Get all network interfaces

ipconfig /all

Ping a hostname

ping <hostname>

Trace route to a hostname or IP address

tracert <hostname\_or\_ip>

Check routing table

route print

Get DNS information

type %SystemRoot%\System32\drivers\etc\hosts

Display ARP table

arp -a

Add a static ARP entry

arp -s <ip\_address> <mac\_address>

Delete an ARP entry

arp -d <ip\_address>

Ping with a specific count

ping -n <count> <ip\_address>

Ping with a specific interval (in milliseconds)

ping -n <count> -w <interval> <ip\_address>

Ping with a specific packet size

ping -n <count> -l <size> <ip\_address>

Connect to a port using Netcat (install Netcat for Windows)

nc <hostname\_or\_ip> <port>

Capture network packets (requires Wireshark or similar)

Example using tshark (Wireshark CLI tool)

tshark -i <interface>

Display open ports and active connections

netstat -ano

Scan for open ports using nmap (if installed)

nmap -p 1-65535 <ip\_address>

Scan for services on a host using nmap

nmap -sV <ip\_address>

Block incoming traffic on a port using Windows Firewall

netsh advfirewall firewall add rule name="Block Port 22" dir=in action=block protocol=TCP localport=22

Block traffic from a specific IP

netsh advfirewall firewall add rule name="Block IP" dir=in action=block remoteip=<ip\_address>

Save current Windows Firewall rules (export to file)

netsh advfirewall export "C:\firewall\_rules.wfw"

Restore Windows Firewall rules

netsh advfirewall import "C:\firewall\_rules.wfw"

Query a domain using nslookup

nslookup <domain>

Query an A record for a domain using nslookup

nslookup -type=A <domain>

Display network connections using netstat

netstat -ano

Check active TCP connections using PowerShell

Get-NetTCPConnection

Set up local port forwarding over SSH (requires SSH client like PuTTY or Windows OpenSSH)

ssh -L <local\_port>:<remote\_host>:<remote\_port> <username>@<ssh\_host>

Forward port 80 to 8080 using netsh (HTTP traffic redirection)

netsh interface portproxy add v4tov4 listenport=80 listenaddress=0.0.0.0 connectport=8080 connectaddress=127.0.0.1

Check SMB vulnerabilities on port 445 using nmap (if installed)

nmap --script smb-vuln-ms17-010 -p 445 <target>

Detect firewall rules using nmap

nmap -sA <target>

Rate limit connections to port 80 (not natively available in Windows; requires 3rd-party tools)

Tools for traffic manipulation (requires installation):

Bettercap (Linux-oriented, may not be directly compatible with Windows)

Ettercap (Windows version may require additional setup)

Run Snort in console mode (requires installation)

snort -A console -q -c "C:\path\to\snort.conf" -i <interface>

Connect to a VPN using OpenVPN (requires installation)

openvpn --config <config\_file.ovpn>

DNS spoofing using dnsspoof (requires installation)

dnsspoof -i <interface>

Set up a SOCKS proxy on port 8080 using SSH

ssh -D 8080 <username>@<remote\_host>

Set up reverse tunneling over SSH

ssh -R 8080:localhost:80 <username>@<remote\_host>

Secure copy a file to a remote host using scp (requires SSH client)

scp <file> <username>@<remote\_host>:<path\_to\_destination>

Synchronize files to a remote host using rsync (requires installation)

rsync -avz <file> <username>@<remote\_host>:<path\_to\_destination>

Wireless tools for Windows:

Use tools like Aircrack-ng, Wireshark, or NetSpot (requires installation)

Manage network namespaces (not natively supported in Windows; requires Hyper-V or WSL)

Example using Hyper-V:

New-VMSwitch -Name <SwitchName> -NetAdapterName <AdapterName> -AllowManagementOS $true