Linux Networking Commands

Agostina A. Ortellado, Juan M. Portillo, Gastón O. Rodríguez, Dayana K. Romero[1]

[1]Facultad de Ciencias Exactas, Naturales y Agrimensura, Universidad Nacional del Nordeste. Corrientes, Argentina.
ortelladoagos@gmail.com, marcelo160891@gmail.com, gastonrodriguez401@gmail.com, karinroom91@gmail.com

Abstract. Linux is widely used in server administration, network infrastructure and Internet of Things (IoT) environments, where command-line tools play a fundamental role in network configuration, monitoring and troubleshooting. This paper presents a didactic study of the main Linux networking commands, organizing them into functional groups according to their role in connectivity testing, interface management, routing, port and service analysis, and domain name resolution. In addition to describing their operation, the study discusses their practical applications, selection criteria and the evolution from traditional utilities to the modern iproute2 suite. To illustrate their use, a case study was carried out in a home network (192.168.0.0/24), where a Smart TV was analyzed as a black-box device. The proposed procedure combines host discovery, connectivity verification and service identification to characterize the target device using standard Linux utilities and Nmap. The results show that the systematic use of these tools enables effective network diagnosis and accurate device identification while promoting a structured methodology for troubleshooting. The proposed approach also provides an educational framework that supports the development of practical skills in Linux networking and system administration.

Keywords: Linux; network administration; network diagnostics; networking commands; Nmap.

Comandos de Red de Linux

Resumen. Linux constituye una de las principales plataformas para la administración de servidores, infraestructuras de red y entornos del Internet de las Cosas (IoT), donde las herramientas de línea de comandos desempeñan un papel fundamental en la configuración, monitoreo y diagnóstico de redes. El presente trabajo analiza las principales herramientas de red disponibles en Linux, organizándolas según su función en tareas de verificación de conectividad, administración de interfaces, enrutamiento, análisis de puertos y servicios, y resolución de nombres. Además de describir su funcionamiento, se presentan criterios

para su utilización y se analiza la evolución desde las utilidades tradicionales hacia el conjunto moderno iproute2. Como aplicación práctica, se desarrolla un caso de estudio en una red doméstica (192.168.0.0/24), donde un Smart TV es analizado como una "caja negra". El procedimiento combina el descubrimiento de hosts, la verificación de conectividad y la identificación de servicios mediante herramientas estándar de Linux y Nmap. Los resultados demuestran que la utilización sistemática de estas herramientas permite realizar diagnósticos de red e identificar dispositivos de manera efectiva, al tiempo que proporciona una metodología de trabajo reproducible con fines didácticos para la formación en administración de sistemas y redes.

Palabras clave: Linux; administración de redes; diagnóstico de redes; comandos de red; Nmap.

1 Introducción

Linux se ha consolidado como una de las principales plataformas para la administración de servidores, infraestructuras de red y dispositivos embebidos, debido a su estabilidad, flexibilidad y amplio conjunto de herramientas orientadas al diagnóstico y la gestión de redes (Bautts et al., 2005; Siever et al., 2009). En este contexto, las utilidades de línea de comandos constituyen recursos fundamentales para verificar la conectividad, inspeccionar interfaces, analizar rutas, identificar servicios y resolver problemas de comunicación de manera precisa y reproducible.

El dominio de estas herramientas resulta especialmente relevante en la formación de administradores de sistemas y profesionales de redes, ya que favorece la comprensión del funcionamiento de los protocolos de comunicación y el desarrollo de metodologías sistemáticas para el diagnóstico de incidentes. Asimismo, el empleo de comandos especializados permite interpretar el comportamiento de la infraestructura de red desde diferentes niveles del modelo TCP/IP, facilitando la identificación de fallas y la toma de decisiones durante las tareas de administración (Stallings, 2004; Kurose y Ross, 2017).

Con este enfoque, el presente trabajo organiza las principales herramientas de red de Linux según la función que desempeñan dentro del proceso de diagnóstico: conectividad, configuración e inspección de interfaces, enrutamiento, análisis de puertos y conexiones, y resolución de nombres. Además de describir su utilización, se incorporan fundamentos teóricos, criterios de selección y una comparación entre herramientas tradicionales y alternativas modernas, como el conjunto iproute2, actualmente recomendado para la administración de redes en sistemas Linux (Siever et al., 2009; The Linux Foundation, 2023).

Finalmente, se presenta un caso práctico desarrollado en una red doméstica bajo el segmento 192.168.0.0/24, cuyo propósito es demostrar un procedimiento reproducible para el descubrimiento, análisis e identificación de un Smart TV tratado como una "caja negra". Mediante la utilización de herramientas estándar de Linux y de Nmap, el

procedimiento permite localizar el dispositivo, verificar su conectividad, analizar sus servicios e identificar el fabricante de su interfaz de red a partir del identificador único de organización (OUI), ilustrando un escenario representativo de diagnóstico y administración de redes (Lyon, 2009; Nmap Project, s.f.).

2 Instrumentación

Para el desarrollo del presente trabajo se implementó un entorno de pruebas virtualizado, con el objetivo de ejecutar y analizar los comandos de red de Linux en un entorno controlado y reproducible.

2.1 Requisitos previos

Antes de comenzar, se requiere disponer de los siguientes recursos y condiciones mínimas:

- Acceso a la línea de comandos o terminal de Linux.
- Una cuenta de usuario con privilegios de sudo (administrador).
- Conexión de red activa y funcional para realizar pruebas de conectividad.

2.2 Plataforma y configuración del entorno

Las pruebas se realizaron en un equipo anfitrión con sistema operativo Windows 11 (64 bits), procesador Intel Core i5, 32 GB de memoria RAM y 256 GB de espacio disponible en disco.

Sobre este host se utilizó el hipervisor Oracle VirtualBox 7.0.20, configurando una máquina virtual (VM) con Ubuntu 22.04.5 LTS (64 bits) como sistema operativo invitado.

La red de la máquina virtual fue configurada en modo Adaptador puente (Bridged Adapter) con la opción Cable connected habilitada, lo que permite que la VM se ubique en la misma subred local del entorno doméstico.

El esquema de red de ejemplo corresponde a 192.168.0.0/24, con puerta de enlace y servicio DHCP en 192.168.0.1.

2.3 Software utilizado

En la máquina virtual se instalaron las herramientas esenciales de red mediante los siguientes comandos:

```
sudo apt update
sudo apt install -y iproute2 traceroute net-tools nmap tcpdump
```

El conjunto iproute2 incluye las utilidades modernas ip y ss, recomendadas para la gestión de interfaces y conexiones. Por motivos de comparación histórica, también se incluyeron los comandos tradicionales ifconfig y netstat (provistos por el paquete net-tools), aunque se encuentran en desuso en distribuciones recientes.

2.4 Herramientas empleadas en el caso práctico

Los principales comandos empleados durante el desarrollo del caso práctico se agrupan según la función que cumplen dentro del proceso de diagnóstico de red:

- **Configuración e inspección de interfaces:** ip addr → permite visualizar las direcciones IP configuradas en las interfaces de red y verificar su estado operativo. Forma parte del paquete iproute2, recomendado para la administración de redes en sistemas Linux modernos.
- **Diagnóstico de conectividad:** ping y traceroute → verifican la disponibilidad del dispositivo objetivo y permiten comprobar el recorrido que siguen los paquetes hasta alcanzar el destino, facilitando la detección de problemas de conectividad y enrutamiento.
- **Descubrimiento de dispositivos:** nmap -sn → realiza un escaneo de la subred para identificar los hosts activos sin efectuar un análisis de puertos, permitiendo localizar el dispositivo objetivo.
- **Identificación de servicios y dispositivos:** nmap y sudo nmap -O → permiten analizar el estado de los puertos más comunes, obtener información adicional del dispositivo y, cuando la ejecución se realiza con privilegios elevados, identificar la dirección MAC y el fabricante de la interfaz de red mediante el identificador único de organización (OUI), contribuyendo a la caracterización del dispositivo analizado.

2.5 Consideraciones sobre distribuciones

Si bien las pruebas se realizaron en Ubuntu 22.04, los comandos descritos son compatibles con la mayoría de las distribuciones Linux modernas, incluyendo Debian, Fedora, CentOS, y OpenSUSE.

Algunas variaciones menores pueden presentarse en la disponibilidad de paquetes o rutas de instalación, pero la sintaxis de las herramientas de red se mantiene consistente en todas ellas, lo que garantiza la reproducibilidad de los resultados obtenidos.

3 Desarrollo

El diagnóstico y la administración de redes constituyen actividades fundamentales para garantizar el correcto funcionamiento de los sistemas informáticos. En Linux, estas tareas se realizan principalmente mediante herramientas de línea de comandos, las cuales permiten inspeccionar interfaces, verificar la conectividad, analizar rutas e identificar servicios. La filosofía heredada de Unix favorece la utilización de utilidades especializadas y combinables, permitiendo construir procedimientos de diagnóstico reproducibles y automatizables (Bautts et al., 2005; Siever et al., 2009). Además, estas herramientas contribuyen a la formación en administración de sistemas y redes al facilitar la comprensión de los protocolos de comunicación y los métodos sistemáticos de resolución de problemas (Kurose y Ross, 2017; Cherry Servers, 2023).

3.1 Comandos de diagnóstico y conectividad

La verificación de la conectividad constituye la primera etapa del diagnóstico de redes. Mediante el empleo de ICMP y mecanismos asociados al campo TTL del protocolo IP, es posible comprobar la accesibilidad de un host y analizar el recorrido seguido por los paquetes (Stallings, 2004; Kurose y Ross, 2017).

ping

Utiliza mensajes ICMP Echo Request y Echo Reply para verificar la disponibilidad de un host y obtener información sobre latencia y pérdida de paquetes, proporcionando una primera aproximación al estado de la conexión (Siever et al., 2009).

Ejemplo: ping 192.168.0.164

traceroute / tracepath

Permiten identificar los routers intermedios recorridos por los paquetes y detectar problemas de enrutamiento o congestión de la red (Bautts et al., 2005; Kurose y Ross, 2017).

3.2 Comandos de configuración e inspección de interfaces

Las interfaces de red constituyen el punto de conexión entre el sistema y el medio de transmisión, por lo que su correcta configuración resulta esencial para garantizar la comunicación (Stallings, 2004).

ifconfig

Herramienta tradicional utilizada para visualizar y modificar parámetros de las interfaces de red. Aunque continúa disponible en algunas distribuciones, actualmente se considera una utilidad heredada (Siever et al., 2009).

ip addr / ip link

Pertenecen al paquete iproute2 y representan la alternativa recomendada para administrar direcciones y enlaces de red, ofreciendo soporte para funcionalidades avanzadas e IPv6 (The Linux Foundation, 2023).

Ejemplo: ip addr show

3.3 Comandos de enrutamiento

El enrutamiento permite determinar el camino que seguirán los paquetes hasta su destino mediante tablas que contienen información sobre redes y puertas de enlace (Stallings, 2004; Kurose y Ross, 2017).

route

Herramienta tradicional para visualizar y modificar la tabla de enrutamiento. Actualmente ha sido reemplazada por las utilidades de iproute2 (Siever et al., 2009).

ip route show

Constituye la alternativa recomendada para administrar rutas en Linux modernos, proporcionando una interfaz más flexible y compatible con IPv6 (The Linux Foundation, 2023).
Ejemplo: ip route show

La información obtenida permite identificar la puerta de enlace predeterminada y diagnosticar problemas de conectividad (Cherry Servers, 2023).

3.4 Comandos de análisis de puertos y conexiones

Los protocolos TCP y UDP utilizan puertos para identificar servicios y permitir múltiples comunicaciones simultáneas (Kurose y Ross, 2017). El análisis de puertos y conexiones resulta fundamental para tareas de administración y seguridad (Stallings, 2004).

netstat

Permite visualizar conexiones activas y puertos abiertos, aunque actualmente es considerada una herramienta heredada (Siever et al., 2009).

ss

Forma parte del paquete iproute2 y proporciona información detallada sobre sockets y conexiones activas con un mejor rendimiento (The Linux Foundation, 2023).

nmap

Herramienta especializada en descubrimiento de hosts e identificación de servicios mediante distintas técnicas de escaneo. Resulta ampliamente utilizada para inventarios y tareas de auditoría (Lyon, 2009; Nmap Project, s.f.).

3.5 Comandos de resolución de nombres (DNS)

El sistema DNS permite asociar nombres de dominio con direcciones IP, simplificando el acceso a los recursos de red (Stallings, 2004). Ante problemas de acceso a servicios, la verificación de la resolución de nombres constituye una tarea habitual (Kurose y Ross, 2017).

nslookup

Permite realizar consultas básicas a servidores DNS y verificar la resolución de nombres (Siever et al., 2009).

dig

Proporciona información más detallada sobre las consultas realizadas y es ampliamente utilizado para tareas de diagnóstico (Bautts et al., 2005).

whois

Permite obtener información administrativa asociada a dominios registrados, complementando las tareas de análisis (Siever et al., 2009).

La utilización conjunta de estas herramientas facilita la identificación de problemas asociados con la resolución de nombres (RedesZone, 2021).

3.6 Análisis comparativo y criterios de utilización

Las herramientas de red en Linux deben considerarse complementarias, ya que cada una proporciona información específica y responde a necesidades particulares (Bautts et al., 2005). Utilidades como ping y traceroute permiten realizar un diagnóstico inicial de conectividad, aunque su alcance se limita al nivel de red y puede verse afectado por políticas de filtrado (Kurose y Ross, 2017).

Las herramientas tradicionales ifconfig, route y netstat han sido progresivamente reemplazadas por el paquete iproute2, el cual ofrece una interfaz unificada y soporte para tecnologías modernas (Siever et al., 2009; The Linux Foundation, 2023). Por su parte, Nmap proporciona información más detallada sobre servicios y dispositivos, aunque los resultados pueden verse condicionados por mecanismos de seguridad y requieren un uso responsable en entornos autorizados (Lyon, 2009; Nmap Project, s.f.).

La selección de una herramienta depende del problema a analizar y del nivel de detalle requerido. En general, el diagnóstico debe realizarse de forma progresiva: primero se verifica la conectividad, luego se inspeccionan interfaces y rutas, y finalmente se analizan servicios y recursos disponibles (Bautts et al., 2005). Desde una perspectiva formativa, el aprendizaje de estas herramientas favorece la comprensión de los protocolos de red y el desarrollo de competencias aplicables a la administración de sistemas, infraestructura, ciberseguridad y automatización (Stallings, 2004; Kurose y Ross, 2017).

4 Caso práctico: Análisis de conectividad en un Smart TV

El procedimiento para el descubrimiento de dispositivos en la infraestructura se desarrolla mediante la ejecución de comandos de red en un entorno doméstico bajo el segmento 192.168.0.0/24. El objetivo consiste en analizar un Smart TV, abordado como 'caja negra', para demostrar cómo un administrador puede identificar su huella y servicios desde un cliente Linux.

4.1 Descubrimiento

En este caso, la ejecución del análisis la realiza un cliente Linux que desconoce la dirección IP del dispositivo objetivo. Se inicia con el comando nmap -sn 192.168.0.0/24 para realizar un sondeo de la red (ver Fig. 1).

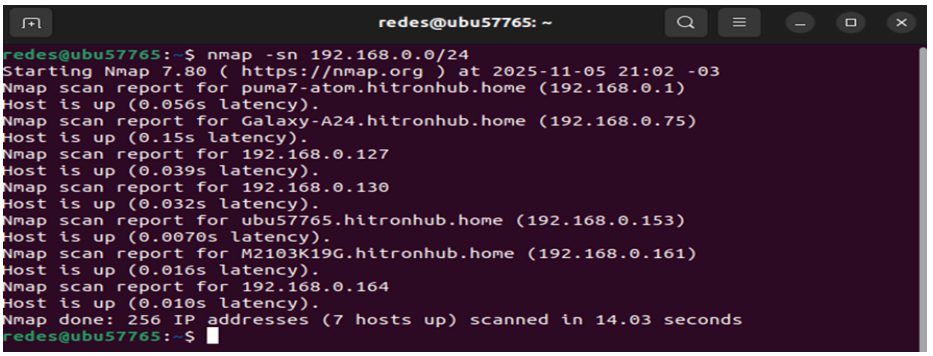

Fig. 1. Escaneo de red con Nmap para el descubrimiento de hosts activos en el segmento 192.168.0.0/24.

Podemos observar cómo la herramienta obtiene información de todos los hosts activos. Si bien la mayoría son identificables por sus nombres, el resultado revela 3 IPs sin descripción. De esta forma, se selecciona la IP 192.168.0.164 para continuar con el análisis.

4.2 Verificación de Conectividad y Ruta

A continuación, se lleva a cabo la ejecución del comando ping 192.168.0.164 para confirmar que el dispositivo está activo y respondiendo. Inmediatamente después, se utiliza traceroute 192.168.0.164 para verificar la ruta. Es posible observar que la interacción define un único salto, correspondiente al destino, validando que pertenece al mismo segmento de red (ver Fig. 2 y Fig. 3).

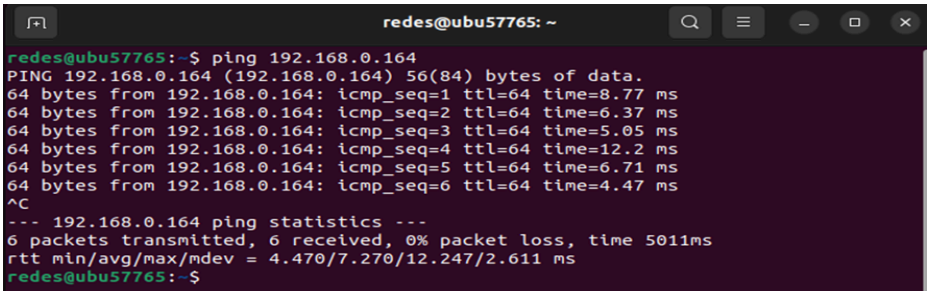

Fig. 2. Verificación de latencia y estado del host mediante el comando ping.

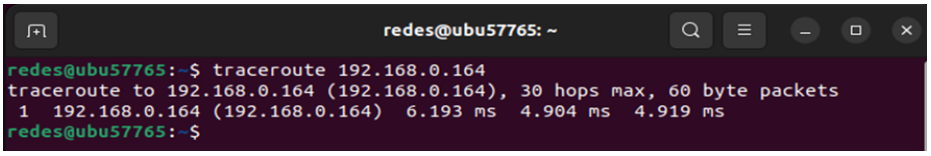

Fig. 3. Identificación de saltos de red y topología hacia el destino con traceroute.

4.3 Verificación de Servicios e Identidad

Como último paso, se procede a la identificación del dispositivo. Mediante la ejecución de nmap 192.168.0.164, se obtiene una respuesta que indica que los 1000 puertos más comunes se encuentran cerrados (ver Fig. 4).

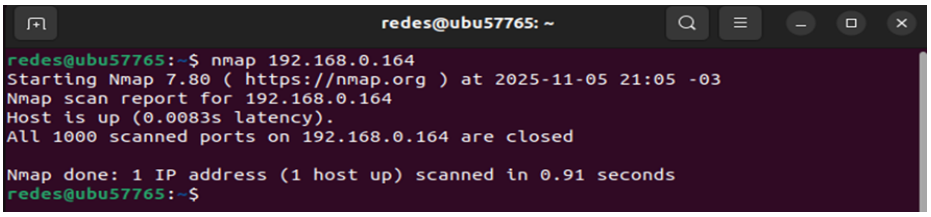

Fig. 4. Escaneo de puertos estándar (TCP) con Nmap sobre el dispositivo objetivo.

Teniendo en cuenta este resultado, se realiza una verificación con mayores privilegios mediante sudo nmap -O 192.168.0.164. Gracias a esta ejecución, la aplicación obtiene la dirección MAC del dispositivo (40:CD:7A:27:4F:59). Mediante esta verificación que se obtuvo de Nmap, se identifica al fabricante como "Qingdao Hisense Communications", permitiendo así confirmar la identidad del Smart TV (ver Fig. 5).

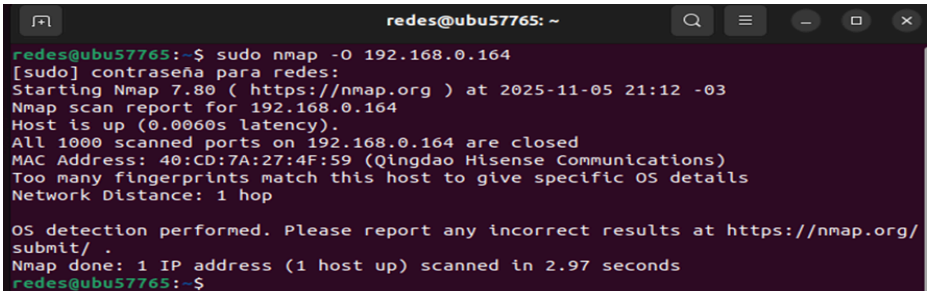

Fig. 5. Detección de dirección MAC y fabricante mediante el uso de Nmap con privilegios de superusuario.

4.4 Discusión de resultados

El caso práctico expuesto verifica la identificación exitosa del dispositivo objetivo basándose en el análisis de puertos y la correlación del identificador único de organización (OUI) de su dirección MAC, identificando efectivamente al fabricante. No obstante, la extrapolación de este procedimiento a infraestructuras heterogéneas requiere considerar escenarios alternativos y limitaciones inherentes a las técnicas de descubrimiento pasivo y activo.

Limitaciones del procedimiento: La eficacia del escaneo con herramientas como Nmap para la obtención de direcciones físicas (MAC) y huellas del sistema operativo se circunscribe al mismo dominio de colisión o segmento de red local de Capa 2 (Lyon, 2009). Si el dispositivo "caja negra" se encontrara en un segmento aislado por un enrutador o detrás de un firewall que bloquee el protocolo ICMP, la visibilidad se reduciría drásticamente, mostrando únicamente la dirección MAC del equipo intermedio o gateway (Kurose y Ross, 2017).

Análisis de posibles falsos positivos: La técnica de correlación OUI/MAC permite descubrir al fabricante de la interfaz de red (NIC), pero no garantiza la identificación del integrador final del dispositivo. En el ecosistema IoT, es frecuente que los componentes de hardware sean manufacturados por terceros (OEM). Esto puede inducir a falsos positivos operativos, donde la identidad de red del equipo no se corresponde con su marca comercial. Asimismo, en entornos de mayor madurez tecnológica, el uso de técnicas de MAC spoofing (falsificación de direcciones) podría alterar deliberadamente los resultados del escaneo (Stallings, 2004).

Consideraciones de seguridad y privacidad: Desde la perspectiva de la ciberseguridad, la ejecución de escaneos activos en redes corporativas o industriales (a diferencia del entorno doméstico evaluado) sin autorización puede desencadenar alertas en sistemas de prevención de intrusos (IPS/IDS) (Lyon, 2009; Stallings, 2004). Por otro lado, desde la perspectiva de la privacidad, la facilidad con la que un dispositivo doméstico expone su huella digital y el fabricante de su hardware evidencia una superficie de ataque que actores malintencionados podrían utilizar para perfilar tecnológicamente un hogar y buscar vulnerabilidades específicas (CVEs)

asociadas a ese fabricante particular.

5 Propuesta de implementación didáctica

El procedimiento documentado en las secciones anteriores fue diseñado con una orientación formativa que permite su aplicación directa como actividad práctica. Se propone su implementación en la asignatura Redes de Datos, correspondiente al cuarto año de la Licenciatura en Sistemas de Información (FACENA – UNNE), que cuenta con una matrícula aproximada de 160 estudiantes por cohorte, distribuidos en comisiones de hasta 40 integrantes en el laboratorio de redes.

La actividad consiste en que cada dupla de estudiantes reproduzca el caso práctico de la Sección 4 en su propia red doméstica, siguiendo la secuencia: descubrimiento con nmap -sn, verificación con ping y traceroute, y caracterización mediante nmap -O con correlación OUI/MAC. Cada dupla deberá identificar al menos un dispositivo IoT tratándolo como "caja negra" y documentar el procedimiento en un informe técnico, complementado con una exposición oral ante sus pares. El trabajo en duplas responde a la estrategia pedagógica de la cátedra, orientada al fortalecimiento de competencias colaborativas y de comunicación técnica.

Esta actividad contribuiría al desarrollo de competencias clave: (a) diagnóstico sistemático de redes mediante línea de comandos; (b) interpretación y correlación de información técnica —puertos, servicios, direcciones MAC— con dispositivos físicos; (c) elaboración de documentación técnica reproducible; y (d) resolución de problemas en escenarios no controlados, dado que cada red doméstica presenta condiciones distintas. La viabilidad de la propuesta se sustenta en que las herramientas empleadas son de libre acceso y no requieren infraestructura más allá de una conexión de red doméstica.

6 Conclusión y posibles mejoras

El presente trabajo permitió analizar las principales herramientas de red disponibles en sistemas Linux y su aplicación en tareas de administración y diagnóstico de conectividad. La organización de los comandos según su función facilita la comprensión de su propósito y pone de manifiesto que el diagnóstico de una red requiere la utilización complementaria de distintas herramientas, cada una orientada a proporcionar información específica sobre el estado de la comunicación, las interfaces, el enrutamiento, los servicios y la resolución de nombres (Bautts et al., 2005; Siever et al., 2009).

El caso práctico demostró que es posible identificar y caracterizar un dispositivo tratado como "caja negra" mediante un procedimiento sistemático basado en herramientas estándar de Linux y Nmap. La secuencia de descubrimiento de hosts, verificación de conectividad y análisis de servicios permitió identificar correctamente un Smart TV dentro de una red doméstica, evidenciando la utilidad de estas herramientas para obtener información relevante sobre dispositivos conectados y apoyar las tareas de diagnóstico y administración (Lyon, 2009; Nmap Project, s.f.).

Asimismo, el estudio evidencia la evolución de las utilidades tradicionales hacia soluciones más completas como el paquete iproute2, cuya integración y soporte para tecnologías actuales lo convierten en la alternativa recomendada para la administración de redes en sistemas Linux modernos (Siever et al., 2009; The Linux Foundation, 2023). Desde una perspectiva formativa, el dominio de estas herramientas favorece la comprensión de los protocolos de comunicación y el desarrollo de competencias aplicables a la administración de sistemas, infraestructura de redes y ciberseguridad (Stallings, 2004; Kurose y Ross, 2017).

Como trabajo futuro, se propone ampliar la experiencia incorporando escenarios de mayor complejidad, como redes segmentadas, infraestructuras virtualizadas o entornos IoT con múltiples dispositivos. Asimismo, el procedimiento podría complementarse con herramientas de análisis de tráfico, como Wireshark, permitiendo correlacionar la información obtenida mediante los comandos de diagnóstico con la captura de paquetes y ampliar las posibilidades de análisis en actividades de administración y enseñanza de redes.

7 Referencias bibliográficas

Bautts, T., Dawson, T. y Purdy, G. (2005). *Linux Network Administrator's Guide* (3rd ed.). O'Reilly Media.

Cherry Servers. (2023). *Linux network commands*. Cherry Servers. https://www.cherryservers.com/blog/linux-network-commands

Kurose, J. F. y Ross, K. W. (2017). *Redes de computadoras: Un enfoque descendente* (7.ª ed.). Pearson Educación.

Lyon, G. F. (2009). *Nmap Network Scanning: The Official Nmap Project Guide to Network Discovery and Security Scanning*. Insecure.Com LLC.

Nmap Project. (s.f.). *Nmap Reference Guide*. https://nmap.org/man.html

RedesZone. (2021). *Comandos básicos para redes en Linux*. RedesZone. https://www.redeszone.net/tutoriales/redes-cable/comandos-basicos-redes-linux/

Siever, E., Figgins, S., Love, R. y Robbins, A. (2009). *Linux in a Nutshell* (6th ed.). O'Reilly Media.

Stallings, W. (2004). *Comunicaciones y Redes de Computadores* (7.ª ed.). Pearson Educación.

The Linux Foundation. (2023). *iproute2*. Linux Foundation Wiki. https://wiki.linuxfoundation.org/networking/iproute2