

SOAD Laboratorio

Lab 5: Security

Adrià Abad Moreno

Raül Sampietro Gutierrez

Group 12

Q2 2021-22

Index

1. Introduction	2
2. Protección de acceso a ficheros y directorios	3
2.1. Entorno de pruebas	3
2.2. Pruebas de protección	4
3. Cifrado de ficheros con el editor Vim	6
4. Evaluación de coste del cifrado de ficheros	8
5. Conclusiones	9

1. Introduction

En la quinta y última práctica de laboratorio estudiaremos diferentes técnicas de seguridad para proteger y acceder al sistema, trabajando con una imagen de Ubuntu.

Primeramente, veremos cómo afectan los permisos que se otorgan a usuarios y grupos al acceso a ficheros y directorios. Seguidamente, encontramos como cifrar ficheros mediante la herramienta vim. Finalmente, hacemos un análisis del coste espacial que tiene la encriptación en ficheros de distintos tamaños.

2. Protección de acceso a ficheros y directorios

2.1. Entorno de pruebas

En una imagen de Ubuntu, hemos creado un directorio llamado **soaddir** y un grupo llamado **soad**, y hemos cambiado el grupo del **soaddir** al grupo **soad**.

```
$ mkdir soaddir
$ sudo addgroup soad
$ sudo chgrp soad soaddir
```

También le hemos dado permisos de escritura a los miembros del grupo sobre el directorio mediante el siguiente comando:

```
$ sudo chgmod g+w soaddir
```

Al hacer el comando `$ ls -la` observamos como efectivamente el grupo de `i` es `soad` y que tanto `samragu`, como `soad` tienen permisos de lectura, escritura y ejecución. Sin embargo los usuarios externos al grupo solo tienen permisos de lectura y ejecución.

```
samragu@samragu-ps42:~/Documents/SOAD/Laboratorios/LAB-5$ ls -la
total 12
drwxrwxr-x 3 samragu samragu 4096 may 18 16:18 .
drwxrwxr-x 6 samragu samragu 4096 may 18 16:18 ..
drwxrwxr-x 2 samragu soad    4096 may 18 16:18 soaddir
```

Resultado del comando `$ ls -la`

Además, hemos creado una serie de usuarios y algunos los hemos asignado al grupo `soad` y otros no.

username	soad group
alumne1	X
alumne2	X
alumne3	

Pertenencia de usuarios al grupo `soad`

2.2. Pruebas de protección

Una vez creado el fichero, el grupo y los usuarios, procedemos a las pruebas.

Primero entramos al directorio como **alumne1** y creamos un archivo llamado `file.txt` en el que escribimos “esto es un texto de prueba” y ejecutamos `$ ls -la`.

```
alumne1@samragu-ps42:/home/samragu/Documents/SOAD/Laboratorios/LAB-5/soaddir$ echo "esto es un texto de p
rueba" > file.txt
alumne1@samragu-ps42:/home/samragu/Documents/SOAD/Laboratorios/LAB-5/soaddir$ ls -la
total 12
drwxrwxr-x 2 samragu soad  4096 may 18 16:54 .
drwxrwxr-x 3 samragu samragu 4096 may 18 16:18 ..
-rw-rw-r-- 1 alumne1 alumne1  27 may 18 16:54 file.txt
```

Escritura y permisos de `file.txt` desde `alumne1`

Vemos que `alumne1` es el propietario de `file.txt` y que tiene permisos de lectura y escritura; y los otros usuarios tienen permiso solo de lectura. Si hacemos `$ cat file.txt` obtenemos como salida el contenido del fichero.

Ahora procedemos a crear otro fichero e intentar leer el fichero como **alumne2**. Con éste usuario obtenemos exactamente la misma salida para el comando `$ ls -la` que con `alumne1`.

```
alumne2@samragu-ps42:/home/samragu/Documents/SOAD/Laboratorios/LAB-5/soaddir$ ls -la
total 12
drwxrwxr-x 2 samragu soad  4096 may 18 16:54 .
drwxrwxr-x 3 samragu samragu 4096 may 18 16:18 ..
-rw-rw-r-- 1 alumne1 alumne1  27 may 18 16:54 file.txt
```

Permisos de `file.txt` desde `alumne2`

```
alumne2@samragu-ps42:/home/samragu/Documents/SOAD/Laboratorios/LAB-5/soaddir$ echo "alumne2 ha estado aqu
i" > file.txt
-bash: file.txt: Permission denied
```

Intento de escritura en `file.txt` desde `alumne2`

Hemos podido leer el archivo pero no hemos podido escribir en `file.txt`. Ésto se debe a que el grupo que figura como propietario de `file.txt` es `alumne1` y no `soad`, ya que si cambiamos el grupo a `soad`, `alumne2` sí que puede escribir en `file.txt`.

```
samragu@samragu-ps42:~/Documents/SOAD/Laboratorios/LAB-5/soaddir$ sudo chgrp soad file.txt
[sudo] password for samragu:
samragu@samragu-ps42:~/Documents/SOAD/Laboratorios/LAB-5/soaddir$ ls -la
total 12
drwxrwxr-x 2 samragu soad  4096 may 18 16:54 .
drwxrwxr-x 3 samragu samragu 4096 may 18 16:18 ..
-rw-rw-r-- 1 alumne1 soad  27 may 18 16:54 file.txt
```

Cambio de grupo de `file.txt` a `soad` desde `samragu`

```
alumne2@samragu-ps42:/home/samragu/Documents/SOAD/Laboratorios/LAB-5/soaddir$ echo "alumne2 ha estado aqu
i" > file.txt
alumne2@samragu-ps42:/home/samragu/Documents/SOAD/Laboratorios/LAB-5/soaddir$ cat file.txt
alumne2 ha estado aquí
```

Escritura en `file.txt` desde `alumne2`

Cómo **alumne2** también podemos crear ficheros, y hemos creado **file2.txt**. El cual tiene como propietario y como grupo a **alumne2**, con permisos de lectura y escritura. Otros usuarios solo pueden leer **file2.txt**.

```
alumne2@samragu-ps42:/home/samragu/Documents/SOAD/Laboratorios/LAB-5/soaddir$ echo "file de alumne2" > file2.txt
alumne2@samragu-ps42:/home/samragu/Documents/SOAD/Laboratorios/LAB-5/soaddir$ ls -la
total 16
drwxrwxr-x 2 samragu soad  4096 may 18 17:52 .
drwxrwxr-x 3 samragu samragu 4096 may 18 16:18 ..
-rw-rw-r-- 1 alumne2 alumne2  16 may 18 17:52 file2.txt
-rw-rw-r-- 1 alumne1 soad    23 may 18 17:12 file.txt
```

Creación y permisos de **file2.txt** desde **alumne2**

Por último, haremos las pruebas de lectura y escritura en el directorio y en los ficheros desde el usuario **alumne3**.

Primero probamos a leer los archivos creados previamente. Como que **alumne3** no pertenece al grupo **soad**, ni es el usuario **alumne2**, solo puede leer los ficheros y no puede escribir en ninguno de los dos.

```
alumne3@samragu-ps42:/home/samragu/Documents/SOAD/Laboratorios/LAB-5/soaddir$ ls -la
total 16
drwxrwxr-x 2 samragu soad  4096 may 18 17:52 .
drwxrwxr-x 3 samragu samragu 4096 may 18 16:18 ..
-rw-rw-r-- 1 alumne2 alumne2  16 may 18 17:52 file2.txt
-rw-rw-r-- 1 alumne1 soad    23 may 18 17:12 file.txt
alumne3@samragu-ps42:/home/samragu/Documents/SOAD/Laboratorios/LAB-5/soaddir$ cat file.txt
alumne2 ha estado aqui
alumne3@samragu-ps42:/home/samragu/Documents/SOAD/Laboratorios/LAB-5/soaddir$ cat file2.txt
file de alumne2
alumne3@samragu-ps42:/home/samragu/Documents/SOAD/Laboratorios/LAB-5/soaddir$ echo "soy alumne3" > file.txt
-bash: file.txt: Permission denied
alumne3@samragu-ps42:/home/samragu/Documents/SOAD/Laboratorios/LAB-5/soaddir$ echo "soy alumne3" > file2.txt
-bash: file2.txt: Permission denied
```

Lectura e intento de escritura de **file.txt** y **file2.txt** desde **alumne3**

De igual forma, tampoco hemos podido crear un archivo dentro de **soaddir** debido a que el directorio solo tiene permisos de escritura para el usuario propietario (**samragu**) y el grupo **soad**.

```
alumne3@samragu-ps42:/home/samragu/Documents/SOAD/Laboratorios/LAB-5/soaddir$ echo "fichero del alumne3" > file3.txt
-bash: file3.txt: Permission denied
```

Intento de creación de un fichero en **soaddir** desde **alumne3**

3. Cifrado de ficheros con el editor Vim

Para probar el cifrado de ficheros del editor Vim hemos usado uno de los ficheros de resultado que se generaban en la sesión 4 del laboratorio. El fichero contiene una serie de 52 líneas con 7 valores numéricos en cada una, separados por tres espacios.

```
...
0.00  3115.00  6230.00  9345.00  12460.00  15575.00  18690.00
0.00  3220.00  6440.00  9660.00  12880.00  16100.00  19320.00
0.00  3640.00  7280.00  10920.00  14560.00  18200.00  21840.00
...
```

Contenido del fichero de ejemplo sin cifrar.

Una vez hemos cifrado el fichero, el contenido de este es el siguiente:

```
VimCrypt~03!CÑ^FZ³^V<9d>^^ (r)Q@òN0P<8b><86>}i^O&Ã|[GV«o¥[^R.@FÕ^DèiTÑÇB'
~<94><93><9b>^O<9f>p¹ÃQð^E`<92>ðð<93>^Z,¹3Û^W:Î¤£ó#<8c>1/2tqÖ,$i1/4·Õ^FÆ
Y Ñ"ñ^£^M·<87>Ó<9c>%ôÅ^Uí¥<92>^<95>Æ>W¢           ê%,)
¢(c)^G<83>ÿ1/4^XÜ«2ö)^M^B^Z
...
```

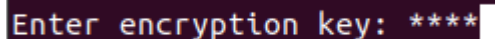
Contenido del fichero de ejemplo cifrado.

Como podemos observar, el contenido del fichero es ilegible. No obstante, podemos ver que al principio de todo aparece “*VimCrypt*”, lo que indica que el cifrado se ha hecho a través de Vim.

Para cifrar el fichero, primero hemos creado una copia del fichero base y entonces lo hemos abierto con el editor Vim usando la opción -x.

```
$ cp ../SOAD_Lab4/v2/results_v2.txt ficheroEjemplo.txt
$ cp ficheroEjemplo.txt ficheroEjemploCifrado.txt
$ vim ficheroEjemploCifrado.txt -x
```

Acto seguido, nos pide que proporcionemos una clave de cifrado. Esta clave será la que tenemos que usar para poder volver a ver el contenido original del fichero al abrirlo con Vim.



Petición de la clave para cifrar

Petición de la clave para descifrar

Para poder comparar el tamaño de los ficheros hemos usado el comando `ls` con la opción `-l`, que nos proporciona el resultado en formato de lista y con una serie de información para cada componente. `$ ls -l`

```
adria@adria-VirtualBox:~/Documents/soad/LAB5$ ls -l
total 8
-rw-rw-r-- 1 adria adria 3854 may 18 16:55 ficheroEjemploCifrado.txt
-rw-rw-r-- 1 adria adria 3826 may 18 16:24 ficheroEjemplo.txt
```

Resultado de ejecutar `ls -l`

Como podemos ver en la quinta columna, el fichero encriptado tiene un tamaño ligeramente superior al que no está encriptado. En concreto, estamos hablando de 28 bytes de diferencia.

4. Evaluación de coste del cifrado de ficheros

Hemos hecho un script en C (**encryption_test.c**) que se encarga de generar archivos normales y sus versiones encriptadas y da como resultado cuanto más grandes son los archivos encriptados que sus versiones no encriptadas. En concreto genera N ficheros sin encriptar (siendo N el parámetro de ejecución), genera las versiones encriptadas mediante el comando `vim -xs`. Para poder pasar las claves de encriptación, usamos el fichero **vim.imp**.

Los archivos que genera són siguiendo la siguiente regla: el primer archivo tiene 1 línea de texto, el segundo tiene 2 y el N-ésimo tiene N líneas. Todas las líneas són iguales.

Para compilar y ejecutar el programa simplemente hay que ejecutar el script bash **exe.sh**, pasando como argumento el número de ficheros que deseamos crear. Este script limpia el directorio de ficheros de ejecuciones anteriores, recompila el ejecutable y lo ejecuta. Si se desea eliminar los ficheros generados simplemente, se puede hacer **make clean**.

Como resultado obtenemos la relación media entre el tamaño de un fichero encriptado respecto a su versión sin encriptar. La siguiente tabla muestra cómo varía el resultado en función de la entrada. La relación indica cuánto más grandes son los archivos encriptados de media.

nFiles	Relación media de tamaño
1	1.947368
2	1.734554
5	1.465228
10	1.305400
20	1.190849
50	1.097105

5. Conclusiones

Como conclusiones del estudio sobre la protección de acceso a ficheros y directorios vemos como solo los usuarios con permisos explícitos pueden escribir en los ficheros y directorios. Además, los permisos de un directorio no los heredan los ficheros y directorios contenidos en éste.

Por otro lado, vemos como el cifrado de ficheros vuelve ininteligible su contenido. Sólo si se conoce la clave de encriptación se podrán recuperar los datos del fichero. Esto resulta de especial utilidad para guardar datos críticos o secretos o para enviarlos por Internet, de manera que si alguien los intercepta, no podrá descifrar-los (o al menos le será difícil).

Del lado del estudio del tamaño de los archivos encriptados, vemos como la encriptación de un archivo pequeño resulta en un archivo de prácticamente el doble de tamaño. En cambio, a medida que incrementamos la media de tamaño de los archivos, vemos como el tamaño de la encriptación disminuye hasta el punto de que casi no supone ningún perjuicio al tamaño.