Autor: Kamil Król

Numer indeksu: 244949

Sprawozdanie

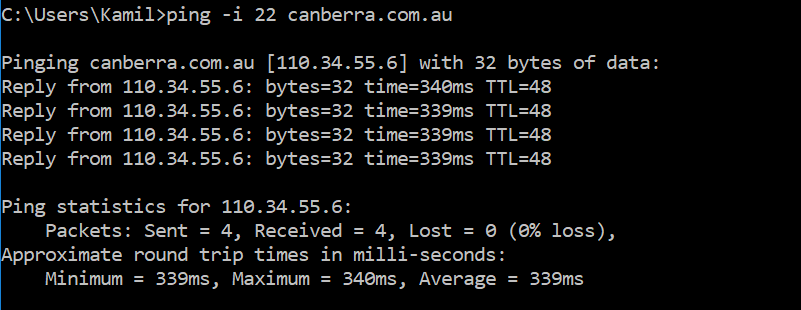
## Ping

* 1. **Opis programu**

**Ping** – jest to program służący do testowania połączeń sieciowych. Jego najczęstszym zastosowaniem jest sprawdzenie czy istnieje połączenie pomiędzy dwoma komputerami. Poza tym pozwala również na sprawdzenie opóźnień podczas wysyłania pakietów czy monitorowania ilości zagubionych pakietów. Podstawą jego działania jest użycie protokołu ICMP (Internet Control Message Protocol). Działanie programu polega na wysłaniu pakietu ‘ICMP echo request’ i oczekiwaniu na odpowiedź ‘ICMP echo reply’. Większość obecnych komputerów wysyła takie odpowiedzi. Jednak część serwisów zdecydowała się na wyłączenie tej opcji ze względów bezpieczeństwa, ponieważ potwierdzenie obecności hosta pod pewnym adresem IP może być potraktowane jako zidentyfikowanie go potencjalny cel ataku. Stąd wniosek, że są takie hosty, które pomimo bycia podłączonymi do sieci nie odpowiedzą na ‘echo request’.

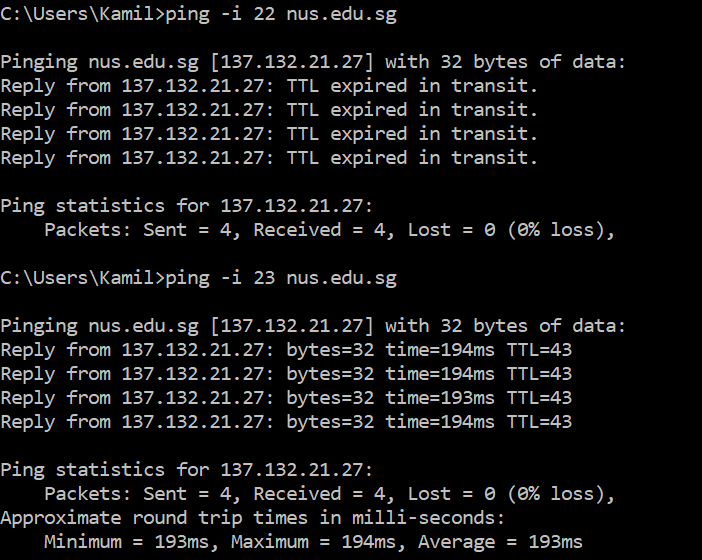
* 1. **Przykładowe użycie**

Pingowanie australijskiego serwera.



* 1. **Liczenie ilości węzłów do danego serwera**

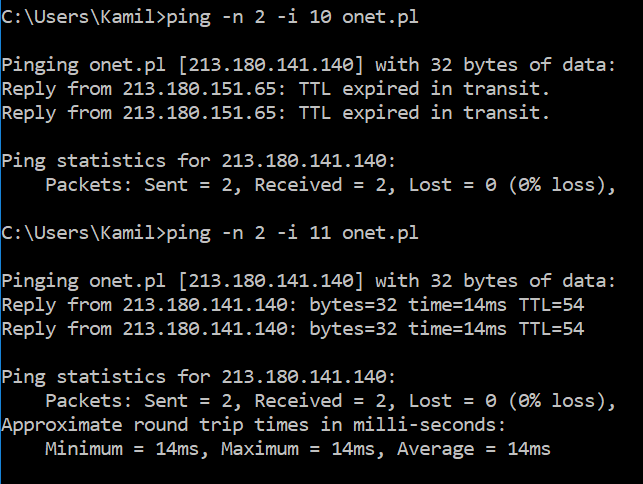
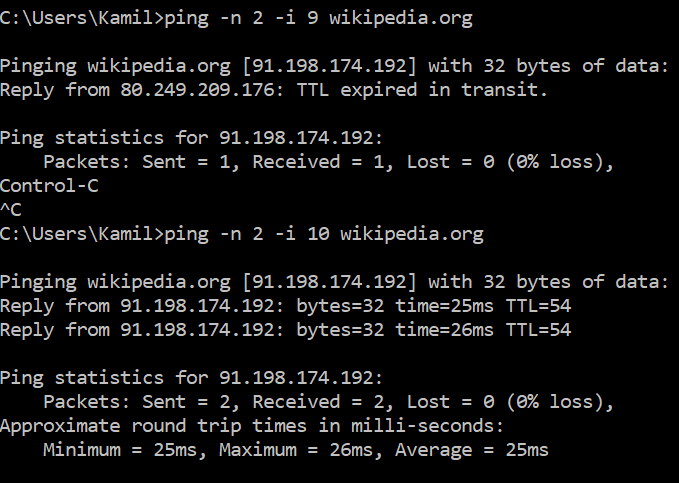
Używając programu ping można sprawdzić ile węzłów znajduje się na trasie do danego serwera. Na każdym węźle wartość TTL jest zmniejszana o 1, więc można ręcznie sprawdzić graniczną wartość TTL dla którego pakiet dotrze do celu. Za pomocą opcji -i można ustalić wartość TTL wysyłanego pakietu. Swoje doświadczenia przeprowadziłem będąc we Wrocławiu.



Postanowiłem sprawdzić ilość węzłów na trasie do serwera nus.edu.sg (National University of Singapore). Wartość TTL równa 22 była za mała aby pakiet dotarł do celu. Jednak wartość 23 okazała się w porządku. Stąd wniosek ze ilość węzłów na trasie do serwera nus.edu.sg wynosi 23.

* 1. **Trasy tam i z powrotem**

Teraz spróbuję ustalić czy trasa do serwera jest taka sama jak trasa powrotna. Najpierw zdecydowałem się sprawdzić ilość węzłów na trasie tam i z powrotem, ponieważ jeśli stwierdzę różnicę miedzy ich ilością to będzie to oznaczać, że trasy te są różne. Do testów postanowiłem użyć dwóch serwerów: onet.pl (Polska) oraz wikipedia.org (USA).

Kiedy serwer wysyła odpowiedź TTL przyjmuje wartość domyślną dla danego serwera. Są to wartości 32/64/128/255. W moim doświadczeniu pakiet zwrotny miał wartość TTL równą 54 (w obu przypadkach), więc stwierdzam, że początkowa wartość TTL była równa 64. Teraz ilość węzłów na trasie powrotnej mogę obliczyć jako różnicę tych wartości

|  |  |  |
| --- | --- | --- |
| Adres hosta | Ilość węzłów na trasie do hosta | Ilość węzłów na drodze powrotnej |
| onet.pl | 11 | 10 |
| wikipedia.org | 10 | 10 |

Wnioskiem z doświadczenia jest, że trasy powrotne nie są takie same, ponieważ ilość węzłów na trasie do serwera onet.pl była inna niż na trasie powrotnej. W przypadku serwera wikipedia.org widząc taką samą ilość węzłów na trasie tam i z powrotem nie jesteśmy w stanie jednoznacznie powiedzieć czy trasa była taka sama, wiemy jedynie, że jej długość była taka sama.

* 1. **Ilość węzłów, czas odpowiedzi, a odległość geograficzna**

Teraz sprawdzę zależność ilości węzłów i czasów odpowiedzi od odległości geograficznej. W tym celu będę sprawdzać ilość węzłów na trasach do serwerów z różnych regionów świata.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| lokalizacja | Adres hosta | Czas(min|max|avg)\* | Ilość węzłów (do hosta) | Ilość węzłów (od hosta)[TTL] |
| Wrocław (Polska) | s.student.pwr.edu.pl | 1|2|1 | 7 | 6[58] |
| Wrocław (Polska) | jsos.pwr.edu.pl | 1|2|1 | 11 | 8[56] |
| Polska | wykop.pl | 52|65|53 | 12 | 11[53] |
| Kraków  (Polska) | krakow.pl | 13|15|13 | 10 | 8[56] |
| Polska | wp.pl | 12|13|12 | 10 | 8[56] |
| Warszawa (Polska) | pw.edu.pl | 10|11|10 | 11 | 9[55] |
| Monachium (Niemcy) | tum.de | 30|32|30 | 13 | 12[243] |
| Mińsk  (Białoruś) | bsu.by | 53|54|55 | 16 | 15[113] |
| Amsterdam (Holandia) | uva.nl | 27|36|28 | 12 | 12[52] |
| Moskwa  (Rosja) | msu.ru | 44|46|45 | 15 | 13[51] |
| Palermo  (Włochy) | palermo.repubblica.it | 23|24|23 | 14 | 9[55] |
| Scottsdale (USA) | franklinepb.com | 164|166|165 | 14 | 12[116] |
| Waszyngton  (USA) | ryanair.com | 110|117|112 | 11 | 11[53] |
| Tokio (Japonia) | tokyotimes.com | 266|266|266 | 15 | 15[49] |
| Chiny | 114.114.115.119 | 125|127|126 | 22 | -[?] |
| Zimbabwe | 77.246.56.247 | 284|285|284 | 13 | 12[243] |
| Demokratyczna Republika Kongo | 31.209.128.129 | 429|444|437 | 14 | 14[50] |
| Argentyna | 2.18.56.0 | 271|274|272 | 14 | 14[50] |
| Palau | 103.30.248.18 | 278|279|278 | 14 | 15[113] |
| Sydney (Australia) | victoria.ac.nz | 317|318|317 | 20 | 16[48] |
| Canberra (Australia) | canberra.com.au | 339|346|339 | 21 | 16[48] |
| Sydney  (Australia) | sydney.edu.au | 314|315|314 | 26 | 25[230] |

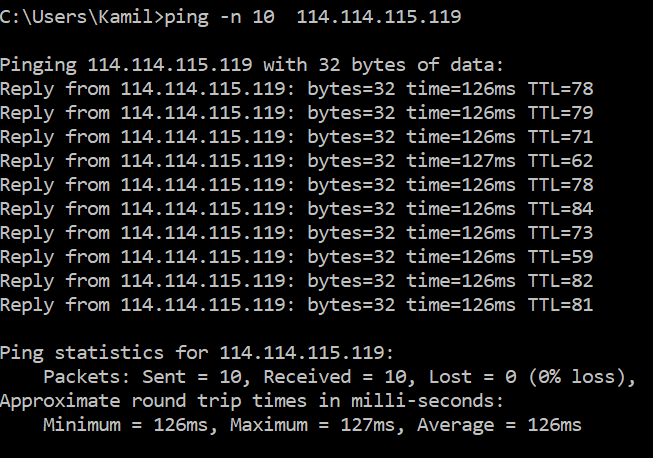
\*wykonywane jest 10 zapytań (opcja -n 10)

Wnioskiem z powyższej tabeli jest to, że ilość węzłów nie zależy w znacznym stopniu od odległości geograficznej. Różnica była widoczna dopiero przy połączeniach z Sydney. Połączenie z Argentyną nie różniło się w ilości pakietów od połączenia z Monachium mimo bardzo znacznej różnicy w odległościach. Widać jednak było nieznacznie zmniejszoną ilość węzłów na trasie do wrocławskich serwerów – tych które są blisko. Połączenie do jednego z nich wymagało przejścia jedynie przez 7 routerów, co może być związane z faktem wykonywania tego doświadczenia na łączu internetowym w akademiku PWr.

Innym ważnym wnioskiem jest zależność czasu odpowiedzi od odległości geograficznej. Czas odpowiedzi był wyraźnie krótszy dla serwerów blisko, i wyraźnie dłuższy dla tych odległych. Jednak występowały też pewne odchylenia – polski serwer wykop.pl odpowiadał w czasie dłuższym niż ten w Palermo. Inny przykład to serwer na małej wyspie na Pacyfiku (Palau), który odpowiadał w podobnym czasie jak ten w Japonii.

* 1. **Chińskie serwery**

Na szczególną uwagę zasługuje zachowanie jednego z chińskich serwerów.



W każdej wiadomości zwrotnej serwer miał inną wartość TTL, co prawdopodobnie jest skutkiem umyślnego zaburzania tej wartości.

* 1. **Wielkość pakietów, a czas przesyłania**

W celu sprawdzenia zależności czasu przesyłania pakietu od wielkości pakietów wysyłałem pakiety o różnych rozmiarach do różnych serwerów. Jednym z serwerów była wikipedia.org (Kalifornia, USA), drugim uni.wroc.pl (Wrocław, Polska),a trzecim canberra.com.au (Sydney, Australia)

Tabela dla wikipedia.org.

|  |  |  |  |
| --- | --- | --- | --- |
| Wielość pakietu w bajtach | Czas(min|max|avg)\* | Trasa do hosta | Trasa od hosta [TTL] |
| 32 | 25|26|25 | 10 | 10[54] |
| 500 | 25|27|25 | 10 | 10[54] |
| 1000 | 26|26|26 | 10 | 10[54] |
| 5000 | 27|28|27 | 10 | 10[54] |
| 8000 | 27|28|27 | 10 | 10[54] |
| 16000 | 29|30|29 | 10 | 10[54] |
| 32000 | 32|33|32 | 10 | 10[54] |
| 48000 | 34|36|34 | 10 | 10[54] |
| 65500 | 37|40|37 | 10 | 10[54] |

\*wykonywane jest 10 zapytań (opcja -n 10)

Różnica między pierwszym, a drugim jest głównie w odległości geograficznej, jeśli chodzi o odległość w ilości węzłów to są one podobne. Trzeci host jest zdecydowanie dalej niż dwa pozostałe zarówno pod względem odległości geograficznej jak i ilości węzłów.

Tabela dla uni.wroc.pl.

|  |  |  |  |
| --- | --- | --- | --- |
| Wielość pakietu w bajtach | Czas(min|max|avg)\* | Trasa do hosta | Trasa od hosta [TTL] |
| 32 | 1|2|1 | 11 | 9[55] |
| 500 | 1|1|1 | 11 | 9[55] |
| 1000 | 1|3|1 | 11 | 9[55] |
| 5000 | 2|3|2 | 11 | 9[55] |
| 8000 | 2|4|2 | 11 | 9[55] |
| 16000 | 4|5|4 | 11 | 9[55] |
| 32000 | 7|8|7 | 11 | 9[55] |
| 48000 | 10|11|10 | 11 | 9[55] |
| 65500 | 13|15|13 | 11 | 9[55] |

\*wykonywane jest 10 zapytań (opcja -n 10)

Tabela dla canberra.com.au.

|  |  |  |  |
| --- | --- | --- | --- |
| Wielość pakietu w bajtach | Czas(min|max|avg)\* | Trasa do hosta | Trasa od hosta [TTL] |
| 32 | 339|340|339 | 21 | 16[48] |
| 500 | 339|349|340 | 21 | 16[48] |
| 1000 | 339|344|339 | 21 | 16[48] |
| 5000 | 339|341|339 | 21 | 16[48] |
| 8000 | 342|343|342 | 21 | 16[48] |
| 16000 | 344|345|344 | 21 | 16[48] |
| 32000 | 348|350|349 | 21 | 16[48] |
| 48000 | 353|354|353 | 21 | 16[48] |
| 65500 | 358|360|358 | 21 | 16[48] |

\*wykonywane jest 10 zapytań (opcja -n 10)

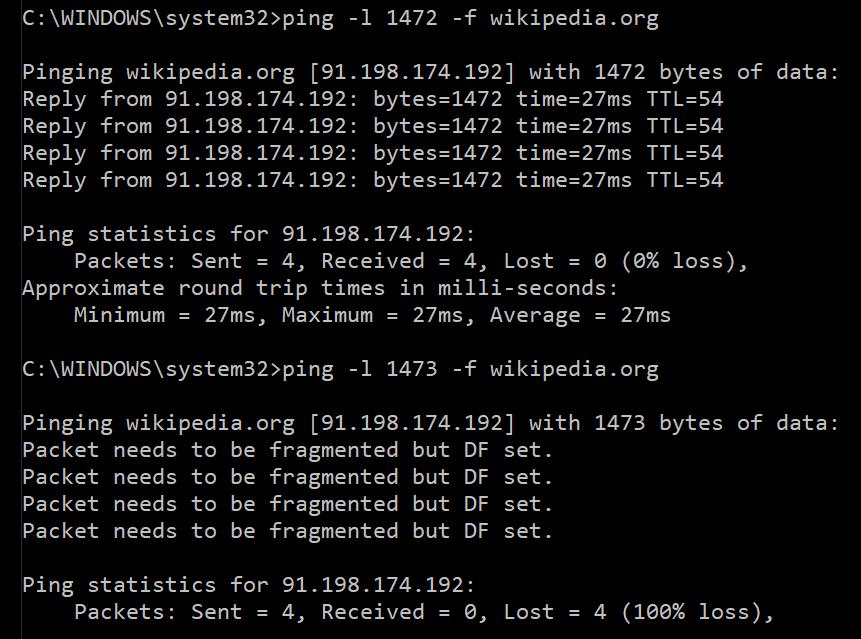
Czas przesyłania dla pakietów poniżej 5000B praktycznie się nie różnił. Drobne różnice zaczęły się pojawiać dla tych od 5000B do 16000B. Dopiero od tych wartości czas wysyłania zaczął rosnąć. Ilość węzłów na trasie do serwera i z powrotem w żadnym przypadku nie uległa zmianie.

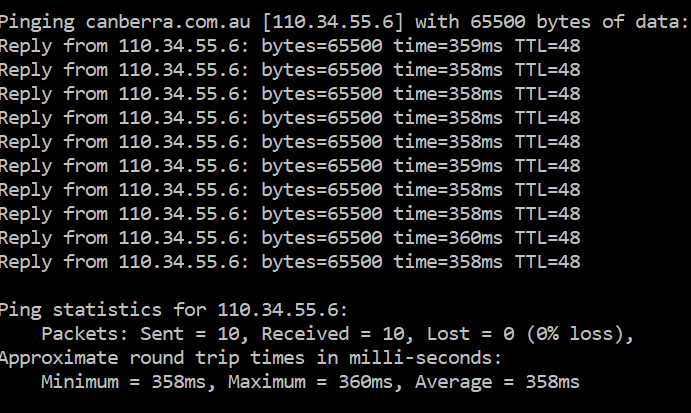
Maksymalny rozmiar wiadomości to 65500B.



Maksymalna wielkość pakietu dla której nie jest potrzebna fragmentacja to 1472B. Wnioskiem z doświadczenia jest to, że fragmentacja wpływa nieznacznie na czas przesyłania pakietu. Dopiero przy bardzo dużych pakietach gdzie nastąpiło większe pofragmentowanie różnica w czasie jest widoczna.

Próba wysłania pakietu większego niż 1472 nie fragmentując go.





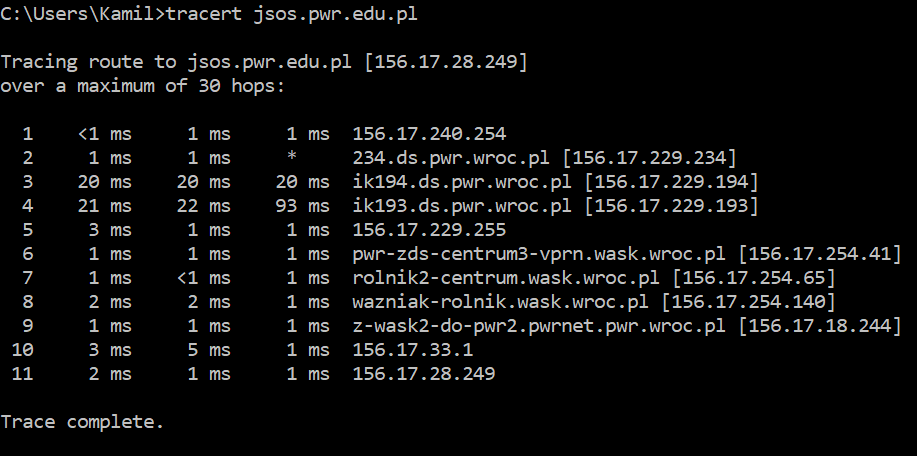
## Traceroute (Tracert)

* 1. **Opis programu**

**Traceroute** (Tracert w systemach Windows) – jest to program służący do badania drogi pakietów w sieci Internet Protocol (IP). Pozwala na wyświetlenie punktów na trasie pakietu wraz z opóźnieniami jakie zaszły na każdym z nich. Używając tego programu można stwierdzić np. przez jakie kraje szedł pakiet. Adresy IP kolejnych routerów ustalane są poprzez wysyłanie pakietów z odpowiednimi wartościami TTL. Przykładowo adres pierwszego routera uzyskiwany jest poprzez wysłanie pakietu z TTL równym 1. Wtedy na pierwszym routerze wartość TTL zostaje zmniejszona do 0, a to spowoduje nadanie informacji o odrzuceniu pakietu. W tej wiadomości znajduje się adres pierwszego routera. Adresy kolejnych routerów pozyskiwane są analogicznie. Jest to ta sama metoda, którą w poprzedniej części używałem ręcznie za pomocą programu ping. Ja w celu realizacji doświadczeń korzystałem z Tracert.

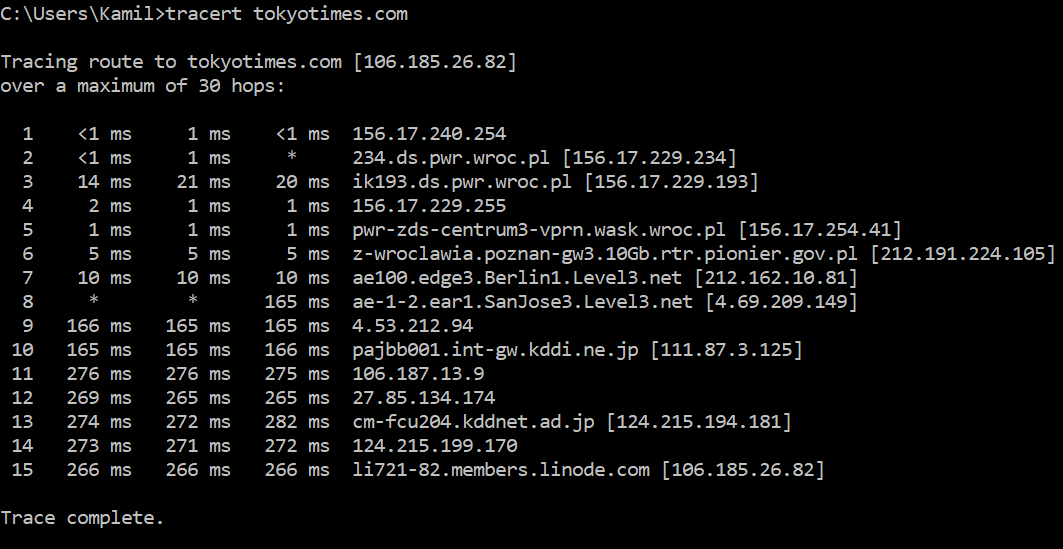
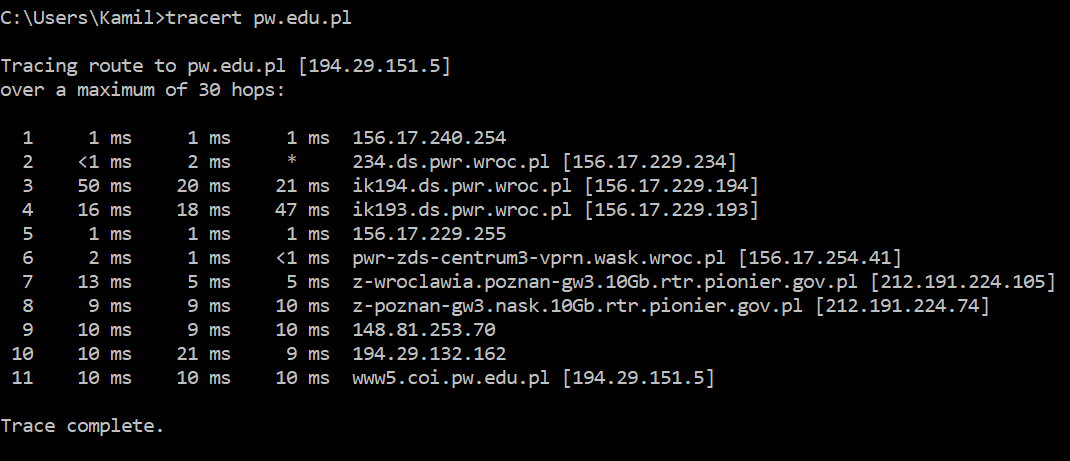
* 1. **Badanie trasy**

Na początku postanowiłem sprawdzić trasę serwera bardzo bliskiego mi geograficznie - jsos.pwr.edu.pl.



Teraz sprawdźmy czy trasy wyznaczone przeze mnie ręcznie w poprzedniej części sprawozdania będą takie same przy użyciu Tracert. Porównam trasy dwóch do trzech serwerów tokyotimes.com, pw.edu.pl i msu.ru.

|  |  |  |
| --- | --- | --- |
| Adres hosta | Trasa do serwera zmierzona ręcznie przy użyciu programu Ping | Trasa do serwera zmierzona przez Tracert |
| tokyotimes.com | 15 | 15 |
| pw.edu.pl | 11 | 11 |
| msu.ru | 15 | 15 |

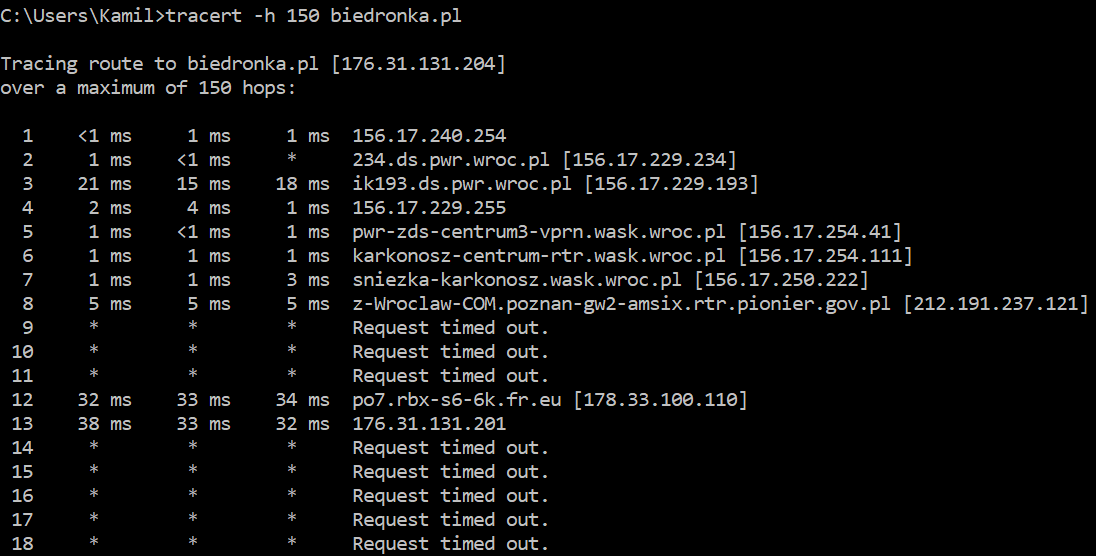


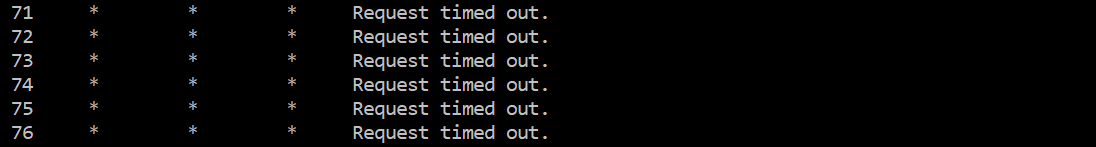
Wnioskiem jest to, że długość tras była taka sama niezależnie od tego czy mierzyłem ją recznie przy użyciu Pinga czy mierzył to Tracert. Trasa do tokyotimes.com biegła przez Berlin, a następnie przez San Jose w Kalifornii lub na Kostaryce\*. Docelowy serwer znajduje się w Japonii.

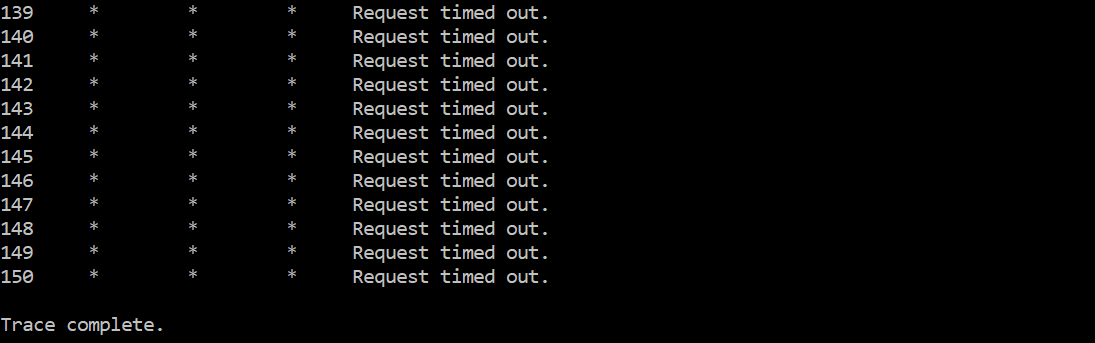
\*miasto o takiej nazwie występuje w obu tych lokalizacjach.

* 1. **Próba znalezienia nietypowych zachowań lub anomalii**

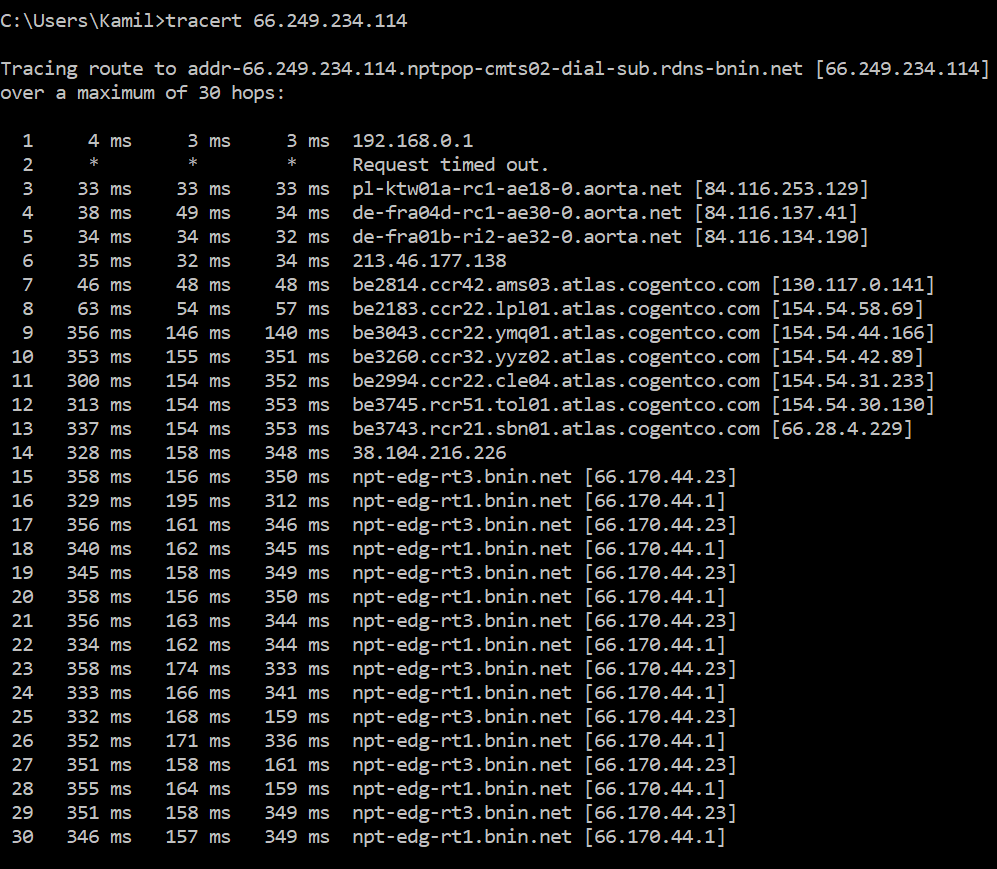
Na koniec zabawy Tracert postanowiłem sprawdzić jak program zachowa się z serwerem w Chinach (114.114.115.119), który za każdym razem zwracał inną wartość TTL w komunikacie zwrotnym. Jednak trasa do wyznaczona przez Tracert okazała się taka sama. Poźniej sprawdziłem jak Tracert zachowa się z adresem biedronka.pl. Trasa okazała się niemożliwa do wyznaczenia.



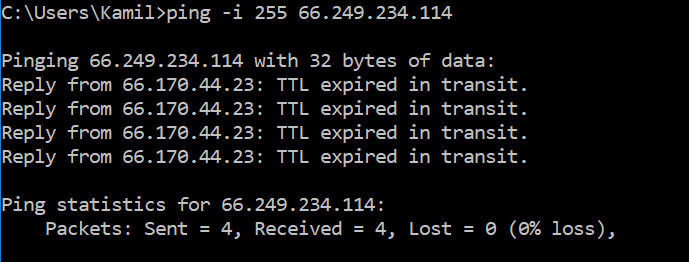




Podejrzewam, że jest przyczyną np. umyślengo zapętlania. Udało mi się znaleźć inny serwer, na którym wyraźnie było widać zapętlanie trasy.

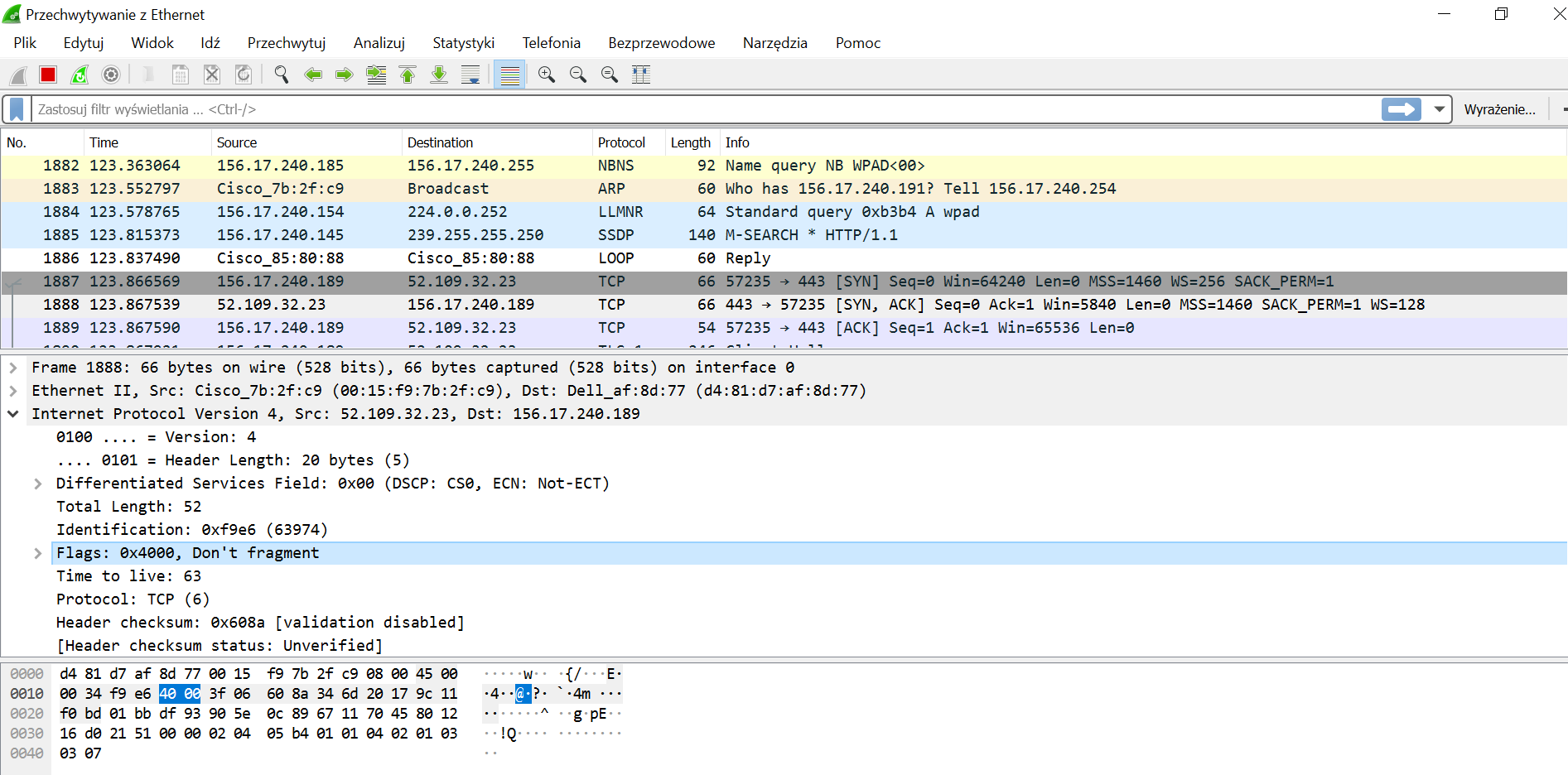


Na poniższym zrzucie ekranu widać, że pakiet nie może dotrzeć do celu nawet przy wartości początkowej TTL równej 255 (maksymalnej).



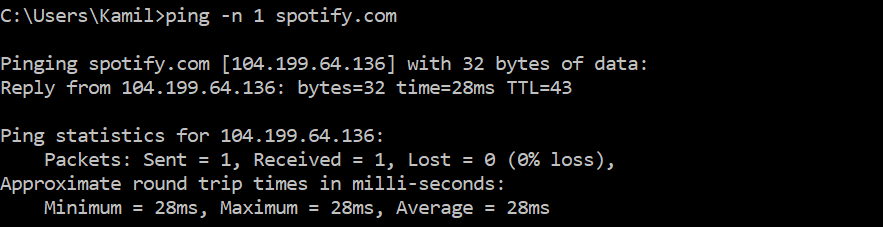
## Whireshark

* 1. **Opis programu**

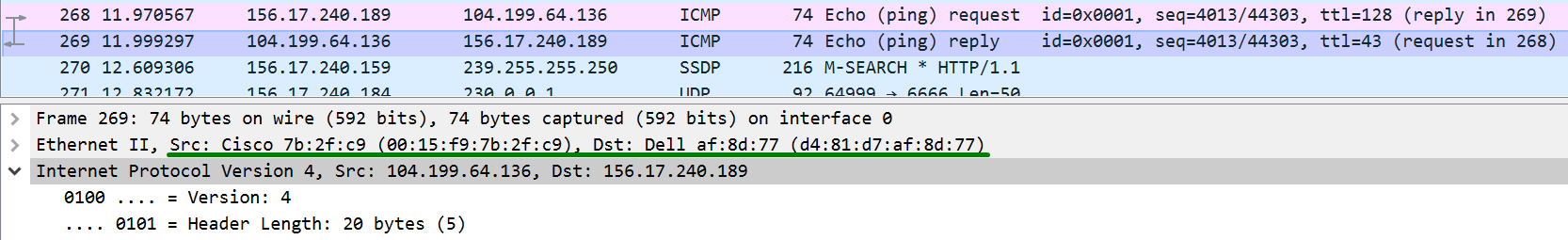
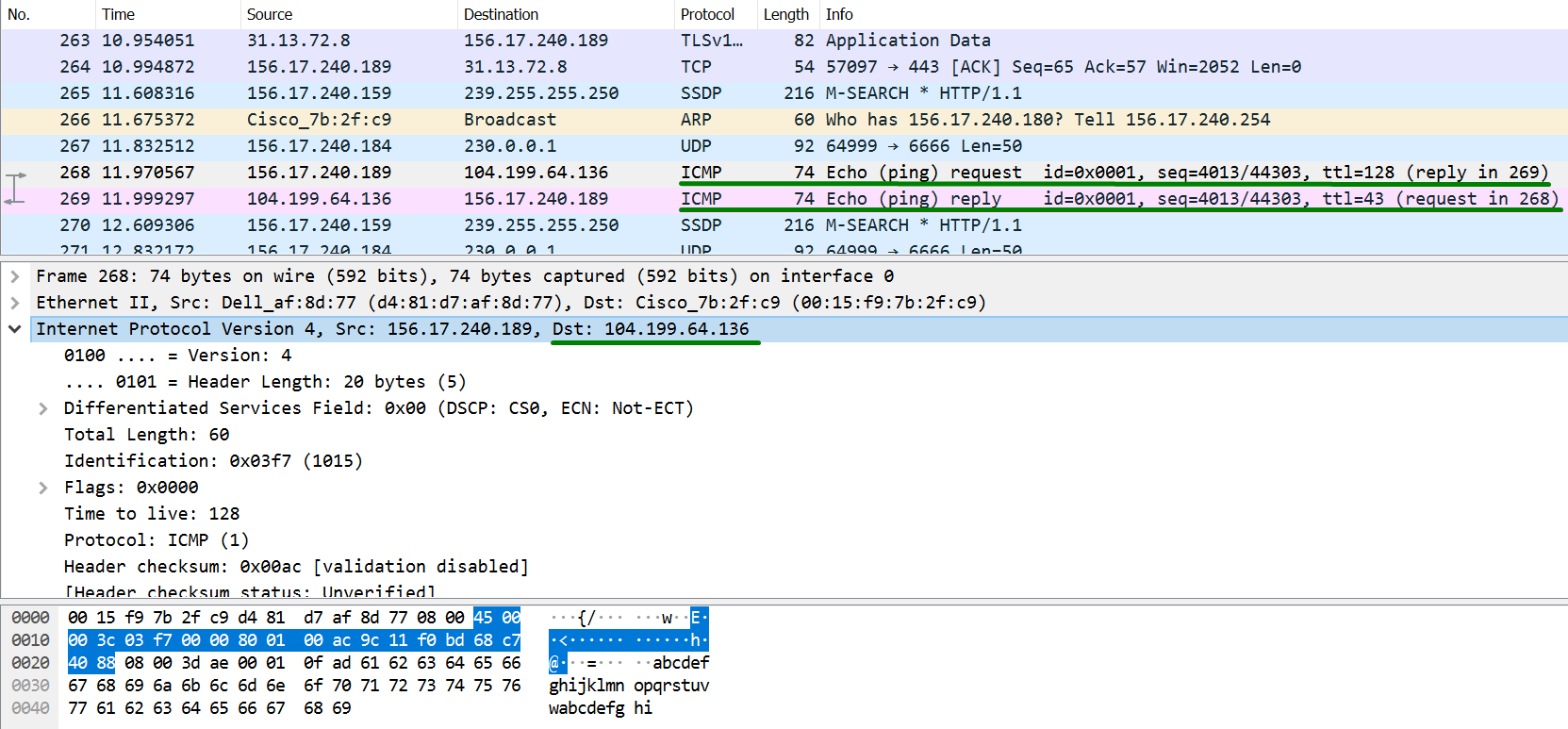
**Whireshark** – jest to program pozwalający analizować wszystkie pakiety wchodzące i wychodzące. Jest w stanie rozpoznać i dekodować wiele różnych protokołów komunikacyjnych. Jedną z jego głównych zalet jest obecność interfejsu graficznego. Pozwala na edytowanie i zapisywanie przechwyconych pakietów. Sam Whireshark nie przechwytuje pakietów, używa do tego innego narzędzia np. npcap (w moim przypadku). Whireshark jest silnym narzędziem wykorzystywanym między innymi do zarządzania sieciami, śledzenia pakietów przez hakerów czy różne służby bezpieczeństwa oraz do prac nad protokołami komunikacyjnymi. Poniżej zrzut ekranu, na którym widać, że przechwycony pakiet TCP został wysłany z flagą mówiącą aby go nie fragmentować. 

* 1. **Przechwycenie pakietów ICMP**

Spróbowałem przechwycić pakiety wysyłane przez program ping.

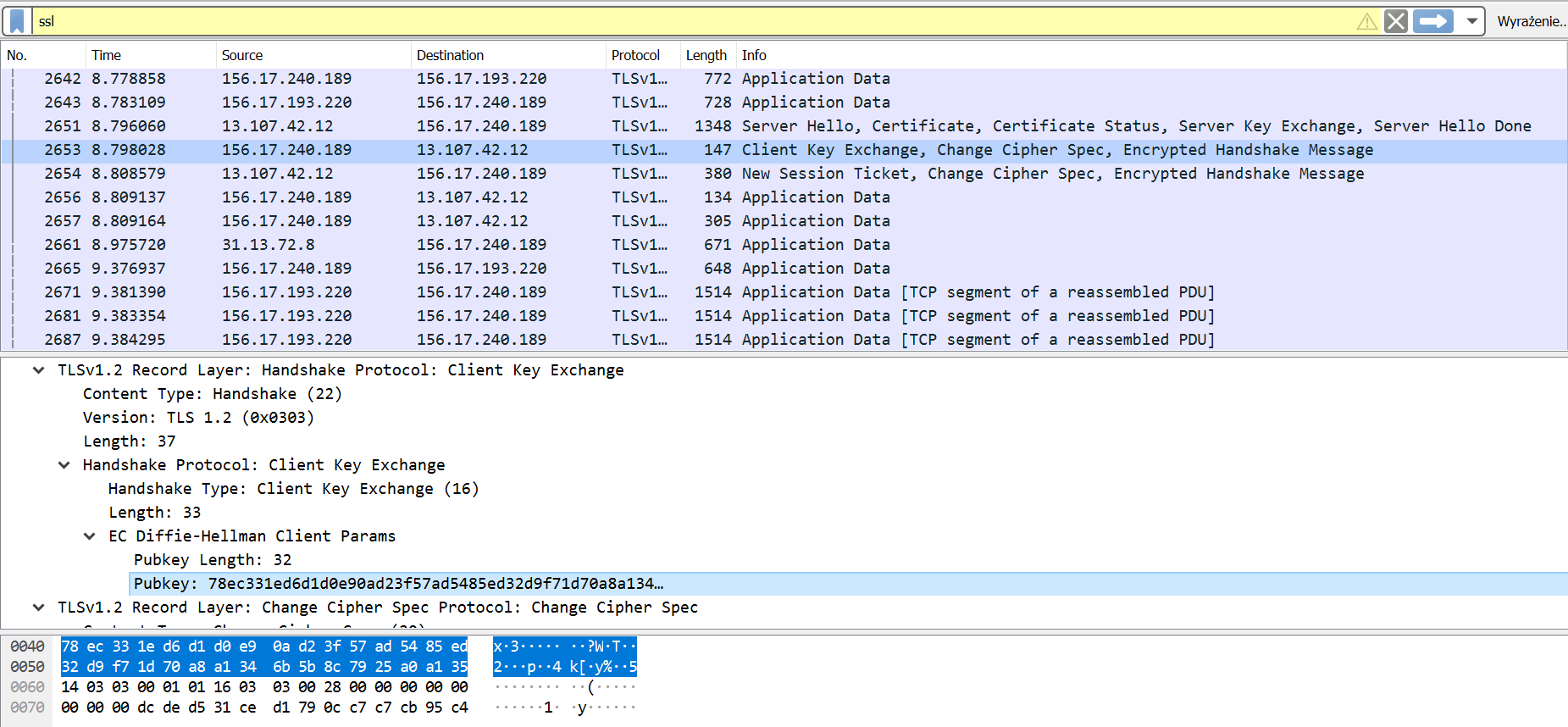


Na poniższym zrzucie ekranu widać wysłany pakiet ‘echo request’ i ‘echo reply’. Widać też, że IP docelowego serwera: 104.199.64.136 na obu zrzutach jest takie same stąd wiem, że przechwyciłem właściwy pakiet.

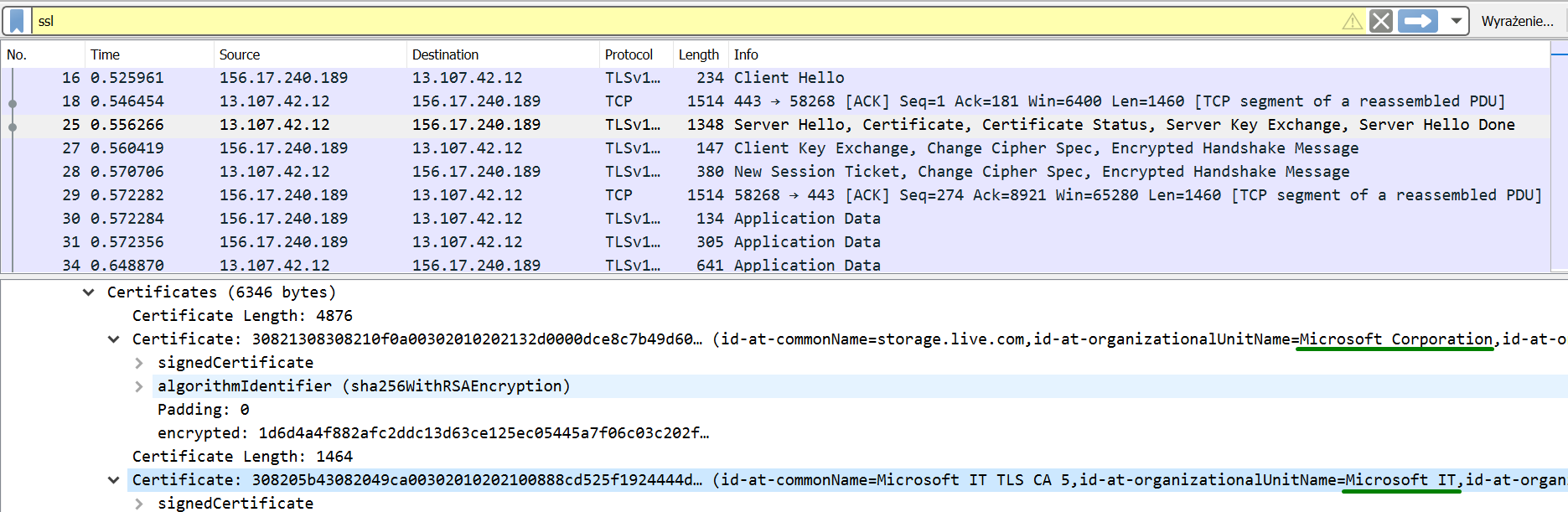


* 1. **Obserwowanie logowania do poczty studenckiej**

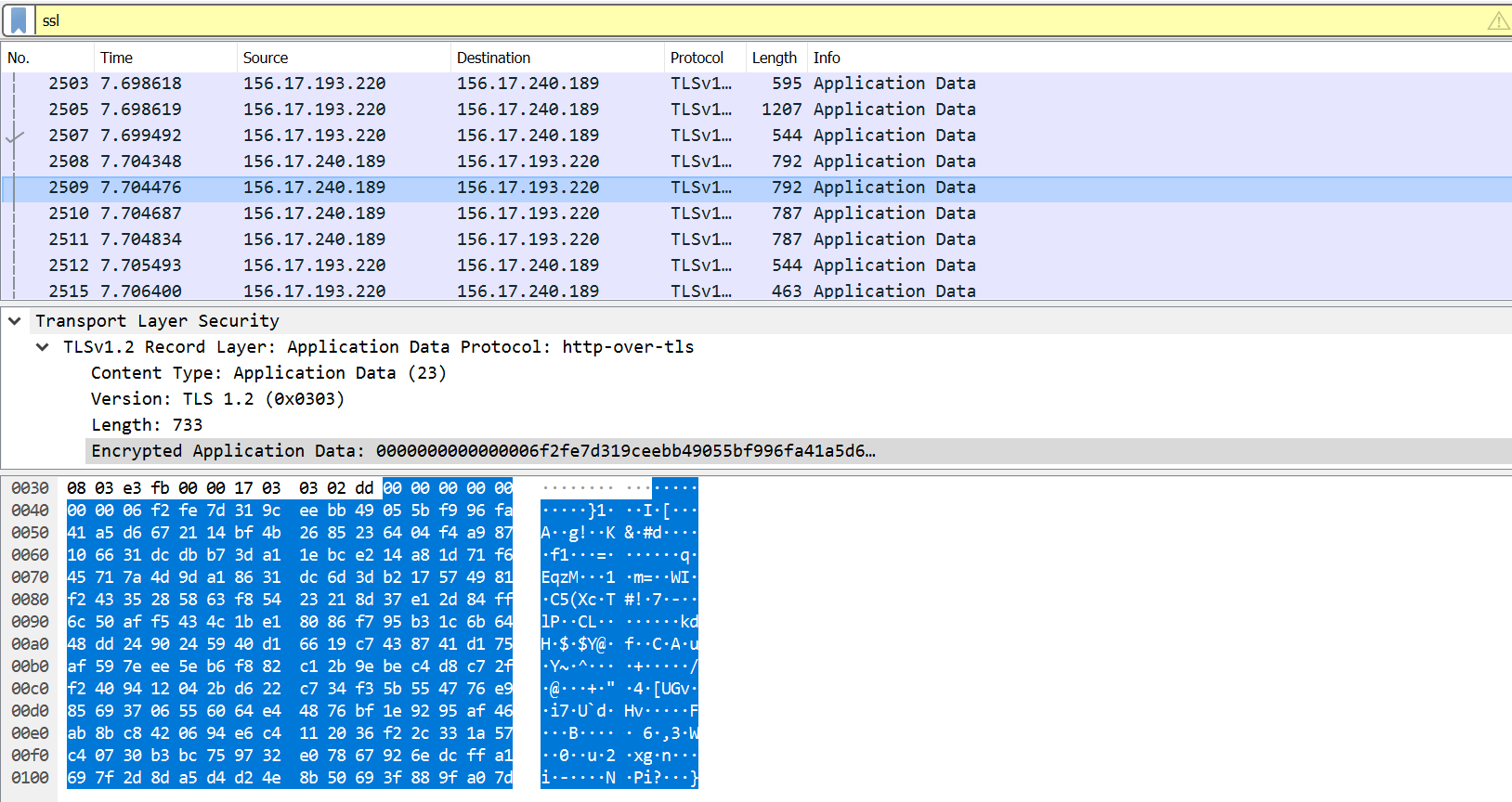
Przed logowaniem włączyłem przechwytywanie pakietów, a następnie użyłem filtrowania wpisując jako kryterium ‘ssl’. Udało mi się w ten sposób uchwycić pakiet z wymianą kluczy. Na zrzucie ekranu widać też klucz publiczny.



Na kolejnym zrzucie ekranu widać 2 certyfikaty Microsoftu.

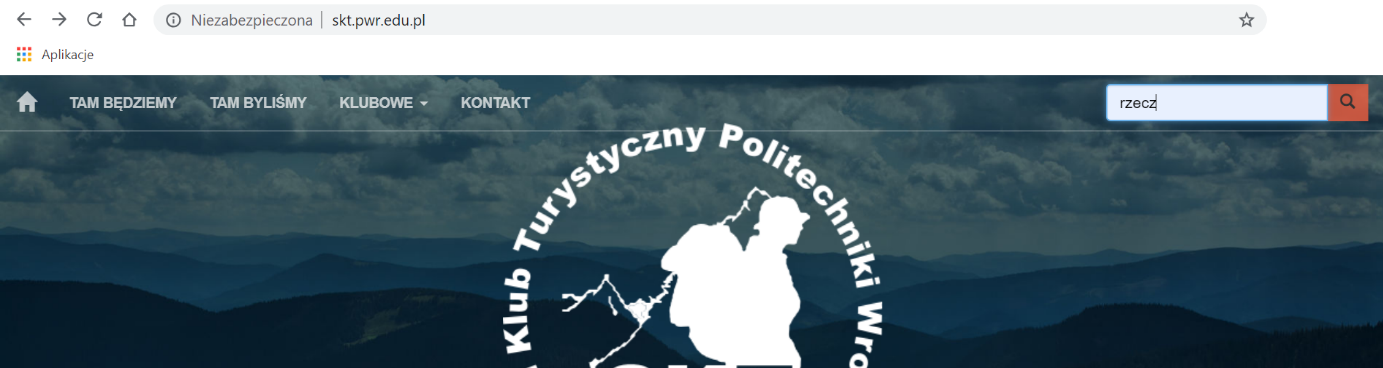


Na kolejnym zrzucie ekranu znajduje się przykładowy pakiet z zaszyfrowanymi danymi.

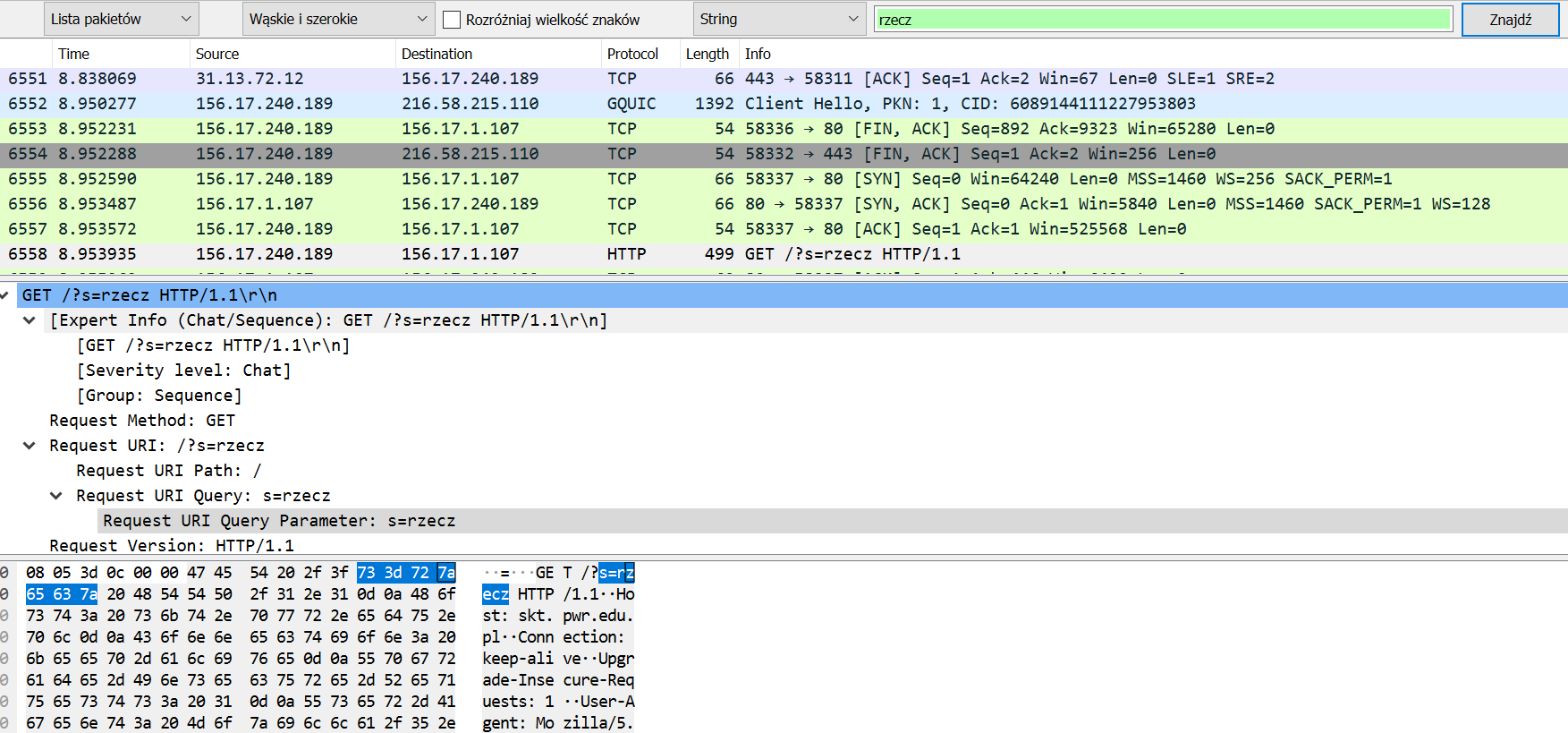


* 1. **Wyszukiwanie na niezaszyfrowanej stronie**

Sprawdziłem czy tekst wpisany w wyszukiwarkę na jakieś stronie nie zabezpieczonej przez https będzie się dało zobaczyć w pakietach. Wybrałem stronę skt.pwr.edu.pl.

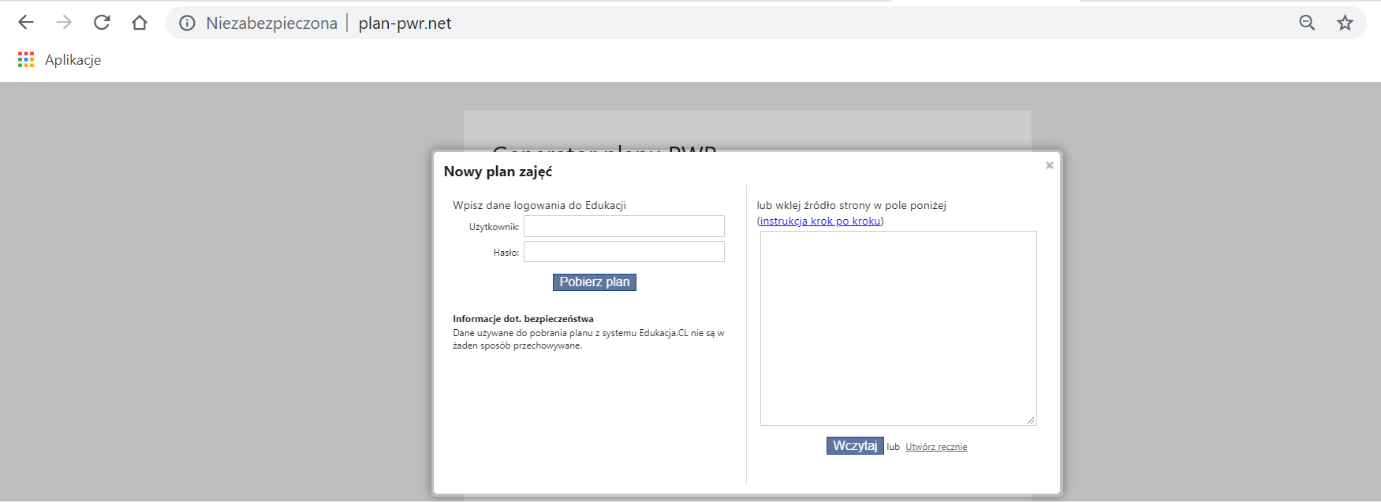


Pakiet z danymi wyszukiwania udało się znaleźć. Widać bezpośrednio treść wpisaną na stronie.

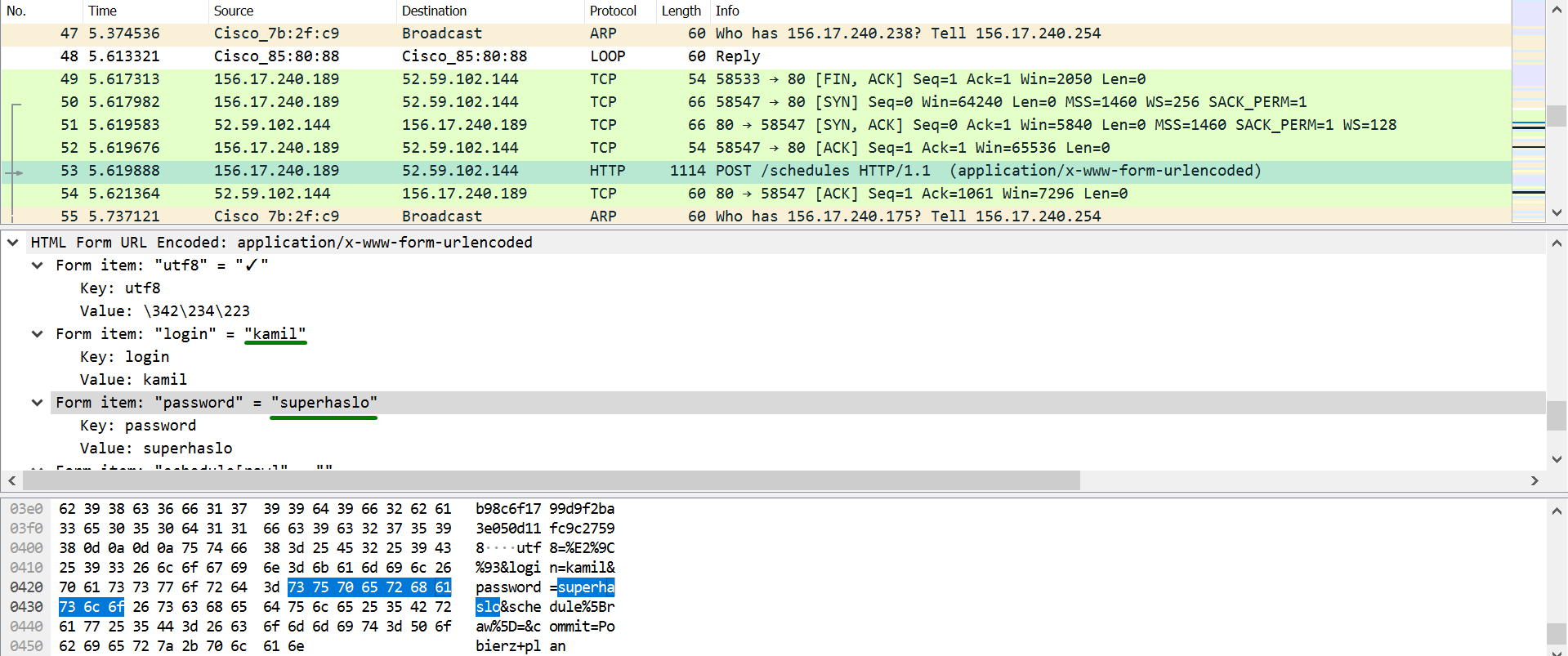


* 1. **Próba przechwycenia danych logowania z niezaszyfrowanej strony**

Po udanej próbie przechwycenia treści wpisanej na stronie zdecydowałem się spróbować przechwycić hasło. Wybrałem stronę niezabezpieczoną https: plan-pwr.net.



Wpisałem (nieprawdziwe) dane logowania i uruchomiłem przechwytywanie pakietów. Rezultat znajduje się na poniższym zrzucie ekranu.



Jak widać zarówno login jak i hasło udało się przechwycić.