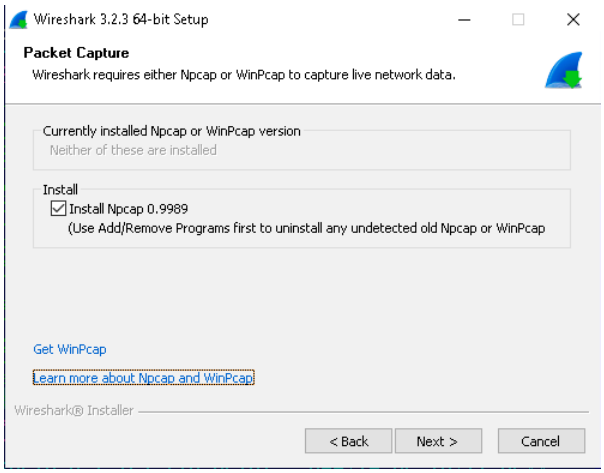
Wprowadzenie do korzystania z narzędzia Wireshark

1. Instalacja programu

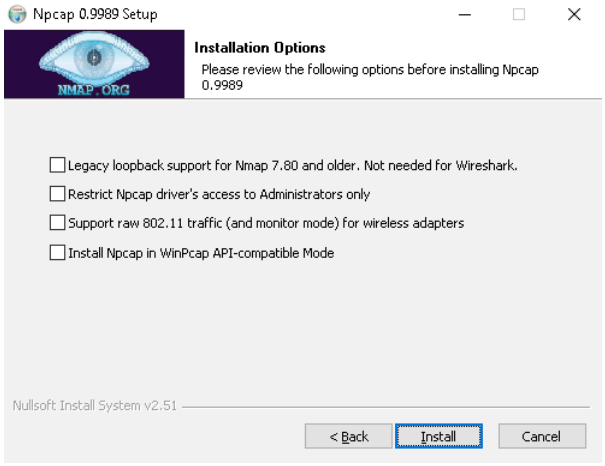
Program Wireshark[1] jest darmowy i można go pobrać ze strony:

```
https://www.wireshark.org/
```

Jest on dostępny dla wielu systemów operacyjnych. Z punktu widzenia przedmiotu OBIR najważniejsze jest aby podczas instalacji wybrać dodatkowo wsparcie dla interfejsu loopback. Wsparcie to realizuje program Npcap, wchodzący w skład instalatora. Podczas instalacji trzeba zaznaczyć odpowiednią opcję, ukazaną na poniższym rysunku:



Program Npcap może podczas swojej instalacji zapytać jeszcze o inne opcje, można pozostawić je nie zaznaczonymi jak na poniższym rysunku:



2.Użytkowanie programu

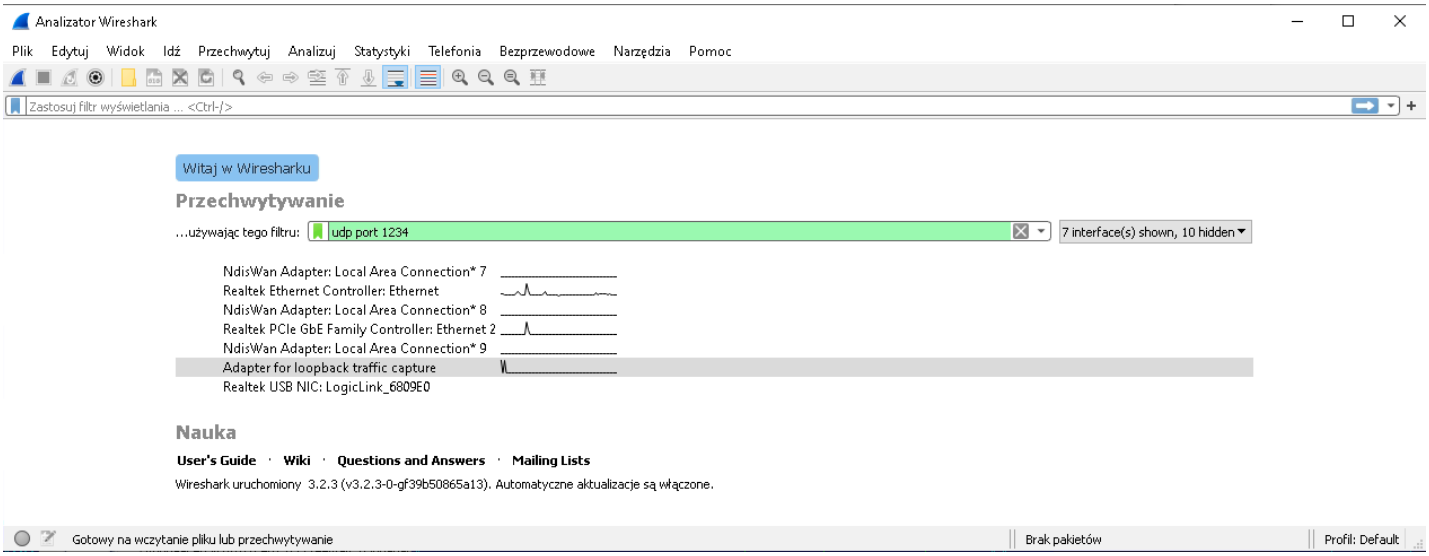
WireShark to narzędzie które pozwala „łapać" pakiety przenoszone przez różne typy sieci, prezentując je w przyjaznej i łatwej do analizy postaci ukazując budowę poszczególnych części danego pakietu. Nioceniona jest także możliwość zapisywania wszystkich lub odfiltrowanych wybranych pakietów na dysk np.: dla późniejszej analizy.

Aby uruchomić program Wireshark należy na komputerach z systemem Linux uruchomić Terminal (ALT+CTRL+T) a następnie wydać polecenie:

```
sudo wireshark
```

Na systemach Windows wystarczy odnaleźć ten program w spisie dostępnych aplikacji.

Dla rozpoczęcia „łapania" pakietów w programie Wireshark wchodzimy w „Przechwytuj" a następnie w „Opcje". Ukaże się nam wtedy lista interfejsów mogących posłużyć do przechwytywania pakietów (nowsze wersje tego programu, opcje te ukazują w głównym oknie zaraz po uruchomieniu):



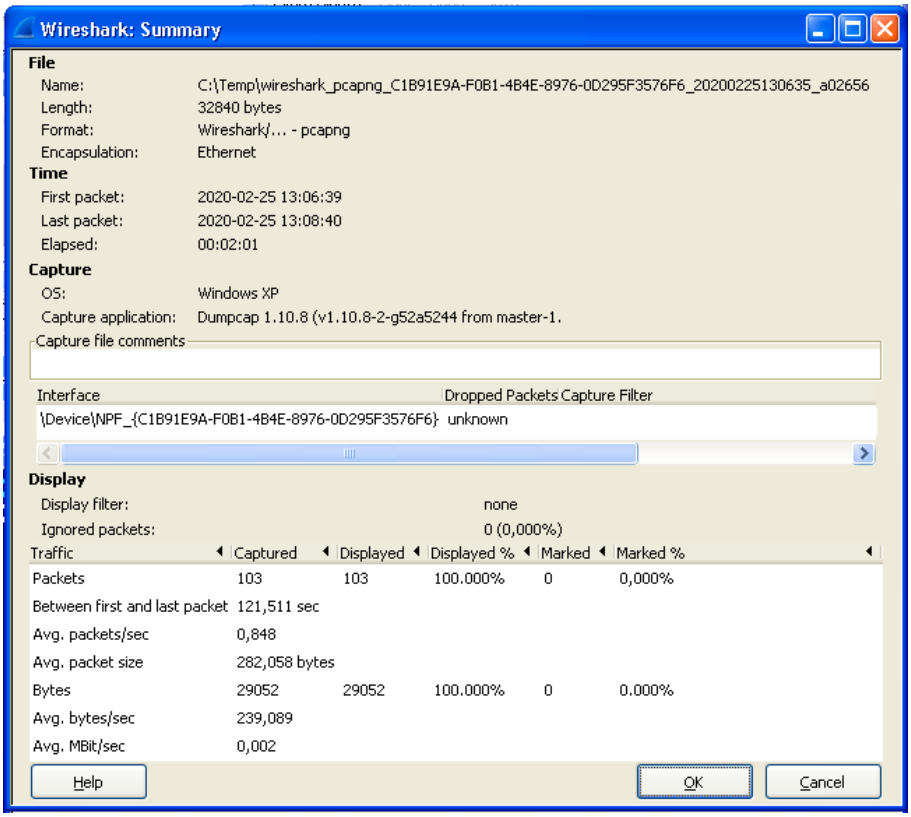
Należy tu jednak wspomnieć, iż ruch lokalny – czyli dotyczący sytuacji gdy np.: właśnie uruchamiamy serwer i chcemy go badać używając klienta uruchamianego na tym samym komputerze – nie może być „złapany” gdy nie zostanie wybrany interfejs „Adapter for loopback traffic capture” (zaznaczony na szaro na powyższym rysunku) lub taki który odpowiada interfejsowi lokalnemu (np.: „lo”, „Loopback: lo”).
Przed przystąpieniem do właściwego procesu „łapania” pakietów trzeba pamiętać, iż ruch sieciowy może być w niektórych przypadkach dość znaczny objętościowo a nie wszystkie pakiety w przyszłości będą nam potrzebne. Dodatkowo w szczególnych przypadkach liczba takich pakietów do zapisania w jednostce czasu może przerosnąć możliwości komputera na jakim działa Wireshark. Warto zatem korzystać z funkcji wstępnego filtrowania co będzie „łapane”. Na powyższym rysunku pokazano w polu wprowadzania filtru wartość: „udp port 1234”. Taka formuła ograniczy łapanie pakietów związanych wyłącznie z protokołem UDP i dodatkowo na porcie 1234.
Badając dla przykładu pakiety ARP i ruch do określonego urządzenia (tu adresu MAC), zasadne może być użycie następującej formuły:

```
(ether src ab:cd:ef:01:23:45 && arp) || (ether dst ab:cd:ef:01:23:45 && arp)
```

Dla sprawdzenia czy Wireshark potrafi poprawnie wykonać powyższe, uruchamiamy przechwytywanie pakietów (klawisz „Start”) i dodatkowo w osobny Terminalu lub tzw. „Command Prompt” (CMD) uruchamiamy polecenie (zakładając, że w naszej sieci istnieje urządzenie 10.17.0.80):

```
ping 10.17.0.80
```

Po krótkiej chwili (około 5sek.) możemy zaobserwować pojawiające się w oknie Wiresharka złapane pakiety. Zanim zaczniemy je analizować, podejrzyjmy za pomocą opcji „Statistics” -> „Summary” w jakim pliku i w jakiej lokalizacja na dyskach komputera, program Wireshark (uruchomiony na Windows XP) umieszcza „złapane” pakiety sieciowe:



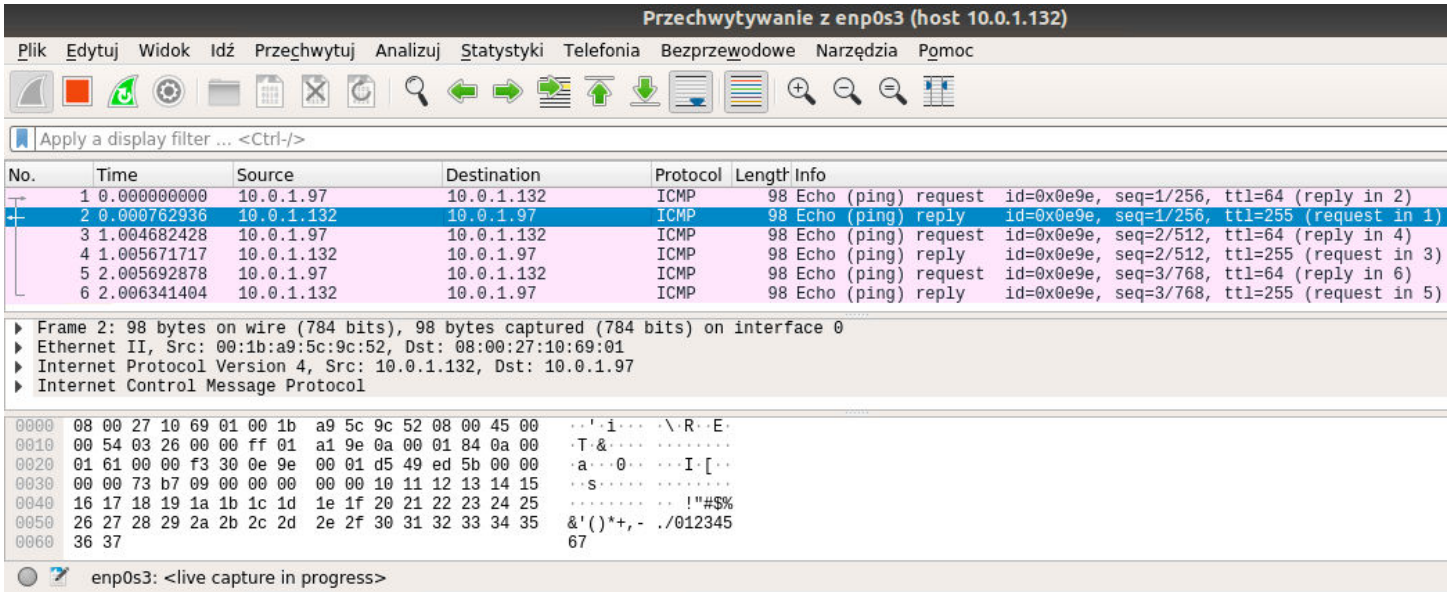
Na powyższym rysunku widać, że w ciągu około 121,5sek. program złapał pakiety i przechowuje je w katalogu C:\Temp\ (domyślny katalog z plikami tymczasowymi) w pliku o wielkości 32840B i o dość dziwnej nazwie zbudowanej na bazie tekstu „wireshark_pcapng" oraz tekstowej nazwy interfejsu sieciowego który złapał pakiety zakończonego stemplem czasowym – kiedy rozpoczęto pracę z tym plikiem. Według podobnej reguły tworzona jest nazwa tego pliku w systemie Linux i innych systemach.
Generalnie budowanie reguł filtracji jest zagadnieniem ciekawym i dość złożonym. Reguły filtracji posiadają własną gramatykę pozwalającą tworzyć skomplikowane reguły. Dla przykładu aby ograniczyć się wyłącznie do wiadomości związanych z protokołem ICMP[2] (używanym przez narzędzie 'ping') i określonym urządzeniem (o IP 10.17.0.80), należy użyć filtr:

```
host 10.17.0.80 && icmp
```

Gdy jednak chcemy aby badany były ruch ICMP związany z dwoma urządzeniami, należy użyć bardziej zaawansowanej reguły filtrowania, np.:

```
((ip host 10.17.0.80) || (ip host 10.17.0.97)) && icmp
```

Widać w niej, że dzięki nawiasom i operatorom „or" („||") oraz „and" („&&") można kontrolować kolejność i zasięg operacji dokonywanych podczas filtrowania. Warte zatem uwagi jest tworzenie reguły filtrowania w taki sposób aby ograniczą ilość „złapanych" informacji wyłącznie do tego co nam jest potrzebne. Więcej informacji na temat „łapania" pakietów i ich filtrowania można znaleźć na stronach dokumentacji projektu Wireshark[3][4].
W trakcie „łapania" pakietów dla obejrzenia co dzieje się z programem Wireshark – wystarczy nawigować po trzech częściach głównego okna tego programu (dla ustawień standardowych): górna część – prezentuje sekwencje „złapanych" pakietów, środkowa – przedstawia interpretację warstwową wybranego pakietu, dolna – ukazuje dane tego pakietu w postaci surowej. Poniższy rysunek pokazuje przykład wyświetlenia „złapanych" pakietów.



Widać, iż w części górnej wybrano pakiet dla którego w części środkowej wypisano z czego się składał (kolejno są przedstawione warstwy: PHY, MAC, IP i ICMP) a w części dolnej przedstawiono jego surową postać w notacji szesnastkowej i ASCII. W części środkowej za pomocą kliknięcia w znaczek „►” możliwe jest dokładniejsze przejrzenie informacji zapisanych w wybranej warstwie sieciowej przenoszonej w danym pakiecie.

3. Bibliografia:

1. https://wiki.wireshark.org, ostatnia wizyta 2020.03.23
2. https://en.wikipedia.org/wiki/Internet_Control_Message_Protocol, ostatnia wizyta 2020.03.23
3. https://wiki.wireshark.org/CaptureFilters, ostatnia wizyta 2020.03.23
4. http://www.tcpdump.org/manpages/pcap-filter.7.html, ostatnia wizyta 2020.03.23