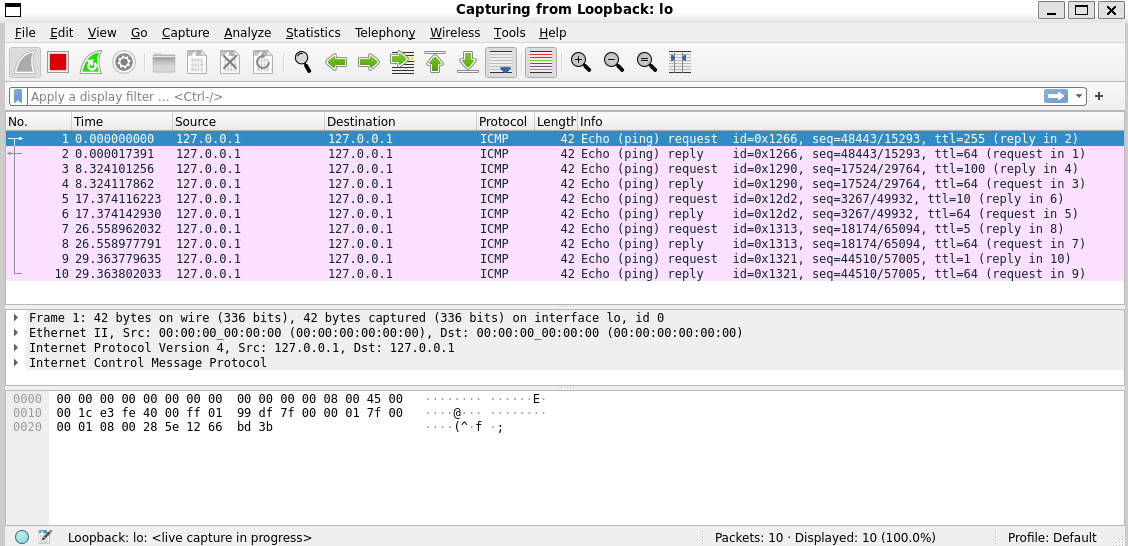
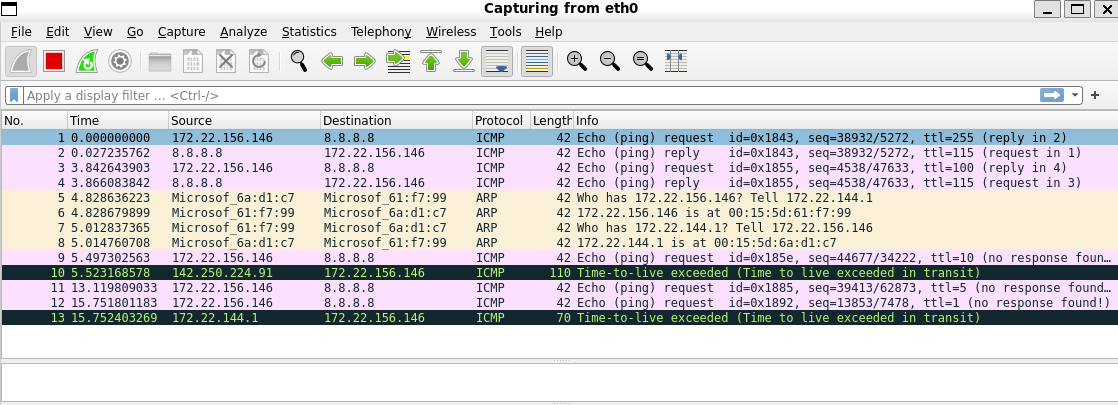
**Opcje IP i gniazda surowe**

**Laboratorium 3**

**Wiktor Zmiendak**

**Zadanie 1**

Na podstawie zaobserwowanych odpowiedzi, nawet z TTL ustawionym na 1, pakiet ICMP Echo Request dotarł do localhost i wygenerował odpowiedź. Oznacza to, że punkt przełamania dla localhost nie został osiągnięty nawet przy wartości TTL = 1.

****

Natomiast w przypadu adresu googla:

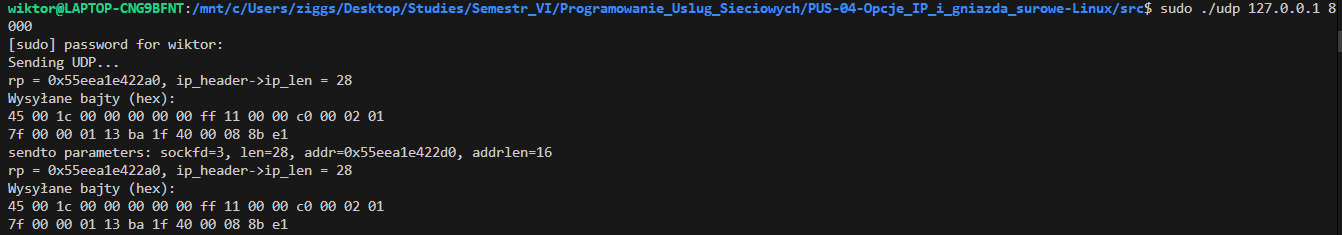
* pakiety 1 i 2: wysłano ICMP Echo Request z TTL=255 do 8.8.8.8 i otrzymano odpowiedź Echo Reply. TTL w odpowiedzi wynosi 115.
* Pakiety 3 i 4: wysłano ICMP Echo Request z TTL=100 do 8.8.8.8 i otrzymano odpowiedź Echo Reply. TTL w odpowiedzi również wynosi 115.
* Pakiet 9: wyłano ICMP Echo Request z TTL=10 do 8.8.8.8 i nie otrzymano bezpośredniej odpowiedzi Echo Reply. Wireshark zaznacza "no response found".

Dzieje się tak, ponieważ ścieżka sieciowa między naszym komputerem a serwerem 8.8.8.8 wymaga więcej niż 10 skoków.

**Zadanie 2**

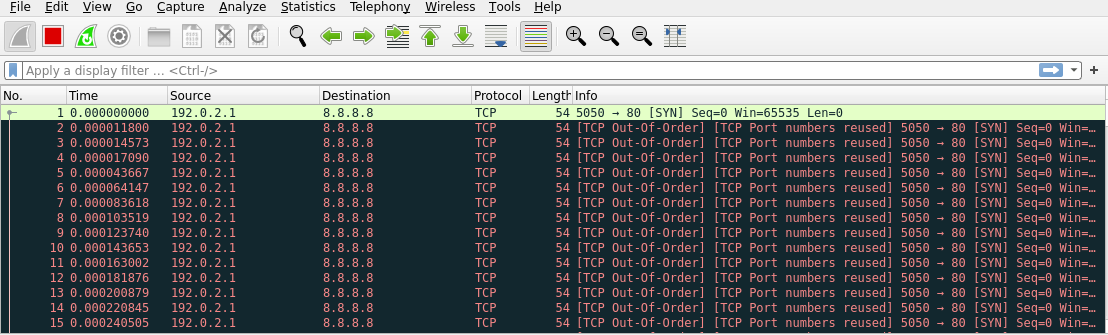
**Obraz zawierający tekst, numer, Czcionka, zrzut ekranu

Zawartość wygenerowana przez sztuczną inteligencję może być niepoprawna.**

****

Program działa prawidłowo, wysyłając pakiety UDP z zdefiniowanym źródłowym adresem IP i portem na docelowy adres IP i port podane jako argumenty. Heksadecymalne wyjście programu przedstawia surowe bajty nagłówków IP i UDP, które są enkapsulowane w pakietach sieciowych przechwytywanych przez Wiresharka. Różnica w długości pakietu między wyjściem programu a Wiresharkiem wynika z dodatkowych nagłówków warstwy 2 dodawanych przez system operacyjny i kartę sieciową.

**Zadanie 3**

****

Na podstawie zrzutu ekranu można stwierdzić, że program działa zgodnie z zamierzeniem i generuje pakiety TCP z ustawioną flagą SYN na zadany adres IP i portu. Jest to potwierdzenie, że implementacja SYN flood przy użyciu surowych gniazd TCP została wykonana poprawnie pod kątem generowania samych pakietów SYN.

Program tworzy surowe pakiety TCP z ustawioną flagą SYN, symulując próbę nawiązania połączenia. Nagłówki IP i TCP są ręcznie wypełniane, a suma kontrolna TCP jest obliczana na podstawie pseudo-nagłówka. Pakiety te są następnie wysyłane w sposób ciągły na zadany adres IP i port, co w efekcie stanowi atak SYN flood

**Zadanie 4**

Obecność adresu IPv6 przypisanego do któregokolwiek z interfejsów jest silnym dowodem na to, że system operacyjny wspiera i ma aktywną obsługę protokołu IPv6

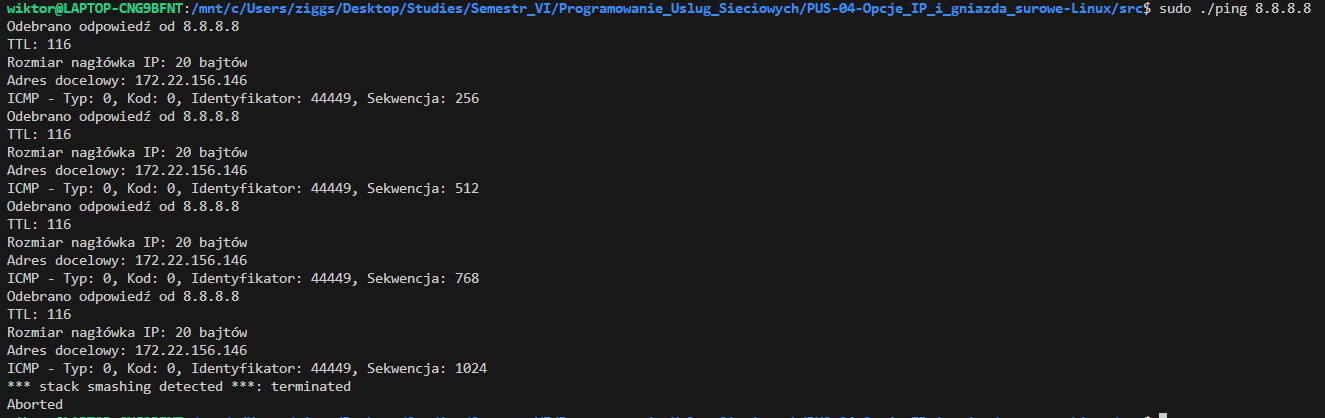
****

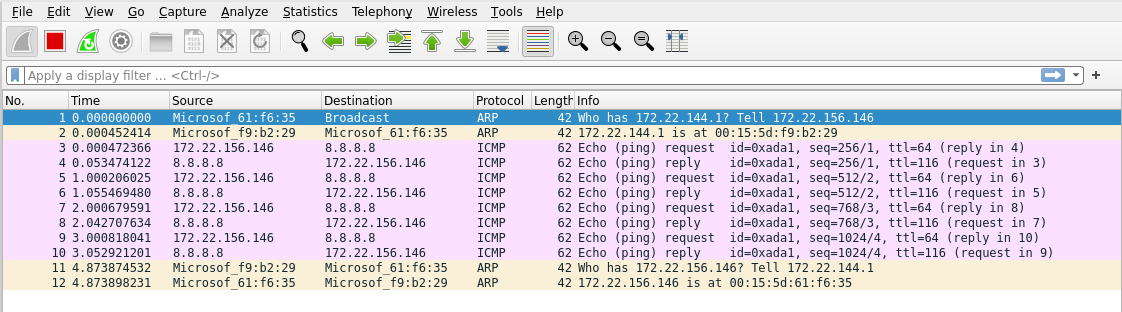
**Obraz zawierający tekst, oprogramowanie, Czcionka, numer

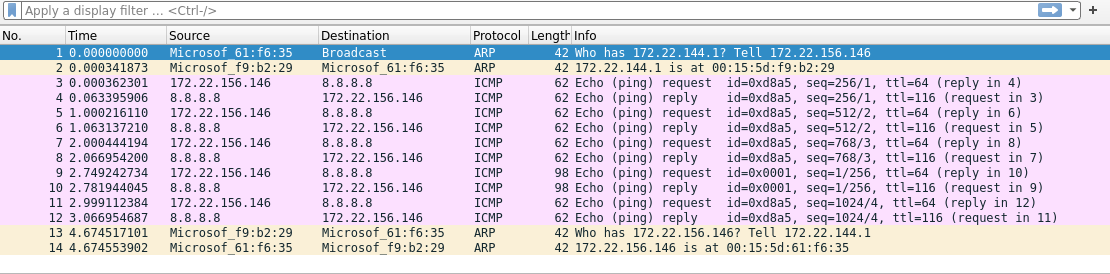
Zawartość wygenerowana przez sztuczną inteligencję może być niepoprawna.**

Program próbuje wysłać puste datagramy UDPv6 na zadany adres i port, wykorzystując surowe gniazdo. Zamiast ręcznie tworzyć nagłówek IP, program polega na systemie operacyjnym w tym zakresie. Wireshark pokazuje pakiety UDPv6 wysyłane z naszego systemu na docelowy adres IPv6 i port.

**Zadanie 5**

****

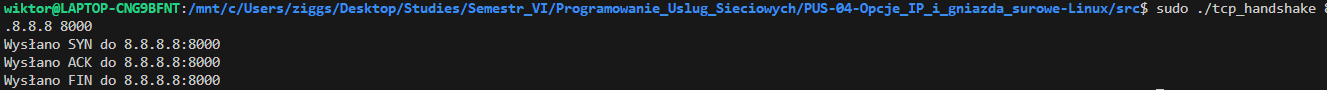
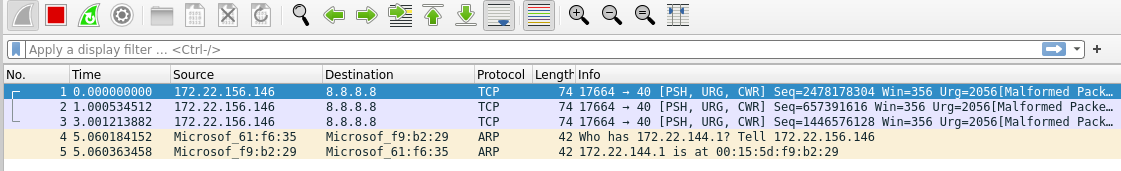
****

****

**Bez wywołania systemowego pinga:** program wysyła 4 komunikaty ICMP Echo Request. Jeśli docelowy host odpowiada, proces potomny odbiera komunikaty ICMP Echo Reply i wypisuje informacje z nagłówków IP i ICMP, w tym adres źródłowy odpowiedzi, TTL, rozmiar nagłówka IP, adres docelowy oraz typ, kod, identyfikator i numer sekwencyjny komunikatu ICMP Reply.

**Po wywołaniu systemowego pinga:** program nadal wysyła swoje komunikaty ICMP z własnym identyfikatorem. Dodatkowo, systemowy ping wysyła swój własny komunikat ICMP Echo Request z innym identyfikatorem. Jeśli docelowy host odpowie, proces potomny w programie odbierze zarówno odpowiedzi na swoje pingi, jak i odpowiedź na ping systemowy, wypisując informacje o obu zestawach odpowiedzi.

**Zadanie 6**

****