|  |  |
| --- | --- |
| # Get all network interfaces | ip a  ifconfig |
| # Ping a hostname | ping <hostname> |
| # Traceroute to a hostname or IP address | traceroute <hostname\_or\_ip> |
| # Check routing table | ip route |
| # Get DNS information | cat /etc/resolv.conf |
| # Display ARP table | arp -a |
| # Add a static ARP entry | sudo arp -s <ip\_address> <mac\_address> |
| # Delete an ARP entry | sudo arp -d <ip\_address> |
| # Ping with a specific count | ping -c <count> <ip\_address> |
| # Ping with a specific interval | ping -i <interval> <ip\_address> |
| # Ping with a specific packet size | ping -s <size> <ip\_address> |
| # Connect to a port using Netcat | nc <hostname\_or\_ip> <port> |
| # Start tcpdump on interface eth0 | tcpdump -i eth0 |
| # Start tcpdump for ICMP packets on eth0 | tcpdump -i eth0 icmp |
| # Save tcpdump output to a file | tcpdump -i eth0 -w capture.pcap |
| # Read a saved tcpdump capture file | tcpdump -r capture.pcap |
| # Scan ports in a specific range using nmap | nmap -p 1-65535 <ip\_address> |
| # Scan for services on a host using nmap | nmap -sV <ip\_address> |
| # Check current iptables rules | sudo iptables -L |
| # Block incoming traffic on port 22 | sudo iptables -A INPUT -p tcp --dport 22 -j DROP |
| # Block traffic from a specific IP | sudo iptables -A INPUT -s <ip\_address> -j DROP |
| # Save iptables rules | sudo iptables-save > /etc/iptables/rules.v4 |
| # Restore iptables rules | sudo iptables-restore < /etc/iptables/rules.v4 |
| # Query a domain using dig | dig <domain> |
| # Query an A record for google.com using dig | dig google.com A |
| # Query a domain using host | host <domain> |
| # Display network connections using netstat | netstat -tuln |
| # Display TCP sockets | ss -t |
| # Display listening ports | ss -l |
| # Set up local port forwarding over SSH | ssh -L <local\_port>:<remote\_host>:<remote\_port> user@<ssh\_host> |
| # Forward port 80 to 8080 using iptables | sudo iptables -t nat -A PREROUTING -p tcp --dport 80 -j REDIRECT --to-port 8080 |
| # Check SMB vulnerabilities on port 445 using nmap | nmap --script smb-vuln-ms17-010 -p 445 <target> |
| # Run an exploit script using nmap | nmap --script <script:exploit> <target> |
| # Detect firewall rules using nmap | nmap -sA <target> |
| # Rate limit connections to port 80 (5/sec) using iptables | sudo iptables -A INPUT -p tcp --dport 80 -m limit --limit 5/sec -j ACCEPT |
| # Tools for traffic manipulation | bettercap  ettercap |
| # Port knocking | knockd |
| # Run Snort in console mode | sudo snort -A console -q -c /etc/snort/snort.conf -i eth0 |
| # Start OpenVPN with a configuration file | sudo openvpn --config <config\_file.ovpn> |
| # Start DNSChef with a fake IP and interface | sudo dnschef --fakeip <fake\_ip> --interface <interface> |
| # DNS spoofing on interface eth0 | dnsspoof -i eth0 |
| # Set up a SOCKS proxy on port 8080 using SSH | ssh -D 8080 user@remote-host |
| # Set up reverse tunneling over SSH  # Secure copy a file to a remote host | ssh -R 8080:localhost:80 user@remote-host  scp <file> user@remote\_host:/path/to/destination |
| # Synchronize a file to a remote host using rsync | rsync -avz <file> user@remote\_host:/path/to/destination |
| # Network utility tools | netcat  hping3  firewalk  iw  nmcli |
| # Wireless tools | airmon-ng  john  pyrit  wifite |
| # Manage network namespaces | ip netns |