



Reconocimiento-CompartirIgual 3.0
España (CC BY-SA 3.0 ES)

Introducción a la Ciberseguridad

Práctica 2: Sistemas Operativos

Marta Beltrán Pardo

Miguel Calvo Matalobos

Agradecimientos (versiones anteriores): Isaac Martín de Diego y Alberto Fernández Isabel

Contenidos

Contenidos.....	2
1. Introducción	3
2. Material de la práctica.....	5
3. Normativa y evaluación	6
4. Enunciado de la práctica	7
Instrucciones	7
1. Linux: Árbol de directorios	17
2. Linux: Comandos básicos	22
3. Linux: Comprimir y descomprimir archivos	26
4. Linux: Administración.....	29
5. Linux: Procesos	35
6. Linux: Instalación de programas, scripts y actualizaciones	40
7. Linux: Logs del sistema	44
8. Control de acceso y contraseñas	47
5. Anexo I	55

1. Introducción

En esta práctica vamos a trabajar con **GNU/Linux**, que es un tipo de sistema operativo (SO) UNIX, normalmente de código abierto, multiplataforma, multiusuario y multitarea. Todo su código fuente puede ser utilizado, modificado y redistribuido libremente por cualquier persona, empresa o institución bajo los términos de la Licencia Pública General de GNU. Se considera un sistema operativo robusto, portable, versátil, escalable, rápido. La mayor parte de su código está escrito en lenguaje de programación C y puede encontrarse en gran cantidad de dispositivos como teléfonos inteligentes, automóviles, supercomputadoras, electrodomésticos, ordenadores domésticos, servidores, etc.

Los sistemas operativos GNU/Linux se hacen llegar a los usuarios en forma de distribuciones (Ubuntu, Debian, CentOS y un largo etcétera), que no son más que el núcleo del sistema operativo y conjuntos de interfaces, aplicaciones y programas con las cuales el sistema cuenta directamente al ser instalado. Si hace falta ampliar este conjunto, se pueden descargar más desde diferentes repositorios. En esta práctica vamos a trabajar con la distribución **Kali Linux**, basada en Debian GNU/Linux y diseñada principalmente la seguridad informática (sobre todo ofensiva).



Figura 1. Algunas distribuciones de GNU/Linux

([https://commons.wikimedia.org/wiki/File:Linux_distro_foto_no_exif_\(1\).jpg](https://commons.wikimedia.org/wiki/File:Linux_distro_foto_no_exif_(1).jpg))



Figura 2. Logotipo de Kali Linux

(https://commons.wikimedia.org/wiki/File:Kali_Linux_2.0_wordmark.svg)

En parte, gracias a que son sistemas de código abierto, su diseño facilita que sus usuarios adquieran, si así lo quieren, conocimientos más profundos sobre el SO, su configuración funcionamiento, etc. Para alcanzar esta meta, es imprescindible leer documentación, consultar los manuales de los propios comandos y, por supuesto, practicar.

En esta práctica, mediante pequeñas explicaciones y ejercicios prácticos, se tratarán diversos temas como el árbol de directorios en Linux, algunos comandos básicos, la administración en Linux, los procesos, la instalación de programas y actualizaciones, los logs del sistema, el control de acceso y contraseñas, etc.

Recurriremos para ello a la virtualización, de manera que tengáis una distribución GNU/Linux disponible para trabajar en vuestros equipos. **VirtualBox** es un software de código abierto, perteneciente a Oracle Corporation, que actúa como hipervisor y, por ende, permite virtualizar diferentes sistemas informáticos creando máquinas virtuales donde el usuario puede ejecutar otros SO. Este software es compatible con Windows, Linux o macOS. Al configurar una nueva máquina virtual, el usuario puede especificar cuántos núcleos de CPU desea adjudicarle a su máquina virtual, cuánta memoria RAM, espacio en disco duro, etc.; en este punto, debe tenerse en cuenta que los valores del hardware virtualizado que vamos a adjudicar a la máquina virtual no deben superar, en ningún caso, los del host físico (además, se debe dejar un margen en los recursos para que este host sea capaz de ejecutar su propio SO y el software que corre en él, como puede ser el propio VirtualBox). Esta herramienta puede descargarse [aquí](#).

2. Material de la práctica

A continuación, se exponen los archivos necesarios para la realización de esta práctica, que pueden descargarse desde el Aula Virtual:

- **Practica2_Guion.pdf:** este documento. Se corresponde con el guion de la práctica y en él están contenidas todas las explicaciones necesarias para su realización.
- **Practica2_CommandReference.pdf:** hoja con los principales comandos para UNIX/Linux y su explicación.
- **Practica2_Material.zip:** archivo comprimido que contiene el material necesario (los ficheros “shadow” y “passwd” para la realización de los ejercicios del apartado “Control de acceso y contraseñas”. Para utilizarlo, se debe descomprimir.

Además, será necesario descargar el programa **VirtualBox** en su última versión. Este programa, servirá para crear la máquina virtual que se utilizará a lo largo del desarrollo de esta práctica. Puede descargarse desde el [siguiente enlace](#). También será necesario el pack de extensiones “Oracle VM VirtualBox”, que permitirá la configuración y el uso de ciertos parámetros y características en las máquinas virtuales (puede descargarse [aquí](#)).

También hay que descargar una distribución del sistema operativo “**Kali Linux**”. Este SO se ofrece en diferentes versiones, entre ellas, una preparada para su ejecución como MV para VirtualBox. Este sistema operativo será el utilizado a lo largo del desarrollo de esta práctica y puede descargarse, en su última versión, de [este](#) enlace.

3. Normativa y evaluación

En este apartado se detalla el formato de entrega de la práctica y la forma en la que se evaluará la misma:

- El porcentaje de la nota final de la asignatura al que corresponde esta práctica puede consultarse en la Guía docente de la propia asignatura.
- La práctica deberá realizarse de manera individual.
- Cada estudiante deberá:
 - Leer, comprender y realizar los ejercicios propuestos en esta guía.
 - Realizar una prueba tipo test que se compartirá en el Aula Virtual al final de la segunda sesión de la práctica.
- La calificación de la práctica dependerá de la realización de esta prueba tipo test a través del Aula Virtual. NO será necesario entregar ninguna memoria ni documento de prácticas, pero sí deberán haberse realizado los ejercicios propuestos en esta guía para comprender las preguntas de la prueba tipo test y poder superarla con éxito.

4. Enunciado de la práctica

En este apartado se describirán las distintas actividades, programas y ejercicios que deberá realizar cada grupo de prácticas.

Instrucciones

Antes de comenzar a realizar la práctica, es necesario descargar tanto la herramienta [VirtualBox](#) (es importante que selecciones la opción asociada al sistema operativo que tengas instalado en tu ordenador y a su arquitectura del sistema), como una distribución del sistema operativo “[Kali Linux](#)” (este SO se ofrece en diferentes versiones, entre ellas y recomendada para la realización de esta práctica, una preparada para su ejecución como MV para VirtualBox) y el pack de “[extensiones de Oracle VM VirtualBox](#)”.

Una vez descargados, se procederá a instalar la aplicación Oracle VM VirtualBox. Para ello, seguiremos la instalación guiada, tal y como se muestra en la Figura 1.

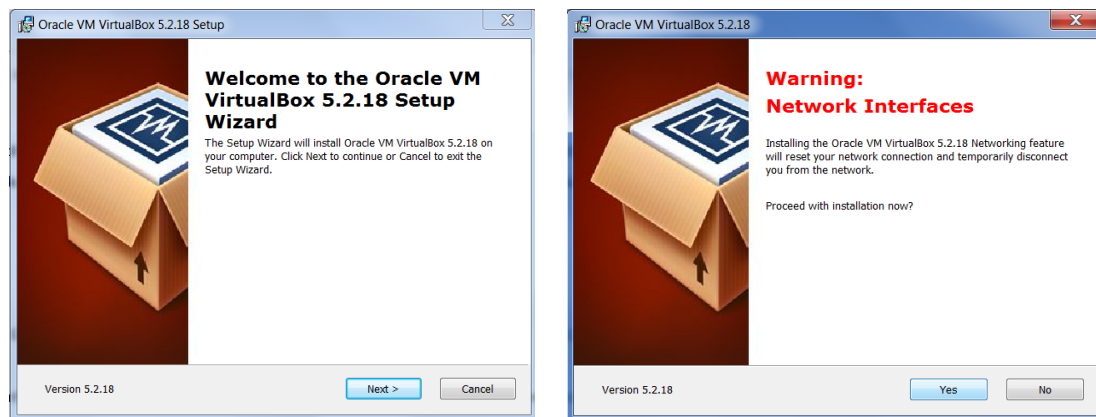


Figura 1. Instalación de Oracle VM VirtualBox.

Una vez instalado VirtualBox, deben instalarse el pack de extensiones “Oracle VM VirtualBox Extension Pack” descargado previamente. Para ello, bastará con hacer doble clic sobre el archivo

descargado y pulsar el botón “Instalar” de la pantalla que esta acción habrá abierto. Tras aceptar el acuerdo de licencia y aceptar los permisos de instalación, el paquete habrá quedado instalado.

El siguiente paso, será la importación de la máquina virtual de “Kali Linux” descargada previamente. Para ello, en primer lugar, se debe abrir el software VirtualBox (véase la Figura 2) e ir al menú “Archivo” → “Importar servicio virtualizado” (Figura 3); esta acción abrirá un desplegable como el que puede observarse en la Figura 4, donde debe seleccionarse, pulsando en el icono del directorio, el archivo “.ova” correspondiente a la máquina virtual de “Kali Linux” que se ha descargado. En caso de haberse descargado una iso de “Kali Linux” en lugar de la máquina virtual ya creada, debe pincharse sobre “Nueva” y configurar todas las opciones que el software nos solicite (nombre, tipo, versión, cantidad de RAM, tamaño de disco duro, etc.).

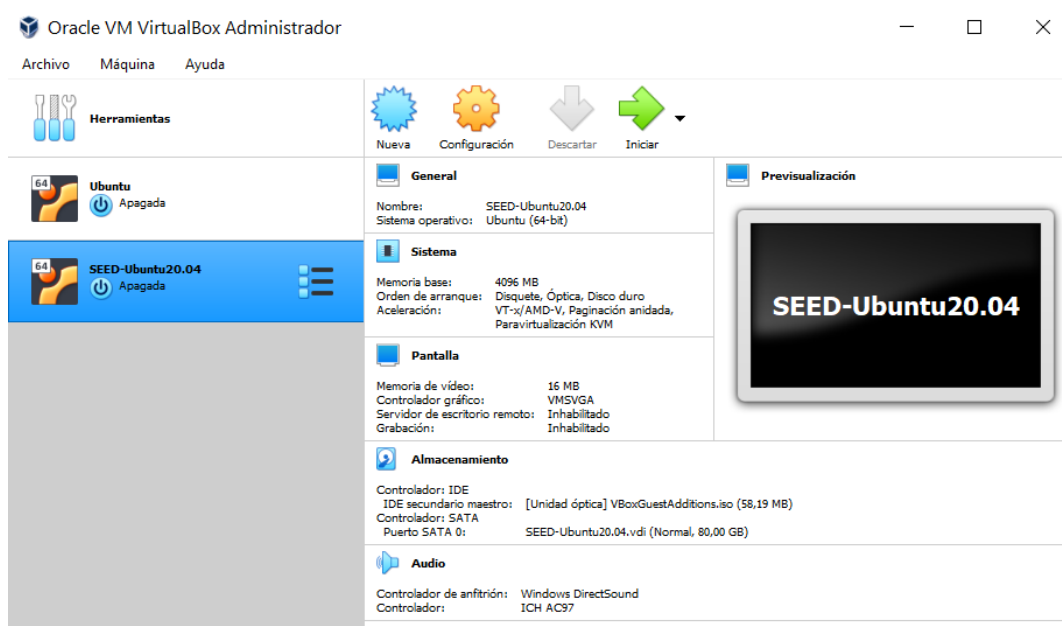


Figura 2. Pantalla principal de VirtualBox.

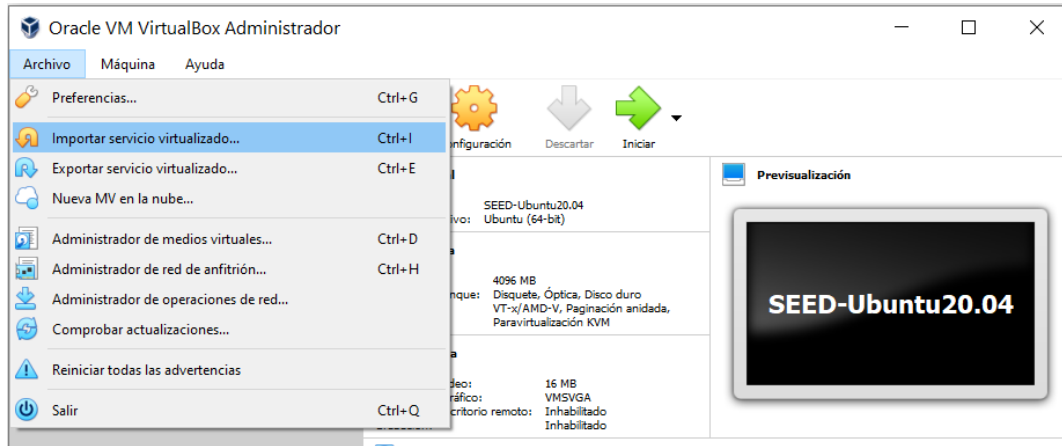


Figura 3. Menú de importación de servicio virtualizado en VirtualBox.

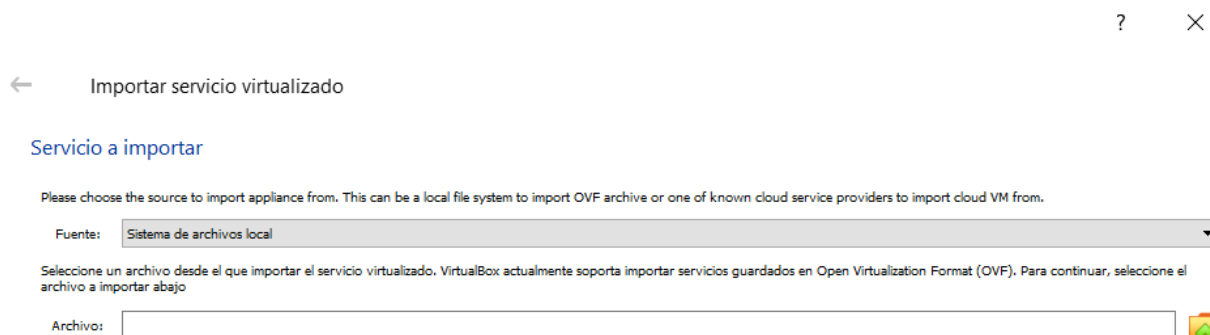


Figura 4. Desplegable para la importación de un servicio virtualizado en VirtualBox.

Una vez revisados todos los parámetros de la máquina virtual que se va a importar (es importante que se tenga en cuenta tanto el número de CPUs como el tamaño de la RAM para, en ningún caso, superar la CPU/RAM del host anfitrión). Véase la Figura 5.

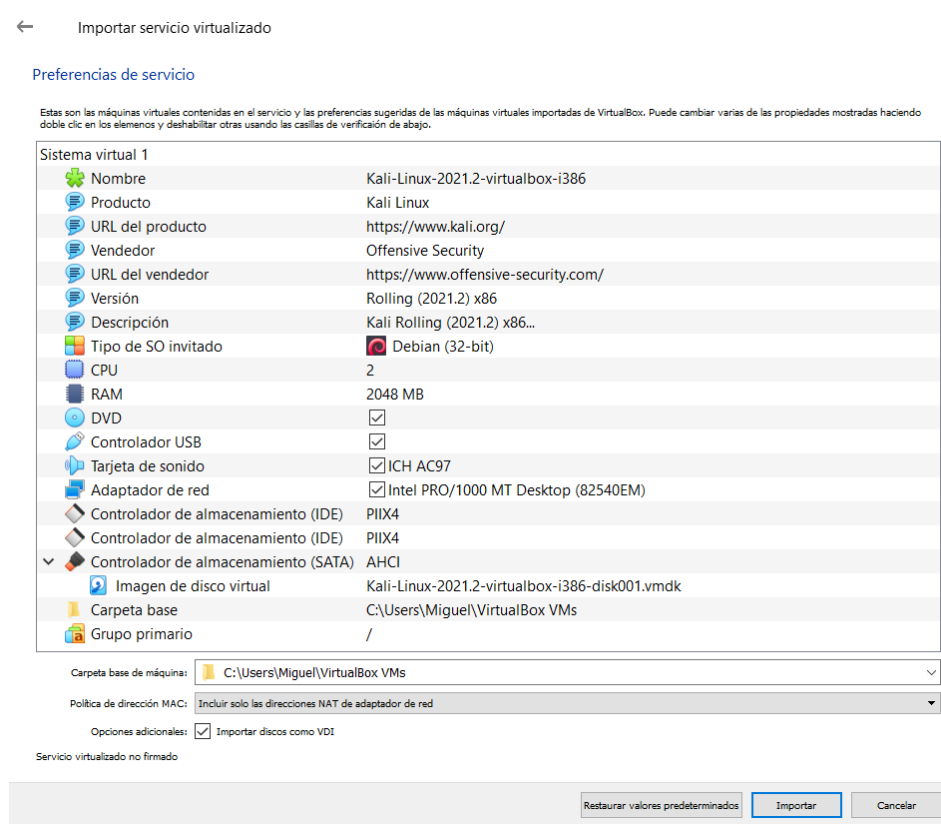


Figura 5. Revisión de la configuración del servicio virtualizado a importar en VirtualBox.

Tras pulsar el botón “Importar”, saltará un mensaje de acuerdo de licencia de software que, después de leerse en la URL indicada en el propio mensaje, debe aceptarse (con el botón “Acepto”); tal y como se muestra en la Figura 6. La aceptación de este acuerdo comenzará la importación de la MV, que puede tardar varios minutos (véase la Figura 7).

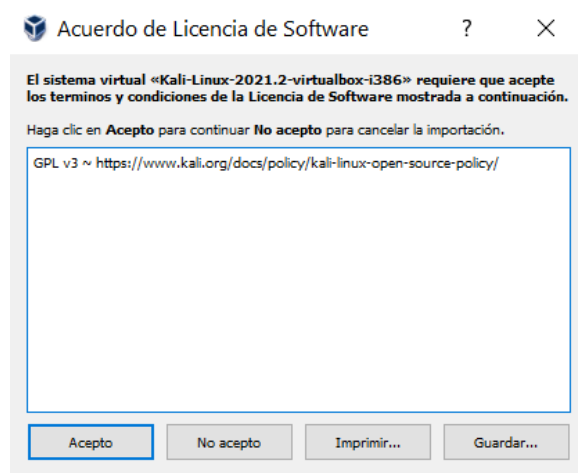


Figura 6. Acuerdo de licencia de software en la importación de la MV “Kali Linux” desde VirtualBox.

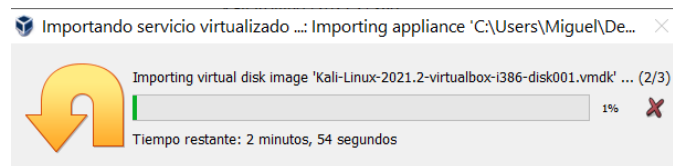


Figura 7. Importación del servicio virtualizado desde VirtualBox.

Al finalizar este proceso, se mostrará de nuevo la pantalla principal del software VirtualBox y podrá observarse en ella la MV que se acaba de importar. Para arrancarla, debe hacerse doble clic sobre ella (véase la Figura 8). Si se desea configurar algún parámetro tras la importación, bastará con hacer clic derecho sobre la máquina a configurar y seleccionar la opción “Configuración” (véase la Figura 9).

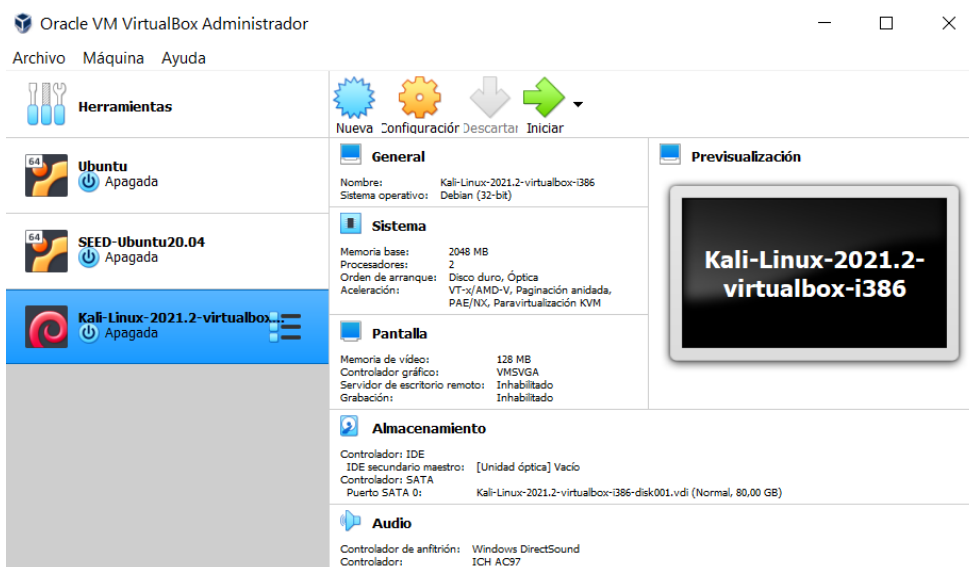


Figura 8. Pantalla principal de VirtualBox con la MV “Kali Linux” importada.

NOTA: Si al tratar de iniciar la máquina, VirtualBox muestra un mensaje de error como el que se observa en la Figura 10, es necesario dirigirse a la máquina virtual en cuestión en la pantalla principal de VirtualBox y hacer clic derecho sobre ella, seleccionando “Configuración” en el menú desplegable. Después, en el menú de la izquierda, seleccionar “USB” y cambiar la opción seleccionada “Controlador USB 2.0 (OHCI + EHCI)” por la opción “Controlador USB 1.1 (OHCI)”. Véase la Figura 11.

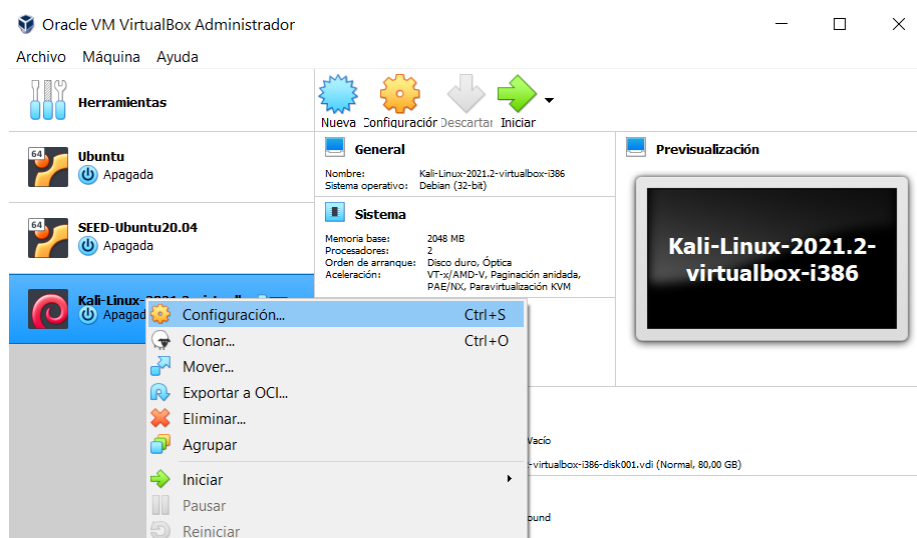


Figura 9. Configuración de una MV ya importada en VirtualBox.



Figura 10. Mensaje de error al iniciar la MV “Kali Linux” en VirtualBox.

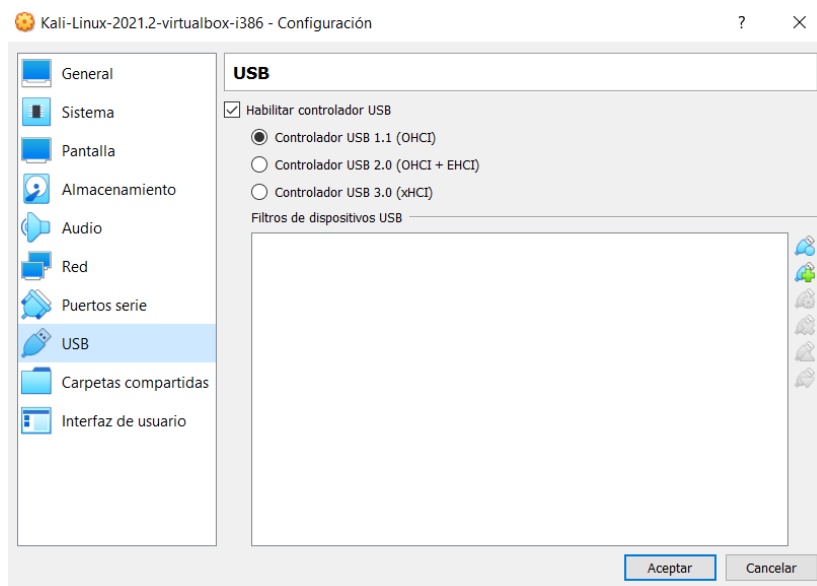


Figura 11. Configuración del controlador USB de una MV en VirtualBox.

Para apagar la máquina, podrá hacerse como con cualquier sistema operativo (“Inicio” → “Apagar”) o utilizando el menú “Archivo” → “Cerrar” (dentro de la pantalla de la propia máquina), tal y como puede observarse en la Figura 12.

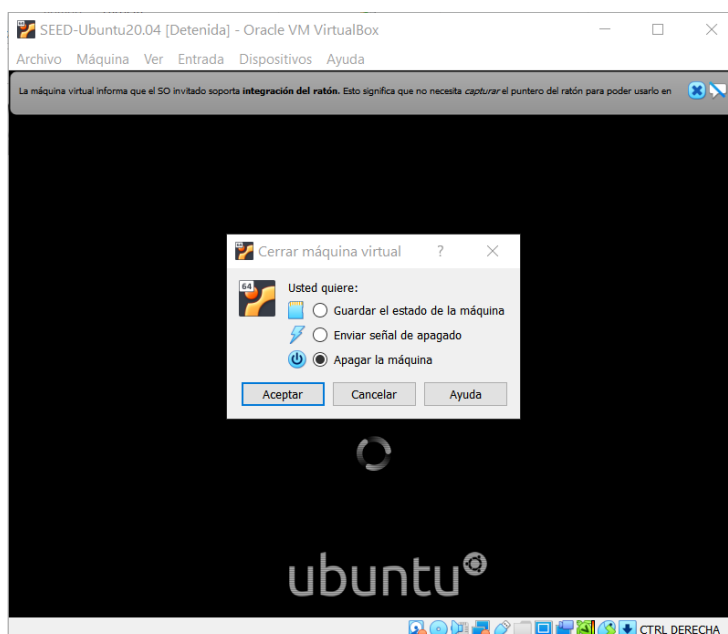


Figura 12. Apagado de una MV en VirtualBox.

Para iniciar sesión dentro de la MV de “Kali Linux” que acaba de importarse, deben utilizarse las credenciales predeterminadas de la misma. Estas son:

- Usuario: kali
- Contraseña: Kali

Tras iniciar sesión, para cambiar la distribución de teclado (por defecto está en inglés), debe hacerse clic sobre la terminal (véase la Figura 13), escribir “`sudo dpkg-reconfigure keyboard-configuration`” (sin las comillas) y pulsar Enter. Este comando nos abrirá el configurador del teclado, a través del cual debemos elegir el lenguaje de este y guardar los cambios.

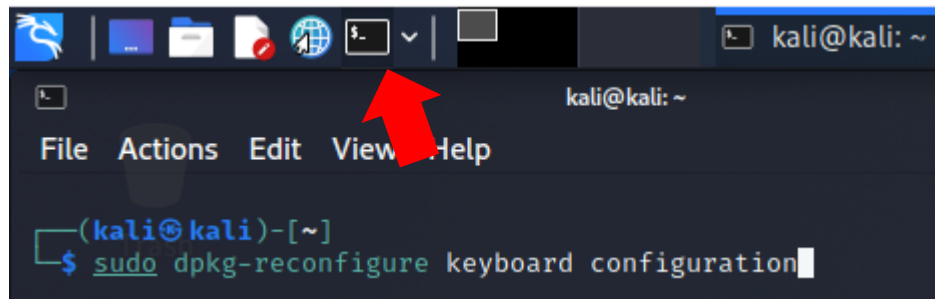


Figura 13. Cambio de distribución de teclado en Kali Linux.

NOTA: Si se utiliza Windows como host anfitrión, para que la pantalla de la MV escale automáticamente y pueda verse a pantalla completa, se recomienda la instalación de “VirtualBox Guest Additions”. Para ello:

- Tras ejecutar la MV, en el menú superior, debe seleccionarse la opción “Dispositivos” → “Insertar imagen de CD de las <<Guest Additions>>”.
- En la máquina, tras iniciar sesión, debe abrirse una terminal (véase la Figura 13), escribirse “`sudo bash /media/cdrom0/VBoxLinuxAdditions.run`” (sin las comillas) y pulsar Enter; esta acción solicitará la contraseña del usuario de la máquina y, tras introducirla, nos pedirá que aceptemos la instalación (debe escribirse “yes” y pulsar Enter).
- Reiniciar la máquina.

- Más información al respecto [aquí](#).

Tras la finalización de esta práctica, el alumno deberá ser capaz de manejarse en un sistema operativo Linux, así como saber utilizarlo mediante la terminal, conocer sus principales comandos, la utilidad de estos y saber aprovecharlos. Se recomienda el uso del comando “man” (comando que muestra el manual de otros comandos; su uso es `man comando` dentro de una terminal). El alumno puede apoyarse también en la hoja de comandos facilitada junto a esta guía (Practica2_CommandReference.pdf) y/o mediante búsquedas en Internet.

1. Linux: Árbol de directorios

En Linux, sin importar el número de discos duros y/o particiones que se tengan, los directorios y los ficheros se organizan en forma de árbol o árbol invertido (véase la Figura 14), siendo la raíz el directorio “/” (barra). Además, este SO, está pensado para poder ser dividido en partes más pequeñas, que, a su vez, pueden estar contenidas en su propio disco duro o partición. Las principales partes divisibles son el sistema de archivos raíz o “/”, /usr, /var y /home, teniendo cada una de ellas un propósito diferente (todos los comandos están en un mismo lugar, los archivos de datos en otro, la documentación en otro, etc.).

Tal y como ya se ha mencionado, en Linux, todos los archivos y dispositivos de almacenamiento se muestran en un único árbol de directorios, mediante una estructura jerárquica. La dirección o ruta de un fichero o directorio dentro de esta estructura, también llamada *path*, puede ser absoluta (cuando se indica la ruta completa; por ejemplo, “/home/luis/trabajo/informe.pdf”) o relativa (cuando se indica la ruta a partir del directorio en el que nos encontramos; por ejemplo, si estamos situados en /home, y queremos indicar la ruta del informe, podríamos especificar “./luis/trabajo/informe.pdf”). Cabe destacar que, en una ruta, el símbolo “.” Indica el directorio actual y “..” el directorio padre. Veamos algunos ejemplos:

- Observando la Figura 14, si nos situamos en el directorio /home/marcos y queremos indicar la ruta relativa de “direcciones.txt”, podemos decir que esta es “../luis/trabajo/direcciones.txt”:

../	luis/	trabajo/	direcciones.txt
Se vuelve al directorio “/home”.	Se avanza al directorio “luis” (/home/luis)	Se avanza al directorio “trabajo” (/home/luis/trabajo)	Se indica la ruta final del fichero “direcciones.txt” (/home/luis/trabajo/direcciones.txt)

- Observando la Figura 14, si nos situamos en el directorio /usr/bin y queremos indicar la ruta relativa de “/”, podemos decir que esta es “../..”:

../	..
Se vuelve al directorio “/usr”.	Se vuelve al directorio “/”.

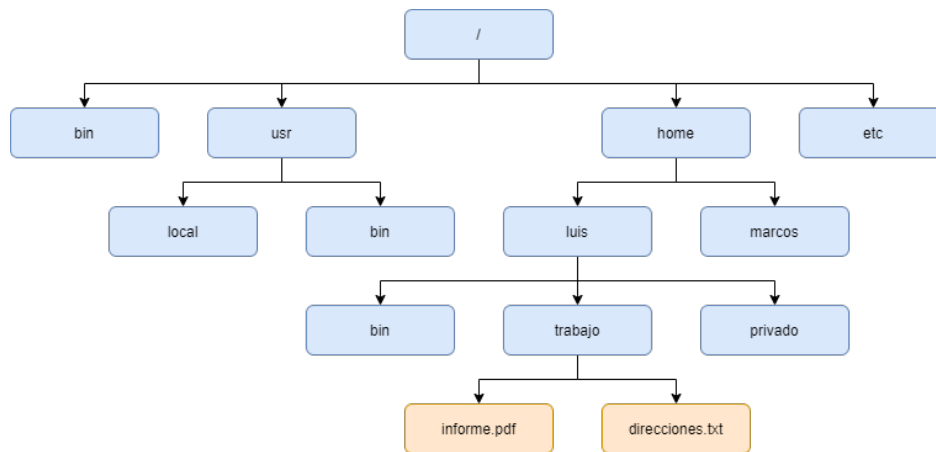


Figura 14. Ejemplo de estructura de árbol en los directorios de Linux.

A continuación, se detallan algunos de los directorios (d) y ficheros (f) más relevantes dentro de un SO Linux:

- **/bin.** Este directorio, contiene los archivos ejecutables que se corresponden con los diferentes comandos que todos los usuarios del sistema pueden utilizar.
- **/boot.** Este otro directorio es el encargado de almacenar todos los archivos que se necesitan para arrancar el sistema (incluido el propio kernel).
- **/dev.** Es el directorio encargado de almacenar los identificadores de los diferentes dispositivos. Entre ellos, se encuentran los de almacenamiento (por ejemplo, /dev/sda1, /dev/sda2, /dev/sda3, etc.), los de entrada/salida (por ejemplo, /dev/tty, /dev/tty0, /dev/tty1, etc.).
- **/home.** Se trata del directorio en el que pueden encontrarse los directorios personales de cada uno de los usuarios del sistema.
- **/root.** Al contrario que el resto de los usuarios, el usuario root tiene su directorio personal en /root (no en /home). Por lo tanto, es el directorio local para el usuario root.

- **/sbin.** Este directorio contiene archivos ejecutables tanto de las utilidades para el arranque del sistema, como para la solución de problemas, mantener el sistema de archivos, etc.
- **/usr.** Principalmente, es un directorio que incluye archivos de comandos (en modo de solo lectura) que son accesibles por todos los usuarios del sistema.
- **/var.** En este directorio se almacenan diferentes subdirectorios y archivos variables (de ahí su nombre) o cambiantes como, por ejemplo, los logs del sistema.
- **/lib.** Es el contenedor de las diferentes librerías que comparten los programas en el sistema de archivos raíz.
- **/proc.** En este directorio se almacena un sistema de archivos virtual que no existe físicamente en el disco duro, está creado únicamente en memoria y se utiliza para ofrecer información relacionada con el propio sistema.
- **/etc.** Este directorio contiene, principalmente, los archivos de configuración (tanto del sistema como de parte del software instalado en él). Entre estos archivos cabe destacar:
 - **/etc/passwd.** En este fichero puede encontrarse una pequeña base de datos de los usuarios del sistema con nombre de usuario, nombre real, contraseña (cifrada (o “hasheada”), etc.
 - **/etc/group.** Se trata de un fichero similar a /etc/passwd, pero con la información de los diferentes grupos de usuarios.
 - **/etc/fstab.** En este fichero están contenidos los diferentes sistemas de archivos que deben montarse en el arranque del sistema.
 - **/etc/shadow.** Se trata de un fichero que contiene contraseñas cifradas (en sistemas donde se encuentre instalado y/o activo el sistema de contraseñas ocultas). A diferencia de /etc/passwd, en este fichero también se almacena la fecha de caducidad de una contraseña, el tiempo mínimo entre cambios de contraseña, etc.
 - **/etc/security.** En este caso, se trata de un directorio que contiene ficheros en los que se establecen una serie de condiciones de seguridad por defecto para el equipo.

- **/etc/shells**. Es un fichero en el que se lista las diferentes rutas de los intérpretes de línea de comandos admitidos en nuestro sistema.

El comando “cd”, desde la terminal de Linux, se utiliza para poder movernos entre directorios. Por otro lado, el comando “ls”, también desde la terminal de Linux, muestra o lista los ficheros o subdirectorios que hay contenidos en un directorio.

Recuerda que, si necesitas ayuda con el funcionamiento o el uso de alguno de los comandos, puedes utilizar el comando “man”, seguido del nombre del comando del que tengas dudas sobre su uso (por ejemplo: `man cd`); esto te mostrará una pequeña explicación del comando y las diferentes opciones para utilizarlo. También puedes utilizar la hoja de comandos facilitada junto a esta guía (Practica2_CommandReference.pdf). Además, como irás aprendiendo a la vez que adquieras experiencia... ¡los buscadores son nuestros amigos!

Ejercicio 1.1

En una terminal dentro de la MV de Kali Linux dirígete hasta la ruta del sistema de archivos raíz (“/”) y lista los subdirectorios que allí están contenidos:

- `cd /`
- `ls`

¿Observas alguno de los directorios y/o archivos que se han mencionado en párrafos anteriores? ¿Cuáles?

Ejercicio 1.2

Utilizando el comando “`cd`”, dirígete hasta el directorio “`/tmp`” (puedes utilizar la ruta absoluta o la ruta relativa), ¿Qué contiene ese directorio?

2. Linux: Comandos básicos

Un intérprete de comandos es un programa informático que permite traducir órdenes provenientes de los usuarios. Estas órdenes se interpretan mediante un conjunto de instrucciones que se envían directamente al conjunto de herramientas que forman el sistema operativo. Es decir, un intérprete de comandos (también llamado *Shell*) es un programa que hace de interfaz de texto entre el usuario y el sistema operativo. La sintaxis típica en un intérprete de comandos es:

- (prompt^{*1}) comando [parámetro 1] ... [parámetro n]

*^{*1} El **prompt** es el carácter (o conjunto de caracteres) que, en una línea de comandos, se muestra al principio. Esto indica que la línea de comandos está a la espera de órdenes por parte del usuario. El carácter o caracteres puede ser diferente dependiendo del intérprete de comandos.*

Algunos comandos útiles en Linux son:

- **cat**. Muestra el contenido de un fichero.
- **cd**. Cambia de directorio. Si no lleva ningún argumento, cambia al directorio home.
- **cp**. Copia un fichero o directorio.
- **echo**. Imprime un mensaje pasado como parámetro.
- **find**. Busca ficheros y directorios.
- **ls**. Muestra el contenido de un directorio.
- **mv**. Renombra o mueve un fichero o directorio.
- **pwd**. Indica cual es el directorio actual.
- **mkdir**. Crea un nuevo directorio.

- **rm**. Borra un fichero o directorio.
- **chmod**. Proporciona permisos de escritura, lectura o ejecución a un fichero.

Recuerda que, si necesitas ayuda con el funcionamiento o el uso de alguno de los comandos, puedes utilizar el comando “man”, seguido del nombre del comando del que tengas dudas sobre su uso (por ejemplo: `man cd`); esto te mostrará una pequeña explicación del comando y las diferentes opciones para utilizarlo. También puedes utilizar la hoja de comandos facilitada junto a esta guía (Practica2_CommandReference.pdf). Además, como irás aprendiendo a la vez que adquieras experiencia... ¡los buscadores son nuestros amigos!

Ejercicio 2.1

Investiga que comando debes utilizar para crear un directorio y cómo debes hacerlo. Una vez sepas hacerlo, crea tres directorios en el directorio “home” del usuario “kali” de tu MV. Llámalos “ practica1”, “ practica2” y “ otros”. ¿Qué comando has utilizado? Para crearlos, ¿has usado una ruta absoluta o una ruta relativa?

Ejercicio 2.2

Mediante el comando “`df`” y apoyándote en el manual del comando (“`man df`”), ¿Qué comando debes utilizar para mostrar el espacio en disco con una presentación “más amigable”? ¿Y si quieres mostrar además el tipo de cada uno de los sistemas de ficheros?

Ejercicio 2.3

El comando “**ls**” muestra el contenido de un directorio. Gracias a alguno de sus parámetros, pueden mostrarse también archivos ocultos, ¿Cuántos archivos ocultos hay en el directorio /home/Kali? ¿Con qué comando los has listado?

Ejercicio 2.4

Dirígete al directorio “practica2” que has creado en ejercicios anteriores. Una vez en él, crea un fichero llamado “README.txt” y edítalo (por ejemplo, con “**nano**”, “**gedit**” o “**vim**”) para escribir en él el nombre de los integrantes del grupo. ¿Qué comando has utilizado para crear el fichero? ¿Y para editarlo?

Ejercicio 2.5

Muestra el contenido del fichero que acabas de crear en la propia consola, sin utilizar ningún editor de texto. ¿Qué comando te permite hacer esto?

Ejercicio 2.6

Apoyándote en el manual del comando “**cp**”, copia tres veces, en el mismo directorio, el fichero “**README.txt**”. El nombre de los ficheros resultantes será: “**info.txt**”, “**datos.txt**” y “**grupo.txt**”.

¿Cómo lo has hecho?

Ejercicio 2.7

Borra el fichero antiguo “**README.txt**” y quédate con el resto de los ficheros. ¿Qué comando te ha permitido borrar el fichero?

Ejercicio 2.8

Borra el directorio “**practica1**” que creaste antes. ¿Qué comando te ha permitido borrar el directorio? ¿En qué se diferencia con el comando que has usado para borrar el fichero?

3. Linux: Comprimir y descomprimir archivos

El objetivo principal de comprimir o empaquetar archivos es reducir su tamaño para que, entre otras, estos ocupen menos y así poderlo transferir, guardar, etc. De una forma más cómoda. En Linux, además de poder hacerse mediante software gráfico, existe la posibilidad de comprimir y descomprimir archivos desde la terminal. Los archivos comprimidos, cada uno con diferentes propiedades, pueden ser muy variados; entre estos destacan los ficheros “tar”, los “gz”, los “bz2”, los “tar.gz” y los ficheros “zip”.

- **tar.**

- Comprimir: `tar -cf archivo-comprimido.tar /directorio/a/comprimir`
 - -c indica que se debe crear un archivo.
 - -f indica que el siguiente argumento es el nombre del fichero final.
- Descomprimir: `tar -xf archivo.comprimido.tar /directorio/a/descomprimir`
 - -x indica que se debe descomprimir un archivo.
 - -f indica que el siguiente argumento es el nombre del fichero a descomprimir.

- **gz.**

- Comprimir: `gzip -9 archivo-comprimido.gz`
 - -9 indica el factor de compresión (9 es el mayor).
- Descomprimir: `gzip -d archivo-comprimido.gz`
 - -d indica que se debe descomprimir un archivo.

- **bz2.**

- Comprimir: `bzip2 archivo-comprimido.bz2`
- Descomprimir: `bzip2 -d archivo-comprimido.bz2`

- -d indica que se debe descomprimir un archivo.
- **tar.gz.**
 - Comprimir: `tar -czf archivo-comprimido.tar.gz /directorio/a/comprimir`
 - -c indica que se debe crear un archivo.
 - -z indica que debe utilizarse el compresor gzip.
 - -f indica que el siguiente argumento es el nombre del fichero final.
 - Descomprimir: `tar -xzf archivo.comprimido.tar.gz /directorio/a/descomprimir`
 - -x indica que se debe descomprimir un archivo.
 - -z indica que está comprimido con gzip.
 - -f indica que el siguiente argumento es el nombre del fichero a descomprimir.
- **zip.**
 - Comprimir: `zip archivo-comprimido.zip /directorio/a/comprimir`
 - Descomprimir: `unzip archivo.comprimido.zip /directorio/a/descomprimir`

Recuerda que, si necesitas ayuda con el funcionamiento o el uso de alguno de los comandos, puedes utilizar el comando “man”, seguido del nombre del comando del que tengas dudas sobre su uso (por ejemplo: `man cd`); esto te mostrará una pequeña explicación del comando y las diferentes opciones para utilizarlo. También puedes utilizar la hoja de comandos facilitada junto a esta guía (Practica2_CommandReference.pdf). Además, como irás aprendiendo a la vez que adquieras experiencia... ¡los buscadores son nuestros amigos!

Ejercicio 3.1

Comprime el directorio “practica2” que creaste en ejercicios anteriores y guárdalo en “/tmp”. El nombre del fichero comprimido debe ser “entrega.tar”. ¿Qué comando has utilizado?

Ejercicio 3.2

Dirígete al directorio “/tmp” y descomprime allí mismo el fichero que has creado en el ejercicio anterior (“entrega.tar”), ¿Con qué comando lo has conseguido hacer?

4. Linux: Administración

En todos los sistemas operativos de tipo UNIX, por defecto, existe un usuario básico llamado “root”. Este usuario, se corresponde con el administrador del sistema o superusuario. Su directorio de trabajo, tal y como se menciona en el apartado “Linux: Árbol de directorios”, es “/root”. Este usuario no tiene limitados sus privilegios (puede borrar, leer, escribir, instalar, etc. Cualquier cosa), por lo tanto, trabajar con este usuario supone un riesgo para la seguridad del sistema.

El resto de los usuarios (ya sean administradores o no), tienen un directorio del cuál son propietarios que está alojado en “/home”. Estos usuarios estarán identificados con un UID (*Unique identifier*), al que va asociado el nombre de usuario en minúsculas del propio usuario (por ejemplo, el usuario root tiene UID 0). Así mismo, todo usuario pertenece al menos a un grupo, que, en combinación, permiten determinar los permisos de acceso a ficheros.

Los tres tipos de permisos que un usuario o un grupo puede tener sobre los ficheros y directorios en Linux son:

- **r.** Permiso de lectura. También se representa con el número 4.
 - En los archivos, permite visualizar el contenido.
 - En directorios, permite saber que archivos y directorios contiene.
- **w.** Permiso de escritura. También se representa con el número 2.
 - En los archivos, permite modificar el contenido del archivo.
 - En directorios, permite crear archivos o subdirectorios dentro del directorio. Además, con este permiso se pueden borrar los directorios, copiar archivos en el directorio, mover, cambiar el nombre, etc.
- **x.** Permiso de ejecución. También se representa con el número 1.
 - En los archivos, permiten, valga la redundancia, ejecutarse (si fuera un programa).
 - En directorios, permite utilizar el nombre del directorio cuando se accede a archivos dentro del directorio (por ejemplo, para buscar con el comando find).

A su vez, estos permisos se asignan en tres categorías y solo el propietario (o el superusuario) puede cambiar estos permisos. Las categorías son:

- **u.** Permisos del propietario.
- **g.** Permisos del grupo.
- **o.** Permisos para otros.

Además, los directorios y los ficheros suelen estar marcados, al inicio de la cadena de caracteres que define sus permisos, con una “d” o un “-” respectivamente.

Si se observa la Figura 15, después del carácter que define el tipo del elemento de la lista (los archivos con “-” y los directorios con “d”), pueden encontrarse los permisos, agrupados de tres en tres (por categorías). En primer lugar, están los permisos del propietario, seguidos de las del grupo y, por último, los de otros. Por ejemplo, si se observa el directorio “midirectorio” de la Figura 15, se indica lo siguiente: “drwxr-xr-x”, donde:

- **d.** Indica que se trata de un directorio.
- **rwX.** Indica que el propietario tiene permisos de lectura, escritura y ejecución sobre el directorio.
- **r-x.** Indica que el grupo tiene permisos de lectura y ejecución sobre el directorio.
- **r-x.** Indica que “otros” tienen permisos de lectura y ejecución sobre el directorio.

También puede encontrarse (de nuevo en la Figura 15) la palabra “kali” o “root”. Esto se corresponde con el usuario y grupo respectivamente, que son los propietarios del archivo o directorio en sí.

```
total 12
drwxr-xr-x  3 kali kali 4096 Jul 23 08:01 .
drwxr-xr-x 17 kali kali 4096 Jul 23 05:47 ..
-rw-r--r--  1 kali kali   0 Jul 23 07:38 info.txt
drwxr-xr-x  2 kali kali 4096 Jul 23 08:01 midirectorio
-rwxr-xr-x  1 root root   0 Jul 23 07:38 test.txt
```

Figura 15. Muestra de los permisos de diferentes archivos en Linux.

Para el **manejo de grupos y usuarios** de un fichero/directorio, suele utilizarse el comando `chown`. Por otro lado, para el **manejo de permisos** de ficheros y/o directorios suele utilizarse el comando `chmod`. Este último comando, puede usarse de diferentes maneras. Algunas de ellas son:

- Por ejemplo, `chmod u=rw,go=r info.txt`, que otorga permisos de lectura (“r”) y escritura (“w”) para el propietario (“u”) y de lectura (“r”) para el grupo y otros (“g” y “o”) sobre el fichero “info.txt”.
- Otro ejemplo es `chmod o+x info.txt`. En este caso, se añade permiso de ejecución (“x”) para otros (“o”) en el fichero “info.txt”. Si se utilizara el símbolo “-” (`chmod o-x info.txt`), se eliminaría el permiso de ejecución para otros.
- Un tercer ejemplo es, `chmod 753 info.txt`, en este caso, cada uno de los dígitos se corresponde con uno de los grupos de permisos (el primero, 7, con el del propietario; el segundo, 5, con el del grupo; y el tercero, 3, con el de otros).

Si nos remontamos a la explicación de los permisos, el de lectura es un 4, el de escritura un 2 y el de ejecución un 1; para calcular el dígito correspondiente a cada grupo, simplemente hay que sumarlos. Por ejemplo, el 7 se correspondería con todos los permisos (4+2+1), el 5 con lectura + ejecución (4+1) y el 3 con escritura + ejecución (2+1).

De esta forma, en el caso del ejemplo, se le estarían otorgando permisos de lectura, escritura y ejecución al propietario; lectura y ejecución al grupo; y escritura y ejecución a otros.

A veces, por los permisos del usuario que utilizamos, no podemos realizar ciertas acciones o ejecutar ciertos comandos. En los sistemas operativos de tipo Unix, existe una palabra reservada (asociada a una utilidad) llamada “sudo” (*Super User Do*), que permite **ejecutar programas con los privilegios de seguridad de otros usuarios** (normalmente el usuario root). Por ejemplo, si quisiéramos crear con el usuario “kali”, dentro de la MV de la práctica, el fichero “/etc/mi-fichero.txt”, deberíamos hacerlo mediante `sudo touch /etc/mi-fichero.txt`, ya que el usuario “kali” por sí solo no tiene permisos para escribir en ese directorio.

Recuerda que, si necesitas ayuda con el funcionamiento o el uso de alguno de los comandos, puedes utilizar el comando “man”, seguido del nombre del comando del que tengas dudas sobre su uso (por ejemplo: `man cd`); esto te mostrará una pequeña explicación del comando y las diferentes opciones para utilizarlo. También puedes utilizar la hoja de comandos facilitada junto a esta guía (Practica2_CommandReference.pdf). Además, como irás aprendiendo a la vez que adquieras experiencia... ¡los buscadores son nuestros amigos!

Ejercicio 4.1

Cambia los permisos sobre el fichero “info.txt” que creaste en ejercicios anteriores para que tanto propietario, como grupo, como otros tengan permisos de lectura, escritura y ejecución. ¿Qué comando has utilizado?

Ejercicio 4.2

Utilizando una técnica diferente a la que has utilizado en el ejercicio anterior, cambia los permisos del fichero “datos.txt” para que tenga permisos de lectura y escritura para el propietario y de solo lectura para grupo y otros. ¿Cómo lo has hecho?

Ejercicio 4.3

Ayudándote de Internet... ¿Cuál es el UID más alto permitido en los sistemas Linux? Aparte del 0, que se reserva para el usuario root, ¿Hay algún otro UID que suele reservarse para usuarios especiales?

Ejercicio 4.4

Cambia el propietario y el grupo del fichero “grupo.txt” por “root” y “root” respectivamente. ¿Cómo has conseguido hacerlo?

Ejercicio 4.5

Elimina el permiso de ejecución al fichero “otros”. ¿Qué comando has utilizado?

Ejercicio 4.6

Utilizando el comando “ls” y con la ayuda de “man”, encuentra la combinación de parámetros necesaria para mostrar los permisos, el propietario y el grupo de todos los ficheros del directorio “/home/kali”. ¿Qué comando has utilizado?

5. Linux: Procesos

Por cada software, comando, aplicación, herramienta, etc. Que se ejecuta en un sistema Linux, se genera, al menos, un proceso. Este proceso suele corresponderse con una tarea, que trabaja de forma independiente, con su propia misión, sus permisos, etc. El tiempo que el proceso se mantenga activo, utilizará ciertos recursos del sistema (como CPU, RAM, etc.).

Cada uno de los procesos posee un PID (*Process ID* o número de identificación de proceso), un número de 5 dígitos único para cada proceso. Este PID no se liberará hasta que el proceso al que se ha destinado termine (muera).

Los procesos pueden ejecutarse en **primer plano o en *background***:

- De forma predeterminada, los procesos iniciados por un usuario se ejecutan en primer plano; se toma la entrada del símbolo del sistema y se muestra la salida en la pantalla del ordenador. Si se está ejecutando un proceso en primer plano, la terminal evita que se inicie un nuevo proceso hasta que el existente termine.
- Los procesos de fondo no requieren entrada de teclado y puede iniciarse otro proceso desde el terminal mientras que el anterior se ejecuta en segundo plano. Para iniciar un proceso en segundo plano, al escribir el comando, un usuario puede añadir un ampersand (&) al final (por ejemplo, `rm -fr /var/www/data &`). Para cambiar entre procesos en primer y segundo plano pueden utilizarse los comandos “fg” y “bg”.

Los procesos, a su vez, pueden ser de **diferentes tipos**:

- **Init.** Se trata del primer proceso que se crea cuando se inicia una máquina Linux o Unix. Todos los procesos de un sistema están asociados a un proceso padre, por lo tanto, todos los demás procesos del sistema son hijos del proceso “init”. Este proceso no se puede eliminar (únicamente cuando se apaga el sistema) y siempre se le asigna el PID número 1.
- **Proceso padre y proceso hijo.** Los procesos, están asociados a un proceso padre o proceso principal.

- **Proceso zombi y proceso huérfano.** Cuando completan su ejecución, los procesos hijo se terminan o mueren, actualizándose entonces el proceso padre para continuar la tarea que se le asignó. Sin embargo, en ocasiones, si el proceso padre muere antes que el proceso hijo, los procesos hijo se convierten en huérfanos, y el proceso padre de todos los procesos "init" se convierte en su nuevo PID. Por otro lado, un proceso que se mata, pero aún muestra su entrada en el estado de los procesos o en la tabla de procesos se denomina proceso zombi (están muertos y no se utilizan).
- **Proceso Daemon.** Se trata de procesos en segundo plano, relacionados con el sistema, que a menudo se ejecutan con permisos de root y solicitudes de servicios de otros procesos. La mayoría de las veces se ejecutan en segundo plano.

Recuerda que, si necesitas ayuda con el funcionamiento o el uso de alguno de los comandos, puedes utilizar el comando "man", seguido del nombre del comando del que tengas dudas sobre su uso (por ejemplo: `man cd`); esto te mostrará una pequeña explicación del comando y las diferentes opciones para utilizarlo. También puedes utilizar la hoja de comandos facilitada junto a esta guía (Practica2_CommandReference.pdf). Además, como irás aprendiendo a la vez que adquieras experiencia... ¡los buscadores son nuestros amigos!

Ejercicio 5.1

¿Qué parámetros muestra el comando "`ps -ef`"? ¿Qué significa la "e" en el comando? ¿Y la "f"?

Ejercicio 5.2

El comando “**sleep**” temporiza un intervalo de tiempo en Linux. Ejecuta el comando para que haga un “**sleep**” de 20 minutos y envía el proceso a *background*. ¿Qué comando has utilizado?

Ejercicio 5.3

¿Cuál es el PID del proceso que se creó en el anterior ejercicio? ¿Y el PID de su proceso padre?
¿En qué comando te has apoyado para averiguarlos?

Ejercicio 5.4

Mata el proceso que se creó al ejecutar el comando “**sleep**” de 20 minutos, ¿Qué comando has utilizado?

Ejercicio 5.5

Lanza un comando para que se haga un “*sleep*” de 5 minutos y envíalo a *background*. Después, localiza su PID y trae el proceso de nuevo a primer plano. ¿Con qué comando has conseguido recuperar el proceso en primer plano?

Ejercicio 5.6

Ayudándote de Internet... ¿Existe la posibilidad de que un proceso se configura para lanzarse al iniciar el sistema operativo? ¿Cómo?

Ejercicio 5.7

Investiga qué comando te muestra, en tiempo real, las estadísticas y el estado de todos los procesos en ejecución. ¿Qué comando es? ¿Qué información muestra?

Ejercicio 5.8

El “pipe” (`|`), en Linux, permite encadenar la ejecución de diferentes comandos, pasando el *output* de uno como el *input* de otro. Esta utilidad, suele ayudar mucho en las búsquedas sobre los resultados de la ejecución de un comando. Teniendo en cuenta que el comando “`grep`” sirve para hacer búsquedas, ¿con qué comando y qué parámetros buscarías todos los procesos que contengan la palabra “net”?

6. Linux: Instalación de programas, scripts y actualizaciones

Instalar programas y aplicaciones en Linux suele ser realmente sencillo, al igual que lo es actualizar el propio sistema operativo. Para ello, existen gran cantidad de gestores de paquetes, se puede instalar software a partir del código fuente, mediante paquetes “.deb”, ejecutar scripts, etc.

- En primer lugar, cabe mencionar los **gestores de paquetes**. Como su propio nombre indica, ayudan a la gestión de paquetes (conjuntos de ficheros, normalmente dependientes de la distribución de Linux para la que han sido creados, utilizados para comprimir aplicaciones en distintos formatos y/o medios de instalación). Los gestores de paquetes permiten mantener un registro del software que está instalado en un ordenador y, además, facilitan la instalación, actualización, desinstalación, etc. De nuevo software. Entre los gestores de paquetes para Linux más destacados se encuentran apt (*Advanced Packaging Tool*; el gestor de paquetes por defecto en GNU/Linux), pacman, yum, entropy, rpm, upkg, Zypp, etc.

Cada gestor de paquetes se utiliza de una forma diferente. Por ejemplo, para usar apt se sigue la siguiente combinación:

- (prompt) apt-get [configuración 1] ... [configuración n] opción

Si por ejemplo se quisiera re-sincronizar el índice de paquetes de apt, se utilizaría el comando “`apt-get update`”, para actualizar todos los paquetes podría hacerse mediante “`apt-get upgrade`”, si hubiera que instalar un paquete concreto bastaría con ejecutar “`apt-get install nombre-paquete`”, etc.

- La instalación de programas desde **código fuente**, en Linux, es algo muy habitual. Al ser un sistema operativo Open Source, muchos de los desarrolladores de software para este SO cuelgan el código directamente en GitHub u otras plataformas similares, y es el propio usuario el que debe compilar e instalar el software (o, al menos, tiene la posibilidad de hacerlo); de esta forma, entre otras, se puede modificar el código a su antojo o revisarlo antes de compilarlo e instalarlo. Para esta acción, una vez descargado el código fuente, descomprimido y situados desde la terminal en el directorio en el que se encuentra, se utilizan principalmente tres comandos:

- `./configure`

- Es importante tener en cuenta que el fichero “configure” debe tener permisos de ejecución.
- Con este comando se configura el modo de compilación.
- `make`
 - Se construye el programa.
- `sudo make install`
 - Se instala el programa.
- Los paquetes **.deb** son paquetes listos para ser ejecutados e instalados (como ocurre con los .exe en Windows). Si se ejecutan en un entorno gráfico, bastará con hacer doble clic sobre el fichero .deb a instalar, lo cual lanzará la aplicación que se encargará de instalarlo de una forma sencilla. Mediante una terminal, gracias al comando “dpkg”, también pueden instalarse (utilizando “`sudo dpkg -i fichero.deb`”) o desinstalarse (con “`sudo dpkg -r fichero.deb`”) estos paquetes.
- En Linux se pueden encontrar diferentes **scripts ejecutables**. Algunos de estos scripts (los más comunes), tienen extensiones como .sh o .py. Para instalar y/o ejecutar estos scripts, es necesario dirigirse al directorio en el que se encuentra el script en sí (utilizando el comando “cd”). Para que pueda ejecutarse, es imprescindible que el fichero tenga permisos de ejecución (véase el apartado “Linux: Administración” de esta misma guía) y, desde la terminal, utilizar la siguiente combinación:
 - (prompt) intérprete nombre_script [parámetro 1] ... [parámetro n]

Donde “intérprete” (del lenguaje de programación al que corresponda el script), será el encargado de ejecutar “nombre_script”, que, a su vez, recibirá tantos parámetros como sean necesarios. Así, por ejemplo, si se quisiera ejecutar un script escrito en lenguaje bash, podría utilizarse el siguiente comando: “`sh nombre_script.sh`”. De la misma forma, si se fuera a ejecutar un script escrito en Python, bastaría con utilizar “`python nombre_script.py`”.

Otra forma de ejecutar scripts es mediante:

- `./nombre_script [parámetro 1] ... [parámetro n]`

En este caso, para que el script pueda ejecutarse, es imprescindible que, en su primera línea de código, se indique la *Shell* que va a interpretar el fichero. Por ejemplo, si se trata de bash deberá contener “#!/bin/bash” o si se está tratando de crear un script escrito en lenguaje Python, “#!/usr/bin/env python”.

Recuerda que, si necesitas ayuda con el funcionamiento o el uso de alguno de los comandos, puedes utilizar el comando “man”, seguido del nombre del comando del que tengas dudas sobre su uso (por ejemplo: `man cd`); esto te mostrará una pequeña explicación del comando y las diferentes opciones para utilizarlo. También puedes utilizar la hoja de comandos facilitada junto a esta guía (Practica2_CommandReference.pdf). Además, como irás aprendiendo a la vez que adquieras experiencia... ¡los buscadores son nuestros amigos!

Ejercicio 6.1

Utilizando el gestor de paquetes de Linux (“apt-get”), instala el paquete “chromium”. ¿Qué comando has utilizado?

Ejercicio 6.2

De nuevo utilizando el gestor de paquetes de Linux (“apt-get”), desinstala (borrando también todos los ficheros de configuración) el paquete “chromium”. Apóyate en el comando “man” para saber todas las opciones de “apt-get”. ¿Qué comando has utilizado para la desinstalación?

Ejercicio 6.3

Crea (y ejecuta) un pequeño script en bash que actualice el índice de paquetes de apt, actualice todos los paquetes a su última versión y elimine los paquetes que se han instalado automáticamente para satisfacer las dependencias de otros paquetes, pero que ya no son necesarios. Por último, el script debe mostrar un mensaje por pantalla indicando que ya se ha terminado de ejecutar. Pega el código de tu script:

Ejercicio 6.4

Siguiendo las instrucciones del creador ^{*1} (puedes encontrarlas en el [README.md del proyecto en GitHub](#)), descarga, compila e instala NotepadQQ en tu máquina virtual. ¿Has encontrado algún problema en la instalación? ¿Cuál?

^{*1} Antes de proceder con la descarga/compilación/instalación de NotepadQQ en tu máquina virtual, debes ejecutar el siguiente comando, que instalará todos los paquetes necesarios: `sudo apt-get install qtbase5-dev qtchooser qt5-qmake qtbase5-dev-tools qtwebengine5-dev qtools5-dev-tools libqt5websockets5-dev libqt5svg5 libqt5svg5-dev libuchardet-dev pkg-config`

7. Linux: Logs del sistema

Los logs o registros del sistema son ficheros de texto en los que se registran, de forma cronológica, la mayoría de las actividades, sucesos e incidencias relevantes que ocurren en un Sistema Operativo. Estos registros son muy útiles para conocer qué ocurre en un ordenador y, por supuesto, poder solucionar problemas y evitar que estos se repitan en el futuro. Entre estos registros se pueden encontrar cosas como:

- Los paquetes que se han instalado o desinstalado.
- Información sobre los diferentes accesos remotos.
- Autenticaciones fallidas.
- Errores de programas o servicios.
- Bloqueos del firewall.
- Etc.

Entre los logs, que en Linux suelen estar almacenados en el directorio “/var/log”, pueden encontrarse los **logs del sistema** (registran información relacionada con el SO, como, por ejemplo, de los servicios, accesos, etc.) y los **logs de programas** (almacenan información y eventos relevantes del software o programa que los crea). Entre los logs del sistema cabe destacar “auth.log” (en él se encuentran todas las acciones que involucran la autenticación), “syslog” (es el cajón de sastre donde se registran gran cantidad de logs, tanto del sistema como de algunos programas), “faillog” (contiene intentos fallidos de autenticación de los usuarios en el sistema, “lastlog” (en él se registran la fecha/hora del último inicio de sesión en el sistema de cada usuario), “boot.log” (registra información relacionada con el arranque del sistema), “daemon.log” (en él pueden encontrarse información relacionada con los diferentes procesos *Daemon*), etcétera.

Recuerda que, si necesitas ayuda con el funcionamiento o el uso de alguno de los comandos, puedes utilizar el comando “man”, seguido del nombre del comando del que tengas dudas sobre su uso (por ejemplo: `man cd`); esto te mostrará una pequeña explicación del comando y las diferentes opciones

para utilizarlo. También puedes utilizar la hoja de comandos facilitada junto a esta guía (Practica2_CommandReference.pdf). Además, como irás aprendiendo a la vez que adquieras experiencia... ¡los buscadores son nuestros amigos!

Ejercicio 7.1

¿En qué fichero podrías encontrar los inicios de sesión que se han producido en la máquina?
¿Y los que se han hecho en remoto con SSH?

Ejercicio 7.2

Apoyándote en los comandos “tail” y “head”, ¿Cuál es la última línea que se ha registrado en el fichero de log “user.log”? ¿Y la primera?

Ejercicio 7.3

Dirígete al directorio “/var/log” y lista su contenido. Ayudándote de Internet... ¿Qué significan los números que aparecen al final de los ficheros de log (por ejemplo, “debug.1”, “auth.log.1”, etc.)?

Ejercicio 7.4

¿Qué comando podrías utilizar para ver las primeras 15 líneas del fichero /var/log/syslog? ¿Y las últimas 20 líneas?

8. Control de acceso y contraseñas

Cabe destacar que, a pesar del falso mito que existe de que Linux o MacOS son SO más seguros que Windows, esto para nada es así. Si bien es cierto que los usuarios de Linux suelen estar más concienciados con la seguridad y que existe un menor número de equipos con este SO (lo cual lo convierte en un objetivo minoritario para los delincuentes), no se debe descuidar la seguridad cuando utilizamos sistemas operativos de este tipo.

Uno de los principales aspectos de seguridad, además de mantener el SO actualizado, es la **contraseña**. Esta debe ser robusta (difícil de descubrir para un programa y/o una persona). Para que una contraseña pueda considerarse robusta, esta debe tener, al menos, una longitud mínima de 8 caracteres; no contener nombres, apellidos, fechas de cumpleaños, nombres de usuario, empresa, etc.; no utilizar palabras del diccionario de ningún idioma; no estar formada con números y/o letras adyacentes en el teclado; contener mayúsculas, minúsculas, números y caracteres especiales; etc.

En Linux, para cambiar la contraseña de acceso para el usuario “root”, desde la terminal, deben seguirse los siguientes pasos:

- Teclear `sudo su`.
- Introducir la contraseña actual.
- Teclear `passwd root` y pulsar enter.
- Escribir la nueva clave y pulsar enter.

De igual forma, para cambiar la contraseña de cualquier otro usuario, por ejemplo, el usuario “kali”, se seguirán los siguientes pasos:

- Teclear `su kali` (de esta forma, se cambiará de usuario al usuario “kali”).
- Introducir la contraseña actual.
- Teclear `passwd kali` y pulsar enter.
- Escribir la nueva clave y pulsar enter.

La herramienta **John the Ripper** sirve para recuperar y auditar la contraseña. Este software es de código abierto y se encuentra disponible para muchos sistemas operativos, entre ellos, Linux. En la distribución “Kali Linux” se encuentra instalada por defecto. Esta herramienta, admite la recuperación y auditoría de cientos de tipos de cifrado y *hash* (entre las que pueden encontrarse contraseñas de usuario, aplicaciones web como WordPress, software colaborativo, bases de datos, claves privadas cifradas, archivos, etc.).

Para comprobar que John the Ripper está correctamente instalado en el equipo y que es capaz de realizar un test de rendimiento del hardware del sistema que disponemos (para saber la capacidad que tendrá la herramienta en su labor de descifrar una contraseña), puede utilizarse el comando `john --test`. Si no estuviera instalada, basta con ejecutar el comando `sudo apt install john` o instalarla desde su código fuente siguiendo los siguientes pasos:

- `sudo apt update && sudo apt full-upgrade -y`
- `reboot`
- `sudo apt install build-essential libssl-dev yasm libgmp-dev libpcap-dev libnss3-dev libkrb5-dev pkg-config`
- `cd /home/kali`
- `wget https://github.com/openwall/john/archive/bleeding-jumbo.zip`
- `unzip bleeding-jumbo.zip`
- `rm bleeding-jumbo.zip`
- `cd john-bleeding-jumbo/src/`
- `./configure && make`
- `cd ../run`
- `./john --test`

Recuerda que, si necesitas ayuda con el funcionamiento o el uso de alguno de los comandos, puedes utilizar el comando “man”, seguido del nombre del comando del que tengas dudas sobre su uso (por ejemplo: `man cd`); esto te mostrará una pequeña explicación del comando y las diferentes opciones para utilizarlo. También puedes utilizar la hoja de comandos facilitada junto a esta guía (Practica2_CommandReference.pdf). Además, como irás aprendiendo a la vez que adquieras experiencia... ¡los buscadores son nuestros amigos!

Ejercicio 8.1

Con la ayuda de Internet, crea un usuario en tu MV llamado “user1” y ponle la contraseña “user1”. ¿Qué comando o comandos has utilizado?

Ejercicio 8.2

Cambia la contraseña del usuario creado en el anterior ejercicio por una más robusta, ¿Con qué comando has modificado la contraseña?

Ejercicio 8.3

Busca el comando adecuado y borra el usuario “user1”, ¿Cómo lo has hecho?

Ejercicio 8.4

Ahora, supongamos que ha llegado a nuestras manos el fichero “/etc/passwd” y el fichero “/etc/shadow” de una máquina y vamos a analizarlo (Practica2_Material.zip).

- a) En primer lugar, debemos crear una carpeta de trabajo dentro de nuestra MV, en “/home/kali”, a la que llamaremos “crack”. Después, accederemos a ella desde la terminal. En esta carpeta copiaremos los ficheros “shadow” y “passwd” adjuntos a esta guía (Practica2_Material.zip).

- b) El siguiente paso será dirigirnos al fichero “passwd” que ha llegado a nuestras manos e investigar su contenido. En este fichero, se puede encontrar información sobre quién puede acceder al sistema y qué puede hacer dentro de él. ¿Puedes encontrar algún usuario “llamativo”? Copia en el siguiente recuadro la línea correspondiente a ese o esos usuarios:

Una línea de ejemplo del fichero “/etc/passwd” es la siguiente, donde:

1	2	3	4	5	6	7
<code>user1:</code>	<code>x:</code>	<code>1001:</code>	<code>1001:</code>	<code>:</code>	<code>/home/user1:</code>	<code>/bin/sh</code>
Nombre de usuario.	Contraseña (el carácter “x” indica que la contraseña cifrada de almacena en /etc/shadow).	Identificador del usuario.	Identificador del grupo al que pertenece el usuario.	Información adicional sobre los usuarios.	Directorio del usuario.	Shell del usuario.

- c) Dirígete ahora al fichero “shadow” y localiza el nombre o nombres de los usuarios que te han parecido más llamativos en el ejercicio anterior. Copia en el siguiente recuadro la línea o líneas correspondientes a dicho usuario junto a su contraseña cifrada/hasheada:

Una línea de ejemplo del fichero “/etc/shadow” es la siguiente, donde:

1	2	3	4	5	6	7	8	9	10
<code>user1:</code>	<code>\$6</code>	<code>\$4w</code> <code>gbT</code> <code>gh2</code>	<code>\$vrupCDR</code> <code>PVxAsmV</code> <code>07ptc1h5E</code> <code>h7.Pcx7o</code> <code>RvBZ/OqJ</code> <code>QiG69I9jO</code> <code>ISutMG1g</code> <code>uIJy7RrA1</code> <code>iOjZJy37E</code> <code>.d.c1HYwc</code> <code>ap1:</code>	<code>17820:</code>	<code>0:</code>	<code>99999:</code>	<code>7:</code>	<code>:</code>	<code>:</code>
Nombre de usuario.	Código asociado al algoritmo de cifrado/hash.	Salt ^{*2} .	Contraseña cifrada/hasheada.	Fecha del último cambio de contraseña (desde el 01/01/1970).	Mínimo nº de días requerido entre cambios de contraseña.	Máximo nº de días requerido entre cambios de contraseña.	Nº de días antes que se avisará cuando expire la contraseña.	Cantidad de días después de que el usuario esté inactivo para deshabilitar la contraseña.	Caducidad de la cuenta (desde el 01/01/1970).

^{*2} El salt es una combinación de bits aleatorios que se usan, junto a la propia contraseña, como las entradas en una función de derivación de claves.

- d) ¿Qué algoritmo de cifrado (elemento nº 2) se ha utilizado para cifrar la contraseña del usuario “seed”? ¿Y para el usuario “test”?

- e) Con la ayuda de Internet, ¿Qué opciones de algoritmos de cifrado existen y a qué códigos se corresponden?

Ejercicio 8.5

Para utilizar John the Ripper, al igual que ocurre con otras herramientas, es necesario introducir los valores necesarios en el formato que la herramienta espera recibirlos.

- Ve al directorio “/home/kali/crack”, donde deberían estar contenidos los ficheros “passwd” y “shadow”.
- Mediante la herramienta “unshadow” (preinstalada en “Kali Linux”), combina estos dos ficheros en el formato que espera recibirlos John the Ripper (`sudo unshadow passwd shadow > passwords.txt`).

¿Qué contiene ahora el fichero “passwords.txt”?

Ejercicio 8.6

Una de las opciones de John the Ripper es tratar de recuperar una contraseña utilizando un diccionario de contraseñas. Puedes encontrar uno dentro del SO “Kali Linux”, en la ruta “/usr/share/john/password.lst”, aunque, si buscas un poco, en Internet podrás encontrar infinidad de ellos (en diferentes idiomas, con distintas combinaciones, etc).

Mediante esta opción, se pueden comparar las contraseñas contenidas en un fichero (cuyo formato se corresponde con la combinación del fichero “/etc/passwd” y “/etc/shadow” (gracias

a la herramienta “unshadow”), frente a las generadas por el software John the Ripper utilizando el diccionario que se le pasa como argumento. Es decir, para cada elemento en el diccionario, se genera un password cifrado/hasheado que se compara con el password contenido en el fichero generado por la herramienta “unshadow”. Si existe una coincidencia, se sabe que el password del diccionario que se ha utilizado para generar el password en el fichero resultante de ejecutar “unshadow”, es el correspondiente a ese usuario.

- a) Ayudándote de la opción `--help` (`john --help`), ¿Con qué opción podríamos indicarle a John the Ripper que lea las diferentes palabras que están contenidas en el diccionario `“/usr/share/john/password.lst”`?

- b) Ejecuta el comando `sudo john wordlist=/usr/share/john/password.lst /home/kali/crack/passwords.txt`, ¿Se ha conseguido descifrar alguna contraseña? ¿Cuál? Puedes observar las contraseñas descifradas con la opción `--show` de John the Ripper, seguida del fichero que contenía dichas contraseñas.

- c) Sobre el mismo fichero, explora la opción de utilizar John the Ripper mediante fuerza bruta, sin usar un diccionario (probando las diferentes combinaciones de caracteres). ¿Cuál es el comando que has utilizado? ¿Has conseguido extraer alguna otra contraseña?

5. Anexo I

En este apartado se incluyen los códigos, scripts y útiles necesarios para la realización de los ejercicios de esta práctica.

passwd

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:100:102:systemd Network Management,,,:/run/systemd:/usr/sbin/nologin
systemd-resolve:x:101:103:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin
systemd-timesync:x:102:104:systemd Time Synchronization,,,:/run/systemd:/usr/sbin/nologin
messagebus:x:103:106:/:/nonexistent:/usr/sbin/nologin
syslog:x:104:110:/:/home/syslog:/usr/sbin/nologin
_apt:x:105:65534:/:/nonexistent:/usr/sbin/nologin
tss:x:106:111:TPM software stack,,,:/var/lib/tpm:/bin/false
uidd:x:107:114:/:/run/uidd:/usr/sbin/nologin
tcpdump:x:108:115:/:/nonexistent:/usr/sbin/nologin
avahi-autoipd:x:109:116:Avahi autoip daemon,,,:/var/lib/avahi-autoipd:/usr/sbin/nologin
usbmux:x:110:46:usbmux daemon,,,:/var/lib/usbmux:/usr/sbin/nologin
rtkit:x:111:117:RealtimeKit,,,:/proc:/usr/sbin/nologin
dnsmasq:x:112:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin
cups-pk-helper:x:113:120:user for cups-pk-helper service,,,:/home/cups-pk-helper:/usr/sbin/nologin
speech-dispatcher:x:114:29:Speech Dispatcher,,,:/run/speech-dispatcher:/bin/false
avahi:x:115:121:Avahi mDNS daemon,,,:/var/run/avahi-daemon:/usr/sbin/nologin
kernoops:x:116:65534:Kernel Oops Tracking Daemon,,,:/usr/sbin/nologin
saned:x:117:123:/:/var/lib/saned:/usr/sbin/nologin
nm-openvpn:x:118:124:NetworkManager OpenVPN,,,:/var/lib/openvpn/chroot:/usr/sbin/nologin
hplip:x:119:7:HPLIP system user,,,:/run/hplip:/bin/false
whoopsie:x:120:125:/:/nonexistent:/bin/false
colord:x:121:126:colord colour management daemon,,,:/var/lib/colord:/usr/sbin/nologin
geoclue:x:122:127:/:/var/lib/geoclue:/usr/sbin/nologin
pulse:x:123:128:PulseAudio daemon,,,:/var/run/pulse:/usr/sbin/nologin
gnome-initial-setup:x:124:65534:/:/run/gnome-initial-setup:/bin/false
gdm:x:125:130:Gnome Display Manager:/var/lib/gdm3:/bin/false
seed:x:1000:1000:SEED,,,:/home/seed:/bin/bash
```

```

systemd-coredump:x:999:999:systemd Core Dumper:/:usr/sbin/nologin
telnetd:x:126:134::/nonexistent:usr/sbin/nologin
ftp:x:127:135:ftp daemon,,,:srv/ftp:usr/sbin/nologin
sshd:x:128:65534:./run/sshd:usr/sbin/nologin
vboxadd:x:998:1:./var/run/vboxadd:/bin/false
test:x:1002:1002:.,,:/home/test:/bin/bash

```

shadow

```

root:!:18590:0:99999:7:::
daemon*:18474:0:99999:7:::
bin*:18474:0:99999:7:::
sys*:18474:0:99999:7:::
sync*:18474:0:99999:7:::
games*:18474:0:99999:7:::
man*:18474:0:99999:7:::
lp*:18474:0:99999:7:::
mail*:18474:0:99999:7:::
news*:18474:0:99999:7:::
uucp*:18474:0:99999:7:::
proxy*:18474:0:99999:7:::
www-data*:18474:0:99999:7:::
backup*:18474:0:99999:7:::
list*:18474:0:99999:7:::
irc*:18474:0:99999:7:::
gnats*:18474:0:99999:7:::
nobody*:18474:0:99999:7:::
systemd-network*:18474:0:99999:7:::
systemd-resolve*:18474:0:99999:7:::
systemd-timesync*:18474:0:99999:7:::
messagebus*:18474:0:99999:7:::
syslog*:18474:0:99999:7:::
_apt*:18474:0:99999:7:::
tss*:18474:0:99999:7:::
uuid*:18474:0:99999:7:::
tcpdump*:18474:0:99999:7:::
avahi-autoipd*:18474:0:99999:7:::
usbmux*:18474:0:99999:7:::
rtkit*:18474:0:99999:7:::
dnsmasq*:18474:0:99999:7:::
cups-pk-helper*:18474:0:99999:7:::
speech-dispatcher:!:18474:0:99999:7:::
avahi*:18474:0:99999:7:::
kernoops*:18474:0:99999:7:::
saned*:18474:0:99999:7:::
nm-openvpn*:18474:0:99999:7:::
hplip*:18474:0:99999:7:::
whoopsie*:18474:0:99999:7:::
colord*:18474:0:99999:7:::
geoclue*:18474:0:99999:7:::
pulse*:18474:0:99999:7:::
gnome-initial-setup*:18474:0:99999:7:::
gdm*:18474:0:99999:7:::
seed:$6$n8DimvsbIglU00xbD$YZ0h1EAS4bGKeUIMQvRhYFvkrMmQZdr/hB.0fe3KFZQTgFTcRgoIoKZd00rhDRxxaITL4b/scpdbTfk/nwFd
0:18590:0:99999:7:::
systemd-coredump:!:18590:0:99999:7:::
telnetd*:18590:0:99999:7:::

```



```
ftp:*.18590:0:99999:7:::  
sshd:*.18590:0:99999:7:::  
vboxadd!:18786:::  
test:$6$IMtT.qDUmzrXfde6$dbHyvoT9GCVJEiST4AwSIHdoLsPMtbiPdX6j.spkDoAF8rXQhp05q1f5uFVmTdsQiGtvag2VkKn9DTVHs4jMu  
1:18834:0:99999:7:::
```