionando-se o **ecyrptfs-umount-private** no `.bash_logout` (no entanto não é possível prever esse funcionamento em quedas de conexões inesperadas).

O pacote `ecryptfs-utils` pode ser instalado no sistema adicionado o repositório da `stretch` em `/etc/apt/sources.list`.

Após isso, execute o comando: **apt-get install ecryptfs-utils**

O padrão do ecryptfs é a montagem automatica da pasta no login dos sistema.

É requerido o suporte a key retention no kernel. Assim como habilitar o suporte criptográfico a CBC, ECB, MD5 e AES. Em `Miscelaneous filesystem`, e também habilitar o suporte a `eCrypt filesystem layer support` (que gera o módulo do kernel **ecryptfs**).

NOTA: O **eCryptfs** é ainda experimental, embora mais seguro que o **EncFS**. Como o formato interno pode mudar, é recomendável manter um backup de arquivos.

Configurando o ecryptfs

Para fazer a configuração `manual` do eCryptfs, primeiro selecione um diretório que receberá o conteúdo criptografado e também um ponto de montagem com o comando:

mount -t ecryptfs ~/origem ~/destino

O sistema perguntará a forma de criptografia desejada:

```
1) Frase Senha
2) tspi
```

Selecione a opção 1, você será perguntado do cifra que será usada (a AES é uma boa escolha):

```
Select cipher:
 1) aes: blocksize = 16; min keysize = 16; max keysize = 32
 2) blowfish: blocksize = 8; min keysize = 16; max keysize = 56
 3) des3_ede: blocksize = 8; min keysize = 24; max keysize = 24
 4) twofish: blocksize = 16; min keysize = 16; max keysize = 32
 5) cast6: blocksize = 16; min keysize = 16; max keysize = 32
 6) cast5: blocksize = 8; min keysize = 5; max keysize = 16
```

Após isso, será mostrada a mensagem pedindo para selecionar o tamanho de bits da chave:

```
Select key bytes:
  1) 16
  2) 32
  3) 24

Selection [16]: 3
```

Após isso, selecione ser texto plano deverá ser ativado:

```
Enable plaintext passthrough (y/n) [n]: n
```

Após, ser o nome do arquivo também deverá ser encriptado (isso é importante para reduzir risco de ataques dirigidos a arquivos específicos):

```
Enable filename encryption (y/n) [n]: n
Attempting to mount with the following options:
  ecryptfs_unlink_sigs
  ecryptfs_key_bytes=56
  ecryptfs_cipher=blowfish
  ecryptfs_sig=7261b2ffab9ae159
WARNING: Based on the contents of [~/.ecryptfs/sig-cache.txt],
it looks like you have never mounted with this key
before. This could mean that you have typed your
passphrase wrong.
```

Em seguida, o ecrypt perguntará se deseja prosseguir com a montagem:

```
Would you like to proceed with the mount (yes/no)? : yes
```

Agora você será perguntado se deseja adicionar a assinatura ao arquivo sig-cache.txt, para evitar o warning de montagem no futuro. Responda 'yes' para evitar novas mensagens sobre esse warning:

```
Would you like to append sig [9c21b2ffab9ae159] to
[/~/.ecryptfs/sig-cache.txt]
in order to avoid this warning in the future (yes/no)? : yes
Successfully appended new sig to user sig cache file
Mounted eCryptfs
```

Note que o cache de assinaturas é adicionado automaticamente em `~/.ecryptfs/sig-cache.txt` evitando o warning de que foi a primeira vez que montou o arquivo. É importante observar as opções de montagem usadas:

```
  ecryptfs_unlink_sigs
  ecryptfs_key_bytes=56
  ecryptfs_cipher=blowfish
  ecryptfs_sig=7261b2ffab9ae159
```

Após isso, o diretório será montado e tudo que for gravado em `~/origem`, será gravado criptografado em `~/destino`:

```
/dev/vda1                           ext2      236M   37M  187M  17% /boot
/dev/mapper/lpi303--debian--vg-home ext4      428M  7.9M  386M   3% /home
tmpfs                               tmpfs     100M     0  100M   0% /run/user/0
/home/guiafoca/origem                 ecryptfs  3.5G  1.7G  1.6G  52% /home/guiafoca/destino
```

Caso tenha obtido o erro: `Error mounting eCryptFS: [-2] No such file or directory`, verifique se tanto o diretório de origem ou destino existem em sua máquina. Caso estiver criando uma nova criptografia, crie os diretórios com o comando **mkdir -p ~/origem ~/destino** e repita os passos para criar os sistema de arquivos criptografado.

Remontando o sistema de arquivos criptografado

Na hora de remontar o sistema, ele perguntará novamente todos os dados, basta preencher corretamente, e seu conteúdo será novamente disponibilizado na pasta especificada. caso digite a frase senha *ERRADA*, o sistema mostrará o seguinte alerta:

```
WARNING: Based on the contents of [~/.ecryptfs/sig-cache.txt],
it looks like you have never mounted with this key
before. This could mean that you have typed your
passphrase wrong.

Would you like to proceed with the mount (yes/no)? : no
Aborting mount.
```

Caso prossiga, os arquivos criptografados anteriormente serão listados, mas ao tentar abrir o conteúdo, o ecryptfs retornará:

```
cat: teste2.txt: Input/output error
```

OBS: O sistema de criptografia garante privacidade nos arquivos, mas eles podem ser removidos por qualquer usuário do sistema permissões aprioriadas.

ecryptfsd

O **ecryptfsd** é um daemon userspace que executa operações sob o ponto de montagem **eCryptfs**. Ele requisita serviços de chave pública do módulo do kernel, enviando os mesmos via `/dev/ecryptfs`. O **ecryptfsd** somente precisa ser executado quando a monatgem é feita usando módulo chave pública.

Todas as chamadas ao

ecryptfsd

são servidas sob o contexto do usuário que rodou o daemon.

Diretório Privado Automático para o usuário

O utilitário **ecryptfs-setup-private** pode ser usado para configurar o diretório privado. Nesse caso, os utilitários **ecryptfs-mount-private** são usados para configurar a criptografia disponibilizando os dados de forma padronizada em `~/Private` e o **ecryptfs-umount-private** para desmontar o filesystem `~/Private`

A seguinte estrutura é usada nessa situação:

- `~/Private` - conteúdo descriptografado do ecrypfs
- `~/.Private` - Contém o conteúdo criptografado, em diretório oculto

A seguinte estrutura é criado em `~/.ecryptfs`:

```
-rw-r--r--   1 gleydson gleydson     0 jul  2 11:25 auto-mount
-rw-r--r--   1 gleydson gleydson     0 jul  2 11:25 auto-umount
-rw-------   1 gleydson gleydson    23 jul  2 11:25 Private.mnt
-rw-------   1 gleydson gleydson    34 jul  2 11:25 Private.sig
-rw-------   1 gleydson gleydson    42 jul  2 11:25 wrapped-passphrase
```

Para criar o diretório privado de usuário, execute o seguinte procedimento:

1. Execute o >**ecryptfs-setup-private**
2. Será solicitado sua senha de login no sistema para continuar

   ```
   Enter your login password [guiafoca]:
   ```

3. Será pedido a senha para montagem do sistema de arquivos criptografado. Selecione uma senha cuidadosamente seguindo os critérios de NNNNNNNN:

   ```
   Enter your mount passwphrase:
   ```

   ATENÇÃO: Guarde sua frase senha e armazene-a em um local seguro. Caso perca a frase-senha, não conseguirá mais ter acesso aos dados!

   ```
   Será exibida as seguintes mensagens de validação da montagem/desmontagem e leitura:
   ************************************************************************
   YOU SHOULD RECORD YOUR MOUNT PASSPHRASE AND STORE IT IN A SAFE LOCATION.
     ecryptfs-unwrap-passphrase ~/.ecryptfs/wrapped-passphrase
   THIS WILL BE REQUIRED IF YOU NEED TO RECOVER YOUR DATA AT A LATER TIME.
   ************************************************************************


   Done configuring.

   Testing mount/write/umount/read...
   Inserted auth tok with sig [cffe18c2df6fd3b2] into the user session keyring
   Inserted auth tok with sig [c70fae191ab3be11] into the user session keyring
   Inserted auth tok with sig [cffe18c2df6fd3b2] into the user session keyring
   Inserted auth tok with sig [c70fae191ab3be11] into the user session keyring
   Testing succeeded.

   Logout, and log back in to begin using your encrypted directory.
   ```

4. Agora, faça logout, e novamente login para poder começar a usar seu diretório `~/Private` montado.
5. Você pode desmontar o diretório no momento que desejar, mas ao invés de usar o comando **umount**, utilize:

ecryptfs-umount-private. Para montar novamente o diretório `~/.Private` sem a necessidade de logout/login, use **ecryptfs-mount-private**.

6. Para validar se a senha e conteúdo podem ser corretamente acessados e descriptografados, use o comando:

   ecryptfs-unwrap-passphrase ~/.ecryptfs/wrapped-passphrase

Migrando o home do usuário para criptografado

Caso deseje migrar um diretório home existente para criptografia, isso é possível com o **ecryptfs**. Quando o home é criptografado, assim que fizer o logon, os dados estarão disponíveis de forma automatica, e o `/home/usuario` deixará de ficar montado assim que o usuário fizer logout (usando o `pam_ecryptfs`):

1. Como root, rode: **ecryptfs-migrate-user -u usuario**
2. Será perguntada a passpharase do usuário, que deverá ser fornecida (siga os critérios de segurança em NNNNNNNN)
3. Peça para o usuário logar no sistema e ver se o sistema montou corretamente seu home. Se estiver tudo ok, o diretório de migração `/home/usuario.XXXXXXX` poderá ser removido
4. Rode `ecryptfs-unwrap-passphrase` e salve o código aleatório gerado

para que a montagem funcione automaticamente com o `pam_ecryptfs`, é necessário que a frase senha de acesso esteja sincronizada com o login de usuário, e que o módulo seja configurado no PAM com:

```
auth    required    pam_ecryptfs.so unwrap
password    optional    pam_ecryptfs.so
session    optional    pam_ecryptfs.so unwrap
```

PAM ecryptfs

Para configurar a montagem automática do eCryptFS, proceda da seguinte forma:

1. Monte o sistema de arquivos atual: **mount -t ecryptfs /root/testecripto /mnt**
2. Pegue os parametros de montagem do `/etc/mtab`:

   grep 'ecryptfs' /etc/mtab /root/testeecryptfs /mnt ecryptfs rw,relatime,ecryptfs_sig=5251a2b3b9ae159,ecryptfs_cipher=blowfish,ecryptfs_key_bytes=56,ecryptfs_unlink_sigs 0 0
3. Adicione a linha no `/etc/fstab` adicionando os parâmetros, `user` e `noauto`:
4. Desmote o compartilhamento montado
5. Adicione sua frase-senha do keyring via utilitário `ecryptfs-manager`, usando a `opção 1`.
6. Monte o diretório com: **mount -i /mnt'** (a opção `-i` impede o **mount** de chamar o helper ecryptfs externo).
7. O diretório deverá agora ser montado automaticamente.
8. Agora que o diretório montou usando o keyring, limpe a chave da sessão de usuário com o **keyctl clear @u**
9. Adicione o comando mount no `~/.bash_profile`: **mount -i /mnt**
10. Finalmente adicione isto ao seu arquivo `/etc/pam/login`, após o `pam_unix.so`:

    ```
    auth required pam_ecryptfs.so
    ```

No RedHat, pode ser usado o script `src/utils/ecryptfs-setup-pam.sh` para fornecer essa montagem automática

Limitações do eCryptfs

O **ecrypt** não deve ser usado para criptografar dispositivos de rede **NFS** . O **EncFS** (veja NNNNNNNN) é uma melhor opção nessa situação.

Nomes maiores que 143 caracteres não podem ser encriptados com a opção *FNEK* (`File Name Encryption Key`).

Caso crie arquivos de imagem com o **truncate** (comando muito usado para criação de imagens em virtualização ou de arquivos vazios, por ex: **truncate -s 10G teste.img**, será produzido um arquivo criptografado de 10Gb no sistema de arquivos origem.

Anterior | Subir | Próximo
Criptografia de arquivos usando ENCFS | Voltar ao Índice | Criptografia usando TrueCrypt/VeraCrypt