

# Diagnostyka sieci

## Sieci komputerowe

Łukasz Szenkiel



COLLEGIUM  
WITELONA  
Uczelnia Państwowa

# ipconfig

**ipconfig** jest narzędziem wiersza poleceń w systemach Windows, które służy do wyświetlania bieżącej konfiguracji sieciowej protokołu TCP/IP komputera. Pozwala na uzyskanie informacji o adresach IP, maskach podsieci, bramach domyślnych, serwerach DNS oraz konfiguracji DHCP i DNS. W systemach Linux jego odpowiednikiem jest polecenie `ip addr` (część pakietu `iproute2`), które oferuje podobną, a często bardziej szczegółową funkcjonalność.

# Co daje?

- ▶ Weryfikacja konfiguracji IP: Pozwala sprawdzić, czy komputer otrzymał poprawny adres IP i inne parametry sieciowe. Jest to kluczowe przy diagnozowaniu problemów z połączeniem sieciowym.
- ▶ Informacje o interfejsach: Wyświetla listę wszystkich interfejsów sieciowych (zarówno fizycznych, jak i wirtualnych) wraz z ich bieżącą konfiguracją.
- ▶ Diagnostyka DHCP: Można sprawdzić, czy interfejs otrzymał konfigurację z serwera DHCP.
- ▶ Informacje o DNS: Wyświetla skonfigurowane serwery DNS, które są używane do tłumaczenia nazw domen na adresy IP.

# Kluczowe zastosowania w rozwiązywaniu problemów sieciowych

- ▶ Sprawdzanie, czy komputer ma adres IP w oczekiwanej podsieci.
- ▶ Weryfikacja poprawności adresu bramy domyślnej, przez którą komputer komunikuje się z innymi sieciami.
- ▶ Upewnienie się, że skonfigurowane są poprawne serwery DNS, umożliwiające przeglądanie stron internetowych.
- ▶ Diagnozowanie problemów z uzyskaniem adresu IP z serwera DHCP.

# Przykład użycia

```
lukasz@szenkiel-ubuntu:~$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp5s0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc fq_codel state DOWN group default qlen 1000
    link/ether 14:da:e9:df:b5:32 brd ff:ff:ff:ff:ff:ff
3: wlx3c64cfcabdd3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 3c:64:cf:ca:bd:d3 brd ff:ff:ff:ff:ff:ff
    inet 192.168.100.10/24 brd 192.168.100.255 scope global dynamic noprefixroute wlx3c64cfcabdd3
        valid_lft 83123sec preferred_lft 83123sec
    inet6 fe80::f72:671a:527d:75fb/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
4: br-41ed7f87e11f: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default
    link/ether fa:c6:08:7a:27:56 brd ff:ff:ff:ff:ff:ff
    inet 172.18.0.1/16 brd 172.18.255.255 scope global br-41ed7f87e11f
        valid_lft forever preferred_lft forever
5: br-76978bebd718: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
    link/ether 4a:7e:0c:a9:b1:f6 brd ff:ff:ff:ff:ff:ff
    inet 172.19.0.1/16 brd 172.19.255.255 scope global br-76978bebd718
        valid_lft forever preferred_lft forever
    inet6 fe80::487e:cff:fea9:b1f6/64 scope link
        valid_lft forever preferred_lft forever
6: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default
    link/ether 2a:09:73:07:ec:cb brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.1/16 brd 172.17.255.255 scope global docker0
        valid_lft forever preferred_lft forever
8: veth4503735@if2: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue master br-76978bebd718 state UP group default
    link/ether 5e:34:5d:af:d7:e6 brd ff:ff:ff:ff:ff:ff
    inet6 fe80::5c34:5dff:feaf:d7e6/64 scope link
        valid_lft forever preferred_lft forever
382: veth55f82bb@if2: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue master br-76978bebd718 state UP group default
    link/ether 5e:c8:f6:d9:7e:0f brd ff:ff:ff:ff:ff:ff
    inet6 fe80::5cc8:f6ff:fed9:7e0f/64 scope link
        valid_lft forever preferred_lft forever
lukasz@szenkiel-ubuntu:~$
```

# ping

**ping** to podstawowe narzędzie diagnostyczne sieciowe, dostępne w większości systemów operacyjnych (w tym Windows i Ubuntu). Działa poprzez wysyłanie pakietów ICMP (Internet Control Message Protocol) Echo Request do określonego hosta w sieci i oczekiwanie na odpowiedź ICMP Echo Reply. Czas odpowiedzi (opóźnienie) oraz informacja o ewentualnej utracie pakietów są kluczowymi wskaźnikami stanu połączenia.

# Co daje?

- ▶ Sprawdzenie dostępności hosta: Pozwala zweryfikować, czy dany host (komputer, serwer, router) jest osiągalny w sieci i odpowiada na żądania.
- ▶ Pomiar opóźnienia (latency): Wyświetla czas odpowiedzi (zazwyczaj w milisekundach), co pozwala ocenić szybkość połączenia między Twoim komputerem a testowanym hostem. Wysokie opóźnienia mogą wskazywać na problemy z siecią.
- ▶ Wykrywanie utraty pakietów: Informuje o procentowej liczbie pakietów, które nie dotarły do celu lub nie wróciły. Utrata pakietów jest silnym wskaźnikiem problemów z niezawodnością połączenia.
- ▶ Podstawowa diagnostyka sieci: Umożliwia szybkie sprawdzenie podstawowej łączności sieciowej przed użyciem bardziej zaawansowanych narzędzi.

# Kluczowe zastosowania w rozwiązywaniu problemów sieciowych

- ▶ Weryfikacja połączenia z Internetem: Sprawdzenie, czy możesz skomunikować się z serwerami w Internecie (np. ping google.com).
- ▶ Diagnostyka problemów z serwerami DNS: Jeśli nie możesz otworzyć stron internetowych, ale pingowanie adresów IP działa, może to wskazywać na problem z serwerami DNS.
- ▶ Lokalizowanie problemów w sieci lokalnej: Pingowanie innych urządzeń w sieci lokalnej (np. routera, innych komputerów) pomaga zidentyfikować, czy problem dotyczy całej sieci, czy tylko konkretnego urządzenia.
- ▶ Ocena jakości połączenia: Wysokie i niestabilne czasy odpowiedzi lub utrata pakietów mogą wskazywać na przeciążenie sieci, problemy z okablowaniem lub inne problemy z infrastrukturą sieciową.
- ▶ Testowanie dostępności urządzeń sieciowych: Szybkie sprawdzenie, czy dany router, przełącznik lub inne urządzenie sieciowe jest online i odpowiada.



# Przykład użycia

```
lukasz@szenkiel-ubuntu: ~  
lukasz@szenkiel-ubuntu:~$ ping chess.com  
PING chess.com (104.18.138.67) 56(84) bytes of data.  
64 bytes from 104.18.138.67 (104.18.138.67): icmp_seq=1 ttl=55 time=17.9 ms  
64 bytes from 104.18.138.67 (104.18.138.67): icmp_seq=2 ttl=55 time=18.0 ms  
64 bytes from 104.18.138.67 (104.18.138.67): icmp_seq=3 ttl=55 time=17.3 ms  
64 bytes from 104.18.138.67 (104.18.138.67): icmp_seq=4 ttl=55 time=17.1 ms  
64 bytes from 104.18.138.67 (104.18.138.67): icmp_seq=5 ttl=55 time=16.6 ms  
64 bytes from 104.18.138.67 (104.18.138.67): icmp_seq=6 ttl=55 time=17.3 ms  
64 bytes from 104.18.138.67 (104.18.138.67): icmp_seq=7 ttl=55 time=16.1 ms  
64 bytes from 104.18.138.67 (104.18.138.67): icmp_seq=8 ttl=55 time=16.5 ms  
64 bytes from 104.18.138.67 (104.18.138.67): icmp_seq=9 ttl=55 time=16.5 ms  
64 bytes from 104.18.138.67 (104.18.138.67): icmp_seq=10 ttl=55 time=16.9 ms  
64 bytes from 104.18.138.67 (104.18.138.67): icmp_seq=11 ttl=55 time=17.5 ms  
64 bytes from 104.18.138.67 (104.18.138.67): icmp_seq=12 ttl=55 time=16.0 ms  
64 bytes from 104.18.138.67 (104.18.138.67): icmp_seq=13 ttl=55 time=16.6 ms  
64 bytes from 104.18.138.67 (104.18.138.67): icmp_seq=14 ttl=55 time=3722 ms  
64 bytes from 104.18.138.67 (104.18.138.67): icmp_seq=15 ttl=55 time=2715 ms  
64 bytes from 104.18.138.67 (104.18.138.67): icmp_seq=16 ttl=55 time=1691 ms  
64 bytes from 104.18.138.67 (104.18.138.67): icmp_seq=17 ttl=55 time=667 ms  
64 bytes from 104.18.138.67 (104.18.138.67): icmp_seq=18 ttl=55 time=16.6 ms  
64 bytes from 104.18.138.67 (104.18.138.67): icmp_seq=19 ttl=55 time=16.5 ms  
64 bytes from 104.18.138.67 (104.18.138.67): icmp_seq=20 ttl=55 time=16.6 ms  
^C  
--- chess.com ping statistics ---  
20 packets transmitted, 20 received, 0% packet loss, time 19074ms  
rtt min/avg/max/mdev = 15.984/453.298/3722.359/1010.660 ms, pipe 4  
lukasz@szenkiel-ubuntu:~$
```

# tracert

**tracert** (w systemach Windows) lub **traceroute** (w systemach Linux i macOS) to narzędzie wiersza poleceń służące do śledzenia trasy pakietów IP do określonego hosta docelowego. Działa poprzez wysyłanie pakietów z sukcesywnie zwiększanym polem TTL (Time To Live) w nagłówku IP. Gdy pakiet osiągnie router, którego TTL wynosi 0, router odsyła komunikat ICMP Time Exceeded. Dzięki temu narzędzie może zidentyfikować wszystkie routery (tzw. "hopy") na ścieżce do celu oraz zmierzyć czas odpowiedzi dla każdego z nich.

# Co daje?

- ▶ Wizualizacja trasy pakietów: Pokazuje dokładną ścieżkę, jaką pakiety IP pokonują od Twojego komputera do serwera docelowego, wylistowując wszystkie pośredniczące routery.
- ▶ Identyfikacja wąskich gardeł: Wysokie czasy odpowiedzi na konkretnych hopach mogą wskazywać na przeciążone lub problematyczne routery na ścieżce.
- ▶ Lokalizowanie awarii sieci: Jeśli śledzenie trasy zatrzymuje się na pewnym hopie, może to sugerować awarię routera lub problem z połączeniem w tym miejscu.
- ▶ Diagnostyka problemów z routingiem: Pozwala zrozumieć, czy pakiety są kierowane oczekiwaną ścieżką i wykryć nieprawidłowe lub nieefektywne trasy.
- ▶ Weryfikacja połączenia z odległymi serwerami: Umożliwia sprawdzenie, czy istnieje fizyczna ścieżka do docelowego serwera i jakie urządzenia sieciowe znajdują się na tej ścieżce.

# Kluczowe zastosowania w rozwiązywaniu problemów sieciowych

- ▶ Diagnostowanie wolnego dostępu do stron internetowych: Jeśli strona ładuje się wolno, tracert może pomóc zidentyfikować, czy opóźnienie występuje na którymś z routerów po drodze.
- ▶ Identyfikacja problemów z połączeniem do serwerów gier online: Wysokie opóźnienia lub utrata pakietów na konkretnych hopach mogą wskazywać na problemy z serwerami gry lub trasą do nich.
- ▶ Weryfikacja poprawności konfiguracji sieci dostawcy internetu: Można sprawdzić, czy ruch przechodzi przez oczekiwane węzły sieci dostawcy.
- ▶ Lokalizowanie problemów z infrastrukturą siecią wewnątrz organizacji: W sieciach firmowych tracert może pomóc zidentyfikować, który router powoduje problemy z połączeniem między różnymi segmentami sieci.
- ▶ Zrozumienie ścieżki ruchu w różnych regionach geograficznych: Śledzenie trasy do serwerów w innych krajach może dać wgląd w infrastrukturę siećową międzykontynentalną.

# Przykład użycia

```
lukasz@szenkiel-ubuntu: ~  
lukasz@szenkiel-ubuntu:~$ traceroute chess.com  
traceroute to chess.com (104.18.139.67), 30 hops max, 60 byte packets  
 1  _gateway (192.168.100.1)  3.273 ms  3.215 ms  3.188 ms  
 2  83-238-252-80.static.inetia.pl (83.238.252.80)  11.475 ms  11.449 ms  11.412 ms  
 3  WROCH002RT91.inetia.pl (83.238.113.18)  13.125 ms  13.101 ms  13.691 ms  
 4  100.95.8.11 (100.95.8.11)  13.670 ms  14.131 ms  14.108 ms  
 5  100.95.8.8 (100.95.8.8)  15.716 ms  15.696 ms  17.306 ms  
 6  JAWOH001RT91.inetia.pl (83.238.249.43)  24.038 ms  21.028 ms  20.964 ms  
 7  WARSH002RT91.inetia.pl (83.238.248.72)  22.296 ms  19.763 ms  19.708 ms  
 8  * * *  
 9  162.158.100.23 (162.158.100.23)  19.595 ms  19.571 ms  162.158.100.27 (162.158.100.27)  21.320 ms  
10  104.18.139.67 (104.18.139.67)  20.374 ms  18.337 ms  18.300 ms  
lukasz@szenkiel-ubuntu:~$
```

# route print

**route print** to narzędzie wiersza poleceń w systemach Windows, które wyświetla tablicę routingu TCP/IP komputera. Tablica routingu zawiera informacje o tym, jak pakiety IP są kierowane do różnych sieci docelowych. Zawiera wpisy definiujące, który interfejs sieciowy i która brama (następny hop) są używane do wysłania pakietów do określonych adresów IP lub sieci. W systemach Linux podobne informacje można uzyskać za pomocą poleceń **ip route show** lub **route -n** (starsze polecenie).

# Co daje?

- ▶ Wgląd w decyzje routingowe: Pozwala zobaczyć, jak system operacyjny decyduje, którą ścieżką wysłać pakiety danych do różnych miejsc w sieci.
- ▶ Identyfikacja bramy domyślnej: Wyświetla adres IP bramy domyślnej, czyli routera, do którego wysyłane są pakiety przeznaczone dla sieci, które nie są bezpośrednio połączone.
- ▶ Sprawdzenie tras do konkretnych sieci: Umożliwia weryfikację, czy w tablicy routingu istnieją wpisy dla oczekiwanych sieci docelowych.
- ▶ Diagnostyka problemów z routingiem: Nieprawidłowe lub brakujące wpisy w tablicy routingu mogą być przyczyną problemów z komunikacją sieciową.
- ▶ Zrozumienie konfiguracji sieci: Daje obraz logicznej topologii sieci z punktu widzenia danego komputera.

# Kluczowe zastosowania w rozwiązywaniu problemów sieciowych

- ▶ Weryfikacja bramy domyślnej: Sprawdzenie, czy komputer ma poprawną bramę do Internetu i innych sieci.
- ▶ Sprawdzanie tras do sieci lokalnych: Upewnienie się, że istnieją trasy do innych segmentów sieci lokalnej.
- ▶ Diagnostyka VPN: Weryfikacja, czy ruch przez VPN jest kierowany prawidłowo.
- ▶ Wykrywanie błędnych konfiguracji IP: Analiza tablicy routingu może ujawnić problemy z adresacją.
- ▶ Sprawdzanie tras statycznych: Potwierdzenie, że ręcznie dodane trasy są aktywne.



# Przykład użycia

```
lukasz@szenkiel-ubuntu: ~  
lukasz@szenkiel-ubuntu:~$ route -n  
Kernel IP routing table  
Destination      Gateway         Genmask         Flags Metric Ref    Use Iface  
0.0.0.0          192.168.100.1  0.0.0.0         UG        600    0      0 wlx3c64cfcabdd3  
169.254.0.0      0.0.0.0        255.255.0.0     U        1000    0      0 br-41ed7f87e11f  
172.17.0.0       0.0.0.0        255.255.0.0     U         0      0      0 docker0  
172.18.0.0       0.0.0.0        255.255.0.0     U         0      0      0 br-41ed7f87e11f  
172.19.0.0       0.0.0.0        255.255.0.0     U         0      0      0 br-76978bebd718  
192.168.100.0    0.0.0.0        255.255.255.0   U        600    0      0 wlx3c64cfcabdd3  
lukasz@szenkiel-ubuntu:~$ ip route show  
default via 192.168.100.1 dev wlx3c64cfcabdd3 proto dhcp metric 600  
169.254.0.0/16 dev br-41ed7f87e11f scope link metric 1000 linkdown  
172.17.0.0/16 dev docker0 proto kernel scope link src 172.17.0.1 linkdown  
172.18.0.0/16 dev br-41ed7f87e11f proto kernel scope link src 172.18.0.1 linkdown  
172.19.0.0/16 dev br-76978bebd718 proto kernel scope link src 172.19.0.1  
192.168.100.0/24 dev wlx3c64cfcabdd3 proto kernel scope link src 192.168.100.10 metric 600  
lukasz@szenkiel-ubuntu:~$
```

# arp

**arp** (Address Resolution Protocol) to narzędzie wiersza poleceń używane do wyświetlania i modyfikowania tabeli ARP (Address Resolution Protocol). Tabela ARP zawiera dynamicznie tworzone i przechowywane mapowania między adresami IP a fizycznymi adresami MAC (Media Access Control) urządzeń znajdujących się w tej samej lokalnej sieci Ethernet. Protokół ARP jest niezbędny do komunikacji w sieci lokalnej, ponieważ warstwa łącza danych (Ethernet) używa adresów MAC do identyfikacji urządzeń, podczas gdy warstwa sieciowa (IP) używa adresów IP. Kiedy urządzenie chce wysłać dane do innego urządzenia w tej samej sieci, znając jego adres IP, musi najpierw poznać jego adres MAC, co osiąga za pomocą zapytań ARP. Narzędzie arp pozwala na inspekcję tych mapowań, a czasami na ich ręczną modyfikację.

# Co daje?

- ▶ Wyświetla powiązania IP z MAC w sieci lokalnej.
- ▶ Pomaga identyfikować urządzenia w sieci.
- ▶ Umożliwia diagnozowanie problemów z komunikacją lokalną (brakujące lub błędne wpisy).
- ▶ Może pomóc w wykrywaniu ARP spoofingu (podejrzane wpisy).
- ▶ Przydatne przy rozwiązywaniu problemów z DHCP.

# Kluczowe zastosowania w rozwiązywaniu problemów sieciowych

- ▶ Weryfikacja połączeń lokalnych: Sprawdź, czy znasz adres MAC innego urządzenia w sieci lokalnej, jeśli masz problem z komunikacją z nim.
- ▶ Identyfikacja intruzów: Nieznane wpisy MAC mogą wskazywać na nieautoryzowane urządzenia w Twojej sieci.
- ▶ Diagnostyka konfliktów IP: Jeśli komunikacja w sieci jest niestabilna, sprawdź, czy ten sam adres IP nie jest powiązany z różnymi adresami MAC.
- ▶ Problemy z bramą: Jeśli nie masz dostępu do Internetu, sprawdź, czy znasz adres MAC Twojego routera (bramy domyślnej).
- ▶ Bezpieczeństwo sieci: Monitoruj tabelę ARP w poszukiwaniu podejrzanych wpisów, które mogą świadczyć o ataku ARP spoofing.

# Przykład użycia

```
lukasz@szenkiel-ubuntu: ~  
lukasz@szenkiel-ubuntu:~$ arp  
Address          HWtype  HWaddress      Flags Mask    Iface  
_gateway          ether    7c:39:85:55:81:3e  C             wlx3c64cfcabdd3  
lukasz@szenkiel-ubuntu:~$ arp -a  
_gateway (192.168.100.1) w 7c:39:85:55:81:3e [ether] na wlx3c64cfcabdd3  
lukasz@szenkiel-ubuntu:~$ arp -av  
_gateway (192.168.100.1) w 7c:39:85:55:81:3e [ether] na wlx3c64cfcabdd3  
Wpisy: 1          Pominięte: 0      Znaleziono: 1  
lukasz@szenkiel-ubuntu:~$
```

# netstat

**netstat** (network statistics) to narzędzie wiersza poleceń, które wyświetla różne statystyki połączeń sieciowych, tablice routingu, statystyki interfejsów, połączenia maskaradowane i członkostwo w grupach multicast. Jest dostępne w większości systemów operacyjnych, w tym Windows i starszych wersjach Linux. W nowszych systemach Linux jego funkcjonalność jest w dużej mierze zastąpiona przez nowsze i bardziej wszechstronne narzędzie o nazwie **ss** (socket statistics). Oba narzędzia pozwalają na monitorowanie aktywnych połączeń sieciowych i powiązanych z nimi informacji.

# Co daje?

- ▶ Wyświetla aktywne połączenia sieciowe: Pokazuje listę wszystkich aktualnie otwartych połączeń TCP i UDP, wraz z adresami lokalnymi i zdalnymi oraz stanem połączenia.
- ▶ Wyświetla nasłuchujące porty: Pokazuje, które porty na Twoim komputerze są aktualnie otwarte i na których aplikacje nasłuchują przychodzących połączeń.
- ▶ Pokazuje tablicę routingu: Podobnie jak route print, może wyświetlić tablicę routingu.
- ▶ Wyświetla statystyki interfejsów sieciowych: Pokazuje informacje o ruchu przychodzącym i wychodzącym na każdym interfejsie sieciowym (ilość wysłanych i odebranych pakietów, błędy itp.).
- ▶ Dostarcza informacji o połączeniach multicast.
- ▶ ss (w Linuxie) oferuje bardziej szczegółowe informacje o gniazdach (socketach) i jest zazwyczaj szybsze.

# Kluczowe zastosowania w rozwiązywaniu problemów sieciowych

- ▶ Sprawdzanie portów nasłuchujących: Upewnij się, że kluczowe usługi (np. serwer WWW na porcie 80/443) prawidłowo nasłuchują na oczekiwanych portach.
- ▶ Wykrywanie niechcianych połączeń: Analizuj aktywne połączenia w poszukiwaniu podejrzanych adresów IP lub nieznanых aplikacji, które mogą wskazywać na problemy bezpieczeństwa.
- ▶ Diagnostowanie problemów z serwerami: Jeśli masz problemy z połączeniem z serwerem, sprawdź stan połączenia (ESTABLISHED, TIME\_WAIT itp.) za pomocą netstat / ss, aby zidentyfikować, na jakim etapie występuje problem.
- ▶ Monitorowanie interfejsów sieciowych: Obserwuj statystyki interfejsów (błędy, odrzucone pakiety), aby wykryć potencjalne problemy z fizycznym połączeniem sieciowym lub sterownikami.
- ▶ Testowanie konfiguracji zapory ogniowej: Sprawdź, czy próby połączeń są odrzucane (np. brak stanu ESTABLISHED), co może wskazywać na blokowanie przez zaporę.



# Przykład użycia

```

lukasz@szenkiel-ubuntu: ~
lukasz@szenkiel-ubuntu:~$ ss -tuln
Netid State  Recv-Q  Send-Q           Local Address:Port       Peer Address:Port Process
udp    UNCONN  0        0           127.0.0.53%lo:53         0.0.0.0:*
udp    UNCONN  0        0           0.0.0.0:50001          0.0.0.0:*
udp    UNCONN  0        0           0.0.0.0:5353           0.0.0.0:*
udp    UNCONN  0        0           0.0.0.0:58680          0.0.0.0:*
udp    UNCONN  0        0           [fe80::f72:671a:527d:75fb]%wlx3c64cfcabdd3:546  [::]:*
udp    UNCONN  0        0           [::]:5353              [::]:*
udp    UNCONN  0        0           [::]:55931             [::]:*
tcp    LISTEN  0        128          0.0.0.0:22             0.0.0.0:*
tcp    LISTEN  0        4096          0.0.0.0:80             0.0.0.0:*
tcp    LISTEN  0        4096          127.0.0.53%lo:53       0.0.0.0:*
tcp    LISTEN  0        511          127.0.0.1:2658         0.0.0.0:*
tcp    LISTEN  0        4096          0.0.0.0:5432           0.0.0.0:*
tcp    LISTEN  0        10          0.0.0.0:7070           0.0.0.0:*
tcp    LISTEN  0        5           127.0.0.1:631          0.0.0.0:*
tcp    LISTEN  0        128          [::]:22                [::]:*
tcp    LISTEN  0        4096          [::]:80                [::]:*
tcp    LISTEN  0        4096          [::]:5432              [::]:*
tcp    LISTEN  0        10          [::]:7070              [::]:*
tcp    LISTEN  0        5           [::1]:631              [::]:*
lukasz@szenkiel-ubuntu:~$

```

# nslookup

**nslookup** (name server lookup) to narzędzie wiersza poleceń służące do wykonywania zapytań do serwerów DNS (Domain Name System). Umożliwia użytkownikom wyszukiwanie rekordów DNS dla określonych nazw domen lub adresów IP. Można go używać do znalezienia adresu IP powiązanego z daną nazwą domeny (forward lookup) lub nazwy domeny powiązanej z danym adresem IP (reverse lookup). Jest to przydatne narzędzie do diagnozowania problemów związanych z rozpoznawaniem nazw domen.

# Co daje?

- ▶ Sprawdzenie adresu IP dla danej domeny: Pozwala znaleźć adres IP serwera, na którym hostowana jest dana strona internetowa lub inna usługa sieciowa.
- ▶ Weryfikacja działania serwerów DNS: Umożliwia sprawdzenie, czy skonfigurowane serwery DNS działają poprawnie i odpowiadają na zapytania.
- ▶ Diagnostyka problemów z rozpoznawaniem nazw: Jeśli nie możesz otworzyć stron internetowych, nslookup może pomóc ustalić, czy problem leży po stronie serwerów DNS.
- ▶ Wyszukiwanie różnych typów rekordów DNS: Pozwala na wyszukiwanie nie tylko rekordów A (mapujących nazwę na IP), ale także MX (dla serwerów poczty), CNAME (aliasów), NS (serwerów nazw dla domeny) i innych.
- ▶ Wykonanie odwrotnego wyszukiwania DNS: Umożliwia znalezienie nazwy domeny powiązanej z danym adresem IP.
- ▶ Możliwość wyboru konkretnego serwera DNS do zapytania: Pozwala na testowanie odpowiedzi z różnych serwerów DNS.

# Kluczowe zastosowania w rozwiązywaniu problemów sieciowych

- ▶ Weryfikacja, czy nazwa strony internetowej (domena) jest poprawnie zamieniana na adres IP.
- ▶ Sprawdzenie, czy Twoje serwery DNS odpowiadają na zapytania o adresy internetowe.
- ▶ Pomoc w ustaleniu, czy problem z otwieraniem stron leży po stronie DNS.
- ▶ Sprawdzanie, gdzie kierowana jest poczta e-mail dla danej domeny (rekordy MX).
- ▶ Możliwość ręcznego zapytania konkretnego serwera DNS o informacje.

# Przykład użycia



```
lukasz@szenkiel-ubuntu: ~  
lukasz@szenkiel-ubuntu:~$ nslookup  
> collegiumwitelona.pl  
Server:          127.0.0.53  
Address:         127.0.0.53#53  
  
Non-authoritative answer:  
Name:   collegiumwitelona.pl  
Address: 156.17.194.39  
> www.zaglebie.com  
Server:          127.0.0.53  
Address:         127.0.0.53#53  
  
Non-authoritative answer:  
www.zaglebie.com      canonical name = zaglebie.com.  
Name:   zaglebie.com  
Address: 89.161.250.52  
> lukasz@szenkiel-ubuntu:~$ ^C  
lukasz@szenkiel-ubuntu:~$
```

# Wireshark

Wireshark to popularny, darmowy i otwarty źródłowo analizator pakietów sieciowych. Pozwala na przechwytywanie i interaktywne przeglądanie danych przesyłanych przez sieć w czasie rzeczywistym. Umożliwia szczegółową analizę każdego pakietu, w tym jego zawartości, nagłówków protokołów (takich jak Ethernet, IP, TCP, UDP, DNS, HTTP i wielu innych) oraz danych. Wireshark jest niezwykle potężnym narzędziem do diagnozowania problemów sieciowych, analizowania protokołów komunikacyjnych, badania bezpieczeństwa sieci i wielu innych zastosowań.

# Co daje?

- ▶ Szczegółowa analiza każdego pakietu sieciowego.
- ▶ Zrozumienie działania protokołów (TCP, UDP, HTTP, DNS itp.).
- ▶ Diagnozowanie problemów z połączeniem (retransmisje, opóźnienia).
- ▶ Badanie wydajności sieci (czas transferu pakietów).
- ▶ Rozwiązywanie problemów z aplikacjami sieciowymi (analiza komunikacji).
- ▶ Wykrywanie podejrzanego ruchu i problemów z bezpieczeństwem.

# Kluczowe zastosowania w rozwiązywaniu problemów sieciowych

- ▶ Identyfikacja problemów z nawiązywaniem połączeń TCP (np. brak synchronizacji SYN/ACK).
- ▶ Analiza problemów z DNS.
- ▶ Diagnozowanie błędów w komunikacji HTTP (np. kody odpowiedzi 4xx, 5xx).
- ▶ Badanie problemów z opóźnieniami w sieci poprzez analizę czasu podróży pakietów.
- ▶ Wykrywanie retransmisji i duplikatów pakietów wskazujących na problemy z siecią fizyczną lub przeciążeniem.
- ▶ Analiza ruchu sieciowego pod kątem potencjalnych problemów z bezpieczeństwem (np. próby nieautoryzowanego dostępu).
- ▶ Rozwiązywanie problemów z działaniem protokołów aplikacyjnych (np. SMTP, POP3, FTP).



# iperf

**iperf** to popularne narzędzie wiersza poleceń służące do pomiaru przepustowości sieci TCP i UDP. Pozwala na testowanie prędkości połączenia między dwoma hostami w sieci. Jeden host działa jako serwer, a drugi jako klient, wysyłając strumień danych do serwera. iperf mierzy przepustowość (throughput), opóźnienie (latency) oraz utratę pakietów podczas tego transferu. Jest to bardzo przydatne narzędzie do diagnozowania problemów z wydajnością sieci i weryfikowania osiągniętych połączeń. Istnieje również nowsza wersja o nazwie iperf3, która oferuje dodatkowe funkcje i jest kompatybilna wstecz.

# Co daje?

- ▶ Pomiar przepustowości (throughput) sieci TCP i UDP.
- ▶ Testowanie prędkości połączenia między dwoma hostami.
- ▶ Identyfikacja wąskich gardeł w sieci.
- ▶ Weryfikacja osiągniętych połączeń sieciowych.
- ▶ Pomiar opóźnienia (latency) i jittera (zmienności opóźnień).
- ▶ Wykrywanie utraty pakietów.
- ▶ Możliwość dostosowania parametrów testu (np. rozmiar okna TCP, protokół).

# Kluczowe zastosowania w rozwiązywaniu problemów sieciowych

- ▶ Weryfikacja przepustowości łącza internetowego lub sieci lokalnej.
- ▶ Identyfikacja spadków wydajności sieci po zmianach konfiguracji lub dodaniu nowych urządzeń.
- ▶ Diagnozowanie problemów z prędkością transferu plików.
- ▶ Testowanie jakości połączenia dla aplikacji czasu rzeczywistego (np. VoIP, wideo streaming).
- ▶ Lokalizowanie wąskich gardeł w infrastrukturze sieciowej (np. przełączniki, routery, okablowanie).
- ▶ Porównywanie rzeczywistej przepustowości z deklarowaną przez dostawcę usług internetowych.
- ▶ Testowanie wpływu różnych ustawień sieciowych (np. MTU, rozmiar okna TCP) na wydajność.

Dziękuję za uwagę