

Polecenie

`ping -R <komp. docelowy> (Linux)`

albo

`ping -r 9 <komp. docelowy> (Windows)`

zapisuje trasę tam i z powrotem w POLU OPCJI nagłówka IPv4. W tym celu wykorzystywana jest opcja zapisu trasy (ang. Record Route). Ponieważ w pierwszym słowie pola opcji jest zapisany kod opcji, a pole opcji ma maksymalnie 40 bajtów, więc zostaje tam miejsce na 9 adresów IPv4. Z tego powodu nie da się tam zapisać całej trasy, jeśli jest zbyt długa.

Zapisywane są adresy interfejsów, przez które pakiet OPUSZCZA kolejne komputery na swojej trasie (oprócz komp. źródłowego i docelowego są to routery), przy czym Linux zapisuje jako pierwszy adres interfejsu źródłowego, a Windows - adres pierwszego routera.

### Ćwiczenie 1

Wyдай polecenie `ping -R` (Linux) albo `ping -r 9` (Windows) do komputera wskazanego przez prowadzącego i zanotuj adresy kolejnych routerów na trasie tam i z powrotem. Następnie wydaj polecenie `tracert` (Linux) lub `tracert` (Windows) z tym samym adresem docelowym i zanotuj adresy kolejnych routerów na trasie tam. Na podstawie uzyskanych danych postaraj się zbudować mapę środowiska sieciowego, w którym były przesyłane pakiety generowane przez powyższe polecenia.

ping -r 9 -n 1 wp.pl (Windows)

213.135.44.133 ->

195.149.232.32 ->

83.238.248.18 ->

83.238.6.9 ->

10.3.101.110 ->

212.77.98.9 ->

212.77.98.9 ->

195.149.232.48 ->

213.135.44.136

tracert -l -n wp.pl wp.pl (Linux)

1 213.135.45.254 0.146 ms 0.122 ms 0.118 ms

2 213.135.44.136 0.307 ms \* \*

3 195.149.232.59 2.229 ms

4 83.238.248.19 1.119 ms 1.130 ms 1.129 ms

5 87.204.224.71 1.181 ms 1.182 ms 1.186 ms

6 83.238.6.13 1.483 ms 1.412 ms 1.405 ms

7 212.77.98.9 1.050 ms 1.335 ms 1.186 ms

□-|.45.21 (1 sieć) .45.254|-⊗-|.44.133 (2 sieć) .44.136|-⊗-|.232.32 (3 sieć) .232.59|-⊗-|.248.18  
(4 sieć)

tracert -> 83.238.248.19

|

4⊗

↓

Chyba nie został wpisany w pole opcji -> ?

(5 sieć)

87.204.224.71

|

5⊗

↓

83.238.6.9

(6 sieć)

83.238.6.13

|

6⊗

↓

ping -r 9 pokazał adres nie z 7 sieci -> ?

(7 sieć)

212.77.98.9

|

□

```
tracert -n fedora.pl
```

```
tracert to fedora.pl (172.67.132.139), 30 hops max, 60 byte packets
```

```
1 213.135.45.254 0.090 ms 0.062 ms 0.061 ms
2 213.135.44.136 0.207 ms 0.190 ms 0.423 ms
3 195.149.232.128 2.106 ms 2.115 ms 2.085 ms
4 172.67.132.139 1.397 ms 1.357 ms 1.359 ms
```

```
ping -r 9 -n 1 fedora.pl
```

```
Pinging fedora.pl [172.67.132.139] with 32 bytes of data:
```

```
Reply from 172.67.132.139: bytes=32 time=3ms TTL=61
```

```
Route: 213.135.44.133 ->
```

```
195.149.232.32 ->
```

```
162.158.100.1 ->
```

```
172.67.132.139 ->
```

```
172.67.132.139 ->
```

```
162.158.100.1 -> dlaczego pakiet wypłynął przez ten sam interfejs trzeciego routera na rysunku,
```

```
przez który wypłynął na trasie tam?
```

```
213.135.44.130 -> z trzeciego routera na rysunku nastąpiło przejście do sieci 213.135.44.??/?
```

```
213.135.45.254
```

```
□-|.45.21 (1 sieć) .45.254|-⊗-|.44.133 (2 sieć) .44.136|-⊗-|.232.32 (3 sieć) .232.128|-⊗-|.100.1
```

```
(4 sieć)
```

```
172.67.132.139
```

```
|
```

```
□
```

## Aplikacja Wireshark

Służy ona do analizy ruchu sieciowego przechodzącego przez lokalny interfejs sieciowy. Działa w trybie graficznym i prezentuje ramki wysyłane i odbierane na wskazanym interfejsie.

Jeśli jest włączony tryb mieszany (promiscuous mode), to prezentowane są wszystkie ramki odbierane przez dany interfejs, a nie tylko ramki do niego zaadresowane i rozgłoszeniowe.

(Interfejs sieciowy może odbierać ramki zaadresowane do innego interfejsu, jeśli, na przykład, jest przyłączony do koncentratora, a nie do przełącznika).

Ramki są prezentowane w podziale na poszczególne warstwy modelu OSI.

Żeby przechwytywać tylko ramki spełniające określone kryteria, należy użyć filtra przechwytywania. Filtr taki wpisujemy w następujące pole:

[Przechwytuj -> Opcje -> Zakładka Wejście -> Filtr przechwytywania](#)

Należy przy tym pamiętać, żeby zaznaczyć właściwy interfejs na liście.

Filtr przechwytywania jest pewnym wyrażeniem logicznym składającym się z wyrażen elementarnych (primitives) połączonych operatorami and, or, not.

Przykłady wyrażen elementarnych:

host w.x.y.z <- adresem IP komp. źr. albo docel. jest w.x.y.z (ramka nie musi zawierać pakietu IP, np. może być ramką ARP)

src host w.x.y.z <- adresem IP komputera źródłowego jest w.x.y.z (ramka nie musi zawierać pakietu IP)

dst host w.x.y.z <- adresem IP komputera docelowego jest w.x.y.z (ramka nie musi zawierać pakietu IP)

ip host w.x.y.z <- adresem IP komp. źr. albo docel. jest w.x.y.z, a ramka musi zawierać pakiet IP

ip src host w.x.y.z <- adresem IP komp. źr. jest w.x.y.z, a ramka musi zawierać pakiet IP

ip dst host w.x.y.z <- adresem IP komp. docel. jest w.x.y.z, a ramka musi zawierać pakiet IP

arp albo ether proto \arp <- ramka (Ethernet) z komunikatem ARP (request albo reply)

ip albo ether proto \ip <- ramka (Ethernet) z pakietem IP

icmp albo ip proto \icmp <- pakiet IP z komunikatem ICMP

ip[2:3]=0xab01c2 <- ramka musi zawierać pakiet IP, a bajty 3, 4 i 5 są odpowiednio równe ab, 01 i c2

tshark - tekstowa wersja Wireshark (prezentacja ramek w trybie tekstowym w mało czytelny sposób)

tcpdump - analizator ruchu sieciowego działający w trybie tekstowym

man tcpdump - opis polecenia tcpdump w podręczniku man; składnia filtrów przechwytywania w Wireshark pochodzi z tcpdump