Sistemas de ficheros criptográficos

Adrián Macías Casado
Argelia Esther Martín Martín

CFS (Cryptographic file system)

- CFS permite encriptación de directorios y su contenido a través de la interfaz de UNIX standard
  - Sistema de ficheros ext2 u otros
- Nunca almacena texto en claro

Introducción (II)

- Los usuarios:
  - Acceden con una clave
  - Encriptación transparente

Flujo de eventos

- El directorio aparece como una unidad NFS (demonio cfsd)
- cfsd
  - Se conecta a localhost
  - Funciona como un servidor NFS
  - Responde a peticiones de lectura/escritura de un cliente

Tareas administrativas (I)

- Cfs requiere una partición “bootstrap” (virtual directory)
  - mkdir /.cfsfs
  - chmod 0 /.cfsfs
- Exportar partición
  - /etc/exports -> /.cfsfs localhost
  - /usr/sbin/exportfs -a
- Crear punto de montaje
  - mkdir /crypt
  - chmod 755 /crypt

Tareas administrativas (II)

- Cargar cfsd
  - /usr/sbin/cfsd *puerto* (3049)
- Montar el directorio virtual
  - /bin/mount -o port=3049,intr localhost:/.cfsfs /crypt

Usando CFS (I)

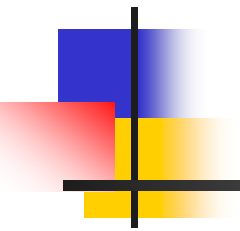
- Crear un directorio
  - cmkdir ~/directorio (introducire clave 16 car.)
  - Se pueden crear en cualquier parte
- Unirlo al directorio /crypt
  - cattach ~/directorio directorio_virtual
    - directorio_virtual: nombre del directorio en /crypt
    - Hay que introducir la clave
- Al terminar:
  - cdetach directorio_virtual

Usando CFS (II)

- Otros comandos:
  - cpasswd: cambiar password de un directorio
  - cmkkey: manejo de más de una clave

Vulnerabilidades de CFS

- Basado en NFS:
  - Incluye posibles vulnerabilidades derivadas de la red
- Aplicación de usuario:
  - Susceptible a análisis de swap
- Errores en la aplicación:
  - Pueden provocar un clave o datos de ficheros sean volcados en claro (core dump)
- Errores derivados de los permisos que se conceden a los usuarios sobre el sistema de ficheros virtual



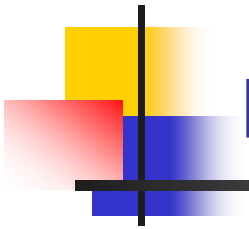
TCFS

Overview of TCFS

- Similar al CFS:
  - NFS para manejo de comunicaciones locales
  - Se interpone entre el usuario y el sistema de ficheros,
  - Garantiza que el texto claro no se almacene en un sistema de ficheros o se envíe sobre la red.

Overview of TCFS

- TCFS:
  - Proporciona un módulo del Kernel:reemplaza el cfsd
  - Incorpora un FS por separado: *tcfs* (directorio en el que se monta el TCFS)
    - Mejora rendimiento
    - todas las claves y conversiones cleartext <-> ciphertext (Kernel)
  - xattrd → ayudar a asociar las claves con directorios NFS montados
  - contraseñas de usuario→ encripta la clave→ encripta y desencripta de ficheros.



Para la instalación

- Bajarse la distribución:
  - Código fuente para el tcfs
  - El parche del núcleo
  - La biblioteca TCFS
  - Mount modificado
  - Shadow
  - Utilidades del ext2
  - Modulo del kernel para la encriptación

Utilidades TCFS

- Para usuarios, grupos y manejo de claves
  - *sbin/tcfsadduser* →(root) añade a la base de datos TCFS
  - *sbin/tcfsrmuser* →(root) elimina de la base de datos TCFS
  - *sbin/tcfsaddgroup* →(root) añade a la base de datos TCFS
  - *sbin/tcfsrmgroup* →(root) eliminina de la base de datos TCFS
  - *bin/tcfsgenkey* →(all user) genera una clave de encrip-desencrip
  - *bin/tcfsputkey* →(all user) poner la clave de usuario en el kernel
  - *bin/tcfsrmkey* →(all user) borra la clave de usuario del kernel
  - *bin/tcfsviewkey* →(all user) ver la clave

Configurando TCFS para uso con PAM

- añadir entradas *pam_tcfs* al fichero */etc/pam.d/passwd*

- Debemos conocer cuando las password de usuarios cambian ya que se usan para encriptar la clave de encriptación y desencriptación de ficheros

Tareas administrativas :Servidor

- Estar seguro de que corre el xattrd (# usr/local/sbin/xattrd)
- Podrías necesitar habilitar un unidad NFS para usarlo:
  - Añadir el recurso /work a /etc/exports:

    /work localhost pluto topcat
- Exportar un recurso como uno normal de NFS

  # exportfs -a
- Crear en /work, directorios cuyos propietarios sean quienes los usan con directorio TCFS (prooveer un direct. de trabajo para user mary)

  # mkdir /work/mary

  # chown mary.mary /work/mary
- El root debe add a la base de datos el user.

Tareas administrativas: Cliente

- Montar el recurso (directorio local de montaje: /mnt/safe)
  - Si el recurso es accedido localmente:

    # mount –t tcfs localhost:/mnt/safe

  - Si el recurso es accedido remotamente :

    [root@pluto]# mount –t tcfs underdog:/work /mnt/safe

Ahora que el recurso está habilitado, los usuarios TCFS pueden usarlo.

Atributos extendidos

- Configurar los atributos apropiados para asegurar la encriptación de los ficheros en el directorio de trabajo. Dos atributos :
  - X: indica que el fichero al que es aplicado va a ser encriptado. (Si es aplicado a un directorio, todos los ficheros y subdirectorios de ese directorio serán encriptados)
  - G: se usa para indicar que el fichero o directorio será accesible (encriptado y desencriptado) para el grupo de TCFS.

```
[mary@underdog]$ chattr +X /mnt/safe/mary
```

(permitiendo que todos los ficheros y directorios sean encriptados después de que /mnt/safe/mary sea montado como un recurso TCFS)

Configurando el directorio encriptado

- Para que Mary pueda crear directorios y fichero encriptados, primero tiene que crearse una clave encripatada para luego añadirse al kernel
  - Generar la clave encriptada:

    [mary@topcat]$ /usr/local/bin/tcfsgenkey

  - Ponerla en el módulo correspondiente en el kernel:

    [mary@topcat]$ /usr/local/bin/tcfsputkey

Configurando el directorio encriptado

Ahora los ficheros y directorios creados por Mary en /mnt/safe/mary podrán ser encriptados para el recurso underdog:/work/mary. Sólo mary será capaz de ver el cleartext por /mnt/safe/mary

- Crear un fichero *secret* en /mnt/safe/mary sobre pluto
  [mary@pluto]$ echo “hola” > /mnt/safe/mary/secret
- El fichero real sobre underdog in /work/mary y su contenido están encriptados
  [mary@underdog]$ ls /work/mary
- no aún el root tendrá acceso a los files encrip. de mary:
  [root@pluto]$ cd /mnt/safe/mary
  [root@pluto]$ ls
  ls: .: permission denied

Grupos TCFS

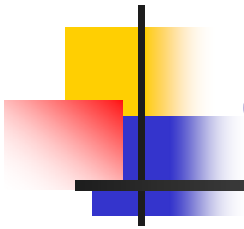
- Los nombres de grupo TCFS deben coincidir con los de */etc/group*
- Pueden ser configurados con cualquier número de usuarios
- Pueden ser configurados para requerir que un número mínimo de usuarios pongan sus claves en el Kernel antes de que el acceso a los ficheros y directorios del grupo sea garantizado.

Gestión de la clave del TCFS

- Ver clave:

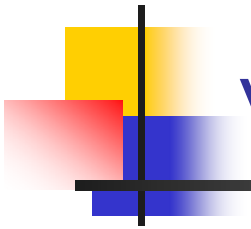
  [mary@topcat]$ /usr/local/bin/tcfsviewkey –k
- Ver si la clave ha sido añadida al kernel con tcfsputkey:

  [mary@topcat]$ /usr/local/bin/tcfsviewkey –c
- Cambiar la clave: eliminar el usuario y añadirlo de nuevo (root). Una vez hecho, el usuario puede generar una nueva clave
- Numerosas claves pueden ser generadas por el usuario.
- Grabar la claves del TCFS generadas a través del comando *tcfsviewkey*
- Cada clave puede ser puesta en el kernel
- La clave usada para encriptar un fichero debe ser puesta dentro del Kernel



Gestión de la clave del TCFS

- Problema fundamental: las claves están atadas a las contraseñas de usuario.
- Si se usan otras claves, deben ser conocidas en orden para desencriptar cualquier fichero encriptado con ellas.
- Esto significa que el uso de múltiples claves, mientras más seguro, requiere la gestión manual de cada clave o el uso de un software específico.



Vulnerabilidades del TCFS

- cleartext : almacenado en la RAM o en la cache de un FS.
- Al contrario que con el CFS, esto ocurre sólo en el espacio del Kernel y durante cortos períodos de tiempo
  → TCFS + seguro
- Vulnerabilidad de TCFS : las claves deencrip/desencrip son accesibles con la contraseña de usuario. No asociar estas claves a la contraseña de usuario es posible, pero engorroso.
- TCFS se encuentra actualmente en desarrollo activo.

Comparación entre CFS y TCFS

- TCFS:
- mejor implementación de rendimiento
- implementación transparente a los usuarios porque sus recursos son accedidos a través de un simple NFS.
- débil método de generación de clave y, por defecto, sólo soporta encriptación triple DES
- carece de una buena documentación
- bastante díficil de instalar y requiere intervención administrativa

Ambas ofrecen beneficios en la encriptación de ficheros

Borrado seguro de ficheros

- Cuando borras un fichero del sistema de ficheros *ext2* , pueden quedar datos temporalmente almacenados en la RAM o en la caché de disco → concierne al CFS y al TCFS.

- Para mejorar la seguridad : usar el atributo de borrado

  # chattr +s secret_file → los bloques en el fichero sean puesto a 0.

- Utilidades de borrado seguro más eficientes : wipe, bcwipe o el secure_delete .