
100 Essentiële Netwerkcommando's (NL)

01	Connectiviteit en basisdiagnose (Windows)	20	Pakketdiagnose
02	Connectiviteit en basisdiagnose (Linux)	21	Hardware en Interfacestatus
03	Routing en padanalyse (Windows)	22	IP-adressen en interfaces
04	Routing en padanalyse (Linux)	23	Routering
05	Routing & Padanalyse (Linux)	24	DNS en naamomzetting
06	Interface- & Adresconfiguratie (Windows)	25	Domein- en IP-registratie
07	Linux · netwerkkaarten en adressen	26	Firewall en Beveiligingsregels
08	Interface- & Adresconfiguratie (Linux)	27	Windows Firewall
09	DNS · Windows en Linux	28	Draadloze netwerken
10	DNS · Linux	29	Lokaal netwerk en ARP
11	DNS & Naamomzetting (Linux)	30	Routering en padanalyse
12	Poorten en verbindingen · Windows	31	Lokale connectiviteitstest
13	Poorten en verbindingen · Linux	32	Lokale socketdiagnose
14	Routing & Padanalyse (Windows)	33	Verkeersdiagnose
15	Poorten en diensten · Linux	34	Bandbreedte
16	Webdiensten · Linux en Windows	35	Interfacebeheer
17	Webdiensten · Windows PowerShell	36	Netwerkadapters
18	Toegang op afstand · Linux	37	IP-configuratie
19	Beheer op afstand	38	DNS-diagnose

39	DNS-configuratie	49	Fysieke verbinding
40	Verbindingsdiagnose	50	Netwerkhardware
41	Poortdiagnose	51	Fysieke Link- & Hardwarestatus
42	Verbindingen en poorten	52	Bandbreedte- & Verkeersmonitoring
43	Poort- en procesdiagnose	53	Netwerkinterface en foutdiagnose
44	Lokaal netwerk	54	Herstel en probleemoplossing
45	Lokale netwerkdiagnose	55	Firewallstatus
46	Netwerkinterfaces	56	Firewalldiagnose
47	Netwerkbeheer	57	Connectiviteit
48	Draadloze netwerkdiagnose		

Connectiviteit en basisdiagnose (Windows)

1 ping (ICMP-echo — Windows)

Controleert de bereikbaarheid op IP-niveau door ICMP-echoverzoeken te versturen.

Syntaxis: ping [-t] [-a] [-n aantal] [-l grootte] host. Voorbeeld: ping -n 4 8.8.8.8 verstuurt vier verzoeken naar dit IP-adres. Bekijk de antwoorden om reactietijden en eventueel pakketverlies te beoordelen.

Connectiviteit en basisdiagnose (Linux)

2 ping (ICMP-echo — Linux)

Controleert de bereikbaarheid van een host met ICMP-echoverzoeken.

Syntaxis: ping [-c aantal] [-i interval] [-s pakketgrootte] bestemming. Voorbeeld: ping -c 5 1.1.1.1 verstuurt vijf verzoeken en stopt daarna, ook als er geen antwoorden komen.

Routing en padanalyse (Windows)

3 tracert (route traceren — Windows)

Toont de hops op de route naar een doeladres.

Syntaxis: tracert [-d] [-h max_hops] [-w time-out] doel. Voorbeeld: tracert -d 192.168.1.1 toont het pad naar de gateway zonder omgekeerde DNS-opzoeken.

4 pathping (pad- en verliesanalyse — Windows)

Traceert een route en verzamelt gegevens waarmee pakketverlies en reactietijden per hop kunnen worden beoordeeld.

Syntaxis: pathping [-n] [-h max_hops] [-p periode] doel. Voorbeeld: pathping -n 1.1.1.1 meet het pad naar dit IP-adres. De analyse kan enige tijd duren.

Routing en padanalyse (Linux)

5 traceroute (route traceren — Linux)

Toont de hops op de route naar een bestemming. De gebruikte probes kunnen per implementatie of optie verschillen.

Syntaxis: `traceroute [-n] [-m max_ttl] [-q probes] bestemming`. Voorbeeld: `traceroute -n 8.8.8.8` toont hopadressen numeriek, zonder DNS-opzoeken.

Routing & Padanalyse (Linux)

6 mtr (Realtime Padanalyse - Linux)

`mtr [-n] [-c aantal] [--report] bestemming`

`mtr --report -c 10 8.8.8.8` genereert een rapport van tien cycli met verliespercentages en responstijden per knooppunt.

7 ip route (Routingstabel - Linux)

`ip route show`

`ip route show` toont de standaard gateway (default route), subnetroutes en de gekoppelde interfaces.

Interface- & Adresconfiguratie (Windows)

8 ipconfig (Interfaceconfiguratie - Windows)

`ipconfig [/all] [/renew] [/release] [/flushdns]`

`ipconfig /all` toont gedetailleerde configuratie inclusief MAC-adres, DHCP-server, leasetijd en actieve DNS-servers.

9 arp -a (ARP-tabel Inspectie - Windows)

`arp -a [-N interface_ip]`

`arp -a` toont de actuele ARP-cache met bekende IP-naar-MAC-toewijzingen op het lokale netwerksegment.

Linux · netwerkkaarten en adressen

10 ip addr (adressen en netwerkkaarten)

Syntaxis: `ip addr show [dev interface]`. Toont IPv4- en IPv6-adressen, prefixlengtes en status van netwerkkaarten. Een prefixlengte zoals `/24` is geen subnetmasker in decimale puntnotatie.

`ip addr show dev eth0` toont de adressen en status van netwerkkaart `eth0`.

11 ip neigh (buurtentabel)

Syntaxis: `ip neigh show [dev interface]`. Toont de IPv4 ARP- en IPv6-buurvermeldingen, waaronder bekende link-adressen en bereikbaarheidstoestand. Statuswaarden zoals `REACHABLE` (bereikbaar) en `STALE` (vermelding mogelijk verouderd) blijven in de uitvoer staan.

`ip neigh show` toont de bekende netwerkburen en hun status; interpreteer `STALE` als een aanwijzing dat de vermelding opnieuw bevestigd kan worden.

Interface- & Adresconfiguratie (Linux)

12 ip link (Interfacestatus - Linux)

`ip link show [dev interface]`

`ip link show eth0` toont de operationele status (UP/DOWN), MTU-grootte en hardwarematig MAC-adres van interface `eth0`.

DNS · Windows en Linux

13 nslookup (DNS-opzoeking)

Syntaxis: `nslookup [-type=record] host [naamserver]`. Vraagt een DNS-record op voor een host; een opgegeven naamserver wordt voor die zoekopdracht gebruikt.

`nslookup -type=MX voorbeeld.nl 8.8.8.8` vraagt de mailserverrecords voor `voorbeeld.nl` op via de opgegeven naamserver.

DNS · Linux

14 dig (DNS-gegevens opvragen)

Syntaxis: `dig [@server] host [recordtype] [+short]`. Vraagt DNS-gegevens op. Met `+short` wordt de uitvoer verkort; zonder recordtype gebruikt dig doorgaans A-records.

| `dig @1.1.1.1 voorbeeld.nl A +short` vraagt het IPv4-adres van `voorbeeld.nl` op via de opgegeven naamserver.

DNS & Naamomzetting (Linux)

15 host (Eenvoudige DNS-lookup - Linux)

`host [-t type] domeinnaam [naamserver]`

| `host -t AAAA voorbeeld.nl` controleert direct of er een actief IPv6-adres geregistreerd staat voor het domein.

Poorten en verbindingen · Windows

16 netstat (verbindingen en poorten)

Syntaxis: `netstat -ano`. Toont netwerkverbindingen en luisterende poorten; de optie `-o` voegt het proces-ID (PID) toe en `-n` toont adressen en poorten numeriek.

| `netstat -ano` geeft onder meer actieve verbindingen, luisterende poorten en bijbehorende proces-ID's weer.

17 Test-NetConnection (TCP-poorttest)

Syntaxis: `Test-NetConnection -ComputerName host -Port poortnummer`. Test of vanaf deze computer een TCP-verbinding met de opgegeven host en poort tot stand kan komen. Een geslaagde test bevestigt niet dat de toepassing of dienst goed werkt.

| `Test-NetConnection -ComputerName 192.168.1.10 -Port 443` controleert of een TCP-verbinding met poort 443 tot stand komt; dit bewijst op zichzelf niet dat HTTPS correct functioneert.

Poorten en verbindingen · Linux

18 ss (socketstatistieken)

Syntaxis: `ss [-t] [-u] [-l] [-p] [-n]`. Toont sockets en netwerkverbindingen. De opties selecteren TCP, UDP, luisterende sockets, processen en numerieke adressen of poorten.

`ss -tulpn` toont luisterende TCP- en UDP-sockets, met numerieke poorten en, indien beschikbaar, procesinformatie.

19 nc (TCP-bereikbaarheid testen)

Syntaxis: `nc -zv host poort`. Probeert een TCP-verbinding met de opgegeven host en poort zonder gegevensuitwisseling. Een geslaagde verbinding duidt op bereikbaarheid, maar beoordeelt niet volledig de dienst achter die poort.

`nc -zv 192.168.1.20 22` test of vanaf deze computer een TCP-verbinding met poort 22 tot stand komt.

Routing & Padanalyse (Windows)

20 route print (Routeringstabel - Windows)

`route print [-4|-6]`

`route print -4` toont alle actieve IPv4-routes, netwerkmaskers, gateways en interface-metriekwaarden.

Poorten en diensten · Linux

21 nmap (poort- en servicediagnose)

Syntaxis: `nmap -sV <ip_of_domein>`. Onderzoekt bereikbare TCP-poorten en probeert diensten en versies te herkennen. Gebruik dit uitsluitend op systemen waarvoor je toestemming hebt; resultaten zijn afhankelijk van netwerk en configuratie.

`nmap -sV 192.168.1.10` inventariseert bereikbare TCP-poorten en herkende diensten op een systeem dat je beheert of waarvoor je toestemming hebt.

Webdiensten · Linux en Windows

22 curl (webverbinding controleren)

Syntaxis: ``curl -I <url>``. Vraagt de HTTP-antwoordkoppen op met een HEAD-verzoek; hiermee kun je onder meer de geretourneerde HTTP-status bekijken. De uitvoer is geen volledige TLS-handshakeanalyse.

``curl -I https://intranet.voorbeeld.nl`` toont de antwoordkoppen en status die de webserver voor dit verzoek terugstuurt.

Webdiensten · Windows PowerShell

23 Invoke-WebRequest (webverzoek controleren)

Syntaxis: ``Invoke-WebRequest -Uri <url> -Method Head``. Doet een HTTP HEAD-verzoek en levert de reactie terug. Welke inhoud in de uitvoer verschijnt, hangt onder meer af van PowerShell-versie en serverreactie.

``Invoke-WebRequest -Uri 'https://intranet.voorbeeld.nl' -Method Head`` vraagt de reactie op zonder een GET-verzoek te doen; controleer de teruggegeven status en koppen.

Toegang op afstand · Linux

24 ssh (SSH-verbinding diagnosticeren)

Syntaxis: ``ssh -v <gebruiker>@<host>``. Probeert een SSH-verbinding en toont extra diagnostische meldingen over het verbindings- en aanmeldproces. Voer dit alleen uit voor een systeem waartoe je bevoegd bent; het kan een interactieve sessie starten.

``ssh -v beheer@10.0.0.50`` toont diagnostische meldingen tijdens het verbinden met een systeem dat je beheert. Stop met ``Ctrl+C`` als je geen sessie wilt openen.

Beheer op afstand

25 scp — Bestanden veilig kopiëren (Linux)

Syntaxis: `scp [-P poort] <bron> <gebruiker>@<host>:<bestemming>`. Kopieert bestanden via SSH. De overdracht is versleuteld; de opdracht zelf meet geen bandbreedte.

`scp -P 22 backup.tar.gz beheer@10.0.0.50:/var/backups/` kopieert het bestand naar de back-upmap op de externe host.

Pakketdiagnose

26 tcpdump

Syntaxis: `tcpdump -i <interface> -nn port <poort>`. Legt pakketten op de opgegeven interface vast en toont adressen en poorten numeriek; inhoud en details zijn niet altijd leesbaar. Gebruik alleen op netwerken waarvoor je toestemming hebt.

`tcpdump -i eth0 -nn port 53`. Bekijk op een bestaande interface of er DNS-verkeer zichtbaar is tijdens een storing.

27 pktmon

Syntaxis: `pktmon filter add -p <poort>`; `pktmon start --etw`; `pktmon stop`. Voegt een poortfilter toe, start een pakketregistratie en stopt die weer. De registratie kan helpen bij het onderzoeken van pakketverlies op deze Windows-computer.

`pktmon filter add -p 443`; `pktmon start --etw`; `pktmon stop`. Gebruik dit voor een beperkte registratie rond een verbindingsprobleem; de poortfilter alleen bewijst niet dat TLS-verkeer de oorzaak is.

28 tshark

Syntaxis: `tshark -i <interface> -f '<capturefilter>' -c <aantal>`. Legt een beperkt aantal pakketten vast met een capturefilter. Captures kunnen gevoelige gegevens bevatten; gebruik dit alleen op interfaces en verkeer waarvoor je toestemming hebt.

`tshark -i eth0 -f 'host 192.168.1.10 and port 53' -c 10`. Leg maximaal tien pakketten vast voor DNS-diagnose van een bevoegd lokaal systeem.

Hardware en Interfacestatus

29 ethtool (Interface- & Snelheidsdiagnose - Linux)

Syntaxis: `ethtool <interface>`. Rapporteer duplexmodus, poortsnelheid en linkstatus van de netwerkkaart.

| `ethtool eth0` (Controleert of een interface daadwerkelijk 'Speed: 1000Mb/s' en 'Link detected: yes' toont).

30 Get-NetAdapter (Status Netwerkinterface - PowerShell)

Syntaxis: `Get-NetAdapter | Format-Table Name, Status, LinkSpeed`. Toont adapterstatus en linksnelheden.

| `Get-NetAdapter` (Controleert direct of de fysieke of virtuele netwerkkaart actief is en up staat).

IP-adressen en interfaces

31 Get-NetIPAddress

Syntaxis: `Get-NetIPAddress -AddressFamily IPv4`. Toont de geconfigureerde IPv4-adressen en prefixlengtes op lokale interfaces.

| `Get-NetIPAddress -InterfaceAlias 'Ethernet'`. Controleer welk IPv4-adres en welke prefixlengte aan de Ethernet-interface zijn toegewezen.

Routing

32 Get-NetRoute

Syntaxis: `Get-NetRoute [-DestinationPrefix <prefix>]`. Toont routes in de Windows-routingtabel. De uitvoer bevat onder meer bestemming, interface en metriek; welke route wordt gebruikt hangt af van de bestemming en routekeuze.

| `Get-NetRoute -DestinationPrefix '0.0.0.0/0'`. Bekijk de routes voor het IPv4-standaardnetwerk; leid uit de uitvoer niet zonder meer af welke route voor elk doel actief is.

DNS en naamomzetting

33 Resolve-DnsName

Syntaxis: `Resolve-DnsName -Name <domeinnaam> -Type <recordtype>`. Vraagt het opgegeven DNS-recordtype op, bijvoorbeeld A, AAAA, MX of CNAME. Beschikbare gegevens hangen af van de DNS-server en het domein.

`Resolve-DnsName -Name example.com -Type MX`. Controleer welke mailserverrecords voor dit domein worden teruggegeven.

34 resolvectl query — DNS-lookup (Linux)

Syntaxis: `resolvectl query <hostnaam>`. Vraagt de lokale systemd-resolved-dienst om de opgegeven naam op te lossen. De getoonde details hangen af van de configuratie en het antwoord.

`resolvectl query voorbeeld.nl` — controleer of de hostnaam via deze computer kan worden omgezet.

Domein- en IP-registratie

35 whois

Syntaxis: `whois <domein-of-IP>`. Raadpleegt beschikbare WHOIS-registratiegegevens; resultaten kunnen beperkt, afgeschermd of verouderd zijn. WHOIS bepaalt niet betrouwbaar het autonome systeem (AS) van een IP-adres.

`whois example.com`. Bekijk welke registratierecords voor het domein beschikbaar zijn; houd er rekening mee dat gegevens ontbreken of privacy-afgeschermd kunnen zijn.

Firewall en Beveiligingsregels

36 iptables (Firewall-regels Inspecteren - Linux)

Syntaxis: `iptables -L -v -n`. Toont gefilterde chains, tellers en actieve blokkades zonder DNS-vertraging.

`iptables -L -n -v --line-numbers` (Spoort blokkerende invoerregels op die verkeer ongewenst laten vallen).

37 nft (Modern Pakketfilter Inspecteren - Linux)

Syntaxis: `nft list ruleset`. Exporteert de volledige actieve nftables-firewallconfiguratie naar het scherm.

| `nft list ruleset` (Controleert status van actieve ketens, NAT-verwijzingen en dropregels).

Windows Firewall

38 netsh advfirewall

Syntaxis: `netsh advfirewall show allprofiles`. Toont per firewallprofiel de configuratiestatus, waaronder of de firewall is ingeschakeld. Dit toont op zichzelf niet welk inkomend verkeer standaard wordt toegestaan of geblokkeerd.

| `netsh advfirewall show allprofiles`. Controleer de status van de Domein-, Privé- en Openbaar-profielen bij een verbindingsprobleem.

39 Get-NetFirewallRule

Syntaxis: `Get-NetFirewallRule -Enabled True -Direction Inbound`. Toont ingeschakelde inkomende firewallregels. Een regel alleen zegt niet altijd of een poort bereikbaar is; profiel, actie en gekoppelde poort- of programmavooraarden tellen mee.

| `Get-NetFirewallRule -Enabled True -Direction Inbound`. Bekijk de ingeschakelde inkomende regels en onderzoek bij een RDP-probleem de relevante regel en bijbehorende poortvoorwaarden.

Draadloze netwerken

40 iw

Syntaxis: `iw dev <interface> link`. Toont de actuele verbindingsstatus van een draadloze interface, waaronder SSID, frequentie en signaalsterkte wanneer verbonden. De opdracht geeft niet noodzakelijk een overdrachtssnelheid weer.

| `iw dev wlan0 link`. Controleer de status en signaalsterkte van de bestaande draadloze interface wlan0.

41 netsh wlan show interfaces

Syntaxis: `netsh wlan show interfaces`. Toont de status van draadloze interfaces en, indien verbonden, gegevens zoals SSID, BSSID, kanaal, radiotype en signaalpercentage.

| `netsh wlan show interfaces`. Bekijk SSID, kanaal en signaalpercentage wanneer een Windows-laptop een instabiele wifi-verbinding heeft.

42 netsh wlan show profile — Wi-Fi-profielen bekijken (Windows)

Syntaxis: ``netsh wlan show profile [name="<profielnaam>"]``. Toont opgeslagen draadloze profielen of details van één profiel. Vermijd opties die opgeslagen sleutels weergeven op gedeelde of onbeveiligde schermen.

| ``netsh wlan show profile name="BedrijfsNetwerk"`` toont de configuratie van dat opgeslagen Wi-Fi-profiel.

43 iw list — Draadloze mogelijkheden bekijken (Linux)

Syntaxis: ``iw list``. Toont de door de draadloze hardware en stuurprogramma gemelde mogelijkheden, waaronder ondersteunde frequenties en modi. De beschikbare details verschillen per adapter.

| ``iw list`` helpt controleren of de adapter de benodigde frequentiebanden ondersteunt.

Lokaal netwerk en ARP

44 arping

Syntaxis: `arping -I <interface> -c <aantal> <IPv4-adres>`. Stuurt een opgegeven aantal ARP-verzoeken op het lokale netwerksegment. Antwoorden tonen dat een apparaat reageert, maar bewijzen op zichzelf geen IP-conflict.

| `arping -I eth0 -c 3 192.168.1.1`. Controleer op een bevoegd lokaal netwerk of het opgegeven adres op ARP-verzoeken antwoordt.

Routering en padanalyse

45 tcptraceroute

Syntaxis: `tcptraceroute <doel> <poort>`. Traceert een netwerkpad met TCP-probes naar de opgegeven doelpoort. Tussengeschakelde routers of filters kunnen antwoorden tegenhouden; gebruik dit alleen voor legitieme diagnose.

| `tcptraceroute example.com 443`. Onderzoek het TCP-pad naar een eigen of bevoegd getest doel op poort 443.

46 **tracpath**

Syntaxis: `tracpath <doel>`. Toont informatie over het netwerkpad en kan, afhankelijk van systeem en antwoorden onderweg, de Path MTU rapporteren. Ontbrekende reacties betekenen niet automatisch dat pakketten worden gefragmenteerd.

| `tracpath example.com`. Vergelijk de gerapporteerde hops en eventuele MTU-informatie bij problemen met bereikbaarheid of grote gegevensoverdracht.

Lokale connectiviteitstest

47 **hping3**

Syntaxis: `hping3 -S -p <poort> -c <aantal> <doel>`. Stuur TCP SYN-probes en toont ontvangen antwoorden. Gebruik dit uitsluitend voor een eigen systeem of een expliciet bevoegd testdoel; het verstuurt actief netwerkverkeer.

| `hping3 -S -p 8080 -c 3 127.0.0.1`. Test op je eigen computer of een lokale dienst op poort 8080 op TCP-probes antwoordt.

Lokale socketdiagnose

48 **socat**

Syntaxis: `socat - TCP:<doel>:<poort>`. Maakt een interactieve TCP-verbinding en koppelt die aan de terminal. Gebruik uitsluitend een eigen of expliciet bevoegd testdoel; invoer kan naar de andere kant worden verzonden.

| `socat - TCP:127.0.0.1:8080`. Verbind met een lokale testdienst op poort 8080 om handmatig te controleren of die reageert.

Verkeersdiagnose

49 **ngrep — zoeken in netwerkverkeer (Linux)**

Syntaxis: ``ngrep [opties] <patroon> [filter]``. Zoekt naar overeenkomende tekst in vastgelegde pakketten. Alleen bruikbaar voor leesbare, niet-versleutelde inhoud; inspectie kan gevoelige gegevens blootleggen. Gebruik uitsluitend op verkeer waarvoor je toestemming hebt.

| ``sudo ngrep -q -W byline 'GET' 'tcp port 80'`` — zoek naar leesbare HTTP-verzoeken op poort 80.

50 tc qdisc show — wachtrijdisciplines bekijken

Syntaxis: `tc qdisc show [dev <interface>]`

Inspecteert de geconfigureerde wachtrijdisciplines voor interfaces. Zonder statistiekoptie geeft dit geen uitgebreide verkeers- of vertragingmetingen; `-s` kan beschikbare tellers toevoegen.

```
| tc -s qdisc show dev eth0
```

51 netstat -s — protocolstatistieken

Syntaxis: `netstat -s [-p <protocol>]`

Toont cumulatieve netwerkstatistieken. Welke protocollen en tellers worden weergegeven, verschilt per besturingssysteem en versie; interpreteer de uitvoer in die context.

```
| netstat -s -p tcp
```

Bandbreedte

52 iftop — verkeer per verbinding bekijken (Linux)

Syntaxis: `iftop [opties]`. Toont live geschatte bandbreedte per netwerkverbinding; de weergave helpt drukke verbindingen onderzoeken, maar stelt niet op zichzelf vast welk proces de oorzaak is.

```
| `sudo iftop -i eth0 -n` — bekijk verbindingen en bandbreedtes op `eth0` zonder hostnamen op te zoeken.
```

53 nload — interfaceverkeer volgen (Linux)

Syntaxis: `nload [interface]`. Toont live inkomend en uitgaand verkeer en overdrachtstotalen voor de gekozen netwerkinterface.

```
| `nload eth0` — volg de actuele verkeerssnelheid van `eth0`.
```

54 bmon — interfacebandbreedte monitoren (Linux)

Syntaxis: `bmon [opties]`; `-p` kiest een interface. Toont interfaceverkeer in tekstuele grafieken. Welke pakket- en foutstatistieken beschikbaar zijn, hangt af van systeem en interface.

```
| `bmon -p eth0` — bekijk de verkeersstatistieken van `eth0`.
```

55 iperf3 — doorvoersnelheid meten (Linux/Windows)

Syntaxis: server ``iperf3 -s``; client ``iperf3 -c <serveradres> [opties]``. Meet doorvoer met verkeer tussen twee systemen die iperf3 draaien. Gebruik alleen systemen en netwerken waarvoor je toestemming hebt. Een TCP-test meet geen jitter; daarvoor gebruik je UDP met ``-u``.

Start op het andere, toegestane systeem ``iperf3 -s``; voer daarna ``iperf3 -c 192.168.1.50 -t 10`` uit voor een TCP-meting van tien seconden.

56 speedtest-cli — internetsnelheid testen

Syntaxis: ``speedtest-cli [opties]``. Meet latentie en download- en uploadsnelheid via een compatibele Speedtest-dienst. Het programma moet geïnstalleerd zijn en de dienst bereikbaar; meetresultaten hangen onder meer af van server en netwerkbelasting.

``speedtest-cli --simple`` — toon een beknopte meting.

Interfacebeheer

57 ip link set — interface-instellingen wijzigen (Linux)

Syntaxis: ``ip link set <interface> <instelling>``. Wijzigt de operationele status of een andere ondersteunde interface-instelling. Wijzigingen kunnen de verbinding onderbreken; voor configuratiewijzigingen zijn vaak beheerdersrechten nodig.

``sudo ip link set eth0 up`` — activeer ``eth0``. Controleer eerst dat dit de juiste interface is.

58 Enable-NetAdapter — adapter inschakelen (PowerShell)

Syntaxis: ``Enable-NetAdapter -Name <naam>``. Schakelt de genoemde Windows-netwerkadaptor in; meestal zijn beheerdersrechten vereist.

``Enable-NetAdapter -Name 'Ethernet 1'`` — schakel de adaptor met die naam in. Bekijk namen zo nodig met ``Get-NetAdapter``.

59 Disable-NetAdapter — adapter uitschakelen (PowerShell)

Syntaxis: ``Disable-NetAdapter -Name <naam>``. Schakelt de adaptor uit en verbreekt daarmee de netwerkcommunicatie via die adaptor. Dit kan ook je beheerverbinding afsluiten; gebruik het niet op de interface waarmee je op afstand werkt.

``Disable-NetAdapter -Name 'Wi-Fi'`` — schakel wifi uit wanneer je lokaal werkt en de gevolgen kunt herstellen.

Netwerkadapters

60 Restart-NetAdapter — Netwerkadapter herstarten (PowerShell)

Syntaxis: ``Restart-NetAdapter -Name <naam>``. Schakelt de opgegeven adapter kort uit en weer in; de verbinding kan tijdelijk wegvallen. Voer dit uit in PowerShell met passende rechten.

| ``Restart-NetAdapter -Name "Ethernet 1"`` herstart die adapter bij een lokale verbindingss storing.

61 Get-NetAdapterStatistics — statistieken van netwerkadapters (PowerShell)

Syntaxis: ``Get-NetAdapterStatistics [-Name] <String>``

Toont verzonden en ontvangen bytes en pakketten, plus weggegooide pakketten en fouten per netwerkadapter. Gebruik dit om transmissieproblemen nader te onderzoeken; de tellers zijn cumulatief sinds de adapter is gestart of de statistieken zijn gereset.

| ``Get-NetAdapterStatistics | Select-Object Name, ReceivedBytes, SentBytes, ReceivedDiscardedPackets, OutboundErrors`` toont kerncijfers voor alle adapters.

IP-configuratie

62 Set-NetIPAddress — IP-configuratie wijzigen (PowerShell)

Syntaxis: ``Set-NetIPAddress -InterfaceAlias <naam> -IPAddress <adres> -PrefixLength <lengte>``. Wijzigt een bestaande IP-adresconfiguratie; deze opdracht stelt geen standaardgateway in. Een wijziging kan bereikbaarheid verbreken. Controleer eerst de bestaande configuratie en gebruik passende waarden voor je netwerk.

| ``Set-NetIPAddress -InterfaceAlias 'Ethernet' -IPAddress 192.168.1.100 -PrefixLength 24`` — pas een bestaande adresconfiguratie aan. Voer dit alleen uit als ``192.168.1.100/24`` bij het netwerk past.

63 New-NetIPAddress — IP-adres toevoegen (PowerShell)

Syntaxis: ``New-NetIPAddress -InterfaceAlias <naam> -IPAddress <adres> -PrefixLength <lengte> [-DefaultGateway <adres>]``. Maakt een IP-adresconfiguratie aan; de gateway is optioneel. Controleer vooraf dat het adres vrij is en bij het netwerk past. Een onjuiste instelling kan de verbinding verstoren.

| ``New-NetIPAddress -InterfaceAlias 'Ethernet' -IPAddress 192.168.1.150 -PrefixLength 24 -DefaultGateway 192.168.1.1`` — voeg deze configuratie toe als adres en gateway voor dat netwerk zijn gereserveerd.

64 Remove-NetIPAddress — IP-adres verwijderen (PowerShell)

Syntaxis: ``Remove-NetIPAddress -InterfaceAlias <naam> -IPAddress <adres>``. Verwijdert de opgegeven adresconfiguratie. Dit kan de verbinding beïnvloeden; laat bevestiging ingeschakeld en controleer interface en adres vooraf.

``Remove-NetIPAddress -InterfaceAlias 'Ethernet' -IPAddress 192.168.1.150`` — controleer de getoonde bevestiging voordat je het adres verwijdert.

65 netsh interface ip show config — IP-configuratie bekijken (Windows)

Syntaxis: `netsh interface ip show config [name="<interfacenaam>"]`

Toont IPv4-instellingen van interfaces, waaronder adres, subnetmasker, gateway en DNS-configuratie; de getoonde gegevens hangen af van de interface.

`netsh interface ip show config name="Ethernet"`

DNS-diagnose

66 Get-DnsClientServerAddress — DNS-servers bekijken (PowerShell)

Syntaxis: ``Get-DnsClientServerAddress [-InterfaceAlias <naam>]``. Toont de ingestelde DNS-serveradressen per Windows-netwerkkinterface.

``Get-DnsClientServerAddress -InterfaceAlias 'Ethernet'`` — bekijk DNS-instellingen voor de adapter Ethernet.

67 Clear-DnsClientCache — lokale DNS-cache wissen (PowerShell)

Syntaxis: ``Clear-DnsClientCache``. Leegt de lokale cache van de Windows DNS-client. Dit verwijdert lokaal bewaarde antwoorden, maar verandert geen records op externe DNS-servers.

``Clear-DnsClientCache`` — wis de lokale cache wanneer je een verouderd lokaal antwoord onderzoekt.

68 ipconfig /flushdns — lokale DNS-cache wissen (Windows)

Syntaxis: ``ipconfig /flushdns``. Leegt de lokale Windows-DNS-cache. Dit verandert geen DNS-records op externe servers en garandeert niet dat gewijzigde records daar al beschikbaar zijn.

``ipconfig /flushdns`` — wis de lokale cache voordat je een naam opnieuw test.

69 `ipconfig /displaydns` — lokale DNS-cache bekijken (Windows)

Syntaxis: ``ipconfig /displaydns``. Toont gegevens uit de lokale DNS-clientcache; dit is geen overzicht van alle DNS-records van een domein.

``ipconfig /displaydns`` — controleer of er lokaal een antwoord voor een recent opgevraagde hostnaam is opgeslagen.

70 `resolvectl flush-caches` — DNS-cache wissen (Linux)

Syntaxis: ``resolvectl flush-caches``. Verzoekt ``systemd-resolved`` de lokale cache te wissen. Dit werkt alleen wanneer die dienst op het systeem aanwezig en actief is; het wist geen cache op externe DNS-servers.

``sudo resolvectl flush-caches`` — wis de cache op een Linux-systeem dat ``systemd-resolved`` gebruikt.

71 `resolvectl status` — DNS-status bekijken (Linux)

Syntaxis: ``resolvectl status [interface]``. Toont DNS-informatie die via ``systemd-resolved`` beschikbaar is, waaronder servers en zoekdomeinen. Uitvoer en DNSSEC-informatie hangen af van systeemconfiguratie en ondersteunde functies.

``resolvectl status`` — bekijk de algemene DNS-status; gebruik bijvoorbeeld ``resolvectl status eth0`` voor een interface.

DNS-configuratie

72 `Set-DnsClientServerAddress` — DNS-servers instellen (PowerShell)

Syntaxis: ``Set-DnsClientServerAddress -InterfaceAlias <naam> -ServerAddresses <adressen>``. Wijzigt de DNS-serverinstellingen van een interface; dit is configuratie, geen alleen-lezen diagnose. Je hebt mogelijk beheerdersrechten nodig. Gebruik alleen resolvers die voor jouw netwerk bedoeld zijn.

``Set-DnsClientServerAddress -InterfaceAlias 'Ethernet' -ServerAddresses ('1.1.1.1','8.8.8.8')`` — wijs deze servers toe als dat past bij je netwerkbeleid.

Verbindingsdiagnose

73 Get-NetTCPConnection — TCP-verbindingen bekijken (PowerShell)

Syntaxis: `Get-NetTCPConnection [-State <status>] [-LocalPort <poort>]`

Toont actieve TCP-verbindingen met lokale en externe adressen, verbindingstatus en proces-ID (PID).

| `Get-NetTCPConnection -State Established`

Poortdiagnose

74 Get-NetUDPEndpoint — UDP-eindpunten bekijken (PowerShell)

Syntaxis: `Get-NetUDPEndpoint [-LocalPort <poort>]`

Toont UDP-eindpunten met lokaal adres, poort en gekoppelde proces-ID. UDP heeft geen verbindingstatus zoals TCP.

| `Get-NetUDPEndpoint -LocalPort 53`

Verbindingen en poorten

75 lsof -i — netwerkbestanden en sockets (Linux)

Syntaxis: ``lsof -i [<protocol>][@<host>][:<poort>]``. Toont open bestanden die bij netwerkverbindingen horen en, afhankelijk van rechten en systeemtoestand, de betrokken processen.

| ``sudo lsof -i :443`` — bekijk welke processen netwerkbestanden of sockets op poort 443 open hebben.

Poort- en procesdiagnose

76 fuser — Processen bij een poort vinden (Linux)

Syntaxis: ``fuser -v <poort>/<protocol>``. Toont processen die de opgegeven TCP- of UDP-poort gebruiken. Zonder ``-k`` inventariseert het alleen; beëindig geen proces voordat je de gevolgen kent.

| ``sudo fuser -v 80/tcp`` toont welke processen TCP-poort 80 gebruiken.

77 netstat -ano — poorten en PID's bekijken (Windows)

Syntaxis: `netstat -ano`

Toont TCP-verbindingen en luisterpoorten, UDP-eindpunten en bijbehorende PID's. Gebruik een filter om uitvoer op een poort te beperken.

| `netstat -ano | findstr :8080`

78 ss — Luisterende sockets tonen (Linux)

Syntaxis: ``ss -tulpn``. Toont luisterende TCP- en UDP-sockets; ``-p`` vraagt procesinformatie op, waarvoor vaak beheerdersrechten nodig zijn.

| ``sudo ss -tulpn`` geeft een overzicht van luisterpoorten en, waar toegestaan, de bijbehorende processen.

Lokaal netwerk

79 arp-scan — Apparaten op het lokale netwerk vinden (Linux)

Syntaxis: ``sudo arp-scan --localnet``. Stuurt ARP-verzoeken op het rechtstreeks aangesloten lokale netwerk en toont reagerende IPv4-adressen en MAC-adressen. Gebruik dit alleen op een netwerk dat je beheert of waarvoor je toestemming hebt.

| ``sudo arp-scan --localnet`` helpt bij het inventariseren van apparaten op je eigen subnet.

Lokale netwerkdiagnose

80 Get-NetNeighbor — neighbor-cache (PowerShell)

Syntaxis: ``Get-NetNeighbor [-AddressFamily IPv4|IPv6] [-InterfaceAlias <naam>]``. Toont cachevermeldingen voor burens op de lokale netwerkverbinding, met onder meer IP-adres, link-layer-adres en status waar beschikbaar.

| ``Get-NetNeighbor -AddressFamily IPv4`` — bekijk de IPv4-vermeldingen in de neighbor-cache.

81 Remove-NetNeighbor — cachevermelding verwijderen (PowerShell)

Syntaxis: ``Remove-NetNeighbor -InterfaceIndex <nummer> -IPAddress <adres> [-Confirm]``. Verwijdert de opgegeven neighbor-cachevermelding op de gekozen interface. Gebruik dit gericht; een nieuwe vermelding kan later opnieuw worden geleerd.

| ``Remove-NetNeighbor -InterfaceIndex 12 -IPAddress 192.168.1.1`` — verwijder die specifieke vermelding nadat je de interface-index hebt gecontroleerd.

82 ip neigh flush — neighbor-cache leegmaken (Linux)

Syntaxis: ``ip neigh flush dev <interface>``. Verwijdert passende vermeldingen uit de neighbor-cache voor die interface; de kernel kan ze opnieuw opbouwen wanneer dat nodig is.

| ``sudo ip neigh flush dev eth0`` — leeg de cache voor ``eth0`` om een lokaal probleem met buuradresresolutie opnieuw te testen.

Netwerkinterfaces

83 nmcli device status — apparaatstatus (Linux)

Syntaxis: ``nmcli device status``. Geeft een beknopt overzicht van netwerkapparaten die NetworkManager beheert, met hun status en eventuele actieve verbinding.

| ``nmcli device status`` — controleer of de wifi- of Ethernetverbinding als verbonden wordt weergegeven.

Netwerkbeheer

84 nmcli connection show — Netwerkprofielen bekijken (Linux)

Syntaxis: ``nmcli connection show [--active]``. Toont opgeslagen NetworkManager-verbindingsprofielen; met ``--active`` beperk je de lijst tot actieve verbindingen.

| ``nmcli connection show --active`` toont de actieve profielen tijdens het controleren van je netwerkverbinding.

Draadloze netwerkdiagnose

85 nmcli device wifi list — waargenomen wifi-netwerken (Linux)

Syntaxis: ``nmcli device wifi list [ifname <interface>]``. Toont wifi-netwerken die de draadloze interface waarneemt; zichtbaarheid hangt onder meer af van bereik en scanresultaten.

| ``nmcli device wifi list ifname wlan0`` — bekijk de waargenomen netwerken voor ``wlan0``.

86 netsh wlan show networks — waargenomen wifi-netwerken (Windows)

Syntaxis: ``netsh wlan show networks [mode=bssid]``. Toont wifi-netwerken die Windows op dat moment waarneemt. De weergegeven beveiligingsinformatie is geen garantie voor welke encryptie een verbinding uiteindelijk gebruikt.

| ``netsh wlan show networks mode=bssid`` — bekijk waargenomen toegangspunten en hun BSSID's.

Fysieke verbinding

87 mii-tool — Ethernet-linkstatus controleren (Linux)

Syntaxis: ``mii-tool [-v] [interface]``. Toont waar ondersteund de Ethernet-linkstatus en onderhandelingsinformatie. Het verouderde hulpmiddel werkt niet met alle moderne adapters; gebruik ``ethtool`` als alternatief wanneer nodig.

| ``sudo mii-tool -v eth0`` controleert of ``eth0`` een link heeft en toont beschikbare statusdetails.

Netwerkhardware

88 Get-NetAdapterHardwareInfo — adapterhardware bekijken (PowerShell)

Syntaxis: `Get-NetAdapterHardwareInfo [-Name <naam>]`

Toont hardwaregegevens van een netwerkadapter, zoals bus- en slotinformatie. De beschikbare velden hangen af van de adapter en het systeem.

| `Get-NetAdapterHardwareInfo -Name "Ethernet"`

Fysieke Link- & Hardwarestatus

89 ip -s link (Interfacestatistieken & Drops - Linux)

Syntaxis: `ip -s link show [interface]`

Toont overdrachtsstatistieken met tellers voor rx/tx pakketten, fouten, dropped packets en carrier-storingen.

| `ip -s link show dev eth0`

Bandbreedte- & Verkeersmonitoring

90 vnstat (Lichtgewicht Bandbreedtemonitor - Linux)

Syntaxis: vnstat [-h | -d | -m] [-i interface]

Genereert overzichtelijke historische verbruiksrapportages van het netwerkverkeer op een gespecificeerde interface.

```
| vnstat -d -i eth0
```

91 nethogs (Bandbreedte per Proces - Linux)

Syntaxis: sudo nethogs [interface]

Biedt een realtime overzicht van welke processen of applicaties de meeste upload- en downloadsnelheid verbruiken.

```
| sudo nethogs eth0
```

92 iptraf-ng (Interactieve IP-LAN Monitor - Linux)

Syntaxis: sudo iptraf-ng [-i interface | -g]

Interactief instrument voor het analyseren van live netwerkactiviteit, pakketgroottes en poortstatistieken.

```
| sudo iptraf-ng -i all
```

93 dstat -n (Systeembrede Netwerkstatistieken - Linux)

Syntaxis: dstat -n [--net-adv]

Toont realtime inkomend en uitgaand netwerkverkeer in een doorlopende kolomweergave voor prestatieanalyse.

```
| dstat -n --net-adv 2 10
```

Netwerkinterface en foutdiagnose

94 ethtool -S — interface-statistieken (Linux)

Syntaxis (Linux): `ethtool -S <interface>`. Toont statistieken die de driver voor de netwerkinterface rapporteert, bijvoorbeeld aantallen fouten of verloren pakketten. De beschikbare tellers verschillen per driver en apparaat.

| Bekijk meldingen over fouten en verloren pakketten op eth0: `sudo ethtool -S eth0 | grep -E 'drop|error'`.

Herstel en probleemoplossing

95 netsh int ip reset — TCP/IP-instellingen herstellen (Windows)

Syntaxis: `netsh int ip reset [<logbestand>]`

Herstelt TCP/IP-instellingen naar de standaardwaarden. Dit kan aangepaste netwerkconfiguratie beïnvloeden; voer het alleen uit als onderdeel van gerichte probleemoplossing. Een herstart kan nodig zijn.

| `netsh int ip reset C:\resetlog.txt`

96 netsh winsock reset — Winsock herstellen (Windows)

Syntaxis: `netsh winsock reset`

Herstelt de Winsock-catalogus naar de standaardconfiguratie. Dit is een wijzigende herstelactie, geen diagnose; noteer relevante configuratie en herstart Windows indien daarom wordt gevraagd.

| `netsh winsock reset`

Firewallstatus

97 conntrack -L — verbindingstabel bekijken

Syntaxis: `sudo conntrack -L [-p <protocol>] [--dport <poort>]`

Geeft een momentopname van vermeldingen in de connection-trackingtabel. De beschikbare filters hangen af van de geïnstalleerde versie. Gebruik geen optie die vermeldingen wist bij diagnose.

| `sudo conntrack -L -p tcp --dport 443`

98 firewall-cmd — actieve firewallzone bekijken

Syntaxis: `firewall-cmd --get-active-zones` | `firewall-cmd [--zone=<zone>] --list-all`

Toont actieve zones of de regels en instellingen van een gekozen zone. Dit is statuscontrole; de opdracht toont geen routingstabel.

```
| sudo firewall-cmd --zone=public --list-all
```

Firewalldiagnose

99 ufw status verbose — firewallstatus bekijken (Linux)

Syntaxis: `sudo ufw status [verbose | numbered]`

Toont of UFW actief is en, met verbose, het standaardbeleid en aanvullende statusinformatie. Dit commando leest de status en wijzigt geen firewallregels.

```
| sudo ufw status verbose
```

Connectiviteit

100 Test-Connection — bereikbaarheid testen

Syntaxis: `Test-Connection -TargetName <host> [-Count <aantal>]`

Stuurt ICMP-echoverzoeken en toont resultaten in PowerShell-objecten. De opdracht test bereikbaarheid; deze syntaxis voert geen routeanalyse uit. Beschikbare objecteigenschappen kunnen per PowerShell-versie verschillen.

```
| Test-Connection -TargetName 1.1.1.1 -Count 3
```
