

Universidad de Costa Rica  
Ciudad Universitaria Rodrigo Facio  
Facultad de Ingeniería  
Escuela de Ciencias de la Computación e Informática

CI-0121 Redes de comunicación de datos  
Grupo 2

*Herramientas para diagnóstico de problemas de red*

Profesor  
Jose Antonio Brenes Carranza

Estudiantes  
María Celeste Herrera Rivera (B83875)  
Kendall Lara Castro (B43707)  
Alessandra Narváez Saravia (B85573)  
Alexandra Siles Alvarado (B56893)

II Semestre, 2020

## **Tabla de contenidos**

<b>1. Introducción</b>	<b>2</b>
<b>2. Desarrollo</b>	<b>3</b>
2.1. Comandos	3
2.2. Sniffers	26
2.3. Conexión Remota	30
<b>3. Conclusiones</b>	<b>34</b>
<b>Referencias</b>	<b>35</b>

## **1. Introducción**

Como muchos de los problemas que involucran el uso de computadoras, es posible que la solución a estos sea simple o complicada de resolver. También, existen muchos factores que influyen en la aparición de inconvenientes: pueden ser tanto de hardware como de software. En el caso de las redes, esta situación es aplicable.

A lo largo del tiempo, se han diseñado diversas herramientas que generan un análisis del estado de la red. Estas son particularmente útiles para encontrar rápidamente el problema y determinar cuál podría ser la solución. La terminal de comandos de un sistema operativo es uno de los medios más comunes para dar uso a estos programas, a través de comandos. Un comando es conocido como una orden que va a realizar un servicio, este se le da al sistema operativo (Rouse, s.f). Cada uno de ellos tiene una serie de parámetros que indican operaciones que están disponibles para realizar sobre determinado puerto y conexión.

Asimismo, estos programas, presentados tanto como comandos de terminal o como una interfaz gráfica, permiten otras funciones con las cuales manejar y aprovechar el alcance de las redes. En primera instancia, se encuentran los sniffers, que se definen como analizadores de tráfico de internet (Goyal & Goyal, 2017). En segunda instancia, se mencionan aquellos que permiten la conexión remota, es decir, la conexión con otras máquinas que resultan en ubicaciones alejadas.

Más allá de proporcionar diagnóstico y conexión, la importancia de conocer el funcionamiento de estos instrumentos radica en la protección de los sistemas y las comunicaciones. Conocer de ellos ayuda a identificar movimientos e intrusiones que hubo en la red, así como proporcionar una conexión para que haya comunicación segura, de modo que se permita transmitir información sensible con cierta protección.

En esta investigación se abordan las diferentes herramientas que ayudan a dar un diagnóstico sobre la red. De todas las posibilidades, se profundiza en comandos, sniffers y de la conexión remota.

## 2. Desarrollo

### 2.1. Comandos

#### 2.1.1.ipconfig

Según la documentación de los comandos por rol de servidor de windows en la documentación de Microsoft (2017),

«IpConfig es un comando que muestra todos los valores de configuración de red TCP/IP actuales y actualiza la configuración del protocolo de configuración dinámica de host (DHCP) y DNS. Si se usa sin parámetros muestra las direcciones de IPv4, IPv6, la máscara de subred y la puerta de enlace predeterminada para todos los adaptadores.»

Dentro de esta misma documentación se muestra los diferentes parámetros y sintaxis que se pueden adjuntar a este comando:

- */all*: muestra la configuración TCP/IP completa de todos los adaptadores.
- */displaydns*: muestra el contenido de la memoria caché de la resolución del cliente DNS.
- */flushdns*: vacía y restablece el contenido de la memoria caché de la resolución del cliente DNS. Para la solución de problemas de DNS, descarta las entradas de caché negativas de la memoria caché.
- */registerdns*: inicia el registro dinámico manual de los nombres DNS y las direcciones IP que se configuran en un equipo. Se puede usar este parámetro para solucionar un error de registro de nombres DNS o resolver un problema de actualización dinámica entre un cliente y el servidor DNS sin reiniciar el equipo cliente.
- */release [<adapter>]*: envía un mensaje DHCPRELEASE al servidor DHCP para liberar la configuración actual de DHCP y descartar la configuración de la dirección IP para todos los adaptadores (si no se especifica un adaptador) o para un adaptador específico, si se incluye el parámetro *adaptador* . Este parámetro deshabilita TCP/IP para los adaptadores configurados para obtener una dirección IP automáticamente.
- */release6 [<adapter>]*: envía un mensaje DHCPRELEASE al servidor DHCPv6 para liberar la configuración actual de DHCP y descartar la configuración de la dirección IPv6

para todos los adaptadores (si no se especifica un adaptador) o para un adaptador específico.

- `/renew [<adapter>]`: renueva la configuración de DHCP para todos los adaptadores (si no se especifica un adaptador).
- `/renew6 [<adapter>]`: renueva la configuración de DHCPv6 para todos los adaptadores (si no se especifica un adaptador).
- `/setclassid <adapter>[<classID>]`: configura el identificador de clase DHCP para un adaptador especificado.
- `/showclassid <adapter>`: muestra el identificador de clase DHCP de un adaptador especificado.

### 2.1.2. ifconfig

Según la Oracle (s.f), este comando visualiza las direcciones IP asignadas, los nombres de dispositivos, la información de vinculación y otra información que podría necesitar durante la resolución de problemas relacionados con la red en dispositivos UNIX. Este es usado en el inicio del sistema (boot) para configurar interfaces de red. El comando se encuentra desactualizado, por eso se recomienda usar el comando *ip*.

Dentro de la documentación del manual de Linux expone la siguiente sintaxis y opciones:

```
ifconfig      [-v] [-a] [-s] [interface]
ifconfig      [-v] interface [aftype] options | address
```

- `-a`: muestra todas las interfaces de red disponibles en el momento incluso si están caídas.
- `-s`: muestra un pequeño resumen de las interfaces de red.
- `<interfaz>`: muestra todos los detalles de una interfaz de red en específico: nombre, estado del dispositivo(flags), la unidad máxima de transferencia (MTU), dirección MAC,dirección inet y inet6, submáscaras de red, longitud de la cola de transmisión, RX packets y TX packets.
- `<interfaz> up/down`: activa o desactiva la interfaz especificada.
- `<interfaz> [-]arp`: habilita o desactiva el uso del protocolo ARP en esta interfaz.

- `<interfaz> [-]allmulti`: habilita o desactiva el modo all-multicast. Si se selecciona, se recibirán todos los paquetes de multicast por la interfaz.
- `<interfaz> mtu N`: establece la Unidad de Transferencia Máxima (MTU) de un interfaz.
- `<interfaz> netmask addr`: establece la máscara de red IP para un interfaz. Este valor es por defecto el de la máscara de red normal de clase A, B o C (tal y como se deriva de la dirección IP del interfaz), pero puede configurarse para cualquier valor.
- `<interfaz> add/del addr/prefixlen`: agrega o remueve una dirección IPv6 a la interfaz.
- `<interfaz> io_addr addr`: establece la dirección inicial en el espacio de E/S para este dispositivo.
- `<interfaz> [-]broadcast [addr]`: si se da un argumento de dirección, establece la dirección de emisión del protocolo del interfaz. De otro modo, establece (o elimina) la opción IFF\_BROADCAST de la interfaz.
- `<interfaz> [-]pointopoint [addr]`: esta opción activa el modo point-to-point (punto a punto) del interfaz, lo cual significa que se trata de una unión directa entre dos máquinas, sin nadie más a la escucha. Si se da también un argumento de dirección, establece la dirección de protocolo del otro lado de la unión, exactamente igual que hace la opción obsoleta `dstaddr`. Si no, establece o elimina la opción IFF\_POINTOPOINT del interfaz.
- `<interfaz> hw class address`: establece la dirección de hardware del interfaz, siempre que el driver del dispositivo lo permita. Esta opción debe ir seguida del nombre de la clase de hardware y el código ASCII imprimible equivalente de la dirección del hardware. Las clases de hardware soportadas actualmente incluyen: ether (Ethernet), ax25 (AMPR AX.25), ARCnet y netrom (AMPR NET/ROM).
- `<interfaz> address`: la dirección IP que se va a asignar al interfaz.
- `<interfaz> txqueuelen length`: establece la longitud de la cola de transmisión del dispositivo. Resulta útil configurar este parámetro con valores pequeños para dispositivos más lentos con latencias altas (uniones de modems, ISDN) para evitar que las transferencias masivas de información interfieran demasiado el tráfico interactivo del tipo telnet.

### 2.1.3.ip

Es un comando exclusivo de UNIX, y el sustituto del comando Ifconfig al ser este obsoleto en versiones más nuevas del sistema operativo. El manual de Linux especifica que “muestra/ manipula el enrutamiento de dispositivos de red, interfaces y túneles”.

Este comando también simplifica algunas funciones del comando arp, netstat y route.

A continuación, se describe la sintaxis del comando y ejemplos de operaciones que pueden utilizarse.

ip [ opciones] objeto{comando}

- *a/addr*: muestra la información de todas las interfaces de red en nuestro dispositivo.
- *a/addr/address show <interfaz>*: información en específico de una interfaz.
- *a add/del <dirección> dev <interfaz>*: agrega o elimina una dirección IP a una interfaz de red.
- *addr flush dev <interfaz>*: remueve la dirección IP de la interfaz de red especificada.
- *-s link*: estadísticas de uso de los dispositivos de red.
- *route*: muestra el enrutamiento de la red.
- *link set <interfaz> down/up*: activar o desactivar una interfaz de red en específico.
- *monitor*: monitoriza el estado de los dispositivos de red (direcciones y rutas).
- *neig/Neighbour*: ver dispositivos MAC conectados.
- *link set dev <interfaz> arp on/off*: activar o desactivar protocolo ARP
- *addr add <direccion> dev <interfaz>*: asigna una dirección IP específica a una interfaz de red.

### 2.1.4. telnet

El comando telnet tiene como función principal establecer comunicación interactiva con otro host al implementar el protocolo Telnet. No obstante, telnet también es una de las principales herramientas para detectar averías en conexiones TCP a cualquier servidor con cualquier puerto. Para poder utilizar el comando en Linux (Ubuntu), es necesario ejecutar “sudo apt-get install telnet” en la línea de comandos. Para el caso de Windows, se debe activar el cliente telnet en la sección de “Activar o desactivar características de Windows”. En las figuras 1

y 2 se ejemplifica el resultado de usar telnet para extraer el código fuente de una página; la primera en el caso de Ubuntu y la segunda, Windows.

Ubuntu Manpage Repository (2019) explica la sintaxis del comando de la siguiente forma:

```
telnet [-8ELadr] [-S tos] [-e escapechar] [-l user] [-n tracefile] [host [port]]
```

- *-8*: negociación para que se utilicen la entrada y salida sean en formato de 8 bits.
- *-S*: establece la opción IP type-of-service para el valor de la conexión que se ingrese con el parámetro.
- *-e*: establece un carácter de escape en específico, aunque si no se introduce ninguno, significa que no habrá carácter de escape.
- *-l*: especifica el usuario con el que se hará la conexión.
- *-n*: especifica el archivo donde se guardará la información de la ruta de los paquetes que se envía y el tiempo en que tarda para llegar hasta su destino.
- *host*: indica con quién se va a intentar la conexión. Si solo se indica el host cuando se llama a telnet, el comando open se ejecutará automáticamente.
- *port*: indica el número de puerto o servicio al que se debe hacer la conexión; el puerto 23 es el predeterminado.

Algunos comandos que se pueden utilizar cuando se establece la conexión se describen a continuación.

- *close (c)*: cierra la conexión actual.
- *display (d)*: despliega los parámetros que especifican las características de la conexión.
- *open (o)*: se acompaña del nombre del host y del port, a menos de que se quiera trabajar con el 23 que es el que se designa por defecto.
- *quit (q)*: para salirse de telnet.
- *set (set)*: especifica las opciones para las conexiones.
- *send (sen)*: envía secuencias de caracteres especiales al host. Los códigos disponibles pueden ser utilizados varios a la vez.
- *status (st)*: muestra el estado actual de telnet.



- *unset (u)*: operación contraria a set; devuelve a la opción por default.
- *help (?/h)*: muestra las posibles.
- *logout*: cierra la conexión forzosamente.
- *z*: sale forzosamente de telnet.

```
macelesteh@Celeste:~$ telnet 163.178.104.187 80
Trying 163.178.104.187...
Connected to 163.178.104.187.
Escape character is '^]'.
GET /HTTP/1.1
<HTML>
  <HEAD>
    <meta property="og:title" content="UCR/ECCI Servidor de recursos">
    <meta property="og:description" content="Recursos para desarrollo de cursos
para la ECCI-UCR, profesor Francisco Arroyo">
    <meta property="og:image" content="https://os.ecci.ucr.ac.cr/ci0122/operati
ng-system.png">
  </HEAD>

  <CENTER>
  <IMG SRC=/RioCeleste-VolcanTenorio.jpg>
  </CENTER>
</HTML>

Connection closed by foreign host.
```

Figura 1. Ejemplo de ejecución de telnet en Ubuntu.

```
Telnet 163.178.104.187
Microsoft Telnet Client
GET /HTTP/1.1
Escape Character is 'CTRL+'
Microsoft Telnet> open 163.178.104.187 80
Connecting To 163.178.104.187...

<HTML>
  <HEAD>
    <meta property="og:title" content="UCR/ECCI Servidor de recurs
para la ECCI-UCR, profesor Francisco Arroyo">
    <meta property="og:image" conte

    =/RioCeleste-VolcanTenorio.jpg>
  </CENTER>
  </HTML>

Connection to host lost.
Press any key to continue...
```

Figura 2. Ejemplo de ejecución de telnet en Windows.

### 2.1.5. netstat

Según la Oracle (s.f), netstat es un comando que presenta el estado de la red y estadísticas del protocolo, dependiendo de la opción usada junto al comando, se muestran más datos de la red. Dentro de la documentación de Microsoft (2017) se muestra la sintaxis dentro de Windows que este comando puede tener y una pequeña descripción de estos mismos. Hay que aclarar que

en la sintaxis en Linux cambia un poco, pero los comandos especificados son importante mencionarlos:

`netstat [-a] [-b] [-e] [-n] [-o] [-p <Protocolo>] [-r] [-s] [<intervalo>]`

- *-a*: en windows muestra conexiones TCP activas. En Linux muestra los sockets que escuchan o no.
- *-b*: en windows muestra el ejecutable al crear una conexión a un puerto que escucha.
- *-e*: en windows muestra las estadísticas del Ethernet. Linux muestra información adicional.
- *-n*: muestra numéricamente el puerto y las direcciones.
- *-o*: en windows muestra las conexiones activas y procesos.
- *-p <protocolo>*: en windows muestra las conexiones que están en x protocolo. Este protocolo se tiene que especificar dentro de <protocolo>.
- *-s*: por protocolo muestra las estadísticas.
- *-r*: muestra lo que hay dentro de la tabla de enrutamiento IP.
- *<intervalo>*: la información la va a mostrar cada x segundos.
- *-i*: muestra las interfaces de red en Linux.

Como se estableció anteriormente, netstat sirve para ver cómo está la red, por lo que se obtiene información de quién está conectada a ella. En caso de que haya algún intruso en la red, esta información se mostrará. Esta herramienta se puede utilizar en Windows como en los diferentes sistemas LINUX. A continuación, en la figura 3 y figura 4 se va a mostrar como se ve el comando netstat en diferentes sistemas operativos.

```
Command Prompt
C:\Users\Alessandra>netstat
Active Connections

```

Proto	Local Address	Foreign Address	State
TCP	192.168.0.7:62616	52.179.224.121:https	ESTABLISHED
TCP	192.168.0.7:62646	37.156.185.137:https	ESTABLISHED
TCP	192.168.0.7:62667	mia04-002:https	ESTABLISHED
TCP	192.168.0.7:62937	vn-in-f108:5228	ESTABLISHED
TCP	192.168.0.7:62947	mia09s21-in-f3:https	ESTABLISHED
TCP	192.168.0.7:62948	yb-in-f109:https	ESTABLISHED
TCP	192.168.0.7:62949	yyz08s13-in-f142:https	ESTABLISHED
TCP	192.168.0.7:63122	mia07s49-in-f19:https	ESTABLISHED
TCP	192.168.0.7:63151	151.101.6.217:https	ESTABLISHED
TCP	192.168.0.7:63182	mia07s48-in-f14:https	ESTABLISHED
TCP	192.168.0.7:63186	104.17.79.107:https	ESTABLISHED
TCP	192.168.0.7:63217	mia07s47-in-f14:https	ESTABLISHED
TCP	192.168.0.7:63295	151.101.6.49:https	ESTABLISHED
TCP	192.168.0.7:63359	yyz08s14-in-f138:https	ESTABLISHED
TCP	192.168.0.7:63364	fra16s42-in-f3:https	ESTABLISHED
TCP	192.168.0.7:63366	mia07s47-in-f14:https	ESTABLISHED
TCP	192.168.0.7:63370	a23-195-96-193:https	ESTABLISHED
TCP	192.168.0.7:63371	r-154-48-62-5:https	ESTABLISHED
TCP	192.168.0.7:63374	64.4.54.254:https	ESTABLISHED
TCP	192.168.0.7:63375	64.4.54.254:https	ESTABLISHED
TCP	192.168.0.7:63378	13.107.21.200:https	ESTABLISHED
TCP	192.168.0.7:63379	13.107.21.200:https	ESTABLISHED
TCP	192.168.0.7:63380	13.107.21.200:https	ESTABLISHED
TCP	192.168.0.7:63381	40.68.211.74:https	ESTABLISHED
TCP	192.168.0.7:63382	ec2-3-220-243-247:https	ESTABLISHED
TCP	:::11:27275	Ale:63372	TIME_WAIT
TCP	:::11:27275	Ale:63373	TIME_WAIT
TCP	:::11:27275	Ale:63376	TIME_WAIT
TCP	:::11:27275	Ale:63377	TIME_WAIT

```
C:\Users\Alessandra>
```

Figura 3. Ejecución del comando netstat en el sistema operativo Windows.

```
File Edit View Search Terminal Help
alessandra@AlessandraUbuntu:~$ netstat
Active Internet connections (w/o servers)

```

Proto	Recv-Q	Send-Q	Local Address	Foreign Address	State
tcp	0	0	AlessandraUbuntu:53528	ord08s13-in-f14.1:https	ESTABLISHED
tcp	0	0	AlessandraUbuntu:41678	mia09s01-in-f3.1e:https	ESTABLISHED
tcp	0	0	AlessandraUbuntu:35200	mia07s49-in-f10.1:https	ESTABLISHED
tcp	0	0	AlessandraUbuntu:53682	mia07s48-in-f13.1:https	ESTABLISHED
tcp	0	0	AlessandraUbuntu:56646	mia07s54-in-f10.1:https	ESTABLISHED
tcp	0	0	AlessandraUbuntu:33188	ec2-52-34-41-81.u:https	ESTABLISHED
tcp	0	0	AlessandraUbuntu:47538	mia09s21-in-f14.1:https	ESTABLISHED
tcp	0	0	AlessandraUbuntu:56240	mia09s19-in-f14.1:https	ESTABLISHED
tcp	0	0	AlessandraUbuntu:56648	mia07s54-in-f10.1:https	ESTABLISHED
tcp	0	0	AlessandraUbuntu:51930	iad23s23-in-f202:https	ESTABLISHED
tcp	0	0	AlessandraUbuntu:45664	yyz08s10-in-f170:https	ESTABLISHED
tcp	0	0	AlessandraUbuntu:34490	mia07s48-in-f4.1e:https	ESTABLISHED
tcp	0	0	AlessandraUbuntu:60028	mia07s48-in-f14.1:https	ESTABLISHED
tcp	0	0	AlessandraUbuntu:36790	mia09s22-in-f3.1e:https	ESTABLISHED
tcp	0	0	AlessandraUbuntu:40472	156.99.224.35.bc.g:http	TIME_WAIT
tcp	0	0	AlessandraUbuntu:32988	mia07s57-in-f14.1:https	ESTABLISHED
tcp	0	0	AlessandraUbuntu:60082	mia07s48-in-f14.1:https	ESTABLISHED
tcp	0	0	AlessandraUbuntu:50598	mia07s49-in-f14.1:https	ESTABLISHED
tcp	0	0	AlessandraUbuntu:35568	mia07s46-in-f3.1e:https	ESTABLISHED
tcp	0	0	AlessandraUbuntu:45662	yyz08s10-in-f170:https	ESTABLISHED
tcp	0	0	AlessandraUbuntu:32994	mia07s57-in-f14.1:https	ESTABLISHED
tcp	0	0	AlessandraUbuntu:54738	mia07s57-in-f3.1e:https	ESTABLISHED
tcp	0	0	AlessandraUbuntu:54422	mia09s01-in-f1.1e:https	ESTABLISHED
tcp	0	0	AlessandraUbuntu:34282	yx-in-f189.1e100:https	ESTABLISHED
tcp	0	0	AlessandraUbuntu:34434	mia07s48-in-f4.1e:https	ESTABLISHED
tcp	0	0	AlessandraUbuntu:34256	mia09s21-in-f3.1e1:http	TIME_WAIT

```
Active UNIX domain sockets (w/o servers)

```

Proto	RefCnt	Flags	Type	State	I-Node	Path
unix	2	[ ]	DGRAM		28841	/run/user/1000/systemd/notify
unix	3	[ ]	DGRAM		16127	/run/systemd/notify
unix	8	[ ]	DGRAM		16142	/run/systemd/journal/socket
unix	18	[ ]	DGRAM		16155	/run/systemd/journal/dev-log
unix	2	[ ]	DGRAM		13864	/run/systemd/journal/syslog
unix	2	[ ]	DGRAM		313366	/run/wpa_supplicant/wlp2s0
unix	2	[ ]	DGRAM		312362	/run/wpa_supplicant/p2p-dev-wlp2s0

Figura 4. Ejecución del comando netstat en el sistema operativo Linux.

Se va a explicar un poco los datos que muestran las figuras 3 y 4 gracias a la documentación de Oracle. Se puede ver en ambas figuras en ambos casos se muestra el protocolo usado para establecer la conexión. Se muestra la dirección local y la dirección destino con su respectivo puerto y finalmente el estado de la conexión. En el caso de la figura 4 (LINUX) también muestran la cantidad de bytes enviados y recibidos que están en cola del socket.

En el caso de Linux este muestra los sockets activos de dominio UNIX con o sin servidores. Según la IBM (s.f) “estos dan una buena comunicación entre procesos que corren en el mismo procesador z/TPF”. Incluyen información como su protocolo, RefCnt (cantidad de procesos conectados a él), banderas, el tipo (DGRAM, STREAM, otros), el estado, I-Node y el path. El manual de Linux (s.f) explica que los inodos contienen metadatos de un archivo.

#### 2.1.6. Tracert (traceroute)

Normalmente, en una red de área local (LAN) cada máquina puede contactar directamente con todas las demás máquinas. Una red más grande, como Internet, se compone de muchas redes más pequeñas, y sería imposible que cada máquina supiera cómo llegar a todas las demás. Para conectar dos redes juntas, podríamos dedicar una computadora (llamada *gateway*) para enrutar el tráfico de una red a otra. Suponga que una computadora en la red A quisiera enviar un paquete a una computadora en la red B. En lugar de almacenar la información de enrutamiento para la red B, la computadora en la red A simplemente enviaría el paquete al *gateway* (puerta de enlace), y la puerta de enlace lo enviaría a su destino. Traceroute es un programa que muestra los enrutadores entre su computadora y una computadora de destino.

Rastrear la ruta que siguen los paquetes de un usuario puede ser difícil, por lo que *traceroute* utiliza el campo "*time to live*" del protocolo IP e intenta obtener una respuesta ICMP TIME\_EXCEEDED de cada *gateway* y el camino a algún *host*. El único parámetro obligatorio es el nombre de host de destino o la dirección IP de este.

```
Kendall — -bash — 81x34
MacBook-Pro-de-Kendal:~ Kendall$ traceroute youtube.com
traceroute to youtube.com (172.217.8.142), 64 hops max, 52 byte packets
 1  192.168.100.1 (192.168.100.1)  23.823 ms  1.102 ms  1.088 ms
 2  100.64.0.1 (100.64.0.1)  3.526 ms  6.738 ms  13.031 ms
 3  10.251.137.33 (10.251.137.33)  3.414 ms  3.354 ms  5.645 ms
 4  10.251.131.137 (10.251.131.137)  43.773 ms  66.187 ms  68.925 ms
 5  mai-b1-link.telia.net (80.239.134.250)  52.674 ms  52.214 ms  53.736 ms
 6  google-ic-333177-mai-b1.c.telia.net (62.115.154.67)  60.111 ms
   google-ic-326616-mai-b1.c.telia.net (80.239.128.35)  45.433 ms  52.229 ms
 7  108.170.249.17 (108.170.249.17)  51.975 ms  59.794 ms  60.066 ms
 8  108.170.225.13 (108.170.225.13)  52.503 ms
   216.239.57.168 (216.239.57.168)  61.771 ms
   108.170.225.13 (108.170.225.13)  48.944 ms
 9  mia07s49-in-f14.1e100.net (172.217.8.142)  57.776 ms
   142.250.58.83 (142.250.58.83)  59.281 ms
   108.170.253.19 (108.170.253.19)  54.899 ms
MacBook-Pro-de-Kendal:~ Kendall$
```

Figura 5. Ejecución del comando traceroute en el sistema operativo MacOS.

Este programa intenta rastrear la ruta que tomaría un paquete a un host final de internet, por ejemplo en la Figura 3 vemos la ruta que se toma para llegar a YouTube, en este caso el host final, desde un dispositivo solicitante. La estrategia tomada es lanzar paquetes de sondeo UDP (User Datagram Protocol) con un pequeño ttl (time to live) y luego escuchar una respuesta ICMP (Internet Control Message Protocol) de "tiempo excedido" desde un gateway, esto se logra estableciendo inicialmente el ttl en uno e ir aumentando este ttl hasta que obtengamos un "puerto inalcanzable" ICMP lo que significa que obtuvimos el host o se alcanzó un máximo de saltos (30 saltos por defecto), los cuales pueden ser cambiados con una bandera.

Se envían tres sondas en cada configuración de ttl y se imprime en consola una línea que muestra la dirección de la puerta de enlace, el ttl, y el tiempo de ida y vuelta de cada sonda. Si las respuestas de la sonda provienen de diferentes gateways, se imprimirá la dirección de cada sistema de respuesta. Si no hay respuesta en tres segundos (configurable con una bandera) en un intervalo de tiempo de espera, se imprime un "\*" para esa sonda.

En este ejemplo hay ocho gateways y el noveno es el destino final, no queremos que el host de destino procese los paquetes de la sonda UDP, por lo que el puerto de destino se establece en un valor poco probable y obtenemos en consola la ruta que se tomó para llegar al host deseado.

### 2.1.7. ping

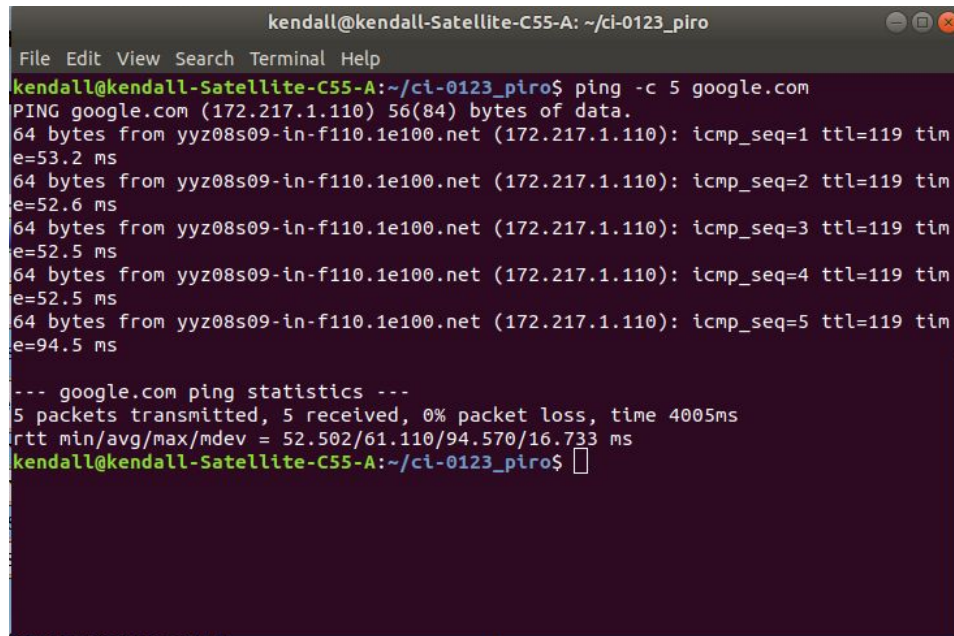
Este comando verifica la conectividad de nivel IP a otra computadora TCP/IP enviando mensajes de solicitud de *echo* ICMP (protocolo de mensajes de control de Internet). Se muestra la recepción de los mensajes de respuesta de *echo* correspondientes, junto con los tiempos de ida y vuelta. *ping* es el comando TCP/IP principal que se utiliza para solucionar problemas de conectividad, accesibilidad y resolución de nombres.

Este comando también puede ser utilizado para probar tanto el nombre de la computadora como la dirección IP de la computadora. Si al hacer *ping* a la dirección IP tiene éxito, pero no lo es el nombre de la computadora, es posible haya un problema de resolución de nombres. En este caso, hay que asegurarse que el nombre de la computadora que está especificando se pueda resolver a través del archivo *Hosts* locales, mediante consultas DNS.

La utilidad de *ping* utiliza el datagrama ECHO\_REQUEST obligatorio del protocolo ICMP para obtener una ECHO\_RESPONSE ICMP de un *host* o *gateway*. Los datagramas ECHO\_REQUEST (pings) tienen un encabezado IP e ICMP, seguido de un "*struct timeval*" y luego un número arbitrario de bytes "*pad*" que se utilizan para completar el paquete. A continuación se muestran algunas de las banderas más utilizadas con *ping*:

- *-a audible*: incluye un carácter de campana (0x07) en la salida cuando se reciba cualquier paquete. Esta opción se ignora si existen otras opciones de formato.
- *-t timeout*: Esta bandera especifica el tiempo de espera, en segundos, antes de que finalice *ping*, sin importar cuántos paquetes haya recibido.
- *-c count*: Se detiene después de enviar (y recibir) los paquetes de conteo ECHO\_RESPONSE.
- *-m ttl*: Establece el ttl de IP para los paquetes salientes.
- *-o*: Salir correctamente después de recibir un paquete de respuesta.
- *-4*: Usa ipv4.
- *-6*: Usa ipv6.



A screenshot of a terminal window titled 'kendall@kendall-Satellite-C55-A: ~/ci-0123\_piro'. The terminal shows the execution of the command 'ping -c 5 google.com'. The output displays five ping attempts with their respective IP addresses, ICMP sequence numbers, TTL values, and round-trip times. The first four attempts are successful with times around 52-53 ms, while the fifth attempt times out at 94.5 ms. Below the individual pings, a summary statistics line shows '5 packets transmitted, 5 received, 0% packet loss, time 4005ms' and 'rtt min/avg/max/mdev = 52.502/61.110/94.570/16.733 ms'. The prompt 'kendall@kendall-Satellite-C55-A:~/ci-0123\_piro\$' is visible at the bottom.

```
kendall@kendall-Satellite-C55-A: ~/ci-0123_piro
File Edit View Search Terminal Help
kendall@kendall-Satellite-C55-A:~/ci-0123_piro$ ping -c 5 google.com
PING google.com (172.217.1.110) 56(84) bytes of data.
64 bytes from yyz08s09-in-f110.1e100.net (172.217.1.110): icmp_seq=1 ttl=119 time=53.2 ms
64 bytes from yyz08s09-in-f110.1e100.net (172.217.1.110): icmp_seq=2 ttl=119 time=52.6 ms
64 bytes from yyz08s09-in-f110.1e100.net (172.217.1.110): icmp_seq=3 ttl=119 time=52.5 ms
64 bytes from yyz08s09-in-f110.1e100.net (172.217.1.110): icmp_seq=4 ttl=119 time=52.5 ms
64 bytes from yyz08s09-in-f110.1e100.net (172.217.1.110): icmp_seq=5 ttl=119 time=94.5 ms

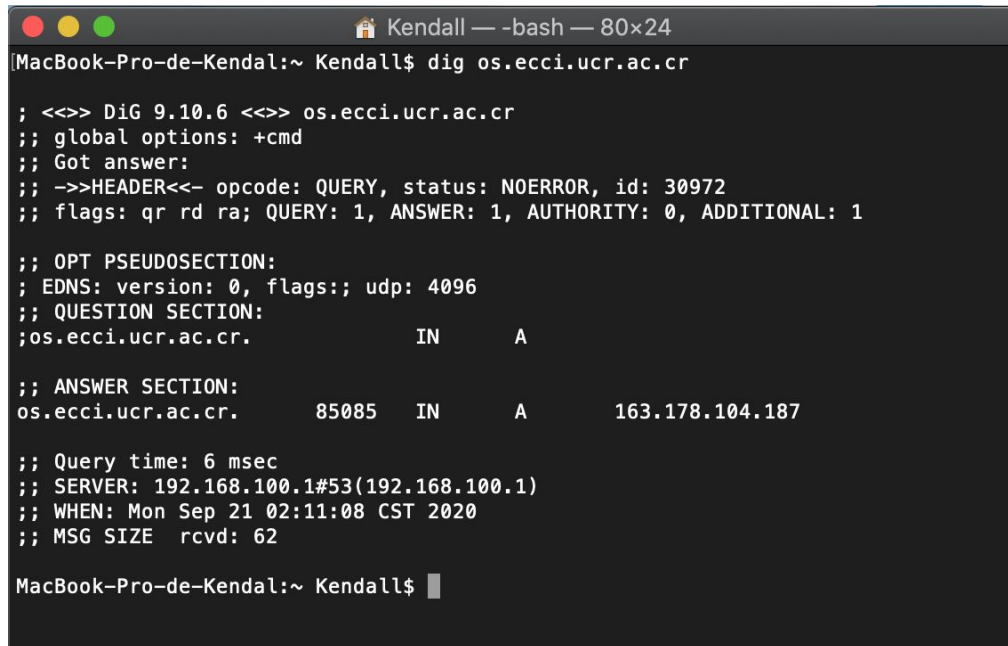
--- google.com ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4005ms
rtt min/avg/max/mdev = 52.502/61.110/94.570/16.733 ms
kendall@kendall-Satellite-C55-A:~/ci-0123_piro$
```

Figura 6. Ejecución del comando ping en el sistema operativo Ubuntu.

#### 2.1.8. dig

Dig es una herramienta de administración de red para consultar el sistema de nombres de dominio (DNS). Es útil para la resolución de problemas de red y es una forma de lidiar con lo que podría decirse que es un defecto en nslookup. Dig no es tan generalizado como nslookup, en dig se especifica todos los aspectos de la consulta que le gustaría enviar en la línea de comando; no hay modo interactivo.

Una diferencia importante entre nslookup y dig, es que dig no aplica la lista de búsqueda, por lo que siempre use nombres de dominio completos como argumentos para hacer "dig". Dig muestra el mensaje de respuesta DNS completo, con las diversas secciones (encabezado, pregunta, respuesta, autoridad y adicionales) claramente indicadas, y con registros de recursos en esas secciones impresas en formato de archivo maestro. Esto puede resultar útil si necesita utilizar algunos de los resultados de su herramienta de resolución de problemas en un archivo de datos de zona o en su archivo de sugerencias raíz. Y se ve de la siguiente manera:



```
Kendall — -bash — 80x24
[MacBook-Pro-de-Kendal:~ Kendall$ dig os.ecci.ucr.ac.cr

; <<>> DiG 9.10.6 <<>> os.ecci.ucr.ac.cr
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 30972
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;os.ecci.ucr.ac.cr.          IN      A

;; ANSWER SECTION:
os.ecci.ucr.ac.cr.          85085   IN      A      163.178.104.187

;; Query time: 6 msec
;; SERVER: 192.168.100.1#53(192.168.100.1)
;; WHEN: Mon Sep 21 02:11:08 CST 2020
;; MSG SIZE rcvd: 62

MacBook-Pro-de-Kendal:~ Kendall$
```

Figura 7. Ejecución del comando dig en el sistema operativo MacOS.

Hay muchas opciones de banderas para introducir en un línea de comandos. Sin embargo, a continuación se muestra una lista de los más importantes y lo que hacen:

- `-x address`: dig asume que el argumento del nombre de dominio que ha especificado es realmente una dirección IP, por lo que invierte los octetos y *tacks* en *in-addr.arpa*. El uso de `-x` también cambia el tipo de registro predeterminado buscado a **cualquiera**, por lo que puede asignar una dirección IP inversa con `dig -x 10.0.0.1`, por ejemplo.
- `-p port`: Envía consultas al puerto especificado en lugar del predeterminado puerto 53.
- `+norec[urse]`: Desactiva la recursividad (la recursividad está activada de forma predeterminada).
- `+vc`: Envía consultas basadas en TCP (las consultas son UDP por defecto).

#### 2.1.9. arp

“ARP” es el acrónimo para Adress Resolution Protocol (Protocolo de Resolución de Direcciones en español). Su función es convertir direcciones de hardware (direcciones físicas o direcciones MAC) a direcciones del protocolo IPv4 y en sentido contrario (Ubuntu Manpage Repository, 2019).



Para asignar direcciones a través de este protocolo, es necesario saber si la dirección IP se encuentra en la red local o en una subred por aparte (IONOS, 2019). Si se trata de una red local, es necesario verificar si existe una entrada en la caché del ARP. En esta última son almacenadas las asociaciones de hardware y las direcciones del protocolo; su tamaño es limitado y tiende a eliminar las que no se han utilizado recientemente.

En el caso de que ya hubiese una dirección física, esta se utiliza para el direccionamiento. De lo contrario, se envía un “ARP request” a los hosts que existan en la red con la dirección IP de destino. La dirección del destinatario, en este caso, sería la dirección de broadcast de ARP. Cada una de las máquinas asociadas compara la dirección enviada con la propia: si no coinciden, entonces rechaza la solicitud, pero si lo hace, se produce un “ARP reply” que contiene la dirección de hardware.

Por otro lado, si el host de destino se ubica en otra subred, el remitente utiliza la puerta de enlace (router) a través de una combinación de las direcciones IP y la física. Seguidamente, la puerta recibe y envía el paquete hasta el destinatario a través de la pasarela de enlace. El proceso para encontrar el host de la otra red se repite indefinidamente hasta que el paquete llegue correctamente (IONOS, 2019).

La longitud de los paquetes enviados bajo este protocolo es de 28 bytes o 224 bits que son distribuidos en nueve espacios. La distribución de los paquetes es la siguiente (Kadhum, s.f.): tipo de dirección de hardware y de protocolo de red, longitud de la dirección de hardware y de la dirección de protocolo, tipo de operación, dirección física e IP del remitente y dirección física e IP del destinatario.

La función “arp” se encarga de interactuar con el caché de la red IPv4 del kernel, de modo que se pueda mostrar el contenido de las tablas, agregar una dirección o eliminarla. Un uso de ese comando es cuando existe una IP duplicada que pueda crear conflictos. En este caso, se puede eliminar la dirección duplicada.

Para utilizar la herramienta, es necesario ejecutar “sudo arp install net-tools” en la línea de comando, para el caso de Linux (Ubuntu). A continuación, se describen las tres principales herramientas que provee “arp” (Dawson y Kirch, 2000).

- `-s`: añade una dirección a la tabla de direcciones. Se utiliza junto con la dirección por añadir junto con la dirección de hardware. Esta función también resulta útil para la implementación de proxys ARP, donde una computadora simula ser una pasarela para otra.
- `-d`: borra las entradas de ARP que hagan referencias a una misma computadora. Puede ser útil para obtener una dirección de Ethernet asociada a una IP. Además, permite la reconfiguración en caso de que alguna información haya sido transferida erróneamente.
- `-a`: muestra el contenido de la tabla. Para Windows, se puede usar tanto la orden “-a” como “-g”. Para Linux, puede invocarse “arp” sin ningún otro acompañamiento o “-e” para establecer una salida por defecto; o “-a” para que la salida sea del estilo BSD.

```

C:\Users\macel>arp -a

Interface: 192.168.56.1 --- 0x4
Internet Address      Physical Address      Type
192.168.56.255        ff-ff-ff-ff-ff-ff    static
224.0.0.2             01-00-5e-00-00-02    static
224.0.0.22            01-00-5e-00-00-16    static
224.0.0.251           01-00-5e-00-00-fb    static
224.0.0.252           01-00-5e-00-00-fc    static
239.255.255.250       01-00-5e-7f-ff-fa    static
255.255.255.255       ff-ff-ff-ff-ff-ff    static

Interface: 192.168.0.7 --- 0xd
Internet Address      Physical Address      Type
192.168.0.1           70-62-b8-82-70-b5    dynamic
192.168.0.2           d0-2b-20-e2-59-76    dynamic
192.168.0.4           c4-2a-d0-66-5b-b6    dynamic
192.168.0.8           e2-f2-ca-cf-90-20    dynamic
192.168.0.255         ff-ff-ff-ff-ff-ff    static
224.0.0.2             01-00-5e-00-00-02    static
224.0.0.22            01-00-5e-00-00-16    static
224.0.0.251           01-00-5e-00-00-fb    static
224.0.0.252           01-00-5e-00-00-fc    static
239.255.255.250       01-00-5e-7f-ff-fa    static
239.255.255.253       01-00-5e-7f-ff-fd    static
255.255.255.255       ff-ff-ff-ff-ff-ff    static

Interface: 169.254.121.107 --- 0x3a
Internet Address      Physical Address      Type
169.254.255.255       ff-ff-ff-ff-ff-ff    static
224.0.0.2             01-00-5e-00-00-02    static
224.0.0.22            01-00-5e-00-00-16    static
224.0.0.251           01-00-5e-00-00-fb    static
224.0.0.252           01-00-5e-00-00-fc    static
239.255.255.250       01-00-5e-7f-ff-fa    static
255.255.255.255       ff-ff-ff-ff-ff-ff    static

C:\Users\macel>

```

Figura 8. Ejecución del comando arp en el sistema operativo Windows.

```

macelesteh@Celeste:~$ arp
Dirección      TipoHW  DirecciónHW      Indic Máscara      Interf
az
_gateway      ether   52:54:00:12:35:02  C                  enp0s3
macelesteh@Celeste:~$ arp -a
_gateway (10.0.2.2) en 52:54:00:12:35:02 [ether] en enp0s3

```

Figura 9. Ejecución del comando arp en el sistema operativo Linux (Ubuntu).

En la figura 8, se muestra la ejecución del comando arp sobre Windows. El formato de impresión es en tabla y se muestran las direcciones según la interfaz. En la figura 9, se muestra la dirección contenida en el caché. Se puede observar que, pese a que se ejecutan operaciones idénticas, el resultado es distinto.

#### 2.1.10. nslookup

Este comando es utilizado en los sistemas operativos Windows y Linux. Dentro de la documentación de Microsoft (2017) y el manual de Linux, se menciona que su principal función es hacer consultas sobre un DNS específico. Estos también mencionan que presenta dos modos que son elegidos antes de realizar la consulta con el comando. En el modo interactivo, la consulta muestra los nombres de servidores para obtener información de los hosts y dominios o la lista de hosts en un dominio. En el modo no interactivo, se muestra el nombre y la información que fue consultada para un dominio o host.

Para poder ingresar al modo interactivo solo basta con el nombre del comando y un enter. En caso de que no se ingrese de esta manera va a modo no interactivo donde se coloca el comando y la dirección. Sintaxis dentro de Windows:

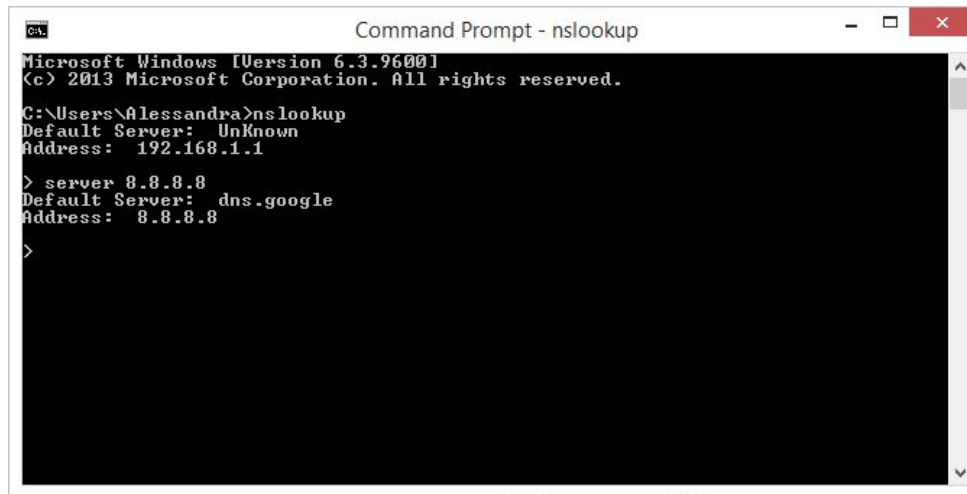
```
nslookup [exit | finger | help | ls | lserver | root | server | set | view] [options]
```

Sintaxis dentro del sistema operativo Linux:

```
nslookup [-option] [name | -] [server]
```

- *exit*: se sale de nslookup.
- *finger*: se conecta al finger server.
- *help*: no se encuentra implementado en Linux pero en Windows muestra los comandos.
- *ls* : muestra en lista información del servidor DNS. Dentro de Linux según su manual no se encuentra implementado.
- *lserver*: cambia el servidor predeterminado al dominio DNS especificado, realiza una consulta con el servidor local.

- *root*: realiza un cambio al servidor raíz del predeterminado. No está implementado en Linux.
- *server*: cambia el servidor predeterminado al dominio DNS especificado.
- *set*: cambia la configuración de las búsquedas.
- *options*: Este se utiliza en conjunto con el comando set. Entre las opciones se encuentran timeout, domain, port, class, querytype, type, entre otros.



```

Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Users\Alessandra>nslookup
Default Server:  Unknown
Address:  192.168.1.1

> server 8.8.8.8
Default Server:  dns.google
Address:  8.8.8.8

>

```

Figura 10. Ejecución del comando nslookup en el sistema operativo Windows.

### 2.1.11. Netcat

El comando para netcat es utilizado tanto en Unix como en Windows. Según Sourceforge (s.f), de donde se puede descargar la herramienta netcat, y Armstrong (2001), explican que es utilizada para escribir y leer datos utilizando conexiones de red con el uso del protocolo UDP y TCP. Esta es una herramienta confiable back-end, además de ser un depurador y explorador de red. Su sintaxis es la siguiente:

nc host puerto

Sourceforge (s.f) menciona que la herramienta crea la conexión al puerto que el usuario haya introducido al igual que el host. Lo que ingresa se envía al host y la respuesta se muestra en la entrada estándar. Además, indica que se puede trabajar como cliente así como servidor, de modo que intercambia la información hasta que la conexión se cierre o no hayan más datos.

Según Villa (2005), esta es una herramienta potente la cual “crea un socket con el destino indicado si es cliente, o en el puerto indicado, si es servidor”. Asimismo, establece que cuando haya una conexión entonces este enviará los datos por el socket, los recibirá y los mostrará en la salida estándar.

En caso de que no se ingresen los datos, estos serán solicitados posteriormente. A continuación, se va a dar una lista de unos cuantos de estos los cuales se extrajo del manual de linux y la página Sourceforge de netcat:

- `-n`: solo acepta IP numéricos.
- `-v`: realiza una búsqueda invertida del nombre y dirección host.
- `-w`: limita el tiempo de tratar de realizar la conexión.
- `-u`: se utiliza para realizar una conexión UDP.
- `-l`: modo escucha.
- `-i`: intervalo de espera entre los datos que se envían y reciben.
- `-p`: utilizar un puerto local en específico.
- `-s`: la IP que es especificado se utiliza para enviar los paquetes.
- `-z`: bandera que se utiliza para saber cuáles puertos están abiertos. Estos pueden ser especificados en un rango.
- `-e`: es utilizado para cuando realiza la conexión ejecuta el comando que fue ingresado.

Para poder utilizar este comando desde la terminal, se debe realizar una descarga del mismo. En Linux (Ubuntu) solo basta con colocar en la terminal ‘`sudo apt-get install netcat`’. En el caso de Windows se descarga dentro de la página SourceForge la versión 1.10.

<pre>File Edit View Search Terminal Help alessandra@AlessandraUbuntu:~\$ nc 192.168.1.13 8080 hola Estoy utilizando la herramienta netcat</pre>	<pre>File Edit View Search Terminal Help alessandra@AlessandraUbuntu:~\$ nc -l 8080 hola</pre>
<pre>File Edit View Search Terminal Help alessandra@AlessandraUbuntu:~\$ nc 192.168.1.13 8080 hola Estoy utilizando la herramienta netcat</pre>	<pre>File Edit View Search Terminal Help alessandra@AlessandraUbuntu:~\$ nc -l 8080 hola Estoy utilizando la herramienta netcat</pre>

Figura 11. Ejecución del comando netcat en el sistema operativo Linux (Ubuntu).

En este ejemplo, se puede ver que al lado derecho se encuentra una terminal como servidor escuchando (-l) en el puerto 8080. Al lado derecho se conecta a ese servidor (local) en el puerto 8080 y comienza a enviar datos, en este caso se puede ver un antes y después de lo que se desea enviar el cual es un texto..

#### 2.1.12. scp

La herramienta SCP (Secure Copy Protocol) tiene como principal función la transferencia de archivos y directorios entre dos computadoras o entre dos sistemas de una misma máquina, similar a RCP utilizada en los sistemas BSD (Lytvyov, 2020). Pese a que en su nombre se hace referencia a un protocolo, no existe ningún “Request for comments” asociado a él. El único protocolo con el que se le vincula es con SSH.

De acuerdo con Lytvynov, al usar el protocolo como tubería para el paso de texto plano, SCP no se preocupa realmente por la seguridad, sino solo por colocar y extraer la información para colocarla donde corresponde. El uso de SSH también obliga a que haya autenticación de ambas computadoras y encriptación de la información, lo que la vuelve una alternativa llamativa.

Recientemente, Harry Sintonen, investigador de ciberseguridad en Finlandia, describió algunos errores en seguridad cuando se implementa este tipo de comunicación (Cimpanu, 2019). En primer lugar, que un cliente de SCP puede permitir que un servidor del mismo tipo modifique los permisos sobre un directorio en específico. En segundo lugar, que un servidor malicioso de SCP puede sobre escribir ciertos archivos y, si la operación “-r” es efectuada, también puede manipular subdirectorios. En tercer lugar, la terminal del cliente para impresión puede ser operada a través de código ANSI, que oculta la operaciones.

SCP no funciona como una conexión, sino como un medio para producir entrada y salida, desde o hacia otra computadora. Para ello, según Lytvynov, existe un servidor que se encuentra corriendo siempre a la espera de solicitudes (sshd); y un cliente que se encarga de recibir o de enviar los archivos (scp). SCP es unidireccional, de modo que, para crear un traslado en sentido contrario con una computadora, se debe abrir otra instancia.

Para poder ser utilizado en Linux (Ubuntu), es necesario que se instale SSH antes, y puede hacerse mediante la ejecución de “sudo apt install openssh-server” en la terminal. En la figura 12 se presenta un ejemplo del uso de scp. Rinne y Ylonen (2018) detallan cómo debe utilizarse el comando y las operaciones que pueden realizarse con él; la información. Dicha información es presentada a continuación.

```
scp [-4BpqrTv] [-c cipher] [-F ssh_config] [-i identity_file] [-l limit] [-o ssh_option] [-P port]
[-S program] source ... target
```

- **-3**: la información transferida entre dos servidores se encamina a través del host local. Esta opción se debe activar o de lo contrario la información se copiará entre ambos.
- **-4 y -6**: fuerza a que se use IPv4 o IPv6 respectivamente.
- **-B**: activa el procesamiento por lotes, en la cual, no se necesita intervención de usuario y, en este caso, no necesitaría el uso de contraseñas.
- **-C**: permite comprimir archivos.
- **-c**: cifrado para encriptar la información para ser transmitida.
- **-F**: especifica un archivo de configuración para ssh.
- **-i**: selecciona el archivo que contiene la llave privada para la llave pública.
- **-l**: limita el ancho de banda.
- **-o**: se usa para asignar opciones a ssh.
- **-P**: especifica el puerto; por defecto, se usa el 22.
- **-p**: conserva datos del archivo original como la cantidad de modificaciones, los tiempos de acceso, etc.
- **-q**: desactiva mensajes tales como nivel de progreso, avisos y diagnósticos del ssh.
- **-r**: utiliza recursión para copiar directorios.
- **-S**: especifica un programa que deberá ser usado para la encriptación, pero que entienda las opciones de ssh.
- **-T**: desactiva la opción que verifica el que no se hayan copiado archivos de más.
- **-v**: hace que se impriman mensajes de depuración durante el proceso.

- Para copiar un archivo en una computadora remota, el comando debe acomodarse de la siguiente forma: “scp <opciones> <archivos o directorios> usuario@host\_romoto:/<folder>”.
- Para copiar un archivo desde una computadora remota a la local, el comando debe colocarse de esta manera: “scp <opciones> usuario@host\_romoto:/archivos <folder de la máquina local>”.



```
macelesteh@Celeste:~$ scp 1.cpp macelesteh@10.0.2.15:/home/macelesteh/Descargas
The authenticity of host '10.0.2.15 (10.0.2.15)' can't be established.
ECDSA key fingerprint is SHA256:/nOFLt0Eki9ZPrqpBQRh/pJB0LoHru1903Kqce7q27o.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.0.2.15' (ECDSA) to the list of known hosts.
macelesteh@10.0.2.15's password:
1.cpp                                100% 812    39.9KB/s   00:00
```

Figura 12. Ejecución del comando scp en el sistema operativo Linux (Ubuntu).

### 2.1.13. nmap

Nmap (Network Mapper) constituye una utilidad que permite la revisión de redes y su seguridad, de acuerdo con Lyon (s.f.), quien es también el creador de la herramienta. El mecanismo tiene muchas funciones, aunque las más populares son: inventario de red, gestión de actualizaciones y monitoreo del tiempo de actividad de los servicios o los hosts. Debido al uso de paquetes de IP, se pueden identificar características de los hosts: disponibilidad; servicios que ofrecen; el sistema operativo que corren; barreras, filtros o firewalls implementados; entre otros.

La salida que produce Nmap es el escaneo de los objetivos especificados, otras características dependen de los otros parámetros insertados. No obstante, siempre se detalla el número de puerto y nombre del protocolo, su estado, el nombre del servicio y versión de este. Existen cuatro estados que puede asignarse Lyon (s.f.): *open*, si la máquina está a la espera de solicitudes; *filtered*, si existe algún tipo de filtro que impide saber si se encuentra abierto o cerrado; *unfiltered*, si responde a las pruebas que hace Nmap, pero sigue siendo incapaz de determinar su verdadero estado, y *closed*, si no se encuentran escuchando, pero pueden ser abiertos en cualquier momento.

Nmap contiene muchas funciones que se clasifican en las siguientes categorías: especificaciones del objetivo, descubrimiento del host, técnicas de escaneo, especificaciones de



puertos y escaneo, detección de versión o servicio, escaneo de datos, detección de sistema operativo, sincronización y desempeño, evasión y engaño de IDs o firewalls, formato de salida y otros.

Para ejecutar esta herramienta, antes se debe instalar el paquete correspondiente. En el caso de Linux es posible a través de lo comando “sudo apt install nmap”. En caso de Windows, el procedimiento es más complejo y se detalla en la página del programa. Mientras que nmap para Linux puede ser ejecutado sin importar el lugar en donde se encuentre, para Windows se debe estar específicamente en el lugar donde se encuentran todos los archivos. La forma de usar el programa en la línea de comando y algunas de las operaciones que se pueden ejecutar se describen a continuación Lyon (s.f.).

nmap [tipo(s) de escaneo] [opciones] {especificación del objetivo}

- *-iL <archivo de entrada>*: obtiene la lista de de hosts o de redes de un archivo.
- *-iR <número de hosts>*: dado un número, se eligen hosts al azar.
- *-sL*: se escanean las direcciones dadas por una lista.
- *-sn*: aplica ping a un puerto.
- *-sO*: escanea el protocolo IP.
- *-p <puertos>*: solo escanea los puertos especificados.
- *-F*: escaneo rápido de solo algunos puertos.
- *-r*: escanea puertos consecutivamente y no de forma aleatoria.
- *-sV*: prueba puertos abiertos para conocer información como el servicio o la versión.
- *-O*: solicita la inspección del sistema operativo.
- *--reason*: muestra la razón por la cual cierto puerto se encuentra en determinado estado.
- *--open*: solo muestra puertos abiertos o que tengan posibilidades de estar abiertos.
- *--packet-trace*: muestra todos los paquetes que fueron enviados y recibidos.
- *--iflist*: imprime la interfaz de los hosts, de forma que sea útil para depurar.
- *--resume <filename>*: reanuda un análisis interrumpido.
- *-A*: permite la detección del sistema operativo, la versión, escaneo de scripts y el traceroute.
- *-V*: imprime el número de versión.

En la figura 13, se observa el empleo del comando en una terminal de Linux sobre una página web, junto a la operación “-A”. En la figura 14, se emplea el comando en Windows, con la diferencia de que el la página es distinta y, en este caso, el host se encuentra apagado o bloqueando la solicitud.

```
macelesteh@Celeste: ~
$ nmap -A es.tldp.org

Starting Nmap 7.60 ( https://nmap.org ) at 2020-09-21 00:25 CST
Nmap scan report for es.tldp.org (150.214.2.36)
Host is up (0.15s latency).
rDNS record for 150.214.2.36: tldp.cica.es
Not shown: 989 filtered ports
PORT      STATE SERVICE VERSION
25/tcp    open  smtp?
|_smtp-commands: Couldn't establish connection on port 25
80/tcp    open  http    Apache httpd
|_http-generator: GNU/Emacs/X11 19.34.1, WML 2.0.8 (30-Oct-2001)
|_http-server-header: Apache
|_http-title: TLDP-ES: P&aacute;gina Principal
110/tcp   open  pop3?
119/tcp   open  nntp?
143/tcp   open  imap?
443/tcp   open  ssl/http Apache httpd
|_http-generator: GNU/Emacs/X11 19.34.1, WML 2.0.8 (30-Oct-2001)
|_http-server-header: Apache
|_http-title: TLDP-ES: P&aacute;gina Principal
|_ssl-cert: Subject: commonName=tldp.cica.es/organizationName=Centro Inform&x3;x
Atitico Client&x3;xADfco de Andaluc&x3;xADa/stateOrProvinceName=Sevilla/countryN
ame=ES
|_ Subject Alternative Name: DNS:tldp.cica.es, DNS:www.tldp.cica.es
|_ Not valid before: 2020-07-09T00:00:00
|_ Not valid after: 2022-07-09T23:59:59
|_ ssl-date: 2020-09-21T06:28:41+00:00; 0s from scanner time.
465/tcp   open  smtps?
|_smtp-commands: Couldn't establish connection on port 465
583/tcp   open  snwps?
587/tcp   open  submission?
|_smtp-commands: Couldn't establish connection on port 587
993/tcp   open  imaps?
995/tcp   open  pop3s?
Service Info: Host: es.tldp.org

Service detection performed. Please report any incorrect results at https://nmap.org/submit/.
Nmap done: 1 IP address (1 host up) scanned in 394.36 seconds
macelesteh@Celeste: ~
```

Figura 13. Ejecución del comando nmap en el sistema operativo Linux (Ubuntu).

```
C:\Program Files\nmap-7.80-win32\nmap-7.80>nmap -v -A os.eccii.ucr.ac.cr
Starting Nmap 7.80 ( https://nmap.org ) at 2020-09-21 00:22 Central America Standard Time
NSE: Loaded 151 scripts for scanning.
NSE: Script Pre-scanning.
Initiating NSE at 00:22
Completed NSE at 00:22, 0.06s elapsed
Initiating NSE at 00:22
Completed NSE at 00:22, 0.00s elapsed
Initiating NSE at 00:22
Completed NSE at 00:22, 0.00s elapsed
Initiating Ping Scan at 00:22
Scanning os.eccii.ucr.ac.cr (163.178.104.187) [4 ports]
Completed Ping Scan at 00:22, 5.34s elapsed (1 total hosts)
Nmap scan report for os.eccii.ucr.ac.cr (163.178.104.187) [host down]
NSE: Script Post-scanning.
Initiating NSE at 00:22
Completed NSE at 00:22, 0.00s elapsed
Initiating NSE at 00:22
Completed NSE at 00:22, 0.00s elapsed
Initiating NSE at 00:22
Completed NSE at 00:22, 0.00s elapsed
Read data files from: C:\Program Files\nmap-7.80-win32\nmap-7.80
Note: Host seems down. If it is really up, but blocking our ping probes, try -Pn
Nmap done: 1 IP address (0 hosts up) scanned in 17.38 seconds
Raw packets sent: 8 (304B) | Rcvd: 0 (0B)

C:\Program Files\nmap-7.80-win32\nmap-7.80>
```

Figura 14. Ejecución del comando nmap en el sistema operativo Windows.

## 2.2. Sniffers

Goyal y Goyal (2017) mencionan que los sniffers son programas que analizan el tráfico de una red, ya sea saliente o entrante. TCPdump y Wireshark son los sniffers más conocidos en el mercado por lo que se van a mostrar a continuación.

### 2.2.1. TCPdump

Esta herramienta se puede adquirir gratis desde su página oficial, está disponible para UNIX ya que para Windows se utiliza la versión WinDUMP. Según Goyal y Goyal (2017) “es una de las mejores herramientas para analizar paquetes de red, se caracteriza por ser intuitiva para los que deseen estudiar acerca de TCP/IP, ya que descarga paquetes sin procesar y sin mucho análisis. Tcpdump proporciona muchas opciones donde se pueden ver los detalles de los paquetes capturados en varios formatos que son fáciles de comprender”.

La principal función de esta herramienta es capturar el tráfico que circula por la red y mostrar una descripción de los contenidos de los paquetes de una interfaz de red. Al termina de ejecutarse el TCPdump hace un conteo de paquetes: capturados (número de paquetes que tcpdump ha recibido y procesado), recibidos por filtro( depende del filtro que se especifique), dejados por el kernel (paquetes dejados a falta de espacio en el buffer). Tcpdump también permite guardar los paquetes deseados para leerlos en cualquier momento.

La información capturada en los paquetes se puede leer como: el primer campo (23:35:07:8268) representa el tiempo de consulta del paquete recibido, seguido por el protocolo de capa de red (en este caso IP), dirección del puerto de origen y después el de destino separado por un “<”. Puede estar seguido de una bandera, un número ack (Acknowledgment) 1 para los que envían información y otra serie de números que representa el siguiente byte esperado de datos, win que representa el número de bytes disponible en el buffer y por último el tamaño del paquete. También pueden contener el número de secuencia de los datos contenidos. Los tipos de banderas se describen a continuación, de acuerdo con Cane(2014).

- [S] - SYN (Start Connection)
- [.] - No Flag Set
- [P] - PSH (Push Data)

- [F] - FIN (Finish Connection)
- [R] - RST (Reset Connection)

```
ale@ale-VirtualBox:~$ tcpdump -i enp0s3
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on enp0s3, link-type EN10MB (Ethernet), capture size 262144 bytes
23:35:07.826856 IP ale-VirtualBox.33706 > 163.178.170.27.https: Flags [P.], seq 199984520:199984559, ack 140045614, win 63900, length 39
23:35:07.827854 IP 163.178.170.27.https > ale-VirtualBox.33706: Flags [.], ack 39, win 65535, length 0
23:35:07.830774 IP ale-VirtualBox.56943 > homerouter.cpe.domain: 25913+ PTR? 27.170.178.163.in-addr.arpa. (45)
23:35:07.919613 IP ale-VirtualBox.32889 > homerouter.cpe.domain: 216+ AAAA? git.ucr.ac.cr. (31)
23:35:07.920524 IP ale-VirtualBox.33706 > 163.178.170.27.https: Flags [P.], seq 39:106, ack 1, win 63900, length 67
```

Figura 15. Ejecución del comando tcpdump en el sistema operativo Linux.

```
ale@ale-VirtualBox:~$ tcpdump -i any -c 5 -vvv
tcpdump: listening on any, link-type LINUX_SLL (Linux cooked v1), capture size 262144 bytes
14:53:13.305550 IP (tos 0x0, ttl 64, id 3090, offset 0, flags [none], proto TCP (6), length 71)
    ec2-54-244-7-161.us-west-2.compute.amazonaws.com.https > ale-VirtualBox.39418: Flags [P.], cksum 0x33e1 (correct), seq 126595955:126595986, ack 252121224, win 65535, length 31
14:53:13.305590 IP (tos 0x0, ttl 64, id 34818, offset 0, flags [DF], proto TCP (6), length 40)
    ale-VirtualBox.39418 > ec2-54-244-7-161.us-west-2.compute.amazonaws.com.https: Flags [.], cksum 0x4abe (incorrect -> 0xf531), seq 1, ack 31, win 63900, length 0
14:53:13.306653 IP (tos 0x0, ttl 64, id 30517, offset 0, flags [DF], proto UDP (17), length 79)
    localhost.35310 > localhost.domain: [bad udp cksum 0xfe82 -> 0x5124!] 28225+ [1au] PTR? 15.2.0.10.in-addr.arpa. ar: . OPT UDPsize=1200 (51)
14:53:13.306880 IP (tos 0x0, ttl 64, id 51974, offset 0, flags [DF], proto UDP (17), length 68)
    ale-VirtualBox.54592 > homerouter.cpe.domain: [bad udp cksum 0xd4f9 -> 0x7ec2!] 63466+ PTR? 15.2.0.10.in-addr.arpa. (40)
14:53:13.307258 IP (tos 0x0, ttl 64, id 34819, offset 0, flags [DF], proto TCP (6), length 75)
    ale-VirtualBox.39418 > ec2-54-244-7-161.us-west-2.compute.amazonaws.com.https: Flags [P.], cksum 0x4ae1 (incorrect -> 0xe0f7), seq 1:36, ack 31, win 63900, length 35
5 packets captured
24 packets received by filter
12 packets dropped by kernel
```

Figura 16. Ejecución del comando tcpdump de forma verbosa en el sistema operativo Linux.

Algunas opciones y comandos:

- *-A*: imprime en ASCII los contenidos del paquete
- *-c* : enumerar, cuenta la cantidad de paquetes.
- *-D* o *--list-interfaces*: muestra las interfaces disponibles del sistema.
- *-i*: identifica la interfaz.
- *-n*: evita que cambie las direcciones por los nombres de los dispositivos asociados.
- *-xx*: imprime en hexadecimal
- *-w archivo.pcap*: almacena los paquetes en la ruta especificada
- *-r archivo.pcap*: lee el archivo de paquetes almacenado
- *-i <interfaz>*: captura paquetes de una interfaz en específico.
- *-c <número> -i <interfaz>* : captura solo un número específico de paquetes.

- *--interface any* : captura los paquetes de todas las interfaces
- *host <dirección>*: encuentra tráfico por IP
- *-i <interfaz> src/dst <dirección>*: captura paquetes de dirección salida o destino IP.
- *src/dst <dirección>*: tráfico en dirección del origen(src) o del destino(dst)
- *net <red>* : cuenta paquetes por una red específica.
- *-i <interfaz> tcp*: captura solo paquetes del puerto TCP
- *-i <interfaz> port <puerto>* : tráfico de un puerto en específico.
- *src port<puerto>*: tráfico de un puerto específico
- *icmp/tcp/udp*: muestra el tráfico de un protocolo
- *ip6*: muestra solo ip6 tráfico.
- *Portrange(xx-xx)*: tráfico en un rango de puertos.

### 2.2.2. Wireshark

Esta herramienta se puede adquirir dentro de su página oficial de Wireshark; se encuentra disponible para los sistemas operativos Windows, MacOS, UNIX y sus derivados. En la página oficial Wireshark (s.f), se describe como una herramienta que analiza protocolos de red, deja ver paquetes que van por la red. Goyal y Goyal (2017) informan que este puede tomar los datos de cualquier red, desde Ethernet, Wifi, Bluetooth y modo monitor. Wireshark no funciona como un detector de intrusión, pero como deja analizar los paquetes, se puede ver si hubo una actividad fuera de lo común.

Esta aplicación contiene información relevante como es la dirección del origen, dirección del destino, el tipo de protocolo usado, el tamaño del paquete, tiempo e información de este. Además tiene un filtro que ayuda a realizar una consulta más específica, como por ejemplo que se haga una búsqueda por el protocolo HTTP. Por otro lado, tiene una interfaz gráfica para que el usuario interactúe con él, al igual se puede usar en una línea de comandos.

Wireshark (s.f) explica que es una herramienta bastante potente ya que en una comunicación TCP puede mostrar los datos intercambiados en ASCII. Todo esto también lo

logra en tiempo real. Dentro del manual de usuario en la aplicación, se menciona que no va a cambiar nada dentro de la red.

Dentro de la pantalla principal, se encuentra una barra superior que contiene las opciones del programa. Estas son explicadas con mayor detalle en el manual de la herramienta. A continuación, se explican alguna de ellas:

- *File*: se pueden abrir archivos capturados.
- *Edit*: se puede copiar, buscar un paquete, marcar paquetes, entre otros.
- *View*: se pueden realizar cambios visuales como los colores de los paquetes.
- *Go*: ir a un paquete en específico.
- *Capture*: iniciar o parar capturas.
- *Analyze*: tiene opciones como manipular filtros, habilitar protocolos.
- *Statistics*: muestra ventanas estadísticas entre ello un resumen de los paquetes que se capturaron.
- *Telephony*: muestra estadísticas relacionadas a la telefonía tipo VoiceIP.
- *Wireless*: muestra estadísticas tipo Bluetooth y wifi.
- *Tools*: herramientas de Wireshark.

Los paquetes pueden presentar diferentes colores, dentro de la opción view > coloring rules se puede ver que significan estos.

The image shows a Wireshark packet capture window titled '\*Wi-Fi'. The main pane displays a list of captured packets with columns for No., Time, Source, Destination, Protocol, Length, and Info. The packets include TCP ACKs, TLSv1.2 application data, ARP requests, and a DNS query. The packet list is as follows:

No.	Time	Source	Destination	Protocol	Length	Info
18	0.142024	192.168.1.13	172.217.3.78	TCP	54	50669 → 443 [ACK] Seq=3239 Ack=1235 Win=6765 Len=0
19	0.145096	192.168.1.13	172.217.3.78	TLSv1.2	93	Application Data
20	0.210732	172.217.3.78	192.168.1.13	TCP	60	443 → 50669 [ACK] Seq=1235 Ack=3270 Win=4415 Len=0
21	0.618804	192.168.1.13	172.217.2.138	TLSv1.2	621	Application Data
22	0.672558	172.217.2.138	192.168.1.13	TCP	54	443 → 50688 [ACK] Seq=1 Ack=568 Win=393 Len=0
23	1.103414	172.217.2.138	192.168.1.13	TLSv1.2	641	Application Data
24	1.103414	172.217.2.138	192.168.1.13	TLSv1.2	256	Application Data
25	1.103414	172.217.2.138	192.168.1.13	TLSv1.2	94	Application Data
26	1.103414	172.217.2.138	192.168.1.13	TLSv1.2	93	Application Data
27	1.103695	192.168.1.13	172.217.2.138	TCP	54	50688 → 443 [ACK] Seq=568 Ack=669 Win=256 Len=0
28	1.120309	192.168.1.13	172.217.2.138	TLSv1.2	93	Application Data
29	1.173071	172.217.2.138	192.168.1.13	TCP	54	443 → 50688 [ACK] Seq=669 Ack=607 Win=393 Len=0
30	2.816391	IntelCor_00:cf:53	Broadcast	ARP	42	Who has 192.168.1.1? Tell 192.168.1.13
31	2.820305	Netgear_2e:b3:ef	IntelCor_00:cf:53	ARP	42	192.168.1.1 is at 28:00:00:2e:b3:ef
32	3.171377	IntelCor_00:cf:53	Broadcast	ARP	42	Who has 192.168.1.1? Tell 192.168.1.13
33	3.174188	192.168.1.13	192.168.1.255	DNS	92	Name query NS 25A7AP<00>
34	3.174756	Netgear_2e:b3:ef	IntelCor_00:cf:53	ARP	42	192.168.1.1 is at 28:00:00:2e:b3:ef

The packet details pane shows the selected packet (Frame 1) with its raw data and protocol structure. The raw data is displayed in hexadecimal and ASCII format.

Figura 17. Ejecución de Wireshark capturando paquetes de Wifi.

### 2.3. Conexión Remota

Con la proliferación de dispositivos conectados a internet, hace algunos años surgió la necesidad de poder acceder a estas unidades, sin importar la ubicación o alcance físico actual que tuviera, siempre y cuando este permaneciera conectado a una red. La introducción de conexiones remotas a internet solventó necesidades como acceso seguro para usuarios y procesos automatizados, transferencias de archivos interactivas y automatizadas, emitir comandos remotos, administrar la infraestructura de red y otros componentes del sistema de misión crítica.

Por ejemplo, una empresa cuya fuerza de trabajo no está unificada en un solo espacio físico, tiene la necesidad de que sus empleados puedan conectarse a su red interna y accedan a datos confidenciales desde cualquier parte del mundo, preferiblemente de una manera segura, por lo que a continuación se mostrarán dos ejemplos de protocolos de conexiones remotas que solucionan este problema,

#### 2.3.1. Telnet

Telnet (Teletype Network) es un protocolo de red para el acceso a una computadora de forma remota. Esta conexión permite el acceso de un cliente a los recursos del servidor, aun cuando haya diferencias entre las terminales de ambos dispositivos (Bonet, s.f., p.1). Este protocolo fue el primero en ser implementado sobre TCP/IP y es el soporte para las interacciones de cliente/servidor de otros protocolos. Además, el puerto predeterminado que le fue asignado fue el 23.

La conexión por Telnet fue la primera aplicación demostrada e instalada en la red de IMPs a finales de 1969. Antes de presentar la primera versión formal del protocolo el 15 de febrero de 1971, se dedicó un año en la formación real del protocolo y otro año para la corrección de errores. La edición final del protocolo fue publicada en 1983 (Khare, 1998, p. 88).

De acuerdo con Postel y Reynolds (1983, p.1), su propósito es garantizar una forma de comunicación general, bidireccional y orientada a bytes. Además, se enfoca en la interconexión entre dispositivos terminales y procesos en terminales, con miras hacia la comunicación de terminal a terminal (vinculación) y la comunicación proceso a proceso (computación distribuida).

El protocolo se sustenta en tres principios: la Terminal Virtual de Red (Network Virtual Terminal o NVT), la negociación de emulación de terminal y la simetría de negociación para procesos y terminales.

La Terminal Virtual de Red es un dispositivo imaginario que proporciona una representación adecuada de una terminal. Es de carácter bidireccional y posee una impresora (responde a datos de entrantes) y un teclado (para datos salientes). La terminal utiliza el conjunto US ASCII de siete bits en campos de ocho bits. A través de ella se efectúa la comunicación, de modo que ni el cliente ni el servidor deben guardar los datos de la terminal del otro (Postel y Reynolds, p.1). Un problema que presenta esta terminal es la latencia, pues es inferior a la de una terminal común debido a la cantidad de puntos de conexión que, en lugar de ser dos, son cuatro.

El proceso de negociación proviene de la necesidad de proporcionar servicios adicionales a los que la NVT provee. Para ello, existen una serie de opciones que pueden ser ejecutadas tanto por el cliente como por el servidor: DO, DON'T, WILL y WON'T; en la Figura 4 se especifica qué sucede con cada par de opciones. Por otro lado, las opciones "SB" y "SE" se utilizan para



iniciar y finalizar una subnegociación, que se utiliza para intercambiar información adicional. Según Bonet (p.7), si ambas partes rechazan todas las formas de negociación, la sesión se mantendrá abierta, pero con las características predeterminadas de la NVT.

Opción	Descripción	Respuesta	Descripción
DO	Pide que se acepte una opción.	WILL	La opción es aceptada y ejecutada.
DO	Pide que se acepte una opción.	WON'T	La opción es rechazada y el estado no cambia.
WILL	Indica la posibilidad de ejecutar una opción.	DO	La posibilidad es aceptada y ejecutada.
WILL	Indica la posibilidad de ejecutar una opción.	DON'T	La posibilidad es rechazada en el instante actual y el estado no cambia.
WILL	Indica la posibilidad de ejecutar una opción.	WON'T	La posibilidad es rechazada para siempre y el estado no cambia.

Figura 18. Descripción de los posibles casos que pueden ocurrir durante la negociación. Bonet, s.f., p.7.

La simetría de la sintaxis de negociación puede inducir a bucles cuando los mensajes de recibo son percibidos como solicitudes. Para evitar esos problemas, se han implementado tres reglas (Postel y Reynolds, p.2). La primera enfatiza en que es preciso solo enviar solicitud para cambiar de estado y no para notificarlo. La segunda indica que, si se recibe una solicitud para cambiar el estado, pero ya se está en él, no debe ser marcada como recibida. La tercera precisa que, si se envía un comando de opción y este tiene repercusiones en el procesamiento de los datos enviados, entonces el comando deberá ser introducido en el flujo de datos, justo en el punto que se desea.

De acuerdo con Postel y Reynolds (p.3), en el caso de los procesos, el ciclos pueden producirse cuando la respuesta a un rechazo es solicitar la opción de nuevo. Para evitar ese problema, lo recomendable es que las solicitudes rechazadas no se repitan hasta que exista un cambio.

Como Telnet fue desarrollado antes del establecimiento del Internet actual, su uso es inseguro. Esta afirmación se debe a que “la autenticación de los usuarios se transmite por la red sin encriptar” (Molina y Polo, 2014, p.423). Además, como Telnet envía el contenido como texto sin formato, cualquiera podría leer lo que se envía. De esta forma, cualquiera podría aprovechar cualquier potencial en el programa para acceder a sistemas restringidos. Es por tanto que se recomienda el uso de SSH que sí encripta los mensajes (Bhutia y Hosch, 2015).

### 2.3.2. SSH

El protocolo SSH (*Secure Shell*) es un método para el inicio de sesión remoto seguro desde una computadora a otra. Proporciona varias opciones alternativas para una autenticación sólida y protege la seguridad e integridad de las comunicaciones con un cifrado robusto. Es una alternativa segura a los protocolos de inicio de sesión no protegidos (como telnet, rlogin) y los métodos de transferencia de archivos inseguros (como FTP), sustituyéndolo con SFTP (*SSH File Transfer Protocol*).

El protocolo funciona en el modelo cliente-servidor, lo que significa que la conexión la establece el cliente SSH que se conecta al servidor SSH. El cliente SSH dirige el proceso de configuración de la conexión y utiliza criptografía, con una clave pública para verificar la identidad del servidor SSH. Después de la fase de configuración, el protocolo SSH utiliza un cifrado simétrico fuerte y algoritmos *hash* para garantizar la privacidad e integridad de los datos que se intercambian entre el cliente y el servidor. Por lo que podemos resumirlos a cuatro simples pasos a un nivel de abstracción muy bajo:

1. El cliente inicia la conexión contactando el servidor.
2. El servidor envía al cliente la llave pública.
3. Bidireccionalmente se negocian los parámetros y se abre un canal de comunicación seguro.
4. El usuario inicia sesión en el servidor que hace *host* al sistema operativo al que se quiere acceder.

Hay varias opciones que se pueden utilizar para la autenticación de usuarios. Los más comunes son las contraseñas y la autenticación de clave pública (SSH key). Inicialmente no se esperaba que el uso de estas llaves fuera tan común pero se han vuelto muy populares, por lo que la administración de llaves SSH se ha vuelto muy importante. Las llaves SSH otorgan acceso como lo hacen los nombres de usuario y las contraseñas y requieren procesos similares de aprovisionamiento y terminación.

### 3. Conclusiones

En la actualidad, existen múltiples herramientas que han tomado popularidad debido a su gran utilidad dentro de las redes de comunicación de datos. A través de la investigación, fue posible identificar la funcionalidad de diversos comandos, sniffers y herramientas para la conexión remota. Por ejemplo, su uso se puede ver reflejado en situaciones como en empresas que no cuentan con instalaciones centralizadas o con la totalidad de sus empleados con acceso a estas, por lo que se debe recurrir a estas alternativas, deseablemente seguras, para la transmisión de datos y acceso de máquinas conectadas a un red específica remotamente.

Asimismo, no se puede dejar de lado la labor de algunas en la resolución de problemas de red. En particular, las herramientas para diagnóstico de problemas de red son importantes para entender a un nivel de abstracción más bajo, cómo son tratados y manejados los datos que se envían y se reciben a través de una red, generalmente internet, y los servicios que podemos acceder mediante ellos.

Los comandos y programas revisados, en su mayoría son amplios: cada uno tiene una cantidad considerable de operaciones a los que el usuario tiene acceso. Si bien en la investigación se mencionan algunos de ellos considerando su importancia y aplicación en el manejo de redes, es importante hacer un estudio a profundidad de su manual de usuario. Además, es importante reconocer que, pese a que una misma herramienta se encuentra disponible para varios sistemas operativos, algunas funciones y banderas pueden tener diferente propósito.

Por último, cabe destacar que muchos de estos instrumentos suelen relacionarse con otros servicios de red. En el caso de nslookup, realiza consultas sobre un DNS específico. Ipconfig puede consultar y eliminar la memoria caché del cliente DNS como así mismo configurar el servicio DHCP. En cuanto a los servicios de almacenamiento, las herramientas TCPdump y Wireshark pueden ver los paquetes que se mueven por la red, este último también analiza los protocolos de red. Por otra parte, además de ser partícipe en la conexión remota, telnet se ha utilizado en ocasiones para trabajar en el diagnóstico de servidores de correo electrónico que utilizan SMTP.

## Referencias

- Albitz, P. y Liu, C. (2006) *DNS and BIND*, 5th Edition. Nutshell Series. O'Reilly and Associates, Inc.
- Armstrong, T. (2001). *Netcat - The TCP/IP Swiss Army Knife*. Recuperado de <https://www.sans.org/reading-room/whitepapers/tools/netcat-tcp-ip-swiss-army-knife-952>
- Berkeley Software Distribution System Manager's Manual. (2020). *Traceroute*. Recuperado de <https://man7.org/linux/man-pages/man8/traceroute.8.html>
- Bhutia, T. y Hosch, W. (2015). *Telnet*. Recuperado de: <https://www.britannica.com/technology/Telnet>
- Bonet, E. (s.f.) *Servicios de acceso remoto I: Telnet*. Recuperado de: [informatica.uv.es/it3guia/AGR/apuntes/teoria/documentos/Telnet.pdf](http://informatica.uv.es/it3guia/AGR/apuntes/teoria/documentos/Telnet.pdf).
- Cane, B. (2014) *A Quick and Practical Reference for tcpdump*. Recuperado de: <https://bencane.com/2014/10/13/quick-and-practical-reference-for-tcpdump/>
- Cimpanu, C. (2019). *SCP implementations impacted by 36-years-old security flaws*. Recuperado de: <https://www.zdnet.com/article/scp-implementations-impacted-by-36-years-old-security-flaws/>
- Dawson, T. y Kirch, O. (2000). *Comprobación de las tablas ARP*. Recuperado de: <http://es.tldp.org/Manuales-LuCAS/GARL2/garl2/x-087-2-iface.verify.arp.html>
- Cherenson, A. (2010). *nslookup(1) - Linux man page*. Recuperado de: <https://linux.die.net/man/1/nslookup>
- Jackson, E. y “Hobbit” (s.f). *nc(1) - Linux man page*. Recuperado de: <https://linux.die.net/man/1/nc>
- Gerardi, R. (2018). *An introduction to using tcpdump at the Linux command line*. Recuperado de: <https://opensource.com/article/18/10/introduction-tcpdump>
- Goyal, P. y Goyal A. (2017). Comparative study of two most popular packet sniffing tools-Tcpdump and Wireshark, *9th International Conference on Computational*

- Intelligence and Communication Networks (CICN)*. Girne, pp. 77-81, doi: 10.1109/CICN.2017.8319360.
- IBM. (s.f). *UNIX domain sockets*. Recuperado de: [https://www.ibm.com/support/knowledgecenter/SSB23S\\_1.1.0.2020/gtpc1/unixsock.html](https://www.ibm.com/support/knowledgecenter/SSB23S_1.1.0.2020/gtpc1/unixsock.html)
- IONOS. (2019). *¿Qué es el ARP (Address Resolution Protocol)?*. Recuperado de: <https://www.ionos.es/digitalguide/servidores/know-how/arp-resolucion-de-direcciones-en-la-red/>
- Khare, R. (1998). TELNET: the mother of all (application) protocols. *IEEE Internet Computing*, 2(3), 88-91, doi: 10.1109/4236.683804.
- Kadhum A. (s.f). *Address Resolution Protocol (ARP)*. Recuperado de: [http://www.uobabylon.edu.iq/eprints/publication\\_5\\_19295\\_300.pdf](http://www.uobabylon.edu.iq/eprints/publication_5_19295_300.pdf)
- Linux manual page. (2008). *ifconfig(8)*. Recuperado de: <https://man7.org/linux/man-pages/man8/ifconfig.8.html>
- Linux manual page. (2020). *inode(7)*. Recuperado de: <https://man7.org/linux/man-pages/man7/inode.7.html>
- Linux manual page. (2011). *ip(8)*. Recuperado de: <https://man7.org/linux/man-pages/man8/ip.8.html>
- Linux manual page. (2014). *netstat(8)*. Recuperado de: <https://man7.org/linux/man-pages/man8/netstat.8.html>
- Lyon, G. (s.f). *Chapter 15. Nmap Reference Guide*. Recuperado de: <https://nmap.org/book/man.html>
- Lytvynov, A. (2020). *SCP - Familiar, Simple, Insecure, and Slow*. Recuperado de: <https://gravitational.com/blog/scp-familiar-simple-insecure-slow/>
- Microsoft. (2017). *ipconfig*. Recuperado de: <https://docs.microsoft.com/es-es/windows-server/administration/windows-commands/ipconfig>
- Microsoft. (2017). *Netstat*. Recuperado de: <https://docs.microsoft.com/en-us/windows-server/administration/windows-commands/netstat>

Microsoft. (2017). *nslookup*. Recuperado de:  
<https://docs.microsoft.com/en-us/windows-server/administration/windows-commands/nslookup>

Microsoft. (2017). *Ping*. Recuperado de:  
<https://docs.microsoft.com/en-us/windows-server/administration/windows-commands/ping>

Molina, F y Polo, E. (2014). *Servicios en Red*. Madrid, España: RA-MA, S.A. Editorial y Publicaciones.

Oracle. (s.f). *Monitoring Network Status With the netstat Command*. Recuperado de:  
[https://docs.oracle.com/cd/E18752\\_01/html/816-4554/ipconfig-142.html](https://docs.oracle.com/cd/E18752_01/html/816-4554/ipconfig-142.html)

Oracle. (s.f). *Man pages section 1M: System Administration Commands - netstat*. Recuperado de  
[https://docs.oracle.com/cd/E26502\\_01/html/E29031/netstat-1m.html#REFMAN1Mnetstat-1m](https://docs.oracle.com/cd/E26502_01/html/E29031/netstat-1m.html#REFMAN1Mnetstat-1m)

Oracle. (2014). *System Administration Commands*. Recuperado de:  
[https://docs.oracle.com/cd/E36784\\_01/html/E36871/ifconfig-1m.html](https://docs.oracle.com/cd/E36784_01/html/E36871/ifconfig-1m.html)

Petters J. (2020). *How to Use Nmap: Commands and Tutorial Guide*. Recuperado de:  
<https://www.varonis.com/blog/nmap-commands/>

Postel, J. y Reynolds J. (Eds.). (1983). *Telnet Protocol Specification*. Recuperado de:  
<https://tools.ietf.org/html/rfc854>

Rinne, T. y Ylonen, T. (2018). *scp(1)*. Recuperado de:  
<https://www.freebsd.org/cgi/man.cgi?query=scp&sektion=1>

Rouse, M. (s.f). *Command*. Recuperado de:  
<https://searchwindowsserver.techtarget.com/definition/command>

Sourceforge. (s.f). *Netcat 1.10*. Recuperado de <https://nc110.sourceforge.io/>

Tcpdump.org (s.f) *TCPDUMP Manual Page*. Recuperado de:  
<https://www.tcpdump.org/manpages/tcpdump.1.html>

Ubuntu Manpage Repository. (2019). *arp*. Recuperado de:  
<http://manpages.ubuntu.com/manpages/bionic/es/man7/arp.7.html>

- Ubuntu Manpage Repository. (2019). *telnet*. Recuperado de:  
<http://manpages.ubuntu.com/manpages/bionic/es/man1/telnet.1.html>
- Ubuntu Manuals. (s.f). *Ubuntu Manpage Repository*. Recuperado de:  
<http://manpages.ubuntu.com/manpages/bionic/es/man8/ifconfig.8.html>
- Villa, D. (2005). *Netcat, la navaja suiza de TCP/IP*. Recuperado de:  
<https://crysol.org/recipe/2005-10-10/netcat-la-navaja-suiza-de-tcp-ip.html#.X2cF4xBKjIU>
- Wijesinghe, N. (2019). *Linux Netstat Commands with Examples*. Recuperado de:  
<https://linuxide.com/linux-how-to/linux-netstat-commands-basic-advanced-examples/>
- Wireshark. (s.f). *Wireshark User's Guide*. Recuperado de:  
[https://www.wireshark.org/docs/wsug\\_html/#ChIntroPurposes](https://www.wireshark.org/docs/wsug_html/#ChIntroPurposes)
- Ylonen, T. (1996). "SSH - Secure Login Connections over the Internet, Proceedings of the 6th USENIX Security Symposium, pp. 37-42, USENIX.