Implementando Seguridad en GNU/Linux

Gabriel Orlando Cuentas Torres
e-mail:gocuentast@unadvirtual.edu.co

RESUMEN: *Este documento se enfoca en la formación de profesionales de Ingeniería de Sistemas en la administración de sistemas GNU/Linux, destacando su relevancia en el panorama actual de la tecnología de la información. A través de dos actividades principales, el trabajo aborda desde la instalación y configuración básica del sistema operativo y la gestión de paquetes, hasta la seguridad perimetral mediante la configuración de NAT.*

El objetivo principal es desarrollar las competencias necesarias para implementar y administrar entornos GNU/Linux en contextos corporativos, promoviendo el uso de software libre y open source. Se subraya la importancia de estas habilidades para optimizar recursos, garantizar la seguridad informática y fomentar la autonomía tecnológica en las organizaciones.

PALABRAS CLAVE: Administración, GNU/Linux, Instalación, NAT

1 INTRODUCCIÓN

En el contexto actual de la tecnología de la información, el conocimiento y dominio de los sistemas operativos GNU/Linux ya no es una opción exclusiva de expertos, sino una competencia esencial para cualquier profesional de Ingeniería de Sistemas que aspire a desenvolverse con solvencia en un entorno laboral competitivo. Lejos de ser una simple alternativa técnica, Linux representa un modelo de autonomía tecnológica, colaboración global y adaptación a necesidades específicas, gracias a su naturaleza de código abierto y su sólida arquitectura.

Este sistema operativo se ha convertido en el motor silencioso de gran parte de la infraestructura tecnológica moderna: desde servidores empresariales y centros de datos, hasta servicios en la nube y dispositivos embebidos. Su estabilidad, seguridad y escalabilidad lo hacen indispensable en contextos donde la eficiencia y la personalización son clave.

Sin embargo, más allá de lo técnico, trabajar con Linux plantea un reto formativo y filosófico: no se trata solo de aprender comandos, sino de entender un ecosistema que promueve la resolución de problemas de forma colaborativa, el pensamiento crítico y la capacidad de adaptar soluciones a contextos cambiantes.

Este articulo busca ir más allá de una simple práctica técnica. Se enfoca en el reconocimiento e implementación de entornos de trabajo en GNU/Linux, abordando desde los aspectos básicos de instalación y configuración, hasta la administración de servicios y recursos en entornos corporativos. La intención no es solo dominar herramientas, sino desarrollar una mentalidad analítica y adaptable, que permita al futuro ingeniero enfrentar desafíos reales con autonomía y criterio profesional.

2 Instalación de Linux y gestión de paquetes

2.1 Diseño del esquema de particionado del disco duro

Trabajar con sistemas GNU/Linux implica comprender no solo cómo funcionan los componentes técnicos, sino también por qué están diseñados de cierta manera y cómo cada decisión de configuración puede impactar la estabilidad y eficiencia del sistema. Uno de los primeros elementos fundamentales en este entorno es el gestor de arranque GRUB, pieza clave para iniciar correctamente el sistema operativo. Sus archivos principales se encuentran en rutas como /boot/grub/ o /boot/grub2/, mientras que la configuración principal suele centralizarse en /etc/default/grub. Aunque esto puede parecer simplemente una ruta más, se trata en realidad de un punto crítico donde se definen parámetros como el kernel predeterminado, las opciones de arranque y la experiencia del usuario desde el primer segundo de encendido.

Históricamente, para sistemas que aún utilizan tablas de particiones MBR, la partición de arranque (/boot) debía ubicarse antes del cilindro 1024 del disco. Esta limitación, aunque obsoleta en arquitecturas modernas, refleja cómo el diseño físico de los discos influyó en el comportamiento del software y aún hoy enseña lecciones sobre compatibilidad y planificación de infraestructura.

En sistemas más actuales con arranque UEFI, se hace necesario contar con una partición EFI (ESP) montada generalmente en /boot/efi. Este cambio marca un avance hacia mayor flexibilidad y seguridad en el arranque, pero también introduce nuevas responsabilidades para el administrador del sistema, quien debe asegurarse de que las configuraciones se mantengan coherentes con los requisitos del firmware moderno.

Cuando se trata de montar sistemas de archivos, el administrador debe distinguir entre montajes temporales —usualmente en /mnt— y dispositivos externos que se montan automáticamente en /media. Esta distinción no es solo una convención, sino una práctica que contribuye al orden, seguridad y automatización del entorno de trabajo.

En cuanto a la gestión de volúmenes lógicos (LVM), el conocimiento de sus componentes —como los extentos físicos (PE), que son la unidad mínima de asignación dentro de un grupo de volúmenes (VG)— permite una administración mucho más flexible y escalable del almacenamiento. El tamaño de un volumen lógico (LV), por ejemplo, no es arbitrario, sino que se calcula como la suma de los PE

asignados, lo que ofrece al administrador un control detallado sobre el uso eficiente del disco.

En esquemas de partición MBR, no es común —ni estándar— el uso de una partición EFI, lo que subraya la necesidad de adaptar estrategias según el entorno y el hardware disponibles. Finalmente, en términos de gestión de memoria, además del uso tradicional de particiones de intercambio, la creación de archivos de intercambio (swap files) se ha convertido en una solución práctica y rápida para ampliar la memoria virtual sin necesidad de redimensionar particiones.

Este enfoque integral no solo permite operar un sistema Linux con eficacia, sino que también promueve una visión crítica y estratégica sobre su arquitectura. El administrador no solo debe saber qué hacer, sino también por qué, anticipando posibles errores y optimizando cada aspecto del entorno de manera proactiva.

2.2 Instalar un gestor de arranque

Comprender el funcionamiento del gestor de arranque GRUB (GRand Unified Bootloader) es fundamental para todo administrador de sistemas Linux, ya que de él depende en gran medida la capacidad del sistema para iniciar correctamente. En su versión moderna, GRUB 2, la configuración principal se almacena en el archivo /boot/grub/grub.cfg. Sin embargo, este archivo no debe editarse directamente, ya que es generado automáticamente a partir de los parámetros definidos en /etc/default/grub y los scripts ubicados en /etc/grub.d/. Para que los cambios surtan efecto, es necesario ejecutar comandos como sudo update-grub o grub2-mkconfig, lo cual resalta una de las buenas prácticas en Linux: automatizar y estandarizar procesos críticos para evitar errores humanos.

Cuando se requiere añadir entradas personalizadas al menú de arranque, el archivo apropiado es /etc/grub.d/40_custom. Esta estrategia modular ofrece una gran ventaja: permite mantener separados los ajustes personalizados del resto del sistema, reduciendo el riesgo de sobreescribir configuraciones con actualizaciones.

En contraste, GRUB Legacy —el antecesor de GRUB 2— utilizaba un archivo mucho más simple y directo: /boot/grub/menu.lst. Aunque más fácil de editar a mano, este enfoque resultaba menos seguro y flexible, especialmente en entornos con múltiples kernels o configuraciones complejas. Además, GRUB Legacy usaba un sistema de numeración de discos y particiones distinto, por ejemplo, la segunda partición del primer disco se identificaba como (hd0,1). Este sistema podía generar confusiones, especialmente para quienes venían de trabajar con identificadores tradicionales como /dev/sda1.

Una de las características que se mantiene tanto en GRUB 2 como en GRUB Legacy es la posibilidad de acceder a la consola de GRUB presionando la tecla c desde el menú de arranque. Esta funcionalidad resulta crucial en situaciones de recuperación, ya que permite al administrador ejecutar comandos directamente para intentar iniciar manualmente el sistema o reinstalar el gestor de arranque.

Para realizar diagnósticos o tareas avanzadas, existen herramientas que permiten identificar con precisión las particiones y sus atributos. Por ejemplo, lsblk -f o sudo parted /dev/sda print ofrecen una vista clara de la estructura del disco, mientras que blkid permite visualizar los UUID de cada partición, lo cual es especialmente útil para configurar GRUB con identificadores únicos que garantizan una mayor estabilidad, incluso si el orden de los discos cambia.

Asimismo, parámetros básicos como el tiempo de espera del menú de GRUB se pueden ajustar modificando la variable GRUB_TIMEOUT en /etc/default/grub, lo cual mejora la experiencia del usuario al permitir más (o menos) tiempo para seleccionar un sistema operativo.

Desde la consola de GRUB Legacy, también es posible instalar manualmente el gestor en una partición específica usando comandos como root (hd1,0) seguido de setup (hd1,0), una técnica que aunque hoy en día es menos común, sigue siendo una herramienta útil en entornos con hardware antiguo o configuraciones no estándar.

A pesar de que GRUB 2 representa una evolución significativa en términos de modularidad, automatización y compatibilidad, también introduce una mayor complejidad que puede intimidar a usuarios menos experimentados. Sin embargo, esta complejidad es el precio de una mayor robustez y escalabilidad. El dominio de estas herramientas no solo permite personalizar el arranque del sistema de manera segura, sino que capacita al profesional en una de las habilidades más críticas de la administración Linux: la recuperación y gestión eficiente del proceso de arranque, que es frecuentemente donde se define la frontera entre un sistema operativo operativo y uno inaccesible.

El sistema de gestión de paquetes en Debian es potente y versátil, pero su efectividad depende en gran medida del conocimiento que tenga el administrador sobre las herramientas disponibles. Aunque comandos como dpkg ofrecen un control directo y bajo nivel sobre los paquetes, su uso indebido puede llevar a inconsistencias o roturas en el sistema. En cambio, las utilidades del ecosistema apt proporcionan mecanismos más seguros, al manejar automáticamente dependencias y realizar validaciones antes de aplicar cambios.

El equilibrio entre control manual y automatización representa un aspecto clave en la administración de sistemas GNU/Linux. Entender cuándo usar dpkg y cuándo recurrir a apt o herramientas complementarias como apt-file permite al profesional tomar decisiones informadas, reduciendo riesgos y optimizando el tiempo de resolución de problemas. Esta capacidad de análisis y selección estratégica es, sin duda, una de las competencias más valoradas en entornos de producción reales.

2.3 Gestión de librerías compartidas

En los sistemas GNU/Linux, las bibliotecas compartidas son esenciales para la ejecución eficiente de programas, ya que permiten reutilizar código común entre múltiples aplicaciones sin duplicarlo en cada binario. Comprender cómo se organizan

y gestionan estas bibliotecas es una competencia fundamental para cualquier administrador de sistemas.

Los nombres de las bibliotecas compartidas generalmente siguen una convención estructurada que incluye el nombre base, un sufijo (normalmente .so de shared object) y, en muchos casos, un número de versión. Esta nomenclatura facilita la gestión de versiones y garantiza que las aplicaciones carguen exactamente la versión que necesitan.

Para que el sistema reconozca nuevas rutas de bibliotecas personalizadas, se pueden crear archivos de configuración como mylib.conf, ubicados en /etc/ld.so.conf.d/. Allí se escriben las rutas absolutas de los directorios que contienen las bibliotecas, permitiendo que el sistema dinámico de carga (ld.so) las incluya en su búsqueda. Después de añadir nuevas rutas, es necesario ejecutar ldconfig para actualizar la caché de bibliotecas compartidas.

Una herramienta útil para diagnosticar qué bibliotecas requiere un binario es ldd, como en el caso de ldd /bin/kill. Este comando muestra una lista de dependencias, lo que permite verificar si las bibliotecas necesarias están disponibles o si hay vínculos rotos (broken links).

Asimismo, utilidades como objdump permiten realizar un análisis más profundo de un ejecutable, incluyendo la inspección del soname (el nombre simbólico que usa el enlazador para referenciar una biblioteca compartida) y las dependencias específicas. Por ejemplo, se puede usar objdump -p /lib/x86_64-linux-gnu/libc.so.6 para obtener información detallada sobre glibc, o analizar las dependencias de Bash, una de las herramientas más críticas del sistema.

La gestión correcta de las bibliotecas compartidas en sistemas Linux no solo contribuye a un uso más eficiente de los recursos, sino que también es clave para evitar errores de ejecución, mejorar la seguridad y asegurar la interoperabilidad entre aplicaciones. Sin embargo, este proceso también puede volverse complejo, especialmente en sistemas con múltiples versiones de bibliotecas o en entornos donde se desarrollan e instalan aplicaciones de forma manual.

Es por esto que los profesionales deben desarrollar una comprensión profunda tanto de los conceptos como de las herramientas involucradas, y mantener buenas prácticas de documentación y control de versiones. El uso de herramientas como ldd y objdump permite auditar el sistema de forma efectiva, anticipando problemas y permitiendo una resolución proactiva. Además, entender cómo Linux localiza e interpreta estas bibliotecas da al administrador la capacidad de personalizar el entorno y mejorar el rendimiento del sistema de acuerdo con las necesidades específicas de cada entorno corporativo o de desarrollo.

2.4 Gestión de paquetes Debian

En el ecosistema de gestión de paquetes de Debian y sus derivados, el uso adecuado de herramientas como dpkg y apt es esencial para mantener un sistema estable y funcional. Uno de los comandos más básicos es dpkg -i package.deb, que permite instalar directamente un paquete .deb descargado manualmente. Sin embargo, esta acción no resuelve automáticamente las dependencias, por lo que debe usarse con cuidado.

Para gestionar paquetes instalados, dpkg -r permite eliminar un paquete, aunque no borra los archivos de configuración. Si se desea una eliminación forzada —por ejemplo, en casos donde existen conflictos de dependencias— puede utilizarse dpkg --force-depends -r, aunque este comando debe emplearse con extremo cuidado, ya que puede comprometer la estabilidad del sistema.

Por otro lado, el comando dpkg-query resulta útil para inspeccionar el estado de los paquetes, como identificar cuál contiene un archivo determinado. Sin embargo, para una solución más potente y flexible, se recomienda el uso de apt-file, una herramienta que permite buscar qué paquete proporciona un archivo específico, incluso si este no está instalado en el sistema. Por ejemplo, apt-file search <archivo> puede ser clave para localizar bibliotecas o archivos de cabecera necesarios para compilar software.

Cuando se trabaja con repositorios, se pueden añadir nuevas fuentes de paquetes modificando el archivo /etc/apt/sources.list, como es el caso al incluir repositorios de paquetes fuente. Este paso debe acompañarse del comando sudo apt update, que actualiza el índice de paquetes y garantiza que el sistema tenga acceso a las versiones más recientes disponibles.

Otra herramienta indispensable es apt-cache show, que ofrece información detallada sobre cualquier paquete, incluyendo su descripción, dependencias, versión y repositorio. Este comando resulta particularmente útil para evaluar si un paquete es adecuado antes de instalarlo.

2.5 Gestión de paquetes RPM y YUM

En los sistemas Linux basados en RPM (Red Hat Package Manager), como Red Hat, CentOS, Fedora, o SUSE, la gestión de paquetes es una tarea central para garantizar el buen funcionamiento del sistema. Cada administrador debe dominar una serie de herramientas que permiten instalar, actualizar, eliminar y auditar los paquetes de software disponibles en el sistema.

El comando rpm -ivh se utiliza para instalar paquetes RPM de forma manual, mostrando además una barra de progreso útil para el seguimiento visual del proceso. Por su parte, el uso de rpm -qf permite identificar a qué paquete pertenece un archivo dado, lo que es clave al investigar conflictos o archivos modificados.

En cuanto a la gestión de actualizaciones, herramientas como yum check-update o dnf check-update permiten verificar si existen nuevas versiones de los paquetes instalados, ofreciendo así una vía sencilla para mantener el sistema actualizado y protegido contra vulnerabilidades conocidas. Para sistemas SUSE, el comando zypper mr -d permite deshabilitar repositorios específicos, útil para evitar conflictos con fuentes no deseadas o temporales.

La organización de los repositorios también es un aspecto clave. En sistemas que utilizan DNF, como Fedora y CentOS

modernos, los archivos .repo se colocan en /etc/yum.repos.d/. Estos archivos definen los repositorios desde los cuales DNF puede descargar paquetes. Si es necesario agregar nuevos repositorios de forma dinámica, se puede usar dnf config-manager --add-repo.

El diagnóstico de qué paquete proporciona un archivo específico se puede realizar con comandos como yum whatprovides, zypper se --provides o apt-file search en sistemas basados en Debian. Esto permite identificar rápidamente dependencias faltantes o resolver problemas de instalación.

Para obtener un listado completo de los paquetes instalados en el sistema, se pueden usar dnf list --installed o zypper se -i, lo cual resulta útil para auditorías, respaldos o validaciones de entorno.

La correcta administración de paquetes es fundamental para garantizar la estabilidad, la seguridad y el rendimiento de los sistemas Linux. Si bien existen múltiples herramientas como RPM, DNF, YUM y Zypper, cada una con su propio conjunto de comandos y sintaxis, todas convergen en un mismo objetivo: facilitar el ciclo de vida del software en entornos empresariales o personales.

El desafío para los administradores no está solo en aprender comandos, sino en entender el contexto en el que cada uno se aplica, cómo interactúan entre sí los repositorios y qué implicaciones puede tener una mala gestión de dependencias o fuentes de software.

Dominar estas herramientas permite no solo reaccionar ante errores, sino anticiparse a ellos, mantener entornos estables y responder con agilidad ante cambios, parches de seguridad o despliegues de nuevas aplicaciones. En entornos productivos, esta competencia marca la diferencia entre una infraestructura robusta y una vulnerable a fallos por configuraciones deficientes o dependencias mal gestionadas.

2.6 Linux como sistema virtualizado

En el contexto actual de las tecnologías de virtualización, contar con un hardware que soporte extensiones como Intel VT-x o AMD-V es esencial para lograr una virtualización completa y eficiente. Estas extensiones permiten ejecutar máquinas virtuales con un rendimiento muy cercano al de un sistema operativo nativo, habilitando la virtualización de hardware asistida directamente por el procesador. Su disponibilidad en la BIOS o UEFI se ha vuelto un requisito clave para entornos de laboratorio, desarrollo y producción.

Para escenarios donde el rendimiento es una prioridad absoluta, se puede optar por alternativas como la virtualización por contenedores (por ejemplo, usando Docker o LXC), o incluso utilizar hipervisores de tipo 1 (bare-metal) como VMware ESXi o Proxmox en modo nativo. Estas opciones reducen significativamente la sobrecarga del sistema, ya que interactúan directamente con el hardware, lo cual maximiza la eficiencia y la escalabilidad.

Un aspecto menos visible, pero crítico al clonar máquinas virtuales, es el manejo de identificadores únicos. Herramientas como dbus-uuidgen permiten verificar o regenerar los UUID de D-Bus en las máquinas clonadas, previniendo conflictos en redes o servicios donde se exige la unicidad de identificación entre nodos.

La Configuración NAT (Network Address Translation) es un componente fundamental en entornos virtualizados, ya que permite a las máquinas virtuales comunicarse con el exterior usando una sola dirección IP pública o una dirección de salida compartida. En muchos entornos de prueba y desarrollo, NAT se utiliza para aislar redes internas mientras se mantiene conectividad con internet, reduciendo así riesgos de exposición directa y simplificando la gestión de redes.

El uso correcto de NAT es especialmente relevante cuando se combinan técnicas de virtualización completas con redes privadas virtuales (VPN) o cuando se implementan soluciones de firewall y seguridad perimetral, como en el caso de proyectos con Endian Firewall o pfSense. Configurar correctamente la traducción de direcciones IP no solo garantiza la conectividad, sino que también permite implementar reglas de control de tráfico, reenvío de puertos y segmentación lógica de la red.

Desde una perspectiva de formación en sistemas operativos open source, comprender la configuración de NAT en entornos virtualizados fortalece competencias clave en redes, seguridad y despliegue de servicios, lo que prepara al estudiante para enfrentar desafíos reales en entornos corporativos y en la administración de infraestructuras TI modernas.

3 Temática 2: Configuración NAT

3.1 Producto esperado:

Previo al desarrollo de la Temática 2, se procedió con la descarga, instalación y configuración de la distribución GNU/Linux Endian Firewall (EFW), una solución robusta y de código abierto diseñada para implementar funciones de firewall, enrutamiento, VPN, y filtrado de contenido en entornos corporativos. Esta distribución se eligió por su interfaz amigable y sus potentes capacidades de configuración de reglas de red, lo cual la convierte en una herramienta ideal para fines educativos y simulaciones de seguridad perimetral.

Una vez instalado el sistema y configuradas las zonas de red (verde para LAN, naranja para DMZ, y roja para WAN), se procedió a configurar reglas de NAT (Network Address Translation) con el objetivo de demostrar la capacidad de EFW para establecer comunicación entre distintas zonas:

- Comunicación LAN hacia la WAN: Se implementó una regla de NAT Masquerading, permitiendo que todos los dispositivos conectados a la zona LAN (verde) pudieran acceder a la red WAN (simulada como Internet) utilizando la IP pública del firewall. Esta configuración es clave para garantizar la conectividad de los usuarios internos sin exponer directamente sus direcciones IP reales al exterior. La funcionalidad fue verificada mediante pruebas de conectividad (ping, navegación) desde clientes en la LAN.
- Comunicación DMZ hacia la WAN: Para permitir que servicios expuestos en la zona DMZ (naranja) también

tengan acceso a Internet —por ejemplo, para actualizaciones de sistema o para permitir salida de tráfico controlado— se creó una segunda regla de NAT Masquerading específica para la interfaz de la DMZ. Esta medida permite mantener segmentados los servicios públicos, pero brindándoles la conectividad necesaria sin comprometer la red LAN.

- Reenvío de puertos (Port Forwarding) / Verificación de Reglas NAT: Finalmente, se implementó y verificó el reenvío de puertos (Port Forwarding), creando reglas específicas en EFW que permiten acceder desde la WAN a servicios alojados en la DMZ, como un servidor web o SSH. Este paso es crucial para exponer servicios de manera controlada y segura, estableciendo las reglas necesarias en el apartado de Firewall > Port Forwarding.

La verificación de estas configuraciones se realizó mediante herramientas de escaneo y monitoreo de tráfico, asegurando que las reglas estuvieran activas, funcionando correctamente, y aplicadas a las interfaces correctas.

3.2 Pasos:

Como parte del proceso de implementación de la solución de firewall perimetral, se realizó la descarga de la imagen ISO de Endian Firewall (EFW) directamente desde su sitio oficial: https://www.endian.com. Esta imagen, compatible con arquitectura de 64 bits (x64), fue utilizada para crear una máquina virtual en el entorno de VirtualBox, herramienta elegida por su facilidad de uso y compatibilidad con múltiples sistemas operativos.

Durante la creación de la máquina virtual, se definieron parámetros esenciales como la cantidad de memoria RAM, tamaño del disco duro virtual y tipo de red. La ISO se montó como disco de arranque, permitiendo iniciar el proceso de instalación del sistema operativo.

Uno de los aspectos críticos en esta fase es la configuración inicial del EFW, particularmente la asignación de direcciones IP para las interfaces de red. Este paso requiere especial atención, ya que de él dependerá el correcto enrutamiento y segmentación de las zonas LAN, DMZ y WAN en etapas posteriores. Se recomienda asignar IPs fijas y bien documentadas desde el inicio, asegurando una topología clara y reproducible.

Al concluir la instalación, el sistema solicita un reinicio. En este punto, se configuró el adaptador de red de la máquina virtual en modo "Adaptador puente" (bridged adapter), lo cual permite que el firewall obtenga acceso directo a la red física del host. Esta configuración es fundamental para simular un entorno realista donde el EFW actúe como un dispositivo perimetral dentro de una infraestructura de red.

Esta fase no solo representa la base para las configuraciones avanzadas que se realizarán posteriormente (como las reglas de NAT y filtrado de tráfico), sino que también permite al estudiante desarrollar competencias prácticas en la instalación y puesta en marcha de soluciones de seguridad Open Source, reforzando el enfoque autónomo y técnico del diplomado.

3.3 Configuración:

Una vez completada la instalación del sistema Endian Firewall (EFW), es fundamental verificar el estado de las interfaces de red para asegurar una correcta asignación de zonas y garantizar el funcionamiento esperado del firewall. Para ello, se emplea el comando ip addr show, que permite observar en tiempo real las interfaces activas del sistema y validar que cada una esté correctamente asociada a su zona correspondiente: GREEN (LAN), ORANGE (DMZ) y RED (WAN).

Posteriormente, se procede a habilitar el reenvío de paquetes IP (IP forwarding), una función esencial para permitir la comunicación entre redes a través del firewall. De manera temporal, se habilita ejecutando echo 1 > /proc/sys/net/ipv4/ip_forward, y para que esta configuración persista tras reinicios, se modifica el archivo /etc/sysctl.conf, asegurando que la línea net.ipv4.ip_forward = 1 esté presente y activa. El cambio se aplica con sysctl -p.

Con la conectividad interzonas activada, se implementan las reglas de NAT (Network Address Translation) utilizando iptables. Para permitir la salida a Internet desde la LAN hacia la WAN, se utiliza la técnica de Masquerading con la siguiente regla:

```
iptables -t nat -A POSTROUTING -o eth2 -s 192.168.1.0/24 -j MASQUERADE
```

De igual forma, se configura la salida desde la zona DMZ hacia la WAN:

```
iptables -t nat -A POSTROUTING -o eth2 -s 189.89.10.0/24 -j MASQUERADE
```

Estas reglas aseguran que las máquinas internas puedan comunicarse con el exterior, ocultando sus direcciones IP privadas mediante la IP pública del firewall.

Para validar la correcta aplicación de estas reglas, se usan los comandos:

```
iptables -t nat -L -n -v (para reglas de NAT)

iptables -L -n -v (para reglas de reenvío de tráfico)
```

La persistencia de estas reglas es un aspecto crítico. Aunque EFW posee scripts propios para gestionar reglas de firewall, se puede forzar una carga automática estándar mediante el siguiente procedimiento:

```
iptables-save > /etc/iptables.rules
echo "iptables-restore < /etc/iptables.rules" >> /etc/rc.local
```

Es importante tener en cuenta que los scripts del propio EFW podrían sobrescribir estas reglas en futuros reinicios, por lo que se recomienda documentar cuidadosamente cualquier modificación manual y, si es posible, integrar estas reglas en la interfaz de configuración web de EFW para asegurar su permanencia.

Finalmente, se realiza una prueba de conectividad desde estaciones dentro de las zonas LAN y DMZ, utilizando herramientas básicas como ping, curl y date, lo que permite comprobar tanto la salida a Internet como la correcta configuración de fecha y hora del sistema:

```
ping 8.8.8.8 -c 4
curl ifconfig.me
date
```

Estas pruebas no solo confirman la operatividad de la red, sino que también permiten detectar posibles errores de configuración en las interfaces o en las reglas de NAT.

4 Conclusiones.

La implementación y administración de entornos basados en GNU/Linux representan una gran oportunidad para las organizaciones que buscan maximizar la flexibilidad, mejorar la seguridad y optimizar el uso de sus recursos tecnológicos. A lo largo de este trabajo, hemos evidenciado que GNU/Linux no es solo una alternativa viable frente a sistemas propietarios, sino un ecosistema vivo, adaptable y en constante evolución, ideal para empresas que requieren escalabilidad y estabilidad.

El análisis de las distribuciones más comunes en servidores y sus aplicaciones ha permitido entender cómo el software libre puede integrarse de manera efectiva en diferentes contextos productivos, fomentando la autonomía tecnológica y una considerable reducción en costos por licenciamiento. Sin embargo, es importante reconocer que esta integración también implica retos, como la necesidad de capacitación continua y la gestión de soporte técnico especializado, aspectos que las organizaciones deben considerar para aprovechar al máximo estas tecnologías.

Los casos prácticos de adopción regional muestran un impacto positivo en la eficiencia operativa y la seguridad informática, pero también resaltan la importancia de una estrategia clara y acompañamiento técnico para una transición exitosa. Finalmente, el desarrollo de habilidades prácticas en instalación y administración no solo fortalece la formación del estudiante, sino que lo prepara para enfrentar con mayor confianza los desafíos que implica gestionar infraestructuras TI modernas.

Este enfoque integral no solo enriquece la formación académica, sino que también posiciona al futuro ingeniero de sistemas como un agente activo en la transformación digital de las organizaciones, capaz de aportar soluciones innovadoras y sostenibles en un entorno tecnológico dinámico.

5 REFERENCIAS

[1] LPI Linux Essentials. (2022). Tema 1: La Comunidad Linux y una carrera en el mundo del código abierto. https://learning.lpi.org/es/learning-materials/010-160/1/

[2] LPI Linux Essentials.(2022). Tema 2: Encontrando el camino en un sistema Linux. https://learning.lpi.org/es/learning-materials/010-160/2/

[3] LPI Linux Essentials.(2022). Tema 3: El poder de la línea de comandos. https://learning.lpi.org/es/learning-materials/010-160/3/

[4] LPI Linux Essentials.(2022). Tema 4: El sistema operativo Linux. https://learning.lpi.org/es/learning-materials/010-160/4/

[5] LPI Linux Essentials.(2022). Tema 5: Seguridad y sistema de permisos de archivos. https://learning.lpi.org/es/learning-materials/010-160/5/

[6] Free Software Foundation (2016). Software Libre y educación. El sistema operativo GNU. http://www.gnu.org/education/education.html

[7] Canonical (2023). Guía del Ubuntu desktop 20.04 LTS. Help Ubuntu. https://help.ubuntu.com/20.04/ubuntu-help/index.html

[8] Debian (2023). El manual del administrador de Debian 12.5.0. Debian https://www.debian.org/releases/stable/amd64/index.es.html

[9] Oracle (2020), Manual de usuario VirtualBox. VirtualBox. https://www.virtualbox.org/manual/

Guzman, D. A. (2017). OVI Unidad I_Nivelacion. [Objeto_virtual_de_información_OVI]. Repositorio Institucional UNAD. http://hdl.handle.net/10596/10570