

Testowanie programów ping, Wireshark, Traceroute.

Sprawdźmy, ile jest węzłów na trasie do wybranego odległego geograficznie serwera. Posłużymy się programem ping. Liczbę węzłów od nas do serwera szacuje się poprzez wyszukanie takiej najmniejszej wartości TTL dla flagi `-t` przy programie ping, że pakiet dojdzie do celu. Czyli dla wartości ping [strona] `-t Q` powinien dojść, a dla ping strona `-t (Q-1)` pakiet już dość nie powinien. W takim przypadku liczę a określamy jako liczbę węzłów. Liczbę węzłów w drodze powrotnej liczymy na podstawie zwróconego ttl przez ping. Należy zastosować algorytm, że dla dużych wartości ttl liczbą węzłów będzie równa $256 - \text{ttl}$, a dla małych $64 - \text{ttl}$.

Przykładowy output z terminala.

```
sparrovsky@sparrovsky-vBox:~$ ping -c 4 rojadirecta.top -t 14
PING rojadirecta.top (104.27.139.106) 56(84) bytes of data.
64 bytes from 104.27.139.106 (104.27.139.106): icmp_seq=1 ttl=50 time=45.8 ms
64 bytes from 104.27.139.106 (104.27.139.106): icmp_seq=2 ttl=50 time=63.0 ms
64 bytes from 104.27.139.106 (104.27.139.106): icmp_seq=3 ttl=50 time=46.7 ms
64 bytes from 104.27.139.106 (104.27.139.106): icmp_seq=4 ttl=50 time=56.1 ms

--- rojadirecta.top ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 6ms
rtt min/avg/max/mdev = 45.805/52.908/63.023/7.089 ms
```

```
sparrovsky@sparrovsky-vBox:~$ ping -c 4 rojadirecta.top -t 13
PING rojadirecta.top (104.27.138.106) 56(84) bytes of data.
From 149.14.83.58 (149.14.83.58) icmp_seq=1 Time to live exceeded
From 149.14.83.58 (149.14.83.58) icmp_seq=2 Time to live exceeded
From 149.14.83.58 (149.14.83.58) icmp_seq=3 Time to live exceeded
From 149.14.83.58 (149.14.83.58) icmp_seq=4 Time to live exceeded
```

Staramy się podstawiać wartości do flagi `-t`. Mamy trochę możliwości jednak nie jest ich tak dużo. Wobec tego analizując kilka przypadków można wysnuć wniosek że dla strony rojadirecta.top oczekiwaną wartością węzłów jest 14. Korzystając ze wskazanego algorytmu liczba węzłów w drodze powrotnej to $64 - 50 = 14$.

Wielkość pakietu a czas propagacji.

Spróbujmy wysłać dokładnie 4 pakiety i zobaczymy statystyki z czasami propagacji.

```
sparrovsky@sparrovsky-vBox:~$ ping -c 4 unixmen.com
PING unixmen.com (104.24.98.177) 56(84) bytes of data.
64 bytes from 104.24.98.177 (104.24.98.177): icmp_seq=1 ttl=51 time=87.9 ms
64 bytes from 104.24.98.177 (104.24.98.177): icmp_seq=2 ttl=51 time=37.4 ms
64 bytes from 104.24.98.177 (104.24.98.177): icmp_seq=3 ttl=51 time=46.5 ms
64 bytes from 104.24.98.177 (104.24.98.177): icmp_seq=4 ttl=51 time=59.1 ms

--- unixmen.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 7ms
rtt min/avg/max/mdev = 37.420/57.730/87.892/19.045 ms
```

Teraz wyślemy 4 pakiety o zwiększonym rozmiarze do 128 bajtów.

```
sparrovsky@sparrovsky-vBox:~$ ping -c 4 -s 100 -c 4 unixmen.com
PING unixmen.com (104.24.99.177) 100(128) bytes of data.
108 bytes from 104.24.99.177 (104.24.99.177): icmp_seq=1 ttl=51 time=48.5 ms
108 bytes from 104.24.99.177 (104.24.99.177): icmp_seq=2 ttl=51 time=49.8 ms
108 bytes from 104.24.99.177 (104.24.99.177): icmp_seq=3 ttl=51 time=45.10 ms
108 bytes from 104.24.99.177 (104.24.99.177): icmp_seq=4 ttl=51 time=45.3 ms

--- unixmen.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 9ms
rtt min/avg/max/mdev = 45.327/47.394/49.773/1.816 ms
```

Wniosek jest prosty. Przy zwiększaniu rozmiaru pakietu czasy propagacji jak różnice pomiędzy wartościami statystycznie największymi i najmniejszymi co do czasu są większe. Jednak rosną one do momentu, gdy pakiet nie przekroczy około 1500 bajtów. Liczba ta bardzo często stanowi granicę MTU (ang. *maximum transmission unit*), która jest graniczą liczbą, po której następuje fragmentacja danych. Pakiety nieprzekraczające MTU są pakietami, które można przesłać bez konieczności fragmentacji. Fragmentacja spowodowana jest tym, że router nie jest w stanie przetworzyć danych o segmencie większym niż ma to zdefiniowane w swoich opcjach. Dlatego też taki pakiet musi zostać podzielony na kilka mniejszych. Po dotarciu do odbiorcy, te fragmenty są składane w całość w oparciu o numer identyfikacyjny pakietu

Dla pakietów przekraczających MTU można zauważyć spadek czasu propagacji co do trendu jaki wyznaczały pakiety o rozmiarze mniejszym niż zadane MTU.

Fragmentacja pakietów jest również wykonywana w celu przyspieszenia przepływu danych w sieci. Pakiety o dużych rozmiarach zwiększyłyby zatory w wolnej sieci (niska przepustowość), więc router bramy do tej sieci ustawi niższą wartość MTU, aby uniknąć wejścia dużych pakietów do sieci

Wielkość pakietu a ilość węzłów na trasie.

Wyślemy teraz różne rozmiary pakietów i sprawdzimy, czy ma to wpływ na długość trasy (liczbę węzłów). Widzimy, że przy znacznym zwiększeniu rozmiaru pakietu ttl, który jednoznacznie pozwala nam określić drogę nie zmienia się. Ostatecznie stwierdzamy, że wielkość pakietu nie ma wpływu na długość trasy.

```
sparrovsky@sparrovsky-vBox:~$ ping -s 56 -c 4 www.uq.edu.au -t 31
PING www.uq.edu.au (130.102.184.3) 56(84) bytes of data.
64 bytes from www.create-change.uq.edu.au (130.102.184.3): icmp_seq=1 ttl=219 time=358 ms
64 bytes from www.create-change.uq.edu.au (130.102.184.3): icmp_seq=2 ttl=219 time=347 ms
64 bytes from www.create-change.uq.edu.au (130.102.184.3): icmp_seq=3 ttl=219 time=369 ms
64 bytes from www.create-change.uq.edu.au (130.102.184.3): icmp_seq=4 ttl=219 time=357 ms

--- www.uq.edu.au ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 384ms
rtt min/avg/max/mdev = 347.341/357.960/369.465/7.886 ms
```

```
sparrovsky@sparrovsky-vBox:~$ ping -s 1016 -c 4 www.uq.edu.au -t 31
PING www.uq.edu.au (130.102.184.3) 1016(1044) bytes of data.
1024 bytes from www.create-change.uq.edu.au (130.102.184.3): icmp_seq=1 ttl=219 time=357 ms
1024 bytes from www.create-change.uq.edu.au (130.102.184.3): icmp_seq=2 ttl=219 time=406 ms
1024 bytes from www.create-change.uq.edu.au (130.102.184.3): icmp_seq=3 ttl=219 time=354 ms
1024 bytes from www.create-change.uq.edu.au (130.102.184.3): icmp_seq=4 ttl=219 time=377 ms

--- www.uq.edu.au ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 4ms
rtt min/avg/max/mdev = 354.036/373.552/405.911/20.614 ms
```

Średnica internetu.

Średnicę internetu wyznaczymy szukając strony, która przy pomocy programu ping będzie charakteryzowana statystycznie największą ilością węzłów. Możemy manipulować wyborami jednak tutaj przytoczę kilka odległych geograficznie przykładów i sprawdzę liczbę węzłów.

University of Kyoto (Japonia) 25 węzłów

```
sparrovsky@sparrovsky-vBox:~$ ping www.kyoto-u.ac.jp -t 24
PING web-front.pr.kyoto-u.ac.jp (133.3.250.141) 56(84) bytes of data.
□
```

```
sparrovsky@sparrovsky-vBox:~$ ping www.kyoto-u.ac.jp -t 25
PING web-front.pr.kyoto-u.ac.jp (133.3.250.141) 56(84) bytes of data.
64 bytes from web-front.pr.kyoto-u.ac.jp (133.3.250.141): icmp_seq=1 ttl=32 time=300 ms
64 bytes from web-front.pr.kyoto-u.ac.jp (133.3.250.141): icmp_seq=2 ttl=32 time=313 ms
64 bytes from web-front.pr.kyoto-u.ac.jp (133.3.250.141): icmp_seq=3 ttl=32 time=300 ms
64 bytes from web-front.pr.kyoto-u.ac.jp (133.3.250.141): icmp_seq=4 ttl=32 time=300 ms
64 bytes from web-front.pr.kyoto-u.ac.jp (133.3.250.141): icmp_seq=5 ttl=32 time=314 ms
```

University of Queensland (Australia) 31 węzłów

```
sparrovsky@sparrovsky-vBox:~$ ping www.uq.edu.au -t 31
PING www.uq.edu.au (130.102.184.3) 56(84) bytes of data.
64 bytes from www.create-change.uq.edu.au (130.102.184.3): icmp_seq=1 ttl=219 time=355 ms
64 bytes from www.create-change.uq.edu.au (130.102.184.3): icmp_seq=2 ttl=219 time=361 ms
64 bytes from www.create-change.uq.edu.au (130.102.184.3): icmp_seq=3 ttl=219 time=350 ms
64 bytes from www.create-change.uq.edu.au (130.102.184.3): icmp_seq=4 ttl=219 time=365 ms
```

```
sparrovsky@sparrovsky-vBox:~$ ping www.uq.edu.au -t 30
PING www.uq.edu.au (130.102.184.3) 56(84) bytes of data.
□
```

Aerotechservices University (Kanada) 28 węzłów

```
sparrovsky@sparrovsky-vBox:~$ ping www.aerotechservices.com -t 28
PING aerotechservices.com (23.235.222.238) 56(84) bytes of data.
64 bytes from ded3500.inmotionhosting.com (23.235.222.238): icmp_seq=1 ttl=38 time=186 ms
64 bytes from ded3500.inmotionhosting.com (23.235.222.238): icmp_seq=2 ttl=38 time=191 ms
64 bytes from ded3500.inmotionhosting.com (23.235.222.238): icmp_seq=3 ttl=38 time=190 ms
64 bytes from ded3500.inmotionhosting.com (23.235.222.238): icmp_seq=4 ttl=38 time=197 ms
^C
--- aerotechservices.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 37ms
rtt min/avg/max/mdev = 186.397/190.983/196.629/3.739 ms
```

```
sparrovsky@sparrovsky-vBox:~$ ping www.aerotechservices.com -t 27
PING aerotechservices.com (23.235.222.238) 56(84) bytes of data.
From csw1.lax1.inmotionhosting.com (198.46.92.26) icmp_seq=1 Time to live exceeded
From csw1.lax1.inmotionhosting.com (198.46.92.26) icmp_seq=2 Time to live exceeded
```

Obserwujemy, że bardzo prawdopodobną liczbą, która jest sugerowaną średnią internetu jest liczba bliska liczby 30. Częstość wynikiem występującym w charakteryzacji były liczby do 28 węzłów jednak zdarzały się i te bliskie liczby 30 albo nawet przekraczające tę liczbę. Dlatego ostatecznym wnioskiem jest, że potencjalną średnią internetu jest liczba większa, lecz dość bliska 30 liczona w jednostkach liczby węzłów na trasie propagacji. W moich próbach wartość ta wynosi 31.

Trasy przez sieci wirtualne

Sieć vlan to wydzielona logicznie sieć urządzeń w ramach innej, większej sieci fizycznej. Urządzenia tworzące sieć vlan, niezależnie od swojej fizycznej lokalizacji (przełącznika, do którego są podłączone), mogą się swobodnie komunikować ze sobą, a jednocześnie są odseparowane od innych sieci vlan, co oznacza, że na poziomie przełącznika nie ma żadnej możliwości skomunikowania urządzeń należących do dwóch różnych sieci vlan.

Korzyści płynące z zastosowania sieci vlan to możliwość ograniczenia ruchu rozgłoszeniowego, bo rozgłaszane ramki trafiają tylko do komputerów w obrębie danej sieci vlan. Stosując sieci vlan administrator jest w stanie wprowadzać zmiany w sieci programowo, a nie sprzętowo, co usprawnia

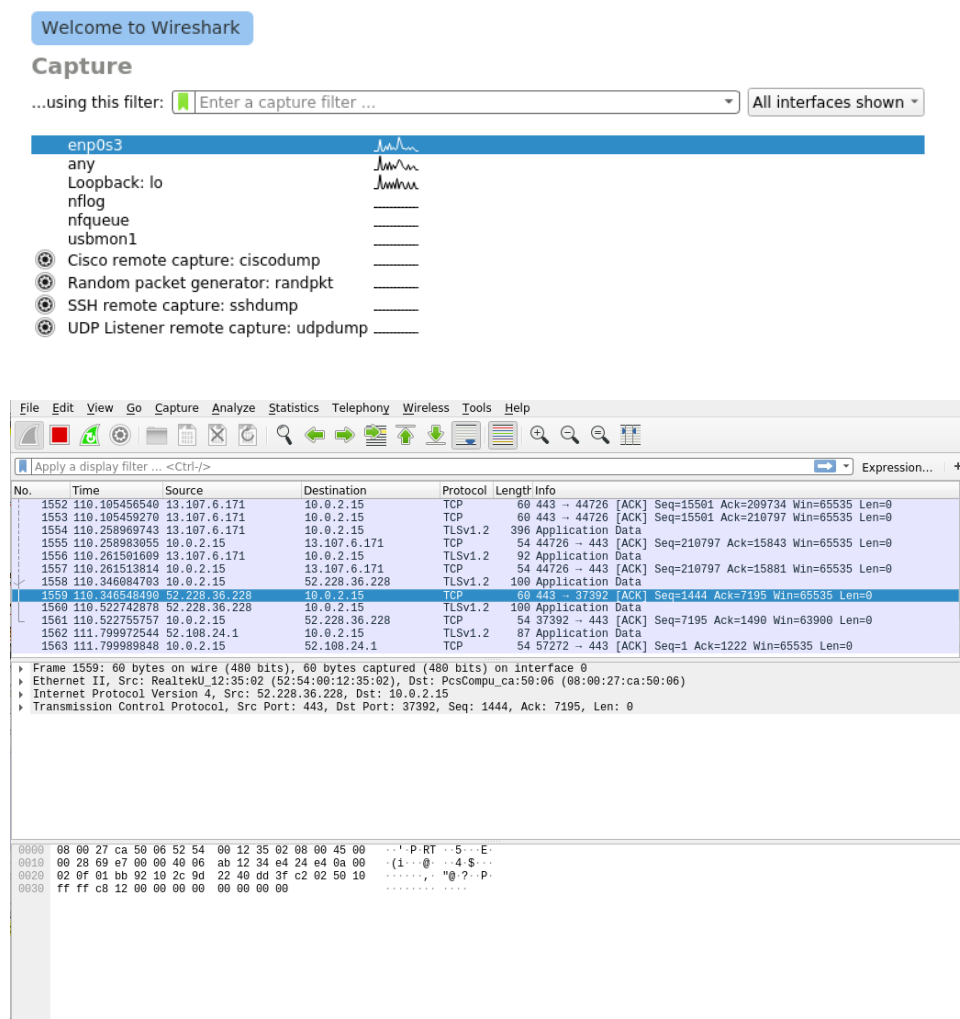
budowanie struktur sieci w danej organizacji oraz zwiększa bezpieczeństwo sieci poprzez ustalenie, które węzły sieci mogą porozumiewać się ze sobą.

Można wyszukać trasy przebiegające przez sieci wirtualne za pomocą Traceroute'a jednak jest to problematyczne. Wówczas sumaryczną liczbą węzłów będzie liczba węzłów na trasie do jak i tych w samej sieci wirtualnej.

Wireshark

Wireshark służy do podsłuchiwania sieci w czasie rzeczywistym oraz analizy ruchu sieciowego, który przechwycony został wcześniej na innym urządzeniu oraz oceny jej prawidłowego funkcjonowania. Wykorzystywany jest głównie do diagnostyki niezawodności i wydajności sieci. Jednak dzięki wielu dodatkom potrafi również zdekodować wiele protokołów komunikacyjnych. Szczegółowy podgląd pakietów pozwala również na analizę wysyłanych danych np. Przechwytywanie nieszyfrowanych haseł lub tych które jeszcze nie zostały poddane szyfrowaniu.

Wireshark'a otwieramy z prawami administratora i wybieramy interfejs sieciowy, z którego chcemy korzystać. Nasłuchiwanie włącza się automatycznie. Po chwili otrzymujemy dostęp do wszystkich przechwyconych pakietów oraz możemy zobaczyć ich specyfikację oraz reprezentację binarną.



Welcome to Wireshark

Capture

...using this filter: All interfaces shown

- enp0s3
- any
- Loopback: lo
- nflog
- nfqueue
- usbmon1
- Cisco remote capture: ciscodump
- Random packet generator: randpkt
- SSH remote capture: sshdump
- UDP Listener remote capture: udpdump

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/> Expression...

| No. | Time | Source | Destination | Protocol | Length | Info |
|------|---------------|---------------|---------------|----------|--------|--------------------------------------------------------|
| 1552 | 110.105456540 | 13.107.6.171 | 10.0.2.15 | TCP | 60 | 443 → 44726 [ACK] Seq=15501 Ack=209734 Win=65535 Len=0 |
| 1553 | 110.105459270 | 13.107.6.171 | 10.0.2.15 | TCP | 60 | 443 → 44726 [ACK] Seq=15501 Ack=210797 Win=65535 Len=0 |
| 1554 | 110.258969743 | 13.107.6.171 | 10.0.2.15 | TLsv1.2 | 396 | Application Data |
| 1555 | 110.258983855 | 10.0.2.15 | 13.107.6.171 | TCP | 54 | 44726 → 443 [ACK] Seq=210797 Ack=15843 Win=65535 Len=0 |
| 1556 | 110.261501089 | 13.107.6.171 | 10.0.2.15 | TLsv1.2 | 92 | Application Data |
| 1557 | 110.261513814 | 10.0.2.15 | 13.107.6.171 | TCP | 54 | 44726 → 443 [ACK] Seq=210797 Ack=15881 Win=65535 Len=0 |
| 1558 | 110.346084703 | 10.0.2.15 | 52.228.36.228 | TLsv1.2 | 100 | Application Data |
| 1559 | 110.346548490 | 52.228.36.228 | 10.0.2.15 | TCP | 60 | 443 → 37392 [ACK] Seq=1444 Ack=7195 Win=65535 Len=0 |
| 1560 | 110.522742878 | 52.228.36.228 | 10.0.2.15 | TLsv1.2 | 100 | Application Data |
| 1561 | 110.522755757 | 10.0.2.15 | 52.228.36.228 | TCP | 54 | 37392 → 443 [ACK] Seq=7195 Ack=1490 Win=63900 Len=0 |
| 1562 | 111.799972544 | 52.108.24.1 | 10.0.2.15 | TLsv1.2 | 87 | Application Data |
| 1563 | 111.799989848 | 10.0.2.15 | 52.108.24.1 | TCP | 54 | 57272 → 443 [ACK] Seq=1 Ack=1222 Win=65535 Len=0 |

Frame 1559: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0

Ethernet II, Src: RealtekU12:35:02 (52:54:00:12:35:02), Dst: PcsCompu_ca:50:06 (08:00:27:ca:50:06)

Internet Protocol Version 4, Src: 52.228.36.228, Dst: 10.0.2.15

Transmission Control Protocol, Src Port: 443, Dst Port: 37392, Seq: 1444, Ack: 7195, Len: 0

0000 08 00 27 ca 50 06 52 54 00 12 35 02 08 00 45 00 ... P RT ... 5 ... E

0010 00 28 69 e7 00 00 40 06 ab 12 34 e4 24 e4 0a 00 ... (1...0... 4 \$...

0020 02 0f 01 bb 19 2c 9d 22 40 dd 3f c2 02 50 10, "0"?..P

0030 ff ff c8 12 00 00 00 00 00 00 00 00 00 00 00

Każdy pakiet opisany jest adresem źródłowym i docelowym. W zależności od protokołu może to być adres IP lub mac. Dla sprecyzowania jaki rodzaj ruchu chcemy nadzorować warto przefiltrować pakiety np. Po nazwie protokołu, bo domyślnie program przechwytuje i wyświetla wszystko co przejdzie przez kartę sieciową. Sposób wyświetlania danych można zmienić poprzez zarządzanie filtrami wyświetlania oraz przechwytywania. ten pierwszy przechwytuje wszystko i wyświetla tylko wybrane dane a ten drugi przechwytuje jedynie szczegółowo wybrane przez nas dane.

| No. | Time | Source | Destination | Protocol | Length | Info |
|-----|--------------|---------------|---------------|----------|--------|--------------------------------------------------------------------|
| 42 | 11.037572945 | 10.0.2.15 | 31.11.202.254 | DNS | 138 | Standard query 0xc263 A odwebpl.trafficmanager.net. |
| 48 | 11.040705940 | 10.0.2.15 | 31.11.202.254 | DNS | 138 | Standard query 0x773c AAAA odwebpl.trafficmanager.net. |
| 49 | 11.067484259 | 31.11.202.254 | 10.0.2.15 | DNS | 168 | Standard query response 0xc263 A odwebpl.trafficmanager.net. |
| 50 | 11.069635499 | 31.11.202.254 | 10.0.2.15 | DNS | 212 | Standard query response 0x773c AAAA odwebpl.trafficmanager.net. |
| 61 | 18.112718997 | 10.0.2.15 | 31.11.202.254 | DNS | 123 | Standard query 0x8ec9 A word-edit.wac.trafficmanager.net. |
| 68 | 18.143237117 | 31.11.202.254 | 10.0.2.15 | DNS | 223 | Standard query response 0x8ec9 A word-edit.wac.trafficmanager.net. |
| 95 | 25.224310561 | 10.0.2.15 | 31.11.202.254 | DNS | 87 | Standard query 0x4938 A www.facebook.com OPT |
| 96 | 25.225066179 | 10.0.2.15 | 31.11.202.254 | DNS | 87 | Standard query 0x00ae AAAA www.facebook.com OPT |
| 97 | 25.250725759 | 31.11.202.254 | 10.0.2.15 | DNS | 132 | Standard query response 0x4938 A www.facebook.com CNAME |
| 99 | 25.256660382 | 31.11.202.254 | 10.0.2.15 | DNS | 144 | Standard query response 0x00ae AAAA www.facebook.com CNAME |

| | |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|
| ▶ Frame 42: 138 bytes on wire (1104 bits), 138 bytes captured (1104 bits) on interface 0 ▶ Ethernet II, Src: PcsCompu.ca:50:06 (08:00:27:ca:50:06), Dst: RealtekU.12:35:02 (52:54:00:12:35:02) ▶ Internet Protocol Version 4, Src: 10.0.2.15, Dst: 31.11.202.254 ▶ User Datagram Protocol, Src Port: 36111, Dst Port: 53 ▶ Domain Name System (query) | |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|

| | | | | |
|------|-------------------------|-------------------------|----------|----------|
| 0000 | 52 54 00 12 35 02 08 00 | 27 ca 50 06 08 00 45 00 | RT-5... | P...E... |
| 0010 | 00 7c 5a bc 40 00 40 11 | e9 9c 0a 00 02 0f 1f 0b | 12 00 | |
| 0020 | ca fe 8d 0f 00 35 00 68 | f6 91 c2 63 01 00 00 01 | ...5 h | ...c.... |
| 0030 | 00 00 00 00 00 01 07 6f | 64 77 65 62 70 6c 0e 74 |o | dwebpl:t |
| 0040 | 72 61 66 66 69 63 6d 61 | 6e 61 67 65 72 03 6e 65 | rafficma | nager-ne |
| 0050 | 74 06 6c 2d 30 30 30 34 | 09 64 63 2d 6d 73 65 64 | t-l-0004 | dc-msed |
| 0060 | 67 65 03 6e 65 74 06 6c | 2d 30 30 30 34 08 6c 2d | ge.net:l | -0004-l- |
| 0070 | 6d 73 65 64 67 65 03 6e | 65 74 00 00 01 00 01 00 | msedge:n | et..... |
| 0080 | 00 29 02 00 00 00 00 00 | 00 00 |)..... | .. |

Program Wireshark jest bardzo przydatnym narzędziem, które dostarcza nam bardzo dużo szczegółowych informacji np. na temat danego protokołu i jego charakterystycznych właściwości jak wielkość nagłówka protokołu, długość całkowitą pakietu, czas życia pakietu oraz informacje czy pakiet został poddany fragmentacji.

| | |
|-----------------------------------------------------------------|-------------------------------|
| Internet Protocol Version 4, Src: 10.0.0.104, Dst: 78.46.80.114 | |
| 0100 | = Version: 4 |
| 0101 | = Header Length: 20 bytes (5) |
| ▶ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT) | |
| Total Length: 452 | |
| Identification: 0x0cd0 (3280) | |
| ▶ Flags: 0x02 (Don't Fragment) | |
| Fragment offset: 0 | |
| Time to live: 128 | |
| Protocol: TCP (6) | |
| Header checksum: 0x0000 [validation disabled] | |
| [Header checksum status: Unverified] | |
| Source: 10.0.0.104 | |
| Destination: 78.46.80.114 | |
| [Source GeoIP: Unknown] | |
| [Destination GeoIP: Unknown] | |

Traceroute

Program służący do badania trasy pakietów oraz diagnozowania potencjalnych błędów na trasie przesyłu danych w sieciach IP. Jest jednym z przydatnych narzędzi do badania sieci, którego działanie pomaga po części w rozwiązywaniu bardziej złożonych problemów oraz analizowaniu ruchu w dużych sieciach, gdzie w celu dotarcia do danego punktu docelowego można użyć kilku tras.

```
wojciech@wojciech-Lenovo-Legion-Y530-15ICH:~$ traceroute youtube.com
traceroute to youtube.com (216.58.209.14), 30 hops max, 60 byte packets
 1 _gateway (192.168.0.1) 3.220 ms 4.514 ms 4.477 ms
 2 10.238.128.1 (10.238.128.1) 17.085 ms 17.057 ms 17.020 ms
 3 172.17.157.5 (172.17.157.5) 16.990 ms 16.957 ms 16.925 ms
 4 172.17.28.186 (172.17.28.186) 33.253 ms 29.716 ms 29.667 ms
 5 172.17.28.182 (172.17.28.182) 33.149 ms 172.17.28.186 (172.17.28.186) 33.078 ms 172.17.28.182 (172.17.28.182) 29.154 ms
 6 031011202146.p2p.business.static.vectranet.pl (31.11.202.146) 33.028 ms 26.936 ms 28.881 ms
 7 108.170.250.209 (108.170.250.209) 28.825 ms 108.170.250.193 (108.170.250.193) 22.437 ms 108.170.250.209 (108.170.250.209) 24.739 ms
 8 172.253.68.29 (172.253.68.29) 28.295 ms 172.253.68.31 (172.253.68.31) 26.677 ms 172.253.68.29 (172.253.68.29) 28.627 ms
 9 sofois12-in-f14.1e100.net (216.58.209.14) 25.761 ms 20.761 ms 21.887 ms

wojciech@wojciech-Lenovo-Legion-Y530-15ICH:~$ traceroute onet.pl
traceroute to onet.pl (213.180.141.140), 30 hops max, 60 byte packets
 1 _gateway (192.168.0.1) 2.272 ms 5.978 ms 8.108 ms
 2 10.238.128.1 (10.238.128.1) 17.232 ms 34.645 ms 53.319 ms
 3 172.17.157.5 (172.17.157.5) 55.561 ms 55.571 ms 55.567 ms
 4 172.17.28.190 (172.17.28.190) 56.636 ms 64.871 ms 65.894 ms
 5 172.17.28.194 (172.17.28.194) 64.025 ms 172.17.28.190 (172.17.28.190) 68.040 ms 76.999 ms
 6 onet.tpix.pl (195.149.232.107) 77.703 ms 27.739 ms 29.051 ms
 7 sdr1.m10r2.z.j.ruc-br1.link1.net.onet.pl (213.180.152.129) 30.852 ms sdr1.cdn1r1.z.j.ruc-br1.link3.net.onet.pl (213.180.151.25) 56.172 ms sdr1.m10r2.z.j.ruc-br1.link3.net.onet.pl (213.180.152.133) 32.698 ms
 8 * * *
 9 * * *
10 * * *
11 * * *
12 * * *
13 * * *
```

Zasada działania

Działanie narzędzia traceroute'a oparte jest na protokołach komunikacyjnych UDP i ICMP. Pierwszy pakiet wysyłany jest z polem TTL (Time To Live) ustawionym na 1. Następnie podczas przechodzenia przez kolejne routery na trasie wartość parametru TTL jest każdorazowo zmniejszana o 1. Natomiast w momencie, gdy pole TTL ma wartość 0 – pakiet jest odrzucany przez router, oraz router ten wysła komunikat ICMP „Time Exceeded”. Należy mieć też na uwadze, że gdy router został skonfigurowany tak aby nie zmniejszał parametru TTL nie będzie on widoczny w wynikach polecenia. Za pomocą narzędzia traceroute można określić, gdzie w sieci zatrzymał się pakiet. Brak sygnału oznaczany jest w terminalu znakiem gwiazdki.

Podsumowanie działania programów

Program ping jest bardzo prostym programem o ograniczonych możliwościach działania. Przydaje się by sprawdzić, czy podany adres IP jest osiągalny oraz możemy na jego podstawie zdobyć podstawowe informacje o drodze i czasie propagacji pakietu. Program traceroute jest wygodnym narzędziem do obserwowania konkretnej trasy (przez konkretne routery) przechodzenia pakietu i ewentualnej jej analizy pod względem poprawności. Wireshark to już bardziej zaawansowane oprogramowanie, którego pożytek można zauważyć na wielu płaszczyznach. Dostarcza wielu informacji o pakietach i przydaje się do szczegółowej analizy protokołów sieciowych jak i sprawdzania bezpieczeństwa danej sieci.