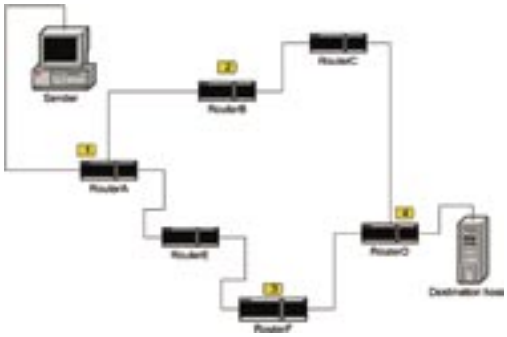
Programy diagnostyczno-narzędziowe do sieci TCP/IP w Windows

Maciej Szmit

Włączamy komputer i po krótszej lub dłuższej chwili związanej z ładowaniem systemu operacyjnego wita nas pulpit ze znajomą ikonką przeglądarki internetowej. Uruchamiamy zatem przeglądarkę i wpisujemy adres strony WWW aby po chwili cieszyć się lekturą zawartości naszych ulubionych stron... A teraz pytanie: co działo się od momentu włączenia zasilania do chwili, kiedy w okienku przeglądarki zaczęły pojawiać się obrazki i napisy – i co ewentualnie mogło pójść źle? Przyjmijmy, że mamy zainstalowany system MS Windows 2000 Professional, a do Internetu przyłączeni jesteśmy za pośrednictwem sieci lokalnej zbudowanej w oparciu o standard ethernet i że korzystamy z usług serwera DNS znajdującego się gdzieś u naszego providera. Żeby było łatwiej w odpowiedzi możemy pominąć rozważania związane z ładowaniem systemu operacyjnego.

Skąd bierze się sieć w komputerze

Choć pytanie wygląda trywialnie, odpowiedź na nie może zająć całkiem sporo czasu. Zacznijmy od początku. Przede wszystkim, aby móc korzystać z usług internetowych nasz komputer musi mieć zainstalowaną obsługę jakichś protokołów warstwy sieci. Zasadniczo możemy sobie wyobrazić kilka rozwiązań. Najprostsze z nich polega na tym, że nasz komputer ma zainstalowaną obsługę rodziny protokołów TCP/IP i przyznany na stałe adres IP wpisany w lokalnych ustawieniach konfiguracyjnych. Jest to rozwiązanie proste, ale dość niebezpieczne, dlatego znacznie bardziej prawdopodobne jest, że administrator naszej sieci zdecydował się na coś innego. Być może uruchomił na którymś z serwerów usługę pozwalającą na dynamiczny przydział adresu IP na żądanie stacji roboczych, jest też całkiem możliwe, że w naszej sieci lokalnej funkcjonują adresy prywatne (nie widziane na zewnątrz), a wszelkie informacje wychodzą z niej z adresem zewnętrznego interface naszego routera, który z kolei tłumaczy przychodzące odpowiedzi na odpowiednie adresy prywatne. W końcu, jeżeli administrator ma naprawdę ważne powody, możliwe jest, że na naszym komputerze zainstalowano obsługę jakiegoś innego stosu protokołów (np. IPX/SPX znanego ze starszych wersji Novell NetWare) wraz z odpowiednim oprogramowaniem, a po drodze z sieci lokalnej do Internetu stoi brama (gateway)



Rysunek 1. *Network Address Translation – zasada działania*

obsługująca translację informacji pomiędzy protokołami i zapewniająca przekazywanie informacji napływającej z Internetu w pakietach IP do odpowiednich maszyn w sieci wewnętrznej (tym razem już w pakietach IPX).

Rozwiązania polegające na zastosowaniu w sieci wewnętrznej zestawu adresów prywatnych (niewidzianych na zewnątrz) nazywane są ogólnie Network Address Translation (NAT), przy czym miłośnicy Linuksa rozróżniają NAT oraz PAT i nazywają to wszystko maskaradą. Samych NATów jest zresztą kilka rodzajów na przykład statyczny, dynamiczny czy negatywny.

NAT działa w ten sposób, że przychodzący na jego wewnętrzny interfejs pakiet (z adresem naszej stacji roboczej, oczywiście wziętym z puli adresów prywatnych) routowany jest na jeden z interfejsów zewnętrznych skąd trafia do sieci z adresem tegoż interfejsu. Przychodząca odpowiedź jest odpowiednio przeadresowywana i przekazywana komu trzeba (Rysunek 1).

Jeżeli adresów prywatnych w sieci wewnętrznej jest więcej niż dostępnych adresów publicznych to jeden adres publiczny powiązany jest z kilkoma adresami prywatnymi, przy czym (żeby było wiadomo, do kogo skierować odpowiedź) każdemu z adresów wewnętrznych przyporządkowywany jest odpowiedni zestaw portów (z puli portów dynamicznych) na adresie zewnętrznym.

To rozwiązanie (ze zmianą portów) wyodrębniane jest czasem po d nazwą PAT. Oczywiście adresy w sieci wewnętrznej mogą być dowolne, ale dobrą praktyką jest stosowanie adresów z zarezerwowanej puli adresów prywatnych (tzw. ten-netting), która obejmuje zakresy adreso-

we 10.0.0.0-10.255.255.255, 172.16.0.0-172.31.255.255 oraz 192.168.0.0-192.168.255.255.

Jeżeli cała operacja zachodzi natomiast oczko wyżej (to znaczy pakiety przerzucane są nie między adresami IP, ale między protokołami – jak w przypadku gatewaya IPX/IP) to całe rozwiązanie nosi już dźwięczną nazwę pośrednika obwodowego (circuit-level proxy). Generalnie gatewayem nazywa się bowiem urządzenie działające w co najmniej czwartej warstwie modelu ISO/OSI (choć i tu w literaturze przedmiotu panuje niezły bałagan).

Najprawdopodobniej nasz administrator zastosował w sieci wewnętrznej adresy prywatne, na wszelki wypadek przydzielane komputerom dynamicznie. Dynamiczny przydział adresów IP może być realizowany przy wykorzystaniu kilku protokołów: RARP – Reverse Address Resolution Protocol, BootP – Bootstrap Protocol, a przede wszystkim – DHCP – Dynamic Host Configuration Protocol, który ma z nich wszystkich największe możliwości przekazywania informacji do stacji roboczych. DHCP przekazywać może nie tylko informacje dotyczące tego, jaki adres IP stacja otrzymuje i jaki jest adres bramy internetowej, ale również szereg informacji dodatkowych (np. o usługach serwerów NetWare dostępnych w naszej sieci).

Uff – uporaliśmy się mniej więcej z pierwszym problemem – wiemy mianowicie w jaki sposób nasz komputer może mieć zapewnione usługi warstwy sieci (III warstwy modelu referencyjnego ISO/OSI).

Żeby się przekonać jak to wygląda w naszym wypadku możemy posłużyć się odpowiednią opcją w panelu sterowania

Rysunek 2. *WinIPcfg – dla lubiących interfejs graficzny uzytkoników Windows 9**

ipconfig – program do wyświetlania konfiguracji protokołu IP

Pomoc: ipconfig `/?`

```
ipconfig [/? | /all | /release [adapter]
   | /renew [adapter]
   | /flushdns | /registerdns
   | /showclassid adapter
   | /setclassid adapter [classidtoset] ]
```

Opcje

- `adapter` – pełna nazwa interfejsu (można używać masek nazw * oraz ?)
- `/?` – wyświetla pomoc
- `/all` – wyświetla pełne informacje o konfiguracji
- `/release` – zwalnia adres IP dla wskazanego interfejsu
- `/renew` – odświeża adres IP dla wskazanego interfejsu (poprzez ponowne odpytanie serwera DHCP)
- `/flushdns` – usuwa wpisy odwzorowana nazw dziedzinowych z pamięci podręcznej
- `/registerdns` – odświeża wszystkie adresy DHCP i nazwy DNS
- `/displaydns` – wyświetla wpisy odwzorowana nazw dziedzinowych z pamięci podręcznej
- `/showclassid` – wyświetla identyfikatory klas DHCP dostępnych dla danego interfejsu
- `/setclassid` – modyfikuje (lub jeśli nie podano identyfikatora – usuwa) identyfikatory klas DHCP dostępnych dla danego interfejsu

Przykład

Polecenie `ipconfig` bez żadnych opcji pokazuje konfigurację interfejsów sieciowych

```
Windows 2000 - konfiguracja IP
0 Ethernet karta :

        Adres IP. . . . . . . . . . . : 192.168.2.4
        Maska podsieci . . . . . . . . : 255.255.255.0
        Domyślna brama . . . . . . . : 192.168.2.1
```

(konfigurację protokołów sieciowych można znaleźć we właściwościach protokołu TCP/IP – Micorosoft konsekwentnie nazywa stos protokołów – protokołem). W systemie mamy też pożyteczny program o nazwie *ipconfig* uruchamiany z konsoli tekstowej, który pozwala nam na wyświetlenie bieżących ustawień programów i bibliotek obsługi stosu protokołów TCP/IP na naszej maszynie i na żądanie odświeżenia całości bądź części konfiguracji poprzez ściągnięcie jej na nowo z serwera DHCP (w Windows z serii 9* mamy dodatkowo do dyspozycji graficzny program winipcfg – patrz Rysunek 2).

Ano właśnie – w jaki sposób nasza maszyna po starcie ma zażądać przesłania informacji konfiguracyjnych od serwera DHCP, którego adresu (ani logicznego ani fizycznego) przecież nie zna, bo i skąd? Oczywiście musi w tym celu

wysłać wiadomość do wszystkich czyli broadcast (dokładniej: umieścić tę wiadomość w ramce broadcastowej). Wiadomość ta nazywana jest DHCPDISCOVER i jest pytaniem do serwerów DHCP, które w odpowiedzi przedstawiają się wiadomością zwaną DHCPOFFER (wysłaną rzecz jasna już w zwykłej unicastowej ramce zaadresowanej do jednego odbiorcy). Klient odczekawszy chwilę (żeby zebrać ofertę od wszystkich zainteresowanych serwerów DHCP) odpowiada z kolei wiadomością DHCPREQUEST (zawierającą identyfikator wybranego serwera, ale broadcastową, żeby inne serwery DHCP, które zgłosiły gotowość świadczenia usług, też wiedziały co się dzieje). Wybrany serwer przesyła teraz wiadomość DHCPACK, w której zapisane są informacje potrzebne dla skonfigurowania klienta. Zatem na samym początku w sieci pojawiają się przynajmniej cztery wiadomości (przy założeniu że mamy jeden serwer DHCP).

Okno na świat

Nasza stacja robocza ma już przynajmniej podstawowe informacje: o własnym adresie logicznym, o używanej w sieci lokalnej masce, o adresach logicznych wyjścia na świat serwerów DNS i jeszcze być może kilka innych. Wpisując adres w przeglądarkę internetową wysyłamy, jak wiadomo, zapytanie do docelowego webserwera... Stop! Skąd mianowicie nasza stacja wie, jaki jest adres IP serwera, na którym umieszczono strony WWW? Żeby się tego dowiedzieć musi skorzystać z usługi odwzorowania nazw, która dla podanej nazwy symbolicznej zwróci odpowiedni adres numeryczny (IP). W tym celu powinna skontaktować się z DNS-em (Domian Name Server – Serwer Nazw Dziedzinowych). Jeżeli, jak to często bywa, serwer nazw dziedzinowych, z którego korzysta nasza sieć znajduje się poza siecią lokalną, to ramka niosąca na sobie pakiet IP, zawierający datagram UDP z zapytaniem DNS (brzmi to strasznie, nieprawdaż?) trafia do bramy wyjściowej z naszej sieci.

Łatwo powiedzieć trafia, tylko jak to robi? Przecież nasza stacja zna wprawdzie adres IP bramy, ale nie ma pojęcia jaki jest jej adres fizyczny (MAC adres), a ramka (która istnieje w warstwie łączenia danych) zaadresowana jest adresem fizycznym właśnie. Nasza stacja musi więc skorzystać z protokołu, który pozwala (broadcastowo) zapytać w sieci o adres

arp – program do obsługi protokołu arp

Pomoc: `arp /?`

```
ARP -s inet_addr eth_addr [if_addr]
ARP -d inet_addr [if_addr]
ARP -a [inet_addr] [-N if_addr]
```

Opcje

- `/?` – wyświetla pomoc
- `-a` – wyświetla bieżące wpisy ARP (można podać dla jakiego interfejsu – opcja – N lub odpytać o rozwiązanie konkretnego adresu IP)
- `-g` – działa jak -a
- `inet_addr` – adres IP
- `-N if_addr` – Wyświetla wpisy dla konkretnego interfejsu
- `-d` – Usuwa wpis dla hosta o podanym adresie (można użyć maski * podówczas będą usunięte wszystkie wpisy)
- `-s` – dodaje wpis do tablicy ARP. Wpisy są dodawane na stałe
- `eth_addr` – adres MAC
- `if_addr` – adres IP dla interfejsu, dla którego chcemy wykonać podaną operację

Przykład

Polecenie:

```
arp -s 157.55.85.212 00-aa-00-62-c6-09
```

dodaje wpis przyporządkowujący adresowi IP 157.55.85.212 adres fizyczny (MAC) 00-aa-00-62-c6-09.

fizyczny (MAC) komputera o podanym adresie logicznym (IP). Protokołem tym jest Address Resolution Protocol – ARP. Samo zapytanie nosi nazwę ARP-Request, zaś udzielana na nie (już normalnie, czyli unicastowo) odpowiedź – ARP-reply. Mamy zatem dwie kolejne ramki w naszej sieci – tym razem służące do ustalenia adresu fizycznego bramy. Oczywiście adres ten jest zapisywany w podręcznej pamięci (ARP cache) w naszym komputerze, żeby nie zapychać sieci niepotrzebnymi pytaniami za każdym razem, kiedy chcemy coś wysłać w świat. W systemie Windows mamy kolejny pożyteczny program pozwalający na obejrzenie i zmianę zawartości ARP cache. Jak łatwo się domyślić nosi on nazwę *arp*.

Ale wracajmy do DNS-a (oczywiście jeśli jest on w sieci lokalnej, to zanim nasza stacja nawiąże z nim połączenie musi przesłać zapytanie ARP o MAC adres jego, a nie bramy). Zakładamy, że DNS, z którego korzystamy, jest poza naszą siecią, zatem nasza brama (czy raczej router stojący na wyjściu z naszej sieci) otrzymuje zaadresowaną do siebie (fizycznie) ramkę zawierającą pakiet IP zaadresowany (logicznie) do DNS-a. Router jest urządzeniem trzeciej warstwy modelu referencyjnego ISO/OSI, zatem umie zajrzeć do nagłówka pakietu IP, stwierdzić, że ten nie jest przeznaczony dla niego, zajrzeć do tablicy routingu aby sprawdzić dokąd (na który ze swoich interfejsów) przekierować taki pakiet, przepakowawszy go przy okazji do ramki odpowiedniej dla tego interfejsu (bo

nslookup – program do obsługi systemu nazw symbolicznych

Pomoc: po uruchomieniu programu bez żadnych parametrów w trybie konwersacyjnym należy wpisać polecenie help

Komendy dostępne w trybie konwersacyjnym:

`NAZWA` – wyświetla informacje o nazwie hosta lub domeny pobrane z domyślnego DNSu

`NAZWA1 NAZWA2` – wyświetla informacje o nazwie hosta lub domeny pobrane z DNSu o adresie NAZWA2

`help` – wyświetla pomoc

`?` – wyświetla pomoc

`set OPCJA` – ustawia opcje (patrz poniżej)

Opcje

- all – wyświetla opcje i informacje o bieżącym hoście i serwerze
- [no]debug – wyświetla informacje debugera
- [no]d2 – wyświetla szczegółowe informacje debugera
- [no]defname – dodaje nazwę domenową do każdego zapytania
- [no]recurse – prosi o rekursywną odpowiedź
- [no]search – używa listy przeszukiwania domen
- [no]vc – zawsze używa obwodu wirtualnego
- domain=NAZWA – ustawia domyślną nazwę domeny na NAZWA
- srchlist=N1[/N2/.../N6] – ustawia domenę na N1, a listę przeszukiwania na N1,N2 itd.
- root=NAZWA – ustawia serwer główny (rootserver) na NAZWA
- retry=X – ustawia liczbę ponawianych prób na X
- timeout=X – ustawia początkowy limit czasu na X sekund
- type=X – ustawia zapytanie o rekord określonego typu (np. A, ANY, CNAME, MX, NS, PTR, SOA,SRV)
- querytype=X – działa tak samo jak opcja type
- class=X – ustawia klasę zapytania (np. IN (Internet), ANY)
- [no]msxfr – używa szybkiego transferu strefy MS
- ixfrver=X – bieżąca wersja do użycia w żądaniu transferu IXFR
- server NAZWA – ustawia domyślny DNS na NAZWA, używając bieżącego
- serwera domyślnego
- lserver NAZWA – ustawia domyślny serwer na NAZWA, używając serwera początkowego
- finger [UŻYTKOWNIK] – uzyskuje informacje o użytkowniku opcjonalnym z bieżącego hosta domyślnego
- root – ustawia bieżący serwer domyślny jako główny
- ls [opt] DOMENA [> PLIK] – wyświetla adresy w domenie DOMENA (opcjonalne: kieruje wyniki do pliku PLIK)
- -a – wyświetla kanoniczne nazwy i aliasy
- -d – wyświetla wszystkie rekordy
- -t TYP – wyświetla rekordy określonego typu (np. A, CNAME, MX, NS, PTR itd.)
- view PLIK – sortuje plik wynikowy polecenia ls i wyświetla go używając pg
- exit – kończy pracę programu

Przykład

Polecenie `nslookup www.polska.pl` wyświetli informacje o hoście *www.polska.pl*

może to przecież być zupełnie inny rodzaj kabelków, czyli protokołów I i II warstwy) i wreszcie wysłać w świat. Przy okazji, jeżeli router obsługuje funkcje NAT albo PAT dokonuje odpowiedniej modyfikacji adresów źródłowych, a jeżeli jest routerem filtrującym (ang. *screening router* – podstawowy komponent ściany przeciwogniowej) to uprzednio sprawdza, czy przypadkiem administrator firewalla nie zakazał wypuszczać pakietów zaadresowanych pod wskazany adres albo pochodzących z danej maszyny. Wszystkie te operacje potrafią zająć routerowi nawet kilka milisekund. Oczywiście przez wysłanie w świat należy rozumieć przesłanie pod adres wskazany w tablicy routingu – adres kolejnego routera, czyli jak wolą niektórzy – następnego hopa (ang. *next hop*). Żeby coś (nasz pakiet) wysłać pod ten adres (logiczny) należy odpowiednio (fizycznie) zaadresować ramkę, która będzie go niosła; jeśli jest więc to pierwszy pakiet od dłuższego czasu, należy sprawdzić adres fizyczny następnego hopa pytając go o to po ARP (ale to już naprawdę czarny scenariusz).

Przez kolejne routery zapytanie dociera (jeżeli wszystko dobrze pójdzie) do naszego DNS-a, który najpierw sprawdza, czy dany adres jest mu znany (może to właśnie nasz DNS obsługuje domenę, do której ów adres należy, a może przed chwilą ktoś już o ten adres pytał, więc odwzorowanie znajduje się jeszcze w pamięci podręcznej – DNS cache), jeżeli nie – wysyła zapytanie do serwera domeny, w której znajduje się podany adres (być może będzie zmuszony zaczynać od góry czyli od rootserwerów obsługujących domeny najwyższe w hierarchii, czyli na przykład *.com* albo *.edu*). Nie wnikając w szczegóły wymiany informacji między DNS-ami, po pewnym czasie do naszej maszyny (znowu za pośrednictwem bramy) trafia odpowiedź odwzorowująca podany adres symboliczny na adres IP. Aha: do ręcznego odpytywanie DNSa w Windows służy program *nslookup*.

Trochę metafizyki

Dalej już wszystko idzie jak już z płatka – przeglądarka wysyła żądanie do serwera WWW. Zapytanie będzie oczywiście przesłane protokołem HTTP (Hyper Text Transfer Protocol). Protokół ten, jak pewnie wiemy, korzysta z połączenia TCP (zestawianego na porcie 80 serwera), zatem zanim zaczniemy wysyłanie danych (czyli w naszym przypadku żądania przesłania nam zawartości stron WWW) należy zestawić wirtualny obwód pomiędzy gniazdem naszej przeglądarki a procesem serwera WWW (w niektórych systemach operacyjnych, żeby było zabawniej, takie procesy nazywa się demonami).

Mechanizm zestawiania wirtualnych obwodów pomiędzy gniazdami demonów noszący nazwę trójstronnego uścisku ręki (three-way handshake) jest znany od przedszkola każdemu demonologowi, to jest chciałem powiedzieć infor-

netstat – program do badania statystyki sieci

Pomoc: `netstat /?`

```
NETSTAT [-a] [-e] [-n] [-s] [-p proto] [-r] [interval]
                    [/?]
```

Opcje

- `/?` – wyświetla pomoc
- `-a` – wyświetla wszystkie połączenia i porty nadsłuchujące.
- `-e` – wyświetla statystyki Ethernetu
- `-n` – wyświetla adresy i numery portów w formie numerycznej
- `-p proto` – wyświetla połączenia dla wyspecyfikowanego protokołu (TCP, IP lub UDP)
- `-r` – wyświetla tabelę routingu
- `-s` – wyświetla połączenia dla protokołów (domyślnie dla TCP, UDP i IP)
- `interval` – wyświetla statystyki co określoną liczbę sekund (zatrzymanie CTRL+C)

Przykład

Polecenie `netstat -n` pokazuje stan połączeń posługując się adresami numerycznymi

```
Aktywne połączenia
Prot.  Adres lokalny        Obcy adres      Stan
TCP    162.87.198.100:1537  61.31.32.28:80  ESTABLISHED
TCP    162.87.198.100:1538     212.51.6.28:80  SYN_SENT
```

Jak widać nasz komputer (162.87.198.100) ma w tej chwili nawiązane połączenie z adresem (61.31.32.28) oraz wysłał pakiet TCP SYN na adres 212.51.6.28. Komunikacja z oboma adresami odbywa się na porcie osiemdziesiątym TCP (po stronie adresata) czyli prawdopodobnie przeglądamy stronę WWW.

Przykład

Polecenie `netstat -a -e -s` pokazuje szczegółowe informacje o ruchu na karcie ethernetowej (w rozbiciu na poszczególne protokoły).

```
Statystyki interfejsu
                           Odebrano            Wysłano

Bajty                       1381188             345886
Pakiety unicast                4303               3904
Pakiety inne niż unicast         94                104
Odrzucone                         0                  0
Błędy                             0                  0
Nieznane protokoły              126

Statystyki IP

  Otrzymane pakiety                    = 4287
  Otrzymane błędy nagłówka             = 0
  Otrzymane błędy adresu               = 36
...
```

matykowi, dla przypomnienia dodajmy jednak, że składa się on z segmentów TCP SYN (od klienta do serwera), SYN ACK (od serwera do klienta) i ACK (od klienta do serwera). Rzecz jasna każdy z segmentów jest wysyłany w pakiecie IP normalną drogą przez bramę z lub do sieci zewnętrznej.

Dzięki tej powitalnej wymianie segmentów ustanawiana jest w pełni dwukierunkowa komunikacja, którą system widzi jako swojego rodzaju osobny dwukierunkowy kanał łączności (zwany właśnie obwodem wirtualnym – ang. *virtual circuit*). Ustanowione połączenia, jak również dodatkowe informacje dotyczące statystyki protokołów możemy w Windows obejrzeć przy użyciu programu *netstat*..

Po zakończeniu powitania nasz klient przesyła protokołem HTTP zapytanie składające się z komendy GET, a w odpowiedzi serwer odsyła kod strony głównej (oczywiście po drodze potwierdziwszy otrzymanie pakietów niosących komendę GET odpowiednimi pakietami ACK, w liczbie i na zasadach określonych przez początkowy rozmiar okna TCP).

Jeżeli chcielibyśmy zobaczyć to na własne oczy możemy skorzystać z programu `telnet`, któremu nakazujemy połączyć się z naszym ulubionym serwerem WWW podając jako port docelowy port o numerze 80. W Windows 9* telnet jest programem graficznym, gdzie numer portu możemy wpisać w odpowiednim okienku, natomiast w Windows 2000 telnet to aplikacja konsolowa, musimy zatem wywołać go z odpowiednim parametrem, na przykład:

```
telnet www.p.lodz.pl 80
```

Uwaga: między nazwą serwera a numerem portu jest spacja (w innych programach czasami numer portu oddziela się dwukropkiem). Połączywszy się piszemy (wielkimi literami)

telnet – zapewnia połączenie ze zdalnym hostem

Pomoc: telnet /?

```
telnet [host [port]] [/?]
```

Dostępne opcje

- `/?` – wyświetla pomoc
- `host` – nazwa lub adres IP zdalnego komputera
- `port` – nazwa usługi lub numer portu

Przykład

Polecenie `telnet www.polska pl 80` uruchomi połączenie z portem 80 serwera *www.polska.pl*

ping – testuje połączenie przy użyciu pakietów ICMP echo

Pomoc: `ping /?`

```
ping [-t] [-a] [-n liczba] [-l rozmiar] [-f] [-i TTL]
[-v TOS] [-r liczba] [-s liczba] [[-j lista_hostów]
| [-k lista_hostów]]
[-w limit_czasu] lista miejsc docelowych
```

Opcje

- `-t` – odpytuje określonego hosta do czasu zatrzymania (CTRL C aby zakończyć lub CTRL BREAK aby zatrzymać a później kontynuować)
- `-a` – tłumaczy adresy na nazwy hostów
- `-n` – określa liczbę wysyłanych pakietów
- `-l` – określa rozmiar przesyłanego pakietu
- `-f` – wysyła pakiety z ustawioną flagą „nie fragmentuj” (do not fragment)
- `-i TTL` – wysyła pakiety z określonym jako parametr TTL czasem życia (time to live)
- `-v TOS` – wysyła pakiety z określonym jako parametr TOS wskaźniekiem „typ usługi” (type of service)
- `-r liczba` – rejestruje kolejne hopy na trasie. Maksymalna wartość parametru liczba może wynosić 9
- `-s liczba` – wysyła pakiety z sygnaturą czasową dla przeskoków
- `-j lista_hostów` – wysyła pakiety z ustawioną wymuszoną trasą routingu wg reguły swobodnej trasy źródłowej (loose) wg listy hostów podanej jako parametr lista_hostów
- `-k lista_hostów` – wysyła pakiety z ustawioną wymuszoną trasą routingu wg reguły ścisłej trasy źródłowej (strict) wg listy hostów podanej jako parametr lista_hostów
- `-w limit_czasu` – określa limit czasu oczekiwania na odpowiedź (w milisekundach)

Przykład

Polecenie `ping www.polska.pl -r 9 -n 1` wysyła pojedynczy pakiet *ICMP echo* na adres *www.polska.pl*, przy czym w pakiecie zarejestrowane zostanie kolejne osiem hopów (routerów) przez które on przechodzi (jest to specyficzna możliwość ICMP pozwalająca śledzić trasę lepiej niż przy użyciu programu *tracert*, ale niestety tylko do ośmiu hopów).

```
Badanie polska.pl [193.59.201.35] z użyciem 32 bajtów
                        danych:
Odpowiedź z 193.59.201.35: bajtów=32 czas=196ms TTL=241
Trasa: 217.30.153.24 ->  odpowiednich tras trwałych.
  217.30.128.62 ->       odpowiednich tras trwałych.
  217.30.129.130 ->      odpowiednich tras trwałych.
(...)
  195.187.254.25         odpowiednich tras trwałych.
Statystyka badania dla 193.59.201.35:
  Pakiety: Wysłane = 1, Odebrane = 1, Utracone = 0 (0%
                        utraconych),
Szacunkowy czas błądzenia pakietów w milisekundach:
  Minimum = 196ms, Maksimum =  196ms, Średnia =  196ms
```

GET (nie przejmując się, że na ekranie tego nie widać) i wciskamy dwukrotnie [*Enter*].

Podsumowując zatem: od uruchomienia komputera do wyświetlenia w okienku przeglądarki zawartości stron WWW zaszły w naszej sieci lokalnej następujące zdarzenia:

- klient wysyła zapytanie DHCPDISCOVER
- serwer dhcp przedstawia się DHCPOFFER
- klient żąda dostarczenia danych DHCPREQUEST
- serwer dhcp dostarcza dane DHCPACK
- klient żąda adresu fizycznego bramy ARP-request
- brama podaje swój adres fizyczny ARP-reply
- klient wysyła zapytanie do DNS-a
- serwer nazw dziedzinowych odpowiada
- klient rozpoczyna handshake TCP SYN z serwerem docelowym
- serwer wysyła do klienta potwierdzenie i żądanie nawiązania połączenia drugostronnego SYN ACK
- klient wysyła do serwera potwierdzenie ACK
- klient przesyła protokołem HTTP polecenie GET
- serwer potwierdza ACK
- serwer protokołem HTTP przesyła kod strony głównej

Powyższy opis obejmuje teoretyczną sytuację, w której wszystko działało jak należy.

Oczywiście nie każda wysłana wiadomość musiała się mieścić w jednym segmencie czy datagramie i nie każdy datagram (segment) musiał być umieszczony w jednej ramce, na dodatek jeżeli doszło do jakichś nieporozumień, mogły być wysłane żądania retransmisji bądź inne informacje związane z obsługą błędów więc rzeczywista liczba ramek mogła być większa.

Jeżeli mamy do dyspozycji sniffer (a jeżeli nie mamy, to zainstalujmy z dołączonej do pisma płyty CD) możemy tę wymianę informacji prześledzić na własne oczy z innej maszyny znajdującej się w tym samym segmencie sieci (czy dokładniej: w tej samej domenie kolizyjnej).

Niestety, w życiu zdarzają się również sytuacje, w których zamiast oczekiwanych tekstów i obrazków otrzymujemy przykrą informację w rodzaju *system nie mógł otworzyć strony*. Na szczęście dysponujemy kilkoma narzędziami, które pozwalają nam na wyszukanie prawdopodobnej przyczyny błędu.

Pierwszym potencjalnym źródłem problemu jest oczywiście fizyczne uszkodzenie karty sieciowej, urządzeń aktywnych lub medium, w szczególności patchcorda łączącego nasz komputer z gniazdkiem w ścianie (choć brzmi to nieprawdopodobnie spotkałem kiedyś administratora, który konfigurując sieć

Sieciowe programy użytkowe systemu Windows związane z protokołami stosu TCP/IP

- FTP – klient file transfer protocol
- TFTP – klient trivial file transfer protocol
- Telnet – narzędzie emulacji terminala
- RCP - klient remote copy protocol
- RSH – remote shell
- REXEC – narzędzie do uruchamiania procesów na komputerze zdalnym
- Finger – klient usługi finger

zapomniał w ogóle o podłączeniu komputerów do gniazdek sieciowych). Jeżeli jednak po rzuceniu na przewody i gniazda okiem (a jeszcze lepiej po sprawdzeniu ich testerem okablowania) nie widzimy nic podejrzanego powinniśmy za pomocą programu *ipconfig* sprawdzić, czy nasz komputer w ogóle ma jakiś adres IP. Jeżeli nie, można spróbować go odświeżyć poleceniem *ipconfig /renew_all* (o ile oczywiście w naszej sieci adresy sa przyznawane dynamicznie przez serwer DHCP). Jeśli i to nie pomoże to błędu szukajmy w serwerze dhcp lub lokalnie (uszkodzenie sprzętu sieciowego, błędna konfiguracja klienta TCP/IP). Jeśli nie mamy problemu z lokalnym adresem IP powinniśmy na wszelki wypadek sprawdzić, czy wszystkie zainstalowane lokalnie składniki obsługi stosu protokołów działają poprawnie. Dobrą metodą jest uruchomienie programu ping z adresem pętli zwrotnej `ping 127.0.0.1` oraz adresem IP interfejsu lokalnego. Jeśli i to się udało – pingujemy nasza bramę a później jakiś host z zewnątrz...

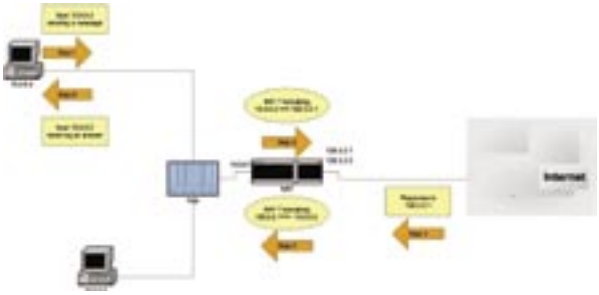
Po prostu ping

Ping jest jednym z najczęściej wykorzystywanych w diagnostyce programów. Jego zadaniem – jak wszyscy wiedzą – jest wysyłanie krótkich pakietów IP do wskazanego komputera, który z kolei odbija je (stąd nazwa) z powrotem do nadawcy. Spróbujmy nieco uściślić: protokół IP posiada wbudowany mechanizm przesyłania komunikatów kontrolnych. Mechanizm ten nosi nazwę Internet Control Message Protocol. Jedną z oferowanych przezeń usług (oznaczoną numerem 8) jest usługa echa. Jeżeli system obsługujący protokół ICMP otrzyma komunikat ICMP, w którego nagłówku w polu *typ* wpisana jest liczba 8 (ICMP echo), to odeśle go z powrotem, umieszczając w nagłówku komunikatu odpowiedzi liczbę 0 (taki komunikat nazywany jest *echo reply*). Ponieważ ICMP jest immanentną częścią IP możemy spodziewać się, że dowolny system operacyjny, w którym zaimplementowano obsługę sieci TCP/IP odpowie na pingi, oczywiście, o ile administrator nie ustawił gdzieś po drodze na zaporze przeciwogniowej filtrowania pakietów ICMP – co nie jest najlepszą praktyką, ale czasami się zdarza. Zdarza się między innymi dlatego, że – jak to bywa z każdym narzędziem – można mechanizm ICMP echo wykorzystać w niecny sposób. Przede wszystkim odpowiadanie na pingi zajmuje nieco zasobów maszyny docelowej, no i pochłania pewną część dostępnego pasma. Nie stanowi to problemu, gdy idzie o okazjonalne zdarzenia, polegające na tym, że jakiś użytkownik sprawdza działanie swojego wyjścia na świat (najlepiej zrobić to pingując jakiś stale włączony webserwer), ewentualnie testuje działanie docelowego hosta, z którego usług nie może z niewiadomych przyczyn skorzystać. Gorzej, gdy na maszynę zaczną napływać bez przerwy tysiące pakietów ICMP. Jedna z dużych firm programistycznych, której główny serwer WWW był wyjątkowo często wykorzystywany w takich celach była zmuszona postawić osobny serwer pingów, bo obsługa komunikatów ICMP pochłaniała zbyt dużo mocy webserwera.

Znacznie ciekawszym (biorąc pod uwagę tytuł naszego czasopisma) i groźniejszym zjawiskiem jest Ping of Death – rodzaj ataku DoS wykorzystujący pakiety ICMP Echo. Zgodnie z RFC 791 maksymalny rozmiar pakietu IP może wynosić 65535 oktetów (wliczając w to długość nagłówka). Komunikat ICMP umieszczony jest wewnątrz pakietu IP, przy czym nagłówek ICMP zajmuje 8 oktetów. Ponieważ nagłówek IP zajmuje zazwyczaj 20 oktetów (jeżeli pole opcje nie jest wykorzystane), w efekcie maksymalny rozmiar treści komunikatu ICMP może wynosić 65535 – 20 – 8 = 65507 oktetów. Oczywiście pakiety o długości przekraczającej wielkość pola danych ramki obowiązującej w danej sieci są fragmentowane (dzielone na mniejsze kawałki i przesyłane do systemu docelowego w kilku ramkach). Na miejscu w drodze system docelowy skleja (defragmentuje) zawartość ramek odzyskując w ten sposób oryginalny pakiet. Proces defragmentacji wykorzystuje pole przesunięcie (offset), które zawarte jest w każdym otrzymywanym fragmencie i które informuje w którym miejscu docelowego pakietu ma on być umieszczony. Możliwe jest takie spreparowanie przesyłanych fragmentów, że otrzymana w efekcie długość komunikatu będzie większa niż dopuszczalna (można w ten sposób bawić się nie tylko ICMP). Choć brzmi to dziwnie, działanie wielu – szczególnie starszych – systemów operacyjnych przy próbie obsłużenia takiego pakietu kończy się błędem krytycznym. Do wygenerowania ataku Ping of Death nie możemy się – na szczęście

Sieciowe programy narzędziowe systemu Windows związane z protokołami stosu TCP/IP

- Ping – testuje połączenie IP korzystając z protokołu ICMP
- ARP – wyświetla i modyfikuje odwzorowania adresów IP na adresy fizyczne
- Ipconfig – wyświetla i pozawla na odświeżenie oraz zwolnienie bieżącej konfiguracji protokołów
- Nbstat – wyświetla statystykę i połączenia korzystające z NetBIOS przez TCP/IP
- Netstat – wyświetla statystyki i połączenia dla protokołów TCP, UDP i IP
- Route – wyświetla i pozwala na modyfikacji lokalnej tabeli routingu
- Hostname – zwraca nazwę hosta dla komputera lokalnego (dla protokołów RCP, RSH i REXEC)



Rysunek 3. *Działanie programu tracert*

– posłużyć programem ping wbudowanym w system operacyjny Windows 2000 – maksymalny rozmiar przesyłanego komunikatu jest tam ograniczony do 65500 oktetów. Również większość dostępnych w sieci narzędzi ma ograniczony rozmiar wysyłanych pakietów, bez trudu jednak można znaleźć odpowiednie (czy raczej nieodpowiednie) exploity.

Uff (po raz drugi) – mamy już metodę sprawdzenia, czy nasz komputer widzi zewnętrzne adresy IP (i co ważniejsze wiemy na czym ona polega). Łatwo też sformułować odpowiednie wnioski: jeżeli uda się nam zapingować na bramę lub dowolny host w sieci lokalnej (tj. jeżeli nadejdzie zeń odpowiedź), a nie udaje się nam wyjść na zewnątrz, to winien jest któryś z routerów położonych między nami a komputerem docelowym lub sam komputer docelowy. Oczywiście cały czas mówimy o używaniu odpowiednich programów z parametrami będącymi adresami numerycznymi (IP). Rzecz jasna, jeżeli potrafimy gdzieś zapingować i prześledzić trasę przy użyciu adresów numerycznych, a nie możemy tego zrobić posługując się adresami symbolicznymi, to znaczy, że coś złego dzieje się z naszym DNS-em (być może po DHCP otrzymaliśmy błędny adres serwera nazw domenowych, a może sam DNS się popsuł, choć na taką ewentualność zazwyczaj mamy podany w naszej konfiguracji jeden lub dwa DNS-y zapasowe).

Tędy i owędy

Aby prześledzić trasę, którą są przesyłane nasze pakiety wykorzystujemy zazwyczaj program *tracert* (*trace route* – śledzenie ścieżki routingu). Działanie programu tracert opiera się na jednym z mechanizmów protokołu IP, który służy do ograniczenia czasu przetwarzania pakietów w intersieci. Mechanizm ten bazuje na informacji umieszczonej w polu TTL (*time to live* – czas życia) nagłówka pakietu IP. Każdy router, przez który przechodzi pakiet ma obowiązek zmniejszyć TTL o liczbę równą liczbie sekund, które zajęło przetwarzanie pakietu, a jeżeli taka informacja nie jest dostępna, to o jedność. Pakiety z TTL równym zero są odrzucane. W ten sposób unika się na przykład przetwarzania w nieskończoność pakietów, jeśli administrator ustawiłby niechcący cykliczną trasę routingu, jak również przechowywania w routerach w nieskończoność pakietów, które z jakichś powodów nie mogą być przez dłuższy czas dostarczone.

Wyznaczanie trasy routingu polega na wysyłaniu serii pakietów zaadresowanych do hosta docelowego, przy czym pole TTL pierwszego z pakietów ustawiane jest na zero. Najbliższy router zatem odrzuci taki pakiet i prześle do nadawcy komunikat (ICMP) o tym fakcie. W ten sposób program tracert uzyskuje informacje o pierwszym z routerów na ścieżce. Kolejne pakiety wysyłane są z powiększanymi o 1 wartościami pola TTL i odrzucane przez kolejne routery. Do hosta źródłowego przychodzą więc komunikaty pozwalające zidentyfikować ścieżkę routingu. Oczywiście ostatni sprawny router prześle informację o swoim adresie IP i o adresie kolejnego – już nieosiągalnego routera. Niestety (czy raczej na szczęście) spora część routerów w Internecie korzysta z dynamicznego wyznaczania tras i wysyłane pakiety nie muszą dochodzić do celu tymi samymi drogami, co powoduje, że informacje otrzymane z programu tracert mogą pokazywać ścieżkę nie istniejącą w rzeczywistości (zobacz Rysunek 3).

I to właściwie tyle, jeżeli idzie o działanie i testowanie działania dolnych warstw stosu protokołów TCP/IP w systemie operacyjnym Windows. Prawda, że proste? O tym, co dzieje się wyżej pomówimy innym razem.■

tracert – śledzi trasę routingu

Pomoc: tracert

tracert [-d] [-h maximum_hops] [-j host-list] [-w timeout] target_name

Dostępne opcje

- `-d` – nie rozwiązuje adresów numerycznych na nazwy hostów
- `-h` maximum_hops – określa maksymalną liczbę przeskoków (hopów) do przeszukania w drodze do celu
- `-j` lista_hostów – wysyła pakiety z ustawioną wymuszoną trasą routingu wg reguły swobodnej trasy źródłowej (loose) wg listy hostów podanej jako parametr *lista_hostów.*
- `-w` czas – czeka liczbę milisekund równą parametrowi *czas* na każdą odpowiedź

Przykład

Polecenie `tracert www.polska.pl -d` wyświetli trasę routingu do hosta *www.polska.pl* nie rozwiązując adresów numerycznych na nazwy symboliczne (porównaj użycie programu ping z opcją `-r`)

```
przewyższa maksymalną liczbę przeskoków 30

  1 156 ms 151 ms 162 ms 62.87.195.1
    odpowiednich tras trwałych
  2 155 ms 160 ms 151 ms 217.30.153.1
    odpowiednich tras trwałych
  4 163 ms 162 ms 168 ms 217.30.129.152
    odpowiednich tras trwałych
(...)
13 178 ms 180 ms 343 ms 193.59.201.35
   odpowiednich tras trwałych
Śledzenie zakończone.
```