Instrukcja laboratoryjna dla „Sieci komputerowe i Internet” – Moduł 4, Zadanie 3 *– Radosław Terelak*

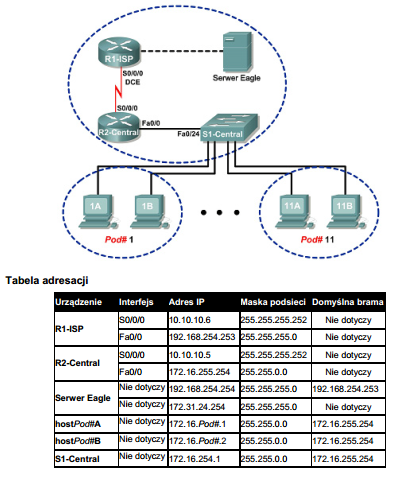
1. Cele zadania laboratoryjnego:

Celem zadania laboratoryjnego jest połączenie dwóch małych lokalnych sieci komputerowych z wykorzystaniem routera i procesu routingu, oraz z wykorzystaniem protokołu IPv6.

1. Przygotowanie środowiska do zajęć:

W ramach zadań wykorzystane zostaną przygotowane już wcześniej pliki *.pcap* (zawierające odpowiednią przechwyconą komunikację sieciową w ramach aplikacji Wireshark), stąd do zadań potrzebna jest aplikacja Wireshark oraz odpowiednie wymieniane w danych krokach pliki *.pcap*.

Same pliki *.pcap* zostały uzyskane w ramach topologii wskazanej na poniższym zdjęciu, której znajomość ma znaczenie w kontekście rozeznania się w logice zadań realizowanych w danych krokach:

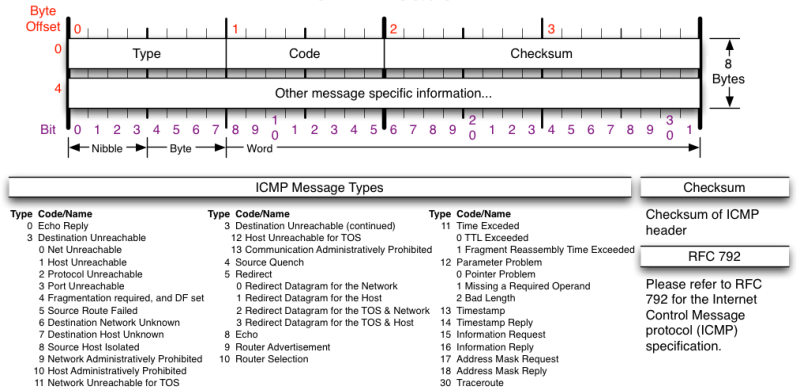


Korzystając z programu Wireshark w ramach zadań przechwytywane będą transmisje danych przesyłane pomiędzy zadanymi urządzeniami sieciowymi w różnych scenariuszach z wykorzystaniem aplikacji **ping**, która z kolei wykorzystuje komunikaty protokołu ICMP.

**Wprowadzenie**

Protokół ICMP (Internet Control Message Protocol) został po raz pierwszy zdefiniowany w dokumencie RFC 792, we wrześniu 1981. Typy komunikatów ICMP opisano później w dokumencie RFC 1700. Protokół ICMP działa w warstwie sieci modelu TCP/IP i jest używany do wymiany informacji kontrolno-diagnostycznych pomiędzy urządzeniami. Przykładem wykorzystania może być sytuacja, kiedy router nie może dostarczyć pakietów do sieci lub hosta docelowego, a informacje o tym fakcie są wtedy zwracane do nadawcy właśnie z wykorzystaniem odpowiednich komunikatów protokołu ICMP. Zarówno znana komenda **ping,** jak i **tracert,** wykorzystują również komunikaty ICMP do realizacji swoich zadań.

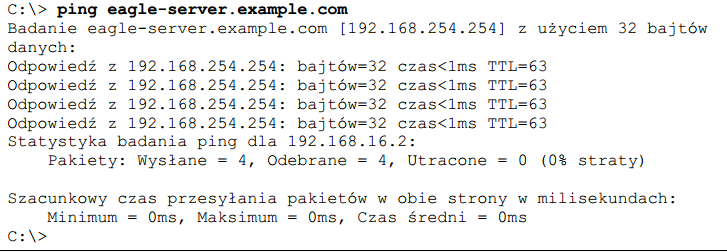
Na poniższym rysunku przedstawione zostały pola nagłówka komunikatu ICMP. Jak można zauważyć, każdy komunikat ICMP rozpoczyna się 8-bitowym polem typu, 8-bitowym polem kodu oraz 16-sto bitową sumą kontrolną. Pozostałe pola zależą od typu komunikatu ICMP. Na rysunku przedstawiono również możliwe wartości i przeznaczenia dla pola **Typ** oraz pola **Kod** dla wiadomości ICMP opisane w dokumencie RFC 792:



Zadania do realizacji

Krok 1: Przechwycenie i analiza wiadomości ICMP echo skierowanych do serwera Eagle.

W tym kroku program Wireshark będzie użyty do przechwycenia i przeanalizowania wiadomości ICMP typu ***echo request/echo reply*** w ramach prawidłowo zrealizowanej komunikacji pomiędzy komputerem „***1A***” a serwer „***Eagle***” z wykorzystaniem aplikacji **ping**.

* + 1. Otworzyć z wykorzystaniem aplikacji Wireshark plik „***ping\_to\_Eagle\_Servera.pcap***”
    2. W ramach przechwyconego ruchu sieciowego jaki się ukaże po otwarciu pliku, wykonane zostało polecenie ***ping*** z komputera ***1A*** na adres IP serwera ***Eagle*** (192.168.254.254). Na poniższym zdjęciu został zawarty wynika wydanego w/w polecenia w wierszu poleceń komputera ***1A***:  
         
       Jak można zauważyć, polecenie ***ping*** wykazało, że transmisja danych została zrealizowana w sposób prawidłowy.
    3. Analizując przechwycony ruch sieciowy w aplikacji Wireshark (dla ułatwienia analizy należy wykorzystać filtr: icmp), wskaż, które z urządzeń sieciowych (tj. urządzenie o jakiej nazwie) odpowiada na wysyłane z komputera 1A żądania ICMP echo (wykorzystując adres IP źródłowy z odpowiedzi):  
       ***Eagle Server***
    4. Zaznacz pierwszą ramkę zawierającą komunikat typu ***icmp echo request***. Rozwiń następnie środkową cześć okna programu Wireshark i rozwiń pole dotyczące protokołu ICMP.  
       Dolne okno będzie potrzebne do zbadania pola danych, tj. zanotuj w poniższej tabelce wskazane informacje odczytane z tegoż pierwszego komunikatu żądania ICMP echo kierowanego do serwera Eagle:

|  |  |
| --- | --- |
| **Pole** | **Wartość** |
| Typ | ***8*** |
| Kod | ***0*** |
| Identyfikator | ***2*** |
| Numer sekwencyjny | ***25*** |

Czy dane zawierają 32 bajty? ***Tak***

***Uwaga:*** *System Windows w ramach identyfikatora i numeru sekwencyjnego stosuje dwa możliwe standardy, tj. „BE” oraz „LE”, stąd wypełniając informacje w tabelkach, należy wybrać jeden z tych standardów i się go potem trzymać.*

* + 1. Zaznacz drugą ramkę zawierającą komunikat typu ***icmp echo reply***. Rozwiń następnie środkową cześć okna programu Wireshark i rozwiń pole dotyczące protokołu ICMP.  
       Zanotuj w poniższej tabelce wskazane informacje odczytane z tegoż drugiego komunikatu odpowiedzi ICMP echo odebranego od serwera Eagle:

|  |  |
| --- | --- |
| **Pole** | **Wartość** |
| Typ | ***0*** |
| Kod | ***0*** |
| Identyfikator | ***2*** |
| Numer sekwencyjny | ***25*** |

Które pola uległy zmianie w porównaniu z żądaniem ICMP echo?   
***Pole Typ uległo zmianie.***

* + 1. Przeanalizuj zawartość pozostałych komunikatów z żądaniami i odpowiedziami ICMP echo, i uzupełnij w tabelce następujące informacje dla każdego z tych komunikatów ping:

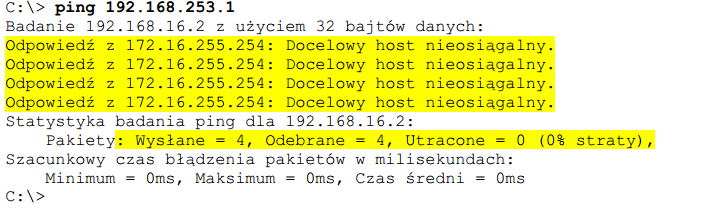
|  |  |  |
| --- | --- | --- |
| **Komunikat** | **Identyfikator** | **Numer sekwencyjny** |
| Żądanie # 2 | ***2*** | ***26*** |
| Odpowiedź # 2 | ***2*** | ***26*** |
| Żądanie # 3 | ***2*** | ***27*** |
| Odpowiedź # 3 | ***2*** | ***27*** |
| Żądanie # 4 | ***2*** | ***28*** |
| Odpowiedź # 4 | ***2*** | ***28*** |

Wskaż, dlaczego wartość sumy kontrolnej zmienia się przy każdym kolejnym żądaniu?

***Każde żądanie ma swoją wartość sumy kontrolnej zapewniającą integralność i spójność wysyłanych i odbieranych danych. Zmienia się dla każdego żądania. Suma kontrolna jest wyliczana na podstawie danych. Jeśli dane są inne to suma kontrolna będzie inna.***

Krok 2:

W tym kroku program Wireshark będzie użyty do przechwycenia i przeanalizowania wiadomości ICMP typu ***echo request/echo reply*** z wykorzystaniem aplikacji **ping**, przy czym wysyłane do fikcyjnego nieistniejącego urządzenia w sieci.

* + 1. Otworzyć z wykorzystaniem aplikacji Wireshark plik „***ping\_to\_fictitious\_host.pcap***”
    2. W ramach przechwyconego ruchu sieciowego jaki się ukaże po otwarciu pliku, wykonane zostało polecenie ***ping*** z komputera ***1A*** na adres IP nieistniejącego urządzenia w sieci (tj. 192.168.253.1). Na poniższym zdjęciu został zawarty wynika wydanego w/w polecenia w wierszu poleceń komputera ***1A***:
    3. Analizując przechwycony ruch sieciowy w aplikacji Wireshark (dla ułatwienia analizy należy wykorzystać filtr: icmp), wskaż, które z urządzeń sieciowych (tj. urządzenie o jakiej nazwie) odpowiada na wysyłane z komputera 1A żądania ICMP echo (wykorzystując adres IP źródłowy z odpowiedzi):  
       ***Router R2 Central***

Wskaż jaki typ wiadomości ICMP jest przesyłany w informacji zwrotnej do nadawcy:  
***Typ: 3 – Cel nieosiągalny***

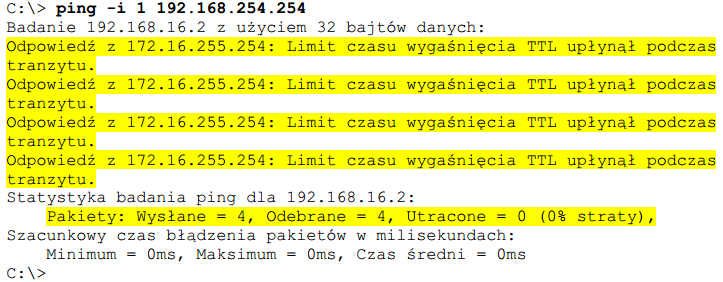
Wskaż jaki kod jest związany z tym typem komunikatu:  
***Kod: 1 - Host nieosiągalny***

Wskaż, dlaczego to takie a nie inne urządzenie zwróciło odpowiedź na wysłane zapytanie *echo request:*

***Odpowiedź zwraca Router R2 Central, ponieważ host docelowy jest spoza sieci lokalnej urządzenia wysyłającego zapytanie ping. Zatem jest ono kierowane do routera, a ten nie znalazł odpowiedniego adresata i odesłał komunikat o błędzie.***

Krok 3:

W tym kroku program Wireshark będzie użyty do przechwycenia i przeanalizowania wiadomości ICMP typu ***echo request/echo reply*** z wykorzystaniem aplikacji **ping**, przy czym wysyłane do serwera Eagle, ale z wartością TTL wysyłanych pakietów ustawioną na „1”.

* + 1. Otworzyć z wykorzystaniem aplikacji Wireshark plik „***ping\_with\_low\_TTL\_to\_Eagle\_Server.pcap***”
    2. W ramach przechwyconego ruchu sieciowego jaki się ukaże po otwarciu pliku, wykonane zostało polecenie ***ping*** z komputera ***1A*** na adres IP serwera ***Eagle***, przy czym z wartością TTL wysyłanych pakietów ustawioną na „1” (wykorzystując przełącznik „**-i 1**”). Na poniższym zdjęciu został zawarty wynika wydanego w/w polecenia w wierszu poleceń komputera ***1A***:

Wskaż jakie jest przeznaczenie/cel stosowania wartości TTL umieszczanej w nagłówkach pakietu IP:

***By źle zaadresowane pakiety danych nie tworzyły niepotrzebnego ruchu sieciowego, poprzez problemy z osiągnięciem adresu docelowego.***

* + 1. Analizując przechwycony ruch sieciowy w aplikacji Wireshark (dla ułatwienia analizy należy wykorzystać filtr: icmp), wskaż, które z urządzeń sieciowych (tj. urządzenie o jakiej nazwie) odpowiada na wysyłane z komputera 1A żądania ICMP echo (wykorzystując adres IP źródłowy z odpowiedzi):  
       ***Router R2 Central***

Wskaż jaki typ wiadomości ICMP jest przesyłany w informacji zwrotnej do nadawcy:  
***Typ: 11 – Time To Live exceeded***

Wskaż jaki kod jest związany z tym typem komunikatu:  
***Kod: 0 – Time to live exceeded in transit***

Wskaż jakie urządzenie sieciowe jest odpowiedzialne za zmniejszanie wartości TTL?

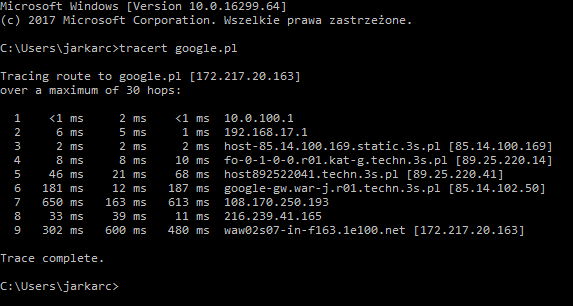
***Router***

Wskaż, dlaczego to takie a nie inne urządzenie zwróciło odpowiedź na wysłane zapytanie *echo request:*

***Zapytanie wykraczało poza pulę adresową sieci lokalnej, a więc zostało skierowane  
do routera R2 Central, który wykrył zbyt małą liczbę TTL.***

Krok 4:

W tym kroku program Wireshark będzie użyty do przechwycenia i przeanalizowania połączeni zrealizowanego z wykorzystaniem aplikacji **tracert**.

* + 1. Otwórz okno linii poleceń Windows na komputerze i uruchom przechwytywanie ramek w aplikacji Wireshark.
    2. Z linii poleceń Windows wyślij **tracert** do serwera **google.pl**. Powinny się pojawiać dane kolejnych węzłów (ruterów) na trasie, przez które przechodzi transmisja danych, np.:   
       

Poczekaj do końca działania polecenia i po zakończeniu (jak się pojawi wyrażenie „Trace complete”), zatrzymaj przechwytywanie w programie Wireshark.  
Jeżeli zamiast pozycji pojawiają się same gwiazdki w kolejnych pozycjach to oznacza, że komunikacja z urządzeniem nie jest aktualnie możliwe, ze względu na problemy z komunikacją sieciową (można spróbować zrealizować połączenie z innym adresem, a jeżeli z żadnym nie działa, to znaczy, że należy zweryfikować możliwość połączenia z siecią Internet z urządzenia na którym się pracuje).

* + 1. Zbadaj przechwycone komunikaty ICMP i wskaż na bazie ich analizy, w jaki sposób  
       komenda **tracert** śledzi ścieżkę do miejsca docelowego:

***Komenda tracert wysyła zapytanie z coraz większą liczbą TTL, tak dużą, jaka pozwala osiągnąć cel docelowy. TTL jest zwiększane o 1, począwszy od wartości 1. Gdy TTL  
się kończy, router, który musi usunąć pakiet z obiegu w sieci wysyła wiadomość zwrotną do nadawcy z informacją o osiągnięciu TTL = 0, dzięki czemu otrzymujemy adres ostatniego routera.***