Ciberseguridad IC Noche

Operaciones Ciberseguras

Axel Lopez 118630567

Exposición 05/03/2025

Módulo 8  
Descripción General de Linux

Índice

[Introducción 3](#__RefHeading___Toc1_2034211389)

[8.1 Nociones Básicas de Linux 4](#__RefHeading___Toc3_2034211389)

[8.1.1 ¿Qué es Linux? 4](#__RefHeading___Toc130_2034211389)

[8.1.2 El Valor de Linux 5](#__RefHeading___Toc132_2034211389)

[8.1.3 Linux en el SOC 6](#__RefHeading___Toc134_2034211389)

[8.1.4 Herramientas de Linux 7](#__RefHeading___Toc136_2034211389)

[8.2 Trabajando en el Shell de Linux 8](#__RefHeading___Toc5_2034211389)

[8.2.1 El Shell de Linux 8](#__RefHeading___Toc138_2034211389)

[8.2.2 Comandos básicos 9](#__RefHeading___Toc140_2034211389)

[8.2.3 Comandos de archivos y directorios 11](#__RefHeading___Toc142_2034211389)

[8.2.4 Trabajando con archivos de texto 11](#__RefHeading___Toc144_2034211389)

[8.2.5 La importancia de los archivos de texto en Linux 12](#__RefHeading___Toc146_2034211389)

[8.3 Servidores y clientes de Linux 13](#__RefHeading___Toc7_2034211389)

[8.3.1 Una introducción a las comunicaciones entre Cliente y Servidor 13](#__RefHeading___Toc148_2034211389)

[8.3.2 Servidores, Servicios y sus puertos 14](#__RefHeading___Toc150_2034211389)

[8.3.3 Clientes 15](#__RefHeading___Toc152_2034211389)

[8.4 Administración básica del servidor 16](#__RefHeading___Toc9_2034211389)

[8.4.1 Archivos de configuración de Servicios 16](#__RefHeading___Toc154_2034211389)

[8.4.2 Fortalecimiento de dispositivos 17](#__RefHeading___Toc156_2034211389)

[8.4.3 Monitoreo de los registros de servicio 19](#__RefHeading___Toc158_2034211389)

[8.5 El sistema de archivos Linux 22](#__RefHeading___Toc11_2034211389)

[8.5.1 Los tipos de sistemas de archivos en Linux 22](#__RefHeading___Toc160_2034211389)

[8.5.2 Roles y permisos de archivo en Linux 25](#__RefHeading___Toc162_2034211389)

[8.5.3 Enlaces rígidos y enlaces simbólicos 28](#__RefHeading___Toc164_2034211389)

[8.6 Trabajando con la GUI de Linux 30](#__RefHeading___Toc13_2034211389)

[8.6.1 Sistema X window 30](#__RefHeading___Toc166_2034211389)

[8.6.2 La GUI de Linux 32](#__RefHeading___Toc168_2034211389)

[8.7 Trabajando en un host Linux 33](#__RefHeading___Toc15_2034211389)

[8.7.1 Instalación y ejecución de aplicaciones en un host de Linux 33](#__RefHeading___Toc170_2034211389)

[8.7.2 Mantener el sistema actualizado 34](#__RefHeading___Toc172_2034211389)

[8.7.3 Procesos y bifurcaciones (*forks*) 35](#__RefHeading___Toc174_2034211389)

[8.7.4 Malware en un host de Linux 36](#__RefHeading___Toc176_2034211389)

[8.7.5 Comprobación de rootkit 37](#__RefHeading___Toc178_2034211389)

[8.7.6 Comandos Piping (*command piping*) 38](#__RefHeading___Toc180_2034211389)

[Conclusión 39](#__RefHeading___Toc3009_1735063907)

# Introducción

Linux es un sistema operativo de código abierto que es rápido, potente y altamente personalizable. Está diseñado para su uso en la red como cliente o servidor. Linux es muy querido por una gran comunidad de usuarios, incluido el personal de ciberseguridad. Las habilidades de Linux son muy deseables en la profesión de operaciones de ciberseguridad.

# Nociones Básicas de Linux

## ¿Qué es Linux?

Linux es un sistema operativo de código abierto, el cual requiere muy pocos recursos de *hardware* y tiene muchas opciones de personalización. Está en constante desarrollo por una comunidad abierta de programadores, y se le puede encontrar implementado en un montón de dispositivos, desde relojes hasta supercomputadoras.

Linux está diseñado para la conectividad en red, por lo que es mucho más fácil de emplear en aplicaciones con base en la red. Como es de *código abierto[[1]](#footnote-2)*, es posible para cualquiera acceder al código fuente del *kernel[[2]](#footnote-3)* para hacer con éste como deseen, y hasta redistribuirlo sin preocupaciones.

Linux se distribuye en—valga la redundancia—distribuciones (o *distros*), las cuales son paquetes o configuraciones creados para un fin específico. Éstas incluyen el *kernel*, además de un conjunto de herramienta y paquetes de software personalizados. Algunas distribuciones son adquiridas por medio de una compra, pero la mayoría se ofrece de forma gratuita.

Algunos ejemplos de *distros* son Debian, Red Hat, Ubuntu, CentOS y SUSE, aunque existe una cantidad exagerada de distribuciones que se desarrollan y publican de manera constante.

## El Valor de Linux

* + - **Código abierto**: Ya que es accesible por cualquiera, Linux puede ser adaptado a las necesidades de cualquiera; esto significa que analistas y administradores pueden, a partir de y por encima de Linux, construir un sistema operativo específicamente para análisis de seguridad.
    - **La interfaz de comandos**: La interfaz de comandos (o CLI) de Linux es muy poderosa—y flexible; con ella, y un conjunto de programas sencillos encadenados, es posible llevar a cabo tareas de manera presencial y remota.
    - **Soberanía del usuario**: El usuario administrador (el *superusuario* o usuario *root*), tiene poder absoluto sobre el sistema. Esto significa que puede hacer lo que quiera con el equipo.
    - **Mayor control en redes**: Debido al nivel de control que se tiene sobre el sistema, es posible crear configuraciones bastante específicas, que se ajusten a cualquier necesidad y gusto. Este aspecto lo hace muy atractivo para el ámbito de redes, aplicación para la cual ya existen muchas herramientas orientadas a la red.

## Linux en el SOC[[3]](#footnote-4)

Debido a la flexibilidad que Linux hace posible, se habilita la posibilidad de crear un sistema operativo con funcionalidad mínima, excepto para un propósito en específico, para el cual es optimizado.

Como parte de una instalación al servicio de un centro de operaciones de seguridad, se suelen encontrar las siguientes herramientas:

* **Captura de paquetes**: Permite observar y analizar cada detalle de los flujos de información en la red, hasta el nivel más bajo: el paquete de datos. Un ejemplo popular es Wireshark.
* **Análisis de *Malware***: Permiten ejecutar y observar de manera segura el funcionamiento de un *malware* sin comprometer el sistema.
* **IDS[[4]](#footnote-5)**: Se utilizan para monitorear e inspeccionar el tráfico en tiempo real. Si se detecta una coincidencia con las reglas establecidas, se espera que se lleve a cabo una acción definida.
* **Firewalls**: Se configuran reglas para permitir o prohibir el tráfico de entrada o de salida de una red o dispositivo.
* **Administradores de registros**: Se usan analizar registros (*logs*) generados por los servicios de la red.
* **SIEM**[[5]](#footnote-6): Proporciona análisis en tiempo real de alertas y entradas de registro generadas por dispositivos de red, como IDS y *Firewalls*.
* **Sistema de tiquetes**: Se suelen asignar tiquetes (*tickets*) a los analistas de seguridad, a través de uno de estos sistemas, los cuales deben de estar integrados para operar junto a las alertas de seguridad generadas por los diferentes sistemas de monitoreo.

## Herramientas de Linux

El *PenTesting[[6]](#footnote-7)* es una función que se le suele dar a algunas distribuciones de Linux orientadas a la gestión y análisis de redes. Un famoso ejemplo de Linux aplicado para el *PenTesting* es Kali Linux, una distribución que viene con un montón de herramientas de penetración instaladas por defecto.

Algunas herramientas para el *PenTesting* son:

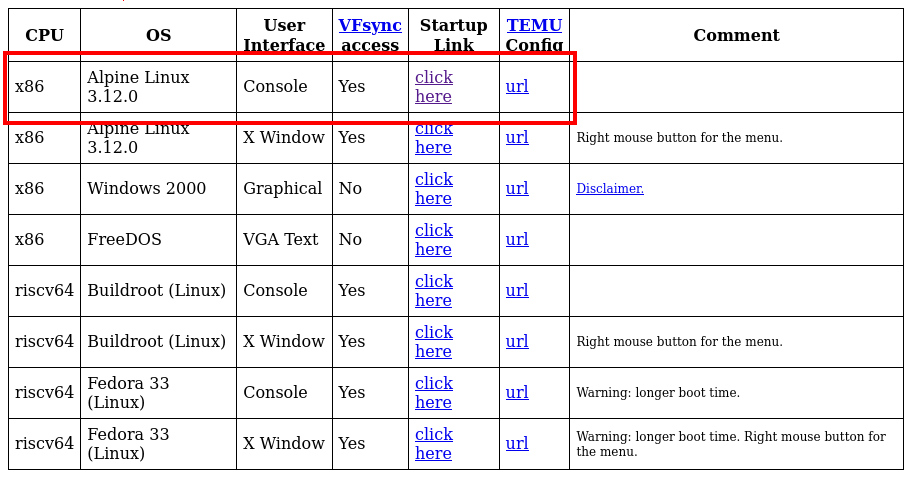
* Generadores de paquetes
* Escáners de puertos
* Ataques de prueba de concepto (*proof-of-concep*t)

# Trabajando en el Shell de Linux

## El Shell de Linux[[7]](#footnote-8)

A diferencia de Windows, cualquier interfaz gráfica de usuario (GUI) en Linux funciona *por encima* de una sesión de la CLI, la cual se suele configurar para iniciar automáticamente, ocultando así la interfaz de texto. Dentro de la GUI, se utilizan programas llamados *emuladores de terminal* para interactuar con una sesión de interfaz de comandos; ejemplos: *Terminator*, *eterm*, *xterm*, *konsole* y *gnome-terminal*. Todos cumplen la misma función, así que no es necesario memorizarlos; sólo saber que tienden a mencionar “terminal” en sus nombres.

Si desea hacer una prueba rápida de la consola de comandos, el curso recomienda visitar la página <https://bellard.org/jslinux/>, en la cual se alberga una versión emulada de Linux.

La versión resaltada en rojo funciona bien.

## Comandos básicos

Cada comando en la interfaz de comandos es un programa. Los programas más comunes ofrecen documentación detallada, la cual se puede encontrar al ejecutar el comando **man*[[8]](#footnote-9)*** antes del nombre del programa; por ejemplo, para ver el manual de **ls**, se ejecuta **man ls**.

A veces, algunos programas no tienen un manual, pero sí un *mensaje de ayuda*, el cual se invoca con el nombre del programa, seguido de **-h** o –**help**; en el caso de **ls**, se usa **ls –help**.

Ya que los llamados *comandos* son realmente programas almacenados en el disco, el *shell* tiene que encontrar el programa, por nombre, antes de poder ejecutarlo. Existen varios directorios[[9]](#footnote-10) en el sistema de archhivos en donde se guardan estos programas; entonces, para saber dónde buscar, el *shell* define una lista de directorios que se denomina *PATH* (ruta). Si el *shell* no encuentra el programa, el usuario tiene que llamar al pograma usando su ubicación exacta, o sino no se podrá ejecutar.

A continuación, una lista de algunos de los comandos básicos que se pueden encontrar en un entorno Linux[[10]](#footnote-11):

|  |  |
| --- | --- |
| **Comando** | **Descripción** |
| mv | Mueve o renombra archivos y directorios |
| chmod | Modifica permisos de archivo |
| chown | Cambia propietario de archivo |
| dd | Copia datos de una entrada (input) a una salida (output) |
| pwd | Escribe el nombre del directorio actual |
| ps | Enumera los procesos que se están ejecutando actualmente |
| su | Permite iniciar sesión como otro usuario o como (s)uper(u)suario |
| sudo | Ejecuta un comando como superusuario |
| grep | Se usa para buscar patrones de texto en un archivo o en la salida de otro programa |
| ifconfig | Se usa para mostrar o configurar información relacionada con la tarjeta de red. Sin parámetros extra, muestra la configuración actual de la tarjeta de red. Actualmente, se preferiere usar **ip address** en su lugar. |
| apt-get | Se usa para instalar, configurar y elimina paquetes de software Debian y distribuciones derivadas de éste (entre otros, Ubuntu). |
| iwconfig | Se usa para mostrar o configurar relacionada con la tarjeta de red inalámbrica. Similar a **ifconfig**, muestra la configuración actual si se ejecuta sin parámetros. |
| shutdown | Dependiendo del sistema, es el comando usado para apagar el sistema. También se puede configurar para que ejecute una serie de tareas relacionadas. |
| passwd | Utilizado para cambiar la contraseña. Si se llama sin parámetros, cambia la contraseña del usuario actual. |
| cat | Se suele utilizar para escribir en pantalla los contenidos de un archivo de texto. |
| man | Busca la documentación de un programa y la muestra en pantalla, si la encuentra. |

## Comandos de archivos y directorios

A continuación se listan algunos de los comandos más comunes relacionados con la gestión de archivos y directorios:

|  |  |
| --- | --- |
| **Comando** | **Descripción** |
| ls | Similar a **DIR**. Muestra los archivos dentro de un directorio. |
| cd | Cambia el directorio actual. |
| mkdir | Crea un directorio en la ubicación especificada. |
| cp | Copia archivos y directorios, de un origen a un destino. |
| mv | Mueve o cambia el nombre de archivos o directorios, de un origen a un destino. |
| rm | Remueve (elimina) archivos, permanentemente. |
| grep | Busca patrones de texto dentro de un archivo o en la salida de otros comandos. |
| cat | Escribe el contenido textual de un archivo. |

## Trabajando con archivos de texto

Existen muchos programas para editar texto en el entorno Linux. Dependiendo de la especialización, contienen características orientadas, por ejemplo, a la programación, con funciones como resaltado de sintaxis.

Ya que la terminal tiende a ser un asunto central en el modelo de trabajo con Linux, se suelen preferir los editores de texto que funcionan desde la línea de comandos.

En muchos casos, además, se prefiere poder tener acceso a la interfaz de texto cuando se accede a una máquina Linux de manera remota, por lo que las herramientas de edición de texto a distancia son herramientas esenciales.

Dos editores de texto populares son ***Nano***, ***Vim***.

## La importancia de los archivos de texto en Linux

Todo en Linux se trata como un archivo de texto: tanto información de diagnóstico del sistema como configuraciones se muestran en pantalla y se manipulan como archivos de texto.

Para modificar el comportamiento de algunos sistemas, se usan editores texto para modificar los archivos relevantes a la configuración de éstos, asumiendo que el usuario tenga los permisos administrativos necesarios para llevar a cabo esta tarea.

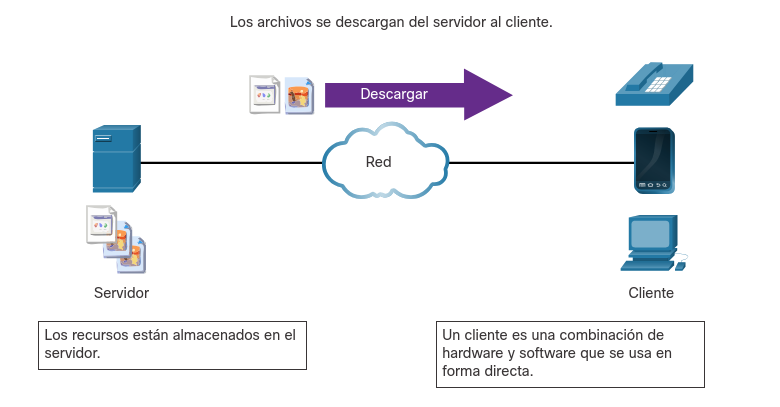
Una vez modificados los diferentes campos de configuración en un archivo dado, el programa relacionado a dicha configuración, comúnmente, debe de ser reiniciado para que detecte los cambios hechos al documento.

# Servidores y clientes de Linux

## Una introducción a las comunicaciones entre Cliente y Servidor

Los servidores son computadoras con programas que les permiten ofrecer servicios a clientes a en la red. Algunos servicios proporcionan recursos externos a los clientes, como archivos, mensajes de correo electrónico o páginas web. Otros servicios ejecutan tareas de mantenimiento, como la administración de registros y de la memoria, análisis de disco y mucho más.

Cada servicio requiere un software de servidor independiente. Por ejemplo, el servidor en la figura utiliza un software para proporcionar al cliente la capacidad de recuperar y enviar archivos.



## Servidores, Servicios y sus puertos

Para que una computadora pueda ofrecer servicios, se usan los puertos. Un puerto es un recurso de red reservado por un servicio. Se dice que un servidor “escucha” cuando se asocia a dicho puerto.

Es posible decidir qué puerto utilizar para cualquier servicio. Aún así, algunos clientes vienen configurados para usar un puerto específico, y es habitual dejar que el servicio se ejecute en el puerto predeterminado.

A continuación, algunos puertos usados comúnmente, conocidos también como *puertos conocidos*:

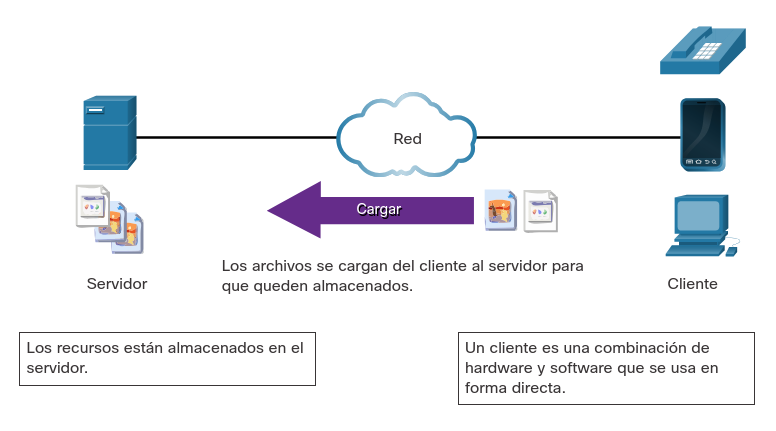
|  |  |
| --- | --- |
| **Puerto** | **Descripción** |
| 20/21 | FTP (File Transfer Protocol) |
| 22 | SSH (Secure Shell) |
| 23 | Telnet |
| 25 | SMTP (Simple Mail Transfer Protocol) |
| 53 | DNS (Domain Name System) |
| 67/68 | DHCP (Dynamic Host Configuration Protocol) |
| 69 | TFTP (Trivial File Transfer Protocol) |
| 80 | HTTP (Hypertext Transfer Protocol) |
| 110 | POP3 (Post Office Protocol version 3) |
| 123 | NTP (Network Time Protocol) |
| 143 | IMAP (Internet Message Access Protocol) |
| 161/162 | SNMP (Simple Network Management Protocol) |
| 443 | HTTPS (HTTP Secure) |

## Clientes

Los clientes son programas que se comunican con servidor en específico. Los clientes usan un protocolo definido para comunicarse con el servidor.

**Ejemplos**:

* Los Navegadores Web (*Web Browsers*) son clientes web que se comunican con servidores web mediante HTTP en el puerto 80.
* El cliente de FTP es un programa utilizado para comunicarse con un servidor FTP. En la figura, se ve a un cliente cargando archivos a un servidor.



# Administración básica del servidor

## Archivos de configuración de Servicios

En Linux, los servicios se administran por medio de archivos de configuración. Opciones comunes en configuraciones son números de puertos, ubicación de los recursos y detalles de autorización del cliente.

Cuando el servicio se inicia, busca sus archivos de configuración, los carga en la memoria y se ajusta conforme a la configuración presente en los archivos. A menudo, las modificaciones del archivo de configuración requieren el reinicio del servicio antes de que los cambios surtan efecto.

Ya que los servicios suelen requerir privilegios de superusuario para ejecutarse, los archivos de configuración de los servicios suelen requerir privilegios de superusuario para editar.

**Ejemplo**: Archivo de configuración de NGINX[[11]](#footnote-12):

#user html;

worker\_processes  1;

#error\_log  logs/error.log;

#error\_log  logs/error.log  notice;

#error\_log  logs/error.log  info;

#pid        logs/nginx.pid;

events {

    worker\_connections 1024;

}

http {

   include      mime.types;

   default\_type  application/octet-stream;

   #log\_format main '$remote\_addr - $remote\_user [$time\_local] "$request"'

   #                  '$status $body\_bytes\_sent "$http\_referer"'

   #                  '"$http\_user\_agent" "$http\_x\_forwarded\_for"';

   #access\_log logs/access.log main;

...

## Fortalecimiento de dispositivos

Consiste en implementar métodos probados para proteger el dispositivo y su acceso administrativo. Algunos de estos métodos implican:

* Mantenimiento de contraseñas.
* Configuración de características de inicio de sesión remoto mejorado
* Implementación de seguridad al iniciar sesión por medio de SSH.

Definir **funciones administrativas** en términos de acceso es otro aspecto importante para proteger los dispositivos de la infraestructura, ya que no todo el personal de TI[[12]](#footnote-13) debe tener el mismo nivel de acceso a los dispositivos de la infraestructura.

Dependiendo de la distribución, muchos **servicios** vienen habilitados de forma predeterminada. Algunas de éstas se habilitaron por motivos históricos, pero ya no son necesarias. Detener estos servicios y garantizar que no se inicien automáticamente al momento del arranque constituye otra técnica de fortalecimiento de dispositivos.

Todos los días se detectan nuevas vulnerabilidades. Los desarrolladores crean y publican correcciones y parches frecuentemente. Las **actualizaciones** del sistema operativo también son fundamentales para mantener un dispositivo fortalecido. Una computadora actualizada corre un riesgo menor de verse afectada.

Algunas *mejores prácticas* para fortalecer un dispositivo son:

* Garantizar la seguridad física.
* Minimizar la cantidad de paquetes instalados.
* Deshabilitar servicios no usados.
* Usar SSH; deshabilitar el inicio de sesión en cuentas *root* en SSH.
* Mantener el sistema actualizado.
* Aplicar contraseñas seguras.
* Forzar cambios de contraseña periódicamente.
* Impedir que los usuarios vuelvan a utilizar contraseñas antiguas.

## Monitoreo de los registros de servicio

Diversos programas en una computadora llevan registro de eventos importantes en lo que llamamos *archivos de registro*, o *logs*. Es importante que un administrador revise periódicamente los registros de un sistema para asegurarse que está libre de problemas.

Al monitorear los *logs* en Linux, el administrador puede obtener una imagen clara del desempeño de la computadora, el estado de la seguridad y cualquier otro problema. El análisis de *logs* le permite protegerse contra futuros problemas que ocurran.

Se pueden clasificar los *logs* del siguiente modo:

* De aplicación
* De eventos
* De servicios
* Del sistema

A continuación, algunos *logs* comunes, y sus funciones:

|  |  |
| --- | --- |
| ***Archivo de registro o log*** | **Descripción** |
| /var/log/messages | * Registros genéricos de la actividad de la computadora. * Mensajes informativos—no críticos—del sistema. * En Debian y derivados, es /var/log/syslog. |
| /var/log/auth.log | * Eventos relacionados con la autenticación en Debian y Ubuntu. * Información relacionada con el mecanismo de autorización. |
| /var/log/secure | * *RedHat* y *CentOS*, en lugar de /var/log/auth.log. * Seguimiento de inicios de sesión sudo, SSH,y errores registrados por SSSD. |
| /var/log/boot.log | * Información relacionada con el arranque y mensajes registrados durante el proceso de inicio de la computadora. |
| /var/log/dmesg | * Mensajes del *kernel ring buffer*[[13]](#footnote-14). * Información relacionada con dispositivos de *hardware* y sus controladores, o *drivers*. * Muy importante. Debido a su *bajo nivel[[14]](#footnote-15)*, |
| /var/log/kern.log | * Información registrada por el *kernel*. |
| /var/log/cron | * Se registran entradas cada vez que *cron[[15]](#footnote-16)* ejecuta una tarea programada (o *cron job*). |
| /var/log/mysqld.log o /var/log/mysql.log | * Registro de MySQL. * Mensajes de depuración, error y éxito relacionados con el proceso *mysqld* y el *demonio*[[16]](#footnote-17) *mysql\_safe*. * En RedHat, CentOS y Fedora, se almacenan *logs* de MySQL en /var/log/mysqld.log, mientras que en los derivados de Debian, el registro se alberga en /var/log/mysql.log. |

Este es un ejemplo del *log* o registro /var/log/messages, el cual se muestra al ejecutar **sudo cat /var/log/messages**:

Mar 20 15:28:45 secOps kernel: Linux version 4.15.10-1-ARCH (builduser@heftig-18961) (gcc version 7.3.1

20180312 (GCC)) #1 SMP PREEMPT Thu Mar 15 12:24:34 UTC 2018

Mar 20 15:28:45 secOps kernel: Command line: BOOT\_IMAGE=/boot/vmlinuz-linux root=UUID=07c6b457-3f39-

4ddf-bfd8-c169e8a877b2 rw quiet

Mar 20 15:28:45 secOps kernel: KERNEL supported cpus:

Mar 20 15:28:45 secOps kernel:   Intel GenuineIntel

Mar 20 15:28:45 secOps kernel:   AMD AuthenticAMD

Mar 20 15:28:45 secOps kernel:   Centaur CentaurHauls

Mar 20 15:28:45 secOps kernel: x86/fpu: Supporting XSAVE feature 0x001: 'x87 floating point registers'

Mar 20 15:28:45 secOps kernel: x86/fpu: Supporting XSAVE feature 0x002: 'SSE registers'

Mar 20 15:28:45 secOps kernel: x86/fpu: Supporting XSAVE feature 0x004: 'AVX registers'

Mar 20 15:28:45 secOps kernel: x86/fpu: xstate\_offset[2]: 576, xstate\_sizes[2]: 256

Mar 20 15:28:45 secOps kernel: x86/fpu: Enabled xstate features 0x7, context size is 832 bytes, using

'standard' format.

Mar 20 15:28:45 secOps kernel: e820: BIOS-provided physical RAM map:

Mar 20 15:28:45 secOps kernel: BIOS-e820: [mem 0x0000000000000000-0x000000000009fbff] usable

Mar 20 15:28:45 secOps kernel: BIOS-e820: [mem 0x000000000009fc00-0x000000000009ffff] reserved

Mar 20 15:28:45 secOps kernel: BIOS-e820: [mem 0x00000000000f0000-0x00000000000fffff] reserved

Mar 20 15:28:45 secOps kernel: BIOS-e820: [mem 0x0000000000100000-0x000000003ffeffff] usable

Mar 20 15:28:45 secOps kernel: BIOS-e820: [mem 0x000000003fff0000-0x000000003fffffff] ACPI data

Mar 20 15:28:45 secOps kernel: BIOS-e820: [mem 0x00000000fec00000-0x00000000fec00fff] reserved

Mar 20 15:28:45 secOps kernel: BIOS-e820: [mem 0x00000000fee00000-0x00000000fee00fff] reserved

Mar 20 15:28:45 secOps kernel: BIOS-e820: [mem 0x00000000fffc0000-0x00000000ffffffff] reserved

Mar 20 15:28:45 secOps kernel: NX (Execute Disable) protection: active

Mar 20 15:28:45 secOps kernel: random: fast init done

Mar 20 15:28:45 secOps kernel: SMBIOS 2.5 present.

Mar 20 15:28:45 secOps kernel: DMI: innotek GmbH VirtualBox/VirtualBox, BIOS VirtualBox 12/01/2006

Mar 20 15:28:45 secOps kernel: Hypervisor detected: KVM

Mar 20 15:28:45 secOps kernel: e820: last\_pfn = 0x3fff0 max\_arch\_pfn = 0x400000000

Mar 20 15:28:45 secOps kernel: MTRR: Disabled

Mar 20 15:28:45 secOps kernel: x86/PAT: MTRRs disabled, skipping PAT initialization too.

Mar 20 15:28:45 secOps kernel: CPU MTRRs all blank - virtualized system.

# El sistema de archivos Linux

## Los tipos de sistemas de archivos en Linux

Hay muchos tipos de sistemas de archivos, que varían en velocidad, flexibilidad, seguridad, tamaño, estructura, lógica y mucho más. El administrador decide qué tipo de sistema de archivos se adapta mejor al sistema operativo y los archivos que almacenará.

Estos son los sistemas de archivos comúnmente reconocidos y admitidos por un sistema Linux:

|  |  |
| --- | --- |
| **Sistema de archivos (o *filesystem*)** | **Descripción** |
| ext2 (extended file system 2) | * El sistema de archivos predeterminado en varias *distros* de Linux, hasta ser sustituido por *ext3*. * Preferido en medios de amacenamiento *flash[[17]](#footnote-18)*. * Como carece de registros de diario o bitácora, aumenta el rendimiento y reduce la cantidad de escrituras. |
| ext3 (extended file system 3) | * Implementa la técnica de *diario* o *bitácora[[18]](#footnote-19)* para mejorar el sistema *ext2*. * El tamaño máximo de un sólo archivo en un sistema *ext3* es de 32TB. |
| ext4 (extended file system 4) | * Creado a partir de una serie de extensiones a *ext3*. * Estas extensiones mejoraban el rendimiento de ext3 y aumentaban el tamaño de los archivos admitidos (los 32TB) * A los desarrolladores del kernel de Linux les preocupaba la estabilidad y se opusieron a añadir las extensiones al sistema ext3 estable. * Entonces el proyecto de ext3 se dividió en ext3, el cual continúa su desarrollo normal; y ext4, el cual incorpora las extensiones mencionadas. |
| NFS (Network File System) | * Basado en la red, permite el acceso a archivos en la red. * Para el usuario no se nota una diferencia entre un archivo almacenado localmente y uno en otra computadora en la red. * NFS es un estándar abierto que permite que cualquier persona lo implemente. |
| CDFS (Compact Disk File System) | * Específicamente para medios de discos ópticos. |
| *Swap* (Intercambio) | * Técnicamente, es una partición[[19]](#footnote-20) que no tiene un sistema de archivos específico, pero debe incluirse en el análisis de sistemas de archivos. * Cuando Linux se queda sin RAM, el kernel mueve el contenido inactivo de la RAM a la partición *swap* en el disco. * Si bien pueden ser útiles para computadoras con una cantidad limitada de memoria, no deben de considerarse una solución principal. * El disco en el que se almacena tiene velocidades de acceso mucho menores a la de la RAM. |
| HFS+ (Hierarchical File System Plus) | * Usado por Apple en la línea Macintosh. * El kernel de Linux incluye un módulo para montar HFS+ para operaciones de lectura y escritura. |
| APFS (Apple File System) | * Utilizado por dispositivos de Apple. Proporciona un cifrado fuerte y está optimizado para unidades de estado sólido (SSD), y *flash*. |
| MBR (Master Boot Record) | * Traducido como “Registro Principal de Arranque”. * Se sitúa en el primer sector de una computadora con particiones. * Almacena toda la información sobre la manera en que está organizado el sistema de archivos. * Cede rápidamente el control a una función de carga para que realice la carga del sistema operativo. |

El proceso de asignación de un directorio a una partición es montaje (*mount*). Después de una operación de montaje exitosa, se puede acceder al sistema de archivos contenido en la partición, a través del directorio especificado.

El directorio (en el cual se montó la partición) se llama el *punto de montaje* (*mount point*) para ese sistema de archivos. Es posible que los usuarios de Windows estén familiarizados con un concepto similar: la letra de unidad (C:, D:, E:, …).

Al ejecutar el comando **mount** sin parámetros adicionales, se produce una lista de los *puntos de montaje* actuales en el sistema:

proc on /proc type proc (rw,nosuid,nodev,noexec,relatime)

sys on /sys type sysfs (rw,nosuid,nodev,noexec,relatime)

dev on /dev type devtmpfs (rw,nosuid,relatime,size=494944k,nr\_inodes=123736,mode=755)

run on /run type tmpfs (rw,nosuid,nodev,relatime,mode=755)

**/dev/sda1 on / type ext4 (rw,relatime)**

securityfs on /sys/kernel/security type securityfs (rw,nosuid,nodev,noexec,relatime)

tmpfs on /dev/shm type tmpfs (rw,nosuid,nodev)

devpts on /dev/pts type devpts (rw,nosuid,noexec,relatime,gid=5,mode=620,ptmxmode=000)

tmpfs on /sys/fs/cgroup type tmpfs (ro,nosuid,nodev,noexec,mode=755)

cgroup2 on /sys/fs/cgroup/unified type cgroup2 (rw,nosuid,nodev,noexec,relatime,nsdelegate)

cgroup on /sys/fs/cgroup/systemd type cgroup (rw,nosuid,nodev,noexec,relatime,xattr,name=systemd)

pstore on /sys/fs/pstore type pstore (rw,nosuid,nodev,noexec,relatime)

none on /sys/fs/bpf type bpf (rw,nosuid,nodev,noexec,relatime,mode=700)

cgroup on /sys/fs/cgroup/rdma type cgroup (rw,nosuid,nodev,noexec,relatime,rdma)

cgroup on /sys/fs/cgroup/cpu,cpuacct type cgroup (rw,nosuid,nodev,noexec,relatime,cpu,cpuacct)

cgroup on /sys/fs/cgroup/blkio type cgroup (rw,nosuid,nodev,noexec,relatime,blkio)

cgroup on /sys/fs/cgroup/hugetlb type cgroup (rw,nosuid,nodev,noexec,relatime,hugetlb)

cgroup on /sys/fs/cgroup/cpuset type cgroup (rw,nosuid,nodev,noexec,relatime,cpuset)

cgroup on /sys/fs/cgroup/devices type cgroup (rw,nosuid,nodev,noexec,relatime,devices)

cgroup on /sys/fs/cgroup/pids type cgroup (rw,nosuid,nodev,noexec,relatime,pids)

...

El sistema de archivos raíz (resaltado en **negrita**) está representado por el símbolo “/” y contiene todos los archivos predeterminados de la computadora. También se muestra en el resultado que el sistema de archivos root fue formateado como ext4 y ocupa la primera partición de la primera unidad (/dev/sda1).

## Roles y permisos de archivo en Linux

En Linux, se trata a la mayoría de las entidades del sistema como archivos. Para organizar el sistema e imponer límites dentro de la computadora, se usan permisos de archivos.

Los permisos de archivos están integrados en la estructura del sistema de archivos y proporcionan un mecanismo para definir los permisos de cada archivo. Cada archivo en Linux trae sus permisos de archivo, los cuales definen las acciones que el propietario, el grupo y otros puede hacer con el archivo.

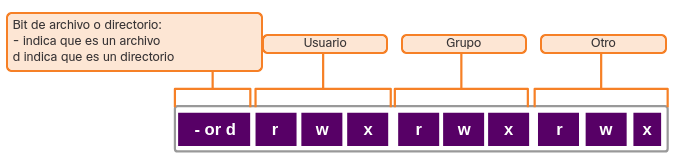
Los posibles permisos son **leer** **((*r*)*ead→r*)**, **escribir ((*w*)*rite*→w)** y **ejecutar (*e*(*x*)*ecute*→x)**. El comando ls con el parámetro -l detalla información adicional sobre el archivo:

-rwxrw-r-- 1 analyst staff 253 May 20 12:49 space.txt

**(1) (2)(3) (4) (5) (6) (7)**

1. **-rwxrw-r--** permite ver los permisos asociados con el archivo:
   * El guión con el que comienza, indica que se trata de un archivo. Si fuera un directorio, en lugar de un guión, se mostraría una letra “*d*”.
   * Los primeros tres caracteres después del primer guión (**rwx**) definen los permisos del usuario.
   * Los tres que siguen (**rw-**), muestran los permisos del grupo.
   * Los siguientes tres (**r--**), muestran los permisos de cualquier otro ususario o grupo que no posea el archivo.
2. **1** define la cantidad de enlaces físicos[[20]](#footnote-21) hacia el archivo.
3. **analyst** muestra el usuario que posee el archivo.
4. **staff** muestra el grupo de usuarios que posee el archivo.
5. **253** muestra el tamaño en bytes del archivo.
6. **May 20 12:49** muestra la fecha y hora de la última modificación.
7. **space.txt** muestra el nombre del archivo.

La figura a continuación muestra un desglose de los permisos de archivos en Linux (el punto **(1)**):



En ocasiones, se usan valores octales para definir los permisos:

|  |  |  |
| --- | --- | --- |
| **Binario** | **Octal** | **Permiso** |
| 000 | 0 | --- |
| 001 | 1 | --x |
| 010 | 2 | -w- |
| 011 | 3 | -wx |
| 100 | 4 | r-- |
| 101 | 5 | r-x |
| 110 | 6 | rw- |
| 111 | 7 | rwx |

Los permisos de archivos son una parte fundamental de Linux. Un usuario tiene tantos derechos sobre un archivo como se lo permiten los permisos de archivo. El único usuario que puede anular el permiso de archivo en una computadora con Linux es el usuario *root*. Como el usuario *root* tiene la capacidad para anular permisos de archivos, puede escribir en cualquier archivo. Dado que todo se trata como un archivo, el usuario root tiene control total sobre la computadora Linux.

El acceso *root* se requiere a menudo antes de realizar mantenimiento y tareas administrativas. Las credenciales *root* deben usar contraseñas seguras y no compartirse con nadie más que los administradores del sistema y otros usuarios de alto nivel.

## Enlaces rígidos y enlaces simbólicos

Un enlace rígido (*hard link*) un archivo que apunta a la misma ubicación de otro archivo. El comando que se usa para formar enlaces rígidos es **ln [archivo a enlazar] [archivo destino]**.

[analyst@secOps ~]$ ln space.txt space.hard.txt

[analyst@secOps ~]$ ls -l space\*

-rw-r--r-- 2 analyst analyst 239 May 7 18:18 space.hard.txt

-rw-r--r-- 2 analyst analyst 239 May 7 18:18 space.txt

Un enlace simbólico, también llamado enlace suave (*soft link*) o *symlink*, es similar a un enlace físico en el sentido en que aplicar cambios a un enlace simbólico también cambia el archivo original.

El comando que se usa para formar enlaces rígidos es   
**ln -s [/ruta/absoluta/al/archivo] [archivo destino]**:

[analyst@secOps ~]$ ln -s /home/user/test.txt mytest.txt

[analyst@secOps ~]$ ls -l mytest.txt

lrwxrwxrwx 1 analyst analyst 8 May 7 20:17 mytest.txt -> /home/user/test.txt

Sin embargo, a diferencia de un enlace físico, eliminar el archivo original text.txt implica que mytext **.txt** ahora está vinculado a un archivo que ya no existe.

Por lo general, es recomendable usar *symlinks* sobre *hard links*. He aquí algunas de sus ventajas sobre los *hard links*:

* Es más difícil localizar enlaces físicos. Los enlaces simbólicos muestran la ubicación del archivo original en el comando **ls -l**.
* Los enlaces físicos están limitados al sistema de archivos en el que se crean. Los enlaces simbólicos pueden estar vinculados a un archivo en otro sistema de archivos.
* Los enlaces físicos no pueden estar vinculados a un directorio porque el propio sistema utiliza enlaces físicos para definir la jerarquía de la estructura de directorios. Sin embargo, los enlaces simbólicos pueden estar vinculados a directorios.

# Trabajando con la GUI de Linux

## Sistema X window

La interfaz gráfica presente en la mayoría de las computadoras Linux tiene como base el *X Window System*. También conocido como X o X11, es un sistema de ventanas diseñado para proporcionar el *framework* básico para una GUI. X incluye funciones para dibujar y mover ventanas en el dispositivo de visualización, e interactuar con un mouse y un teclado.

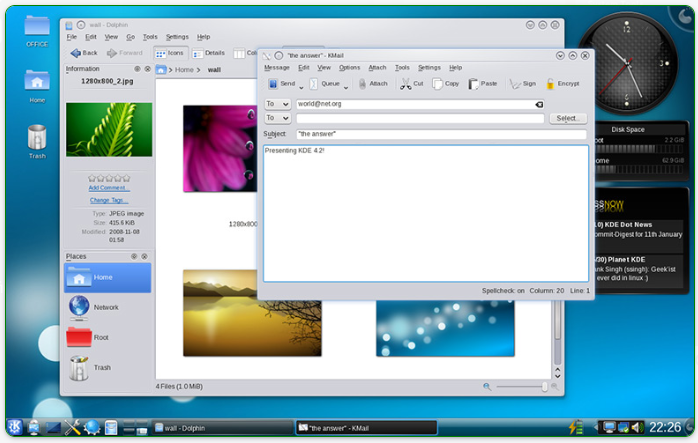
X funciona como un servidor, el cual le permite a un usuario remoto utilizar la red para conectarse, iniciar una aplicación gráfica y mantener la ventana de gráficos abierta en el terminal remoto. Si bien la aplicación propiamente dicha se ejecuta en el servidor, la apariencia gráfica se envía por X a través de la red y se ve en la computadora remota.

X no especifica la interfaz de usuario y deja que otros programas, como los gestores de ventanas (*Window Managers*), definan todos los componentes gráficos. Esta abstracción otorga gran flexibilidad y personalización, ya que es la aplicación del usuario la que define los componentes gráficos, como botones, fuentes, iconos, bordes de ventanas y esquemas de color.

Debido a esta separación, la GUI de Linux varía enormemente de una distribución a otra. Ejemplos de gestores de ventanas son Gnome y KDE. Mientras que el aspecto de los gestores de ventanas varía, los componentes principales permanecen presentes.

Un par de ejemplos de gestores de ventanas:

 Gnome

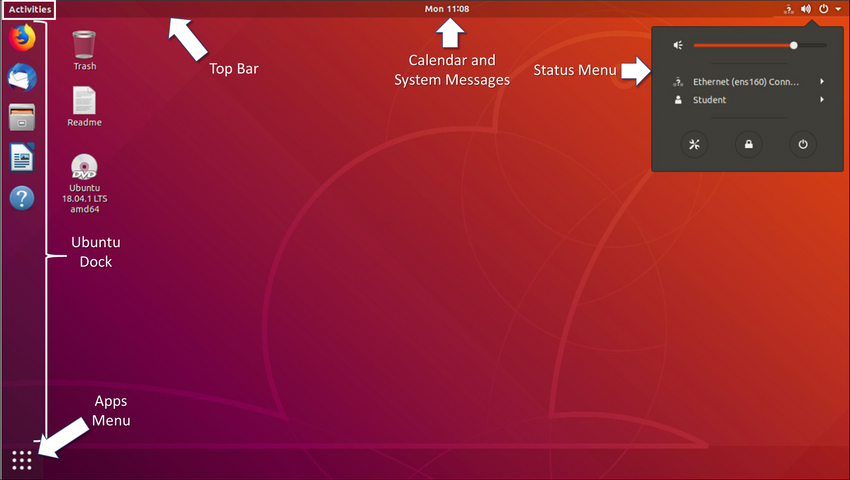
KDE

## La GUI de Linux

Aunque un sistema operativo no necesita una GUI para funcionar, las GUI se consideran más fáciles de usar que la CLI. El usuario puede reemplazar fácilmente la GUI de Linux en su totalidad. Existen muchas distribuciones y gestores de ventanas. Este ejemplo es pertinente a Ubuntu, ya que es una distribución popular y fácil de usar.

Ubuntu Linux, por lo general, utiliza Gnome 3 como la GUI predeterminada. El objetivo de Gnome 3 es hacer que Ubuntu sea todavía más “fácil” de usar.

La figura muestra la ubicación de algunas de las características del escritorio Ubuntu Gnome 3.



# Trabajando en un host Linux

## Instalación y ejecución de aplicaciones en un host de Linux

Para ayudar en el proceso de instalación, Linux suele incluir programas llamados ***administradores de paquetes***. “*Paquete*” es el término utilizado para referirse a un programa y a todos sus archivos compatibles. El administrador de paquetes permite colocar todos los archivos necesarios en la ubicación correcta en el sistema de archivos.

Los administradores de paquetes varían dependiendo de las distribuciones de Linux: **pacman** es utilizado por Arch Linux mientras que **dpkg** (paquete Debian) y **apt** ( Advanced Packaging Tool) se utilizan en las distribuciones Debian y Ubuntu Linux.

## Mantener el sistema actualizado

Las actualizaciones del sistema son hechas disponibles periódicamente para abordar cualquier vulnerabilidad conocida en los sistemas. En el nivel empresarial, es posible que se dén actualizaciones programadas, además de actualizaciones no programadas en caso de querer evadir una amenaza. Es posible que, dependiendo del sistema, se reciban alertas sobre nuevas actualizaciones, entre otros detalles pertinentes.

La tabla presenta los comandos para ejecutar algunas tareas relacionadas a la actualización de los paquetes del sistema, según distribución (Arch Linux y Debian/Ubuntu), más o menos:

|  |  |  |
| --- | --- | --- |
| **Tarea** | **Arch** | **Debian/Ubuntu** |
| Instalar un paquete | pacman -S [paquete] | apt install [paquete] |
| Eliminar un paquete | pacman -Rs [paquete] | apt remove [paquete] |
| Actualizar un paquete | pacman -Syy [paquete] | apt-get update [paquete] |
| Actualizar todos los paquetes | pacman -Syu [paquete] | apt-get upgrade |

También existen herramientas de GUI que abstraen y facilitan la gestión de paquetes.

## Procesos y bifurcaciones (*forks*)

Un proceso es una instancia en ejecución de un programa. Los sistemas operativos *multitarea* son capaces de ejecutar muchos procesos al mismo tiempo.

El *forking* es un método utilizado por el kernel para permitir que un proceso cree una copia de sí mismo. En Linux, es la única manera en la que un proceso puede crear un nuevo proceso. El *forking* aporta al aspecto de la escalabilidad de procesos.

Cuando un proceso invoca un *fork*, éste se convierte en el proceso madre, y su nuevo *fork* se denomina su hijo. Luego de este intercambio, cada cual proceso es inependiente del otro, hasta cierto punto; ambos tienen el mismo ID de proceso (PID), pero ejecutan el mismo código.

La tabla muestra algunos comandos para gestionar procesos:

|  |  |
| --- | --- |
| **Comando** | **Descripción** |
| ps | * Lista los procesos actualmente en ejecución en la computadora. * Ofrece opciones para mostrar procesos que pertenecen al usuario. * No requiere privilegios *root*. |
| top | * Lista procesos en ejecución, en tiempo real, en una pantalla cambiante. * Para salir, se presiona “q”. |
| kill | * Se usa para modificar el comportamiento de un proceso en específico. * Dependiendo de los parámetros, mata, reinica o pausa un proceso. * Se necesita conocer el PID del proceso cuyo comportamiento se desea modificar. |

## Malware en un host de Linux

Incluye virus, troyanos, gusanos y otros tipos de malware que pueden afectar el sistema operativo.

Aunque se podría decir que está mejor protegido, Linux no es inmune al malware. Se han encontrado y aprovechado muchas vulnerabilidades en Linux. Debido a la naturaleza de código abierto de Linux, los parches y las correcciones suelen estar disponibles apenas horas después de detectar este tipo de problemas.

Si se ejecuta un programa malicioso, causará daños, independientemente de la plataforma. Un vector de ataque común en Linux son sus servicios y procesos. Con frecuencia, las vulnerabilidades se detectan en el código de servidores y procesos ejecutándose en computadoras conectadas a la red.

A menudo, los atacantes sondean puertos abiertos para determinar la versión y naturaleza del servidor en ese puerto. Con ese conocimiento, los atacantes pueden investigar si hay algún problema conocido con esa versión particular de ese servidor específico para mejorar el ataque.

Mantener la computadora actualizada y cerrar todos los servicios y puertos no utilizados es una buena manera de reducir las posibilidades de ataque en una computadora con Linux.

## Comprobación de rootkit

Un *rootkit* es un tipo de malware diseñado para aumentar los privilegios de un usuario no autorizado u otorgar acceso a partes del software que no suelen estar disponibles. Los rootkit también suelen usarse para asegurar el ingreso a una puerta trasera de una computadora atacada. Es posible automatizar la instalación de un rootkit (como parte de una infección) o un atacante puede instalarlo manualmente después de atacar una computadora.

Un rootkit cambia el código del kernel y sus módulos, lo que afecta las operaciones más fundamentales del sistema operativo propiamente dicho. Pueden ocultar la intrusión, eliminar cualquier rastro de su instalación y manipular las herramientas de solución de problemas y diagnóstico para que oculten la presencia del rootkit en sus resultados. La gran mayoría de los ataques con rootkit requieren acceso de root o administrador.

La detección de rootkits puede ser muy difícil. Los métodos de detección típicos suelen incluir arrancar la computadora desde medios confiables. Se monta la unidad afectada y, desde el conjunto seguro de herramientas del sistema, es posible iniciar herramientas de diagnóstico confiables para inspeccionar el sistema de archivos atacado.

La eliminación de rootkits puede ser complicada y, a menudo, es imposible. La reinstalación del sistema operativo suele ser la única solución real. Normalmente, los rootkits de firmware requieren la sustitución de hardware.

***chkrootkit*** es un software popular con base en Linux diseñado para comprobar si la computadora tiene rootkits conocidos. Aunque son útiles, los programas para detectar rootkits no son infalibles.

## Comandos Piping[[21]](#footnote-22) (*command piping*)

Muchos comandos se pueden combinar para realizar tareas más complejas mediante una técnica conocida como *tuberías*. El carácter que define el proceso de tubería es la barra vertical (|); consiste en vincular comandos entre sí, de manera que el resultado (*output*) de un comando alimente la entrada (*input*) de otro.

Los dos comandos, **ls** y **grep,** pueden ser entubados para filtrar el resultado de **ls** . Esto se muestra en la salida del comando **ls -l | grep host**:

**[analyst@secOps ~]$ ls -l**

total 40

drwxr-xr-x 2 analyst analyst    4096 Mar  22  2018 Desktop

drwxr-xr-x 3 analyst analyst    4096 April 2 14:44 Downloads

-rw-r--r-- 1 analyst analyst    9 May 20 10:51 hostfile1.txt

-rw-r--r-- 1 analyst analyst    9 May 20 10:51 hostfile2.txt

-rw-r--r-- 1 analyst analyst    9 May 20 10:52 hostfile3.txt

drwxr-xr-x 9 analyst analyst    4096 Jul 19 2018 lab.support.files

-rw-r--r-- 1 analyst analyst    19 May 20 10:53 mytest.com

-rw-r--r-- 1 analyst analyst    228844 May 20 10:54 rkhunter-1.4.6-1-any.pkg.tar.xz

drwxr-xr-x 2 analyst analyst     4096 Mar 21 2018 second\_drive

-rw-r--r-- 1 analyst analyst    257 May 20 10:52 space.txt

**[analyst@secOps ~]$ ls -l | grep host**

-rw-r--r-- 1 analyst analyst    9 May 20 10:51 hostfile1.txt

-rw-r--r-- 1 analyst analyst    9 May 20 10:51 hostfile2.txt

-rw-r--r-- 1 analyst analyst    9 May 20 10:52 hostfile3.txt

# Conclusión

* Linux es un sistema operativo de código abierto rápido, confiable y pequeño. Requiere pocos recursos de hardware para ejecutarse y es altamente personalizable. Está diseñado para ser utilizado en redes.
* Los comandos de Linux son programas que realizan una tarea específica. El comando man, seguido de un comando específico, proporciona documentación para ese comando.
* En Linux todo se trata como si fuera un archivo, incluyendo la memoria, los discos, el monitor y los directorios.
* Los servidores son computadoras que tienen software instalado que les permite brindar servicios a las computadoras cliente a través de la red. Las aplicaciones de software de cliente están diseñadas para comunicarse con tipos específicos de servidores.
* En Linux, los servidores se administran mediante archivos de configuración. Se pueden modificar y guardar varios ajustes en archivos de configuración. Cuando se inicia un servicio, mira sus archivos de configuración para saber cómo debe ejecutarse.
* Los dispositivos Linux deben protegerse mediante el uso de métodos probados para proteger el dispositivo y el acceso administrativo.
* Algunos de los sistemas de archivos compatibles con Linux son ext2, ext3, ext4, NFS y CDFS. Los sistemas de archivos se montan en particiones y se accede a ellos a través de puntos de montaje o directorios.
* Linux usa permisos de archivo para controlar quién puede tener diferentes tipos de acceso a archivos y directorios. Los permisos incluyen lectura (r), escritura (w) y ejecución (x).
* Los enlaces duros se crean con el comando ln. Los cambios en uno de los archivos vinculados también se realizan en el archivo original. Los enlaces simbólicos, o symlinks, son similares a los enlaces duros en el sentido de que un cambio en el archivo vinculado se refleja en el archivo original. Los enlaces simbólicos tienen varias ventajas sobre los enlaces duros.
* El Sistema X Window, o X11, es un marco de software básico que incluye funciones para crear, controlar y configurar una GUI de ventanas en una interfaz de apuntar y hacer clic. Diferentes proveedores utilizan el Sistema X Window para crear diferentes GUI de administrador de ventanas para Linux.
* Para instalar aplicaciones en hosts Linux, se utilizan programas llamados administradores de paquetes. Los paquetes son aplicaciones de software y todos sus archivos de soporte.
* Los procesos de software son instancias de programas de computadora que se están ejecutando. Los sistemas operativos multitarea pueden ejecutar muchos procesos al mismo tiempo. La bifurcación es un método que utiliza el kernel para permitir que un proceso en ejecución se copie a sí mismo.
* Si bien se considera que Linux está mejor protegido contra software malicioso (malware) que otros sistemas operativos, aún es susceptible a troyanos, gusanos y otros tipos de malware. Linux suele ser atacado a través de sus servicios y procesos.
* La combinación de comandos utiliza el símbolo “|” para encadenar diferentes comandos usando la salida de un comando como entrada para otro.

# Preguntas

1. ¿Qué es código abierto?
2. ¿Qué hace el comando mv?
3. ¿Qué hace el comando chmod?
4. ¿Qué hace el comando chown?
5. Qué distribución de Linux se suele usar para *PenTesting*?
6. ¿Qué es un servidor?
7. ¿Qué es un cliente?
8. ¿Cómo trata Linux a la mayoría de sus componentes?
9. ¿Linux puede ser atacado?
10. ¿Cuáles son los tres permisos con los que se marca un documento en Linux?
11. Al menos tres sistemas de archivos
12. ¿Cómo se llama un servidor comúnmente usado para la construcción de gráficos en Linux?

1. **Código abierto**: Es posible acceder, leer y modificar el código sin compilar de un programa. [↑](#footnote-ref-2)
2. ***Kernel***: El centro de todo sistema operativo. Linux es, en realidad, solamente el kernel. [↑](#footnote-ref-3)
3. **SOC**: Security Operations Center (Centro de operaciones de seguridad). [↑](#footnote-ref-4)
4. **IDS**: Intrusion Detection System (Sistema de detección de intrusiones). Visto anteriormente. [↑](#footnote-ref-5)
5. **SIEM**: Security Information and Event Management (Administración de información y eventos de seguridad). [↑](#footnote-ref-6)
6. **PenTesting**: Pruebas de penetración. Son el proceso de atacar una red o computadora en busca de vulnerabilidades. En otras palabras, son puebas de *hackeo*. [↑](#footnote-ref-7)
7. **Nota**: Aunque los distintos nombres implican diferentes connotaciones, los términos usados para referirse a esta heramienta suelen ser *shell*, *consola*, *terminal*, y *CLI*. [↑](#footnote-ref-8)
8. ***man***: Abreviación de *manual*. [↑](#footnote-ref-9)
9. **Directorio**: carpeta. [↑](#footnote-ref-10)
10. **Nota**: Algunos de los comandos listados asumen que el usuario tiene permisos de administrador. [↑](#footnote-ref-11)
11. **NGINX**: Un ligero servidor web (HTTP) para Linux. [↑](#footnote-ref-12)
12. **TI**: Tecnología de la Información, también conocido como *IT*. Disciplina general dedicada las computadoras y su manejo. [↑](#footnote-ref-13)
13. ***Kernel ring buffer***: Un búfer especial, de bajo nivel, dedicado al *kernel*. [↑](#footnote-ref-14)
14. **Bajo nivel**: Si visualizamos a un sistema operativo desde su cercanía al *hardware—*es decir, qué tan directamente opera en relación a éste—podemos imaginar que los niveles más bajos descansan por encima del metal de la máquina, por así decirlo. [↑](#footnote-ref-15)
15. ***cron****: Cron es un servicio para programar tareas automatizadas.* [↑](#footnote-ref-16)
16. **Demonio**: Más comúnmente llamados *daemon*, son los procesos o servicios que trabajan *en el fondo* (*background*). Su connotación no es satánica, sino que deriva de la mitología griega. [↑](#footnote-ref-17)
17. **Memoria flash**: Medio de memoria, ejemplificado por las memorias USB o llaves Maya. [↑](#footnote-ref-18)
18. **Diario, bitácora, journal**: Es una técnica que se utiliza para minimizar el riesgo de corrupción del sistema de archivos en caso de una interrupación repentina en el suminstro de corriente. En síntesis, el sistema mantiene un registro de todos los cambios que están por efectuarse en el sistema de archivos; si la computadora deja de funcionar antes de que el cambio se complete, el diario puede usarse para corregir cualquier posible problema causado por la interrupción [↑](#footnote-ref-19)
19. **Parcición**: Una porción de varias entre las cuales se divide, lógicamente, al disco duro. [↑](#footnote-ref-20)
20. **Enlace físico**: Crea otro archivo con un nombre diferente vinculado al mismo lugar en el sistema de archivos (se lo denomina *inode*). [↑](#footnote-ref-21)
21. **Comandos Piping**: Mal título, la traducción correcta sería algo así como “tuberías de comandos”. [↑](#footnote-ref-22)