📚Networking Commands 📚

1. nmap:

nmap scanme.nmap.org

1. This command scans the host “scanme.nmap.org” to discover open ports and services.

2. tcpdump:

sudo tcpdump -i eth0 tcp port 80

2. This command captures and displays traffic on interface eth0, specifically for TCP traffic on port 80 (HTTP).

3. lsof:

lsof -i TCP:22

3. This command lists processes that are using TCP port 22 (SSH).

4. iftop:

sudo iftop -n -i eth0

4. This command displays real-time network bandwidth usage on interface eth0.

5. iwconfig (Linux):

iwconfig wlan0

5. This command displays the configuration of the wireless interface wlan0.

6. dig:

dig example.com

6. This command queries DNS servers to retrieve information about the domain “example.com”.

7. route (Linux):

route -n

7. This command displays the routing table on a Linux system.

8. ss:

ss -tuln

8. This command shows all TCP and UDP listening ports on the system.

9. telnet:

telnet example.com 80

9. This command establishes a connection to the HTTP service (port 80) on the server “example.com” using Telnet.

10. netsh (Windows):

netsh interface ipv4 show addresses

11. arp:

arp -a

11. This command displays the ARP cache, showing the mappings between IP addresses and MAC addresses on your local network.

12. ipconfig (Windows) / ifconfig (Linux/macOS):

ipconfig /all

12. or

ifconfig -a

12. These commands display detailed information about all network interfaces on your system.

13. traceroute (Windows) / traceroute (Linux/macOS):

traceroute google.com

13. This command traces the route that packets take to reach the specified destination (in this case, “google.com”).

14. curl:

curl https://www.example.com

14. This command retrieves the content of a web page from the specified URL using the HTTP protocol.

15. wget:

wget https://www.example.com/file.txt

15. This command downloads a file from the specified URL using HTTP, HTTPS, or FTP.

16. ssh:

ssh username@hostname

16. This command establishes an SSH connection to the specified hostname with the provided username.

17. netstat:

netstat -ano

17. This command displays active network connections, listening ports, and related information.

18. ip (Linux):

ip addr show

18. This command displays IP addresses and related information for all network interfaces.

19. iwconfig (Linux):

iwconfig

19. This command displays wireless network interface configuration.

20. nslookup:

```
nslookup example.com
```

20. This command queries DNS servers to retrieve information about the specified domain.

21. iftop:

```
sudo iftop -i eth0
```

21. This command displays a real-time view of network bandwidth usage on interface eth0.

22. nmap:

```
nmap -sS target_ip
```

22. This command performs a TCP SYN scan on the specified target IP address.

23. tcpdump:

```
sudo tcpdump -i eth0 port 80
```

23. This command captures and displays HTTP traffic on interface eth0.

24. lsof:

lsof -i :port_number

24. This command lists processes that are using the specified port number.

25. ss:

ss -tulnp

25. This command shows TCP and UDP listening ports along with the associated process names.

26. dig:

dig +short example.com

26. This command retrieves the IP address of the specified domain in a concise format.

27. route (Linux):

route -n

27. This command displays the kernel routing table in a numeric format.

28. telnet:

telnet example.com 22

28. This command establishes a Telnet connection to the specified hostname and port number.

29. netsh (Windows):

netsh interface ipv4 show interfaces

29. This command displays a list of network interfaces with their state and metrics.

30. ipconfig (Windows) / ifconfig (Linux/macOS):

ipconfig /flushdns

30. or

sudo ifconfig eth0 down

30. These commands respectively flush the DNS resolver cache on Windows or bring down the eth0 network interface on Linux/macOS.

31. scp:

scp file.txt username@hostname:/remote/directory

31. This command securely copies a file from the local system to a remote system using SSH.

32. sftp:

sftp username@hostname

32. This command establishes an interactive FTP-like session for transferring files securely over SSH.

33. iptraf-ng:

sudo iptraf-ng

33. This command starts an interactive ncurses-based tool for monitoring network traffic in real-time.

34. mtr:

mtr google.com

34. This command combines the functionality of traceroute and ping to provide detailed network diagnostics.

35. arping:

```
arping -c 3 192.168.1.1
```

35. This command sends ARP requests to a specific IP address to determine if it’s reachable on the local network.

36. route (Windows):

```
route print
```

36. This command displays the routing table on a Windows system.

37. nsupdate:

```
nsupdate -k /path/to/keyfile
```

37. This command interactively updates DNS records using the DNS UPDATE protocol.

38. nmcli (Linux):

```
nmcli device show
```

38. This command displays information about network devices and their configuration using NetworkManager.

39. host:

host 8.8.8.8

39. This command performs DNS lookups to retrieve domain names associated with an IP address.

40. curl with headers:

curl -I https://www.example.com

40. This command retrieves only the HTTP headers from the specified URL using curl.

41. wget with bandwidth limit:

wget --limit-rate=100k https://www.example.com/file.txt

41. This command downloads a file from the specified URL with a specified bandwidth limit.

42. ssh with port forwarding:

ssh -L 8080:localhost:80 username@hostname

42. This command establishes an SSH connection with local port forwarding, forwarding traffic from port 8080 on the local machine to port 80 on the remote machine.

43. netcat (nc):

```
nc -l -p 1234
```

43. This command listens on port 1234 for incoming connections, useful for testing network connectivity.

44. traceroute with ICMP:

```
traceroute -I google.com
```

44. This command performs traceroute using ICMP echo requests instead of UDP packets.

45. tcpdump with specific source IP:

```
sudo tcpdump src host 192.168.1.100
```

45. This command captures packets with a specific source IP address.

46. iftop with filtering:

```
sudo iftop -i eth0 -f "src net 192.168.0.0/16"
```

46. This command displays bandwidth usage only for traffic originating from the specified network.

47. nmap with OS detection:

```
nmap -O target_ip
```

47. This command performs an Nmap scan with operating system detection on the specified target IP address.

48. nslookup with specific DNS server:

```
nslookup example.com 8.8.8.8
```

48. This command performs a DNS lookup using the specified DNS server (in this case, Google’s public DNS server).

49. iptraf-ng with filter:

```
sudo iptraf-ng -B -L /path/to/logfile
```

49. This command starts iptraf-ng with bandwidth monitoring and logs the output to a specified file.

50. arping with interface:

arping -I eth0 192.168.1.1

50. This command sends ARP requests through the specified network interface.

51. iptables (Linux):

sudo iptables -L

51. This command lists all current firewall rules configured using iptables on a Linux system.

52. route print (Windows):

route print

52. This command displays the IPv4 routing table on a Windows system, showing the network destinations, gateways, and interface metrics.

53. ipfw (macOS):

sudo ipfw list

53. This command lists the current firewall rules configured using ipfw on a macOS system.

54. ipfw (FreeBSD):

```
sudo ipfw list
```

54. This command lists the firewall rules configured using ipfw on a FreeBSD system.

55. netcat (nc) file transfer:

```
nc -l -p 1234 > received_file
```

55. This command listens on port 1234 and saves the received data to a file named “received_file”.

56. scp with specific port:

```
scp -P 2222 file.txt username@hostname:/remote/directory
```

56. This command securely copies a file to a remote system using SSH on port 2222.

57. ipset:

```
sudo ipset list
```

57. This command displays the current IP sets configured on the system.

58. ssh-keygen:

```
ssh-keygen -t rsa -b 4096
```

58. This command generates an RSA SSH key pair with a key size of 4096 bits.

59. sshd_config (OpenSSH):

```
sudo nano /etc/ssh/sshd_config
```

59. This command opens the OpenSSH server configuration file for editing.

60. dig with specific DNS server:

```
dig @8.8.4.4 example.com
```

60. This command performs a DNS lookup using the specified DNS server (in this case, Google's public DNS server 8.8.4.4).

61. netstat with specific protocol:

```
netstat -tuln
```

61. This command displays listening TCP and UDP ports.

62. ss with specific state:

```
ss -t state established
```

62. This command displays established TCP connections.

63. traceroute with maximum hops:

```
traceroute -m 20 google.com
```

63. This command traces the route to Google with a maximum of 20 hops.

64. curl with output to file:

```
curl -o output.txt https://www.example.com
```

64. This command downloads the content of a web page and saves it to a file named “output.txt”.

65. wget with recursive download:

```
wget -r -np https://www.example.com/directory/
```

65. This command recursively downloads all files from the specified directory on a website.

66. scp with recursive copy:

```
scp -r local_directory username@hostname:/remote/directory
```

66. This command securely copies a directory and its contents to a remote system using SSH.

67. sftp with batch mode:

```
sftp -b batchfile.txt username@hostname
```

67. This command performs batch mode file transfers using SFTP.

68. nmcli (NetworkManager) with connection status:

```
nmcli connection show
```

68. This command displays the status of NetworkManager connections.

69. ipfs (InterPlanetary File System):

```
ipfs cat /ipfs/QmHash/file.txt
```

69. This command retrieves a file from the InterPlanetary File System (IPFS) using its hash.

70. iperf3 (network performance testing):

```
iperf3 -c server_ip
```

70. This command tests network bandwidth between the local machine and a specified server.

71. curl with POST request:

```
curl -X POST -d 'param1=value1&param2=value2' https://api.example.com
```

71. This command sends a POST request with form data to a specified API endpoint.

72. wget with user-agent header:

```
wget --user-agent="Mozilla/5.0" https://www.example.com
```

72. This command downloads a web page pretending to be a Mozilla browser.

73. ssh with specific private key:

```
ssh -i /path/to/private_key username@hostname
```

73. This command connects to a remote server using SSH with a specific private key.

74. netcat (nc) with listening mode:

```
nc -l -p 1234
```

74. This command listens for incoming connections on port 1234.

75. traceroute with IPv6:

```
traceroute6 ipv6.google.com
```

75. This command traces the route to Google’s IPv6 address.

76. nslookup with reverse lookup:

```
nslookup 8.8.8.8
```

76. This command performs a reverse DNS lookup for the IP address 8.8.8.8.

77. ipfw with port forwarding:

```
sudo ipfw add 100 fwd 192.168.1.2,80 tcp from any to any 8080
```

77. This command forwards incoming TCP traffic on port 8080 to port 80 of the specified IP address.

78. iptables with NAT:

```
sudo iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
```

78. This command configures iptables to perform network address translation (NAT) on outgoing packets on interface eth0.

79. arp-scan:

```
sudo arp-scan --localnet
```

79. This command scans the local network for live hosts using ARP.

80. sshd with password authentication disabled:

```
sudo nano /etc/ssh/sshd_config
```

80. Edit the SSH server configuration file to disable password authentication.

81. netcat (nc) with file transfer:

```
nc -l -p 1234 < file.txt
```

81. This command listens on port 1234 and sends the contents of file.txt to any connecting client.

82. scp with specific cipher:

```
scp -c aes256 file.txt username@hostname:/remote/directory
```

82. This command securely copies a file to a remote system using the AES-256 encryption cipher.

83. sftp with public key authentication:

```
sftp -i /path/to/private_key username@hostname
```

83. This command performs SFTP file transfers using public key authentication.

84. nmcli with Wi-Fi connection details:

```
nmcli device wifi list
```

84. This command lists available Wi-Fi networks and their signal strengths.

85. ipfs with file sharing:

```
ipfs add file.txt
```

85. This command adds a file to the IPFS network for sharing.

86. iperf3 with specific port:

```
iperf3 -c server_ip -p 5001
```

86. This command tests network bandwidth using port 5001 instead of the default port.

87. curl with cookies:

```
curl --cookie "session_id=123456" https://www.example.com
```

87. This command sends an HTTP request with a cookie named “session_id” set to “123456”.

88. wget with retry option:

```
wget --tries=3 https://www.example.com/file.txt
```

88. This command retries the download three times if it fails.

89. ssh with X11 forwarding:

```
ssh -X username@hostname
```

89. This command establishes an SSH connection with X11 forwarding enabled for GUI applications.

90. netcat (nc) with UDP:

```
nc -u -l -p 1234
```

90. This command listens for incoming UDP packets on port 1234.

91. traceroute with specific interface:

```
traceroute -i eth0 google.com
```

91. This command traces the route to Google using the specified network interface.

92. nslookup with specific DNS server and record type:

```
nslookup -type=mx example.com 8.8.8.8
```

93. ipfw with port range:

```
sudo ipfw add 100 allow tcp from any to any 8000-9000
```

93. This command allows TCP traffic on ports 8000 to 9000.

94. iptables with specific source IP:

```
sudo iptables -A INPUT -s 192.168.1.100 -j DROP
```

94. This command drops all incoming packets from the specified source IP address.

95. arp-scan with specific network interface:

```
sudo arp-scan --interface=eth0 --localnet
```

95. This command scans the local network using the eth0 interface.

96. sshd with specific listening address:

```
sudo nano /etc/ssh/sshd_config
```

96. Edit the SSH server configuration file to listen on a specific address.

97. netcat (nc) with verbose output:

```
nc -l -v -p 1234
```

97. This command listens on port 1234 and provides verbose output.

98. scp with verbose output:

```
scp -v file.txt username@hostname:/remote/directory
```

98. This command securely copies a file to a remote system and provides verbose output.

99. sftp with verbose output:

```
sftp -v username@hostname
```

99. This command establishes an SFTP session and provides verbose output.

100. nmcli with VPN connections:

```
nmcli connection show --active
```

100. This command displays active network connections, including VPN connections.

101. ipfs with file sharing and pinning:

```
ipfs pin add QmHash
```

101. This command pins a file to ensure it remains available on the IPFS network even if the original uploader goes offline.

102. iperf3 with parallel streams:

```
iperf3 -c server_ip -P 5
```

102. This command tests network bandwidth using 5 parallel streams.

103. curl with basic authentication:

```
curl -u username:password https://api.example.com
```

103. This command sends an HTTP request with basic authentication credentials.

104. wget with quiet mode:

```
wget -q https://www.example.com/file.txt
```

104. This command downloads a file quietly without showing progress or messages.

105. ssh with port forwarding and SOCKS proxy:

```
ssh -D 8080 -f -C -q -N username@hostname
```

105. This command establishes an SSH connection with dynamic port forwarding and starts a SOCKS proxy on port 8080.

106. netcat (nc) with verbose output and listening on all interfaces:

```
nc -l -v -p 1234 -k
```

106. This command listens on port 1234 on all available interfaces and provides verbose output.

107. traceroute with ICMP and specific source address:

```
traceroute -i eth0 -s 192.168.1.100 google.com
```

107. This command traces the route to Google using ICMP packets with a specific source IP address.

108. nslookup with reverse lookup and specific DNS server:

```
nslookup 8.8.8.8 8.8.8.8
```

108. This command performs a reverse DNS lookup for the IP address 8.8.8.8 using Google's public DNS server.

109. ipfw with logging:

```
sudo ipfw add 100 allow tcp from any to any 22 log
```

109. This command allows TCP traffic on port 22 and logs it.

110. iptables with rate limiting:

```
sudo iptables -A INPUT -p tcp --dport 22 -m state --state NEW -m limit --limit 3/minute --limit-burst 3 -j ACCEPT
```

110. This command limits the rate of incoming SSH connections to 3 per minute.

111. arp-scan with output to file:

```
sudo arp-scan --localnet > scan_results.txt
```

111. This command scans the local network and saves the results to a file named “scan_results.txt”.

112. sshd with specific listening port:

```
sudo nano /etc/ssh/sshd_config
```

112. Edit the SSH server configuration file to listen on a specific port.

113. netcat (nc) with data transfer in hex format:

```
echo "48656C6C6F20576F726C64" | xxd -r -p | nc -l -p 1234
```

113. This command listens on port 1234 and transfers data in hexadecimal format.

114. scp with compression:

```
scp -C file.txt username@hostname:/remote/directory
```

114. This command securely copies a file to a remote system using compression.

115. sftp with specific SSH key:

```
sftp -i /path/to/private_key username@hostname
```

115. This command establishes an SFTP session using a specific SSH key for authentication.

116. nmcli with specific connection details:

```
nmcli connection show "Wired connection 1"
```

116. This command displays details about a specific NetworkManager connection.

117. ipfs with file sharing and encryption:

```
ipfs add --encrypt file.txt
```

117. This command adds a file to the IPFS network with encryption.

118. iperf3 with UDP and specific port:

```
iperf3 -c server_ip -u -p 5001
```

118. This command tests UDP bandwidth using port 5001.

119. curl with multipart form data:

```
curl -F "file=@/path/to/file.txt" https://api.example.com/upload
```

119. This command sends a POST request with multipart form data, including a file upload.

120. wget with timestamping:

```
wget -N https://www.example.com/file.txt
```

120. This command downloads a file only if it is newer than the local copy, based on timestamps.

121. ssh with X11 forwarding and specific display:

```
ssh -X -o "ForwardX11Display=localhost:0" username@hostname
```

121. This command establishes an SSH connection with X11 forwarding and specifies the display.

122. netcat (nc) with verbose output and listening on specific IP address:

```
nc -l -v -p 1234 -s 192.168.1.100
```

122. This command listens on port 1234 on the specified IP address and provides verbose output.

123. traceroute with specific timeout:

```
traceroute -w 2 google.com
```

123. This command traces the route to Google with a timeout of 2 seconds for each probe.

124. nslookup with specific DNS server and record type:

```
nslookup -type=txt example.com 8.8.8.8
```

124. This command performs a DNS lookup for TXT records for the domain “example