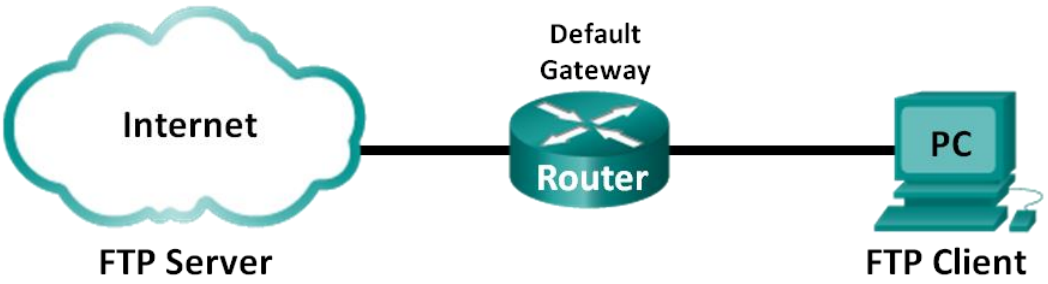
Laboratorium - Użycie z programu Wireshark do przechwytywania danych pochodzących z protokołu FTP.

Topologia – FTP

W tym ćwiczeniu omówimy przechwytywanie danych TCP w sesji FTP . Topologia składa się z komputera z dostępem do Internetu.



Cele

Identyfikacja pól w nagłówku TCP oraz operacji przychwytywania przez Wireshark sesji FTP

Scenariusz

Dwa protokoły w warstwie transportowej TCP/IP, to TCP zdefiniowane w RFC 761 oraz UDP zdefiniowane w RFC 768. Oba protokoły obsługują komunikację protokołów wyższych warstw. Na przykład, TCP jest używany do obsługi warstwy transportowej między innymi, dla protokołów Hypertext Transfer Protocol (HTTP) i FTP. Protokół UDP działa w warstwie transportowej i współpracuje między innymi z protokołami Domain Name System (DNS).

Uwaga: Zrozumienie zawartości nagłówków TCP i UDP oraz ich działania jest bardzo istotne dla inżynierów sieciowych.

W laboratorium będziesz używał programu typu Open Source Wireshark w celu przechwytywania i analizowania pól nagłówka protokołu TCP dla sesji FTP przesyłania plików pomiędzy komputerem hosta i anonimowym serwerem FTP. Aby połączyć się z anonimowym serwerem FTP i pobrać plik będzie używany Wiersz poleceń systemu Windows.

Wymagane zasoby - FTP

1 PC (Windows 7, Vista, lub XP z dostępem do wiersza poleceń, dostępem do Internetu i zainstalowanym programem Wireshark)

Część 1: Identyfikacja pól w nagłówku segmentu TCP oraz obserwacja działania w sesji FTP za pomocą programu Wireshark.

W części 1 należy użyć programu Wireshark do przechwytywania sesji FTP i sprawdzenia pól nagłówka TCP.

Krok 1: W programie Wireshark uruchom przechwytywanie.

a. Zamknij wszystkie zbędne komunikacje w sieci, takie jak na przykład przeglądarki WWW, aby ograniczyć ilość ruch podczas przechwytywania pakietów w programie Wireshark.

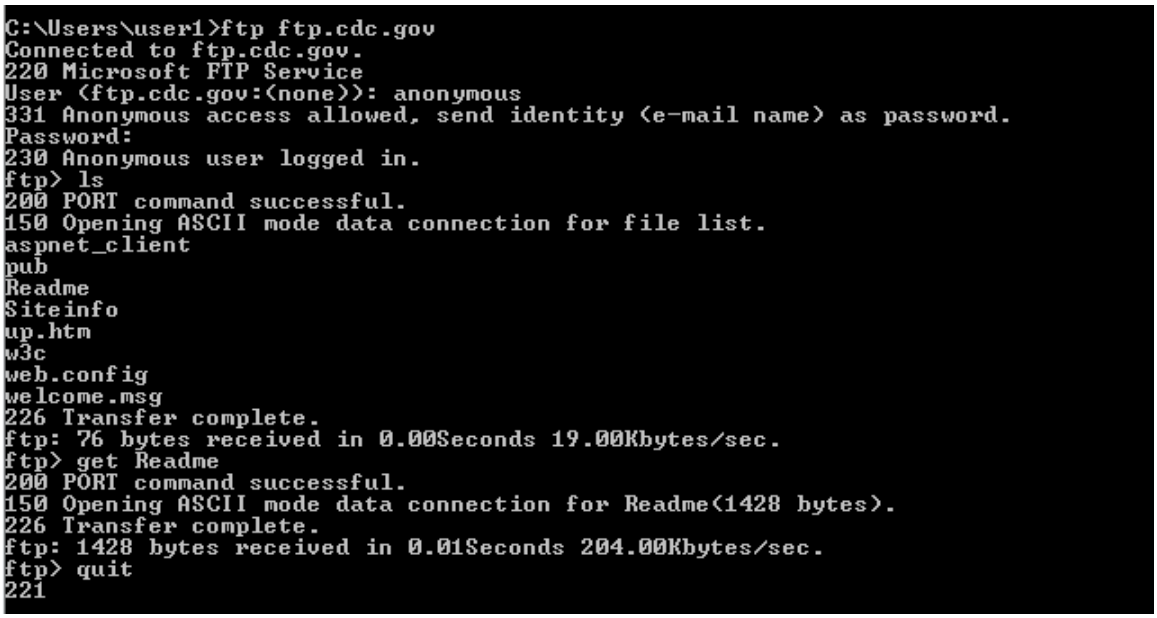
b. W programie Wireshark uruchom przechwytywanie.

Krok 2: Pobierz plik Readme.

a. W wierszu poleceń wpisz **ftp ftp.cdc.gov**.

b. Zaloguj się na stronę FTP Centers for Disease Control and Prevention (CDC) używając konta **anonymous** bez hasła.

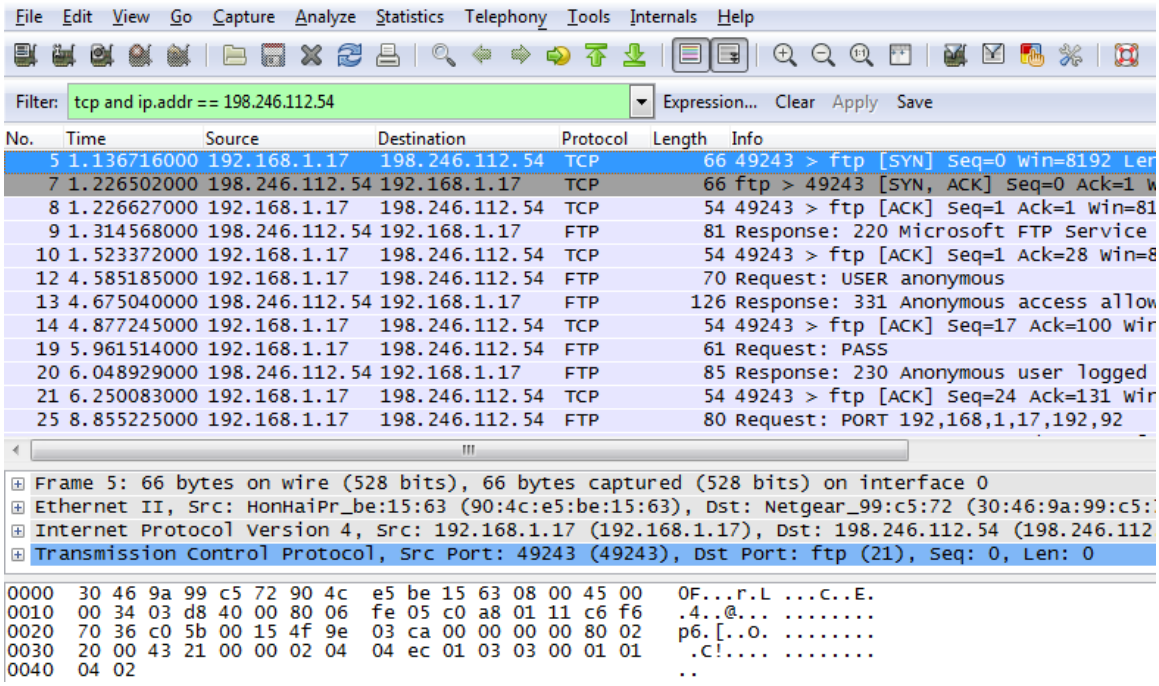
c. Znajdź i pobierz plik Readme.



Krok 3: Zatrzymaj przechwytywanie w programie Wireshark.

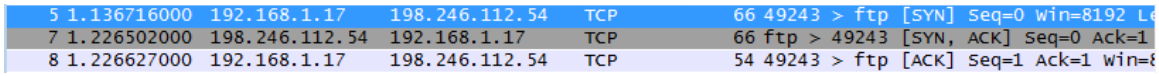
Krok 4: Przejdź do okna głównego programu Wireshark.

Wireshark przechwycił wiele pakietów w trakcie sesji FTP do strony ftp.cdc.gov. Aby ograniczyć ilość danych do analizy, w polu **Filter: entry** wpisz **tcp oraz ip.addr == 198.246.112.54** i kliknij **Apply**. Adres IP 198.246.112.54 jest adresem strony ftp.cdc.gov.



Krok 5: Przeanalizuj zawartości pól TCP.

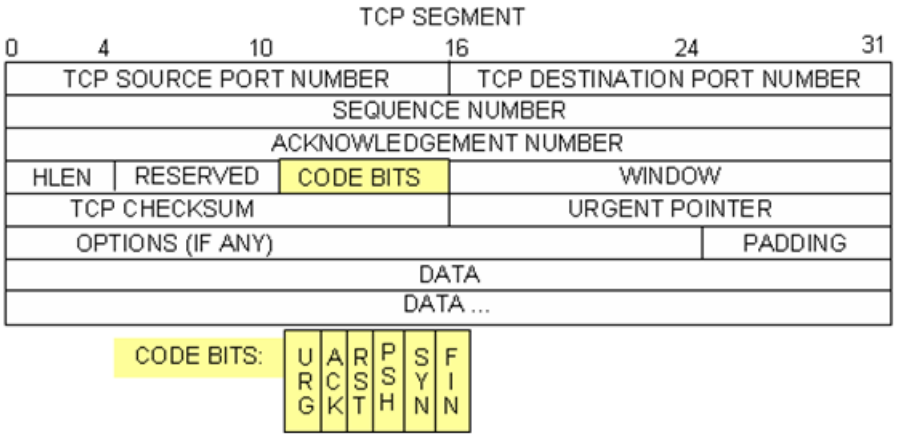
Po zastosowaniu filtru TCP pierwsze trzy ramki w okienku listy pakietów (górna sekcja) wyświetla w warstwie transportowej protokół TCP służący do tworzenia niezawodnej sesji. Sekwencja [SYN], [SYN, ACK], [ACK] ilustruje 3-etapowe uzgodnienie.



TCP jest rutynowo używany podczas sesji do kontroli dostarczenia datagramu, weryfikacji jego dotarcia i zarządzania rozmiarem okna. Dla każdej wymiany danych pomiędzy klientem a serwerem FTP jest uruchomiana nowa sesja TCP. Na zakończenie wymiany danych sesja TCP jest zamykana. Gdy sesja FTP jest zakończona, to TCP wykonuje procedurę zamknięcia i zakończenia połączenia.

Szczegółowe informacje na temat TCP są dostępne w panelu szczegółów pakietów programu Wireshark (sekcja środkowa). Podświetl pierwszy datagram TCP z komputera hosta i rozwiń rekord TCP. Rozwinięty datagram TCP wydaje się być podobny do okienka szczegółów pakietu (packet detail) widocznego poniżej.

```
⊞ Frame 5: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface 0
⊞ Ethernet II, Src: HonHaiPr_be:15:63 (90:4c:e5:be:15:63), Dst: Netgear_99:c5:72 (30:46:9a:99:c5:72)
⊞ Internet Protocol Version 4, Src: 192.168.1.17 (192.168.1.17), Dst: 198.246.112.54 (198.246.112.54)
⊟ Transmission Control Protocol, Src Port: 49243 (49243), Dst Port: ftp (21), Seq: 0, Len: 0
    Source port: 49243 (49243)
    Destination port: ftp (21)
    [Stream index: 0]
    Sequence number: 0    (relative sequence number)
    Header length: 32 bytes
  ⊟ Flags: 0x002 (SYN)
      000. .... .... = Reserved: Not set
      ...0 .... .... = Nonce: Not set
      .... 0... .... = Congestion Window Reduced (CWR): Not set
      .... .0.. .... = ECN-Echo: Not set
      .... ..0. .... = Urgent: Not set
      .... ...0 .... = Acknowledgment: Not set
      .... .... 0... = Push: Not set
      .... .... .0.. = Reset: Not set
    ⊞ .... .... ..1. = Syn: Set
      .... .... ...0 = Fin: Not set
    Window size value: 8192
    [Calculated window size: 8192]
  ⊞ Checksum: 0x4321 [validation disabled]
  ⊞ Options: (12 bytes), Maximum segment size, No-Operation (NOP), Window scale, No-Operation (NOP), Nc
```



Powyższy rysunek przedstawia schemat datagramu TCP. Opis każdego pola znajduje się w dokumencie:

- **Numer portu źródłowego TCP** przypisany jest do sesji TCP hosta, który otworzył połączenie. Liczba ta jest zwykle wartością losową powyżej 1023.
- **Numer portu docelowego** jest używany w celu określenia protokołu warstwy wyższej bądź aplikacji na komputerze docelowym (serwerze). Wartości z zakresu 0-1023 reprezentują "dobrze znane porty" i związane są z popularnymi usługami i aplikacjami (w sposób opisany w dokumencie RFC 1700, takimi jak Telnet, FTP, HTTP oraz innymi). Kombinacja czterech wartości (źródłowego adresu IP, źródłowego numeru portu, docelowego adresu IP, docelowego numeru portu) identyfikuje w sposób unikalny sesję dla obu hostów: klienta i serwera.

Uwaga: W programie Wireshark przechwycony poniżej port docelowy to 21, co oznacza że jest to FTP. Serwery FTP na porcie 21 nasłuchują połączenia od klienta FTP.

- **Numer sekwencyjny** określa numer ostatniego oktetu w segmencie.
- **Numer potwierdzenia** określa numer następnego oktetu oczekiwanego przez odbiorcę.
- **Bity kontrolne** (flagi) mają specjalne znaczenie w zarządzaniu sesją i w określaniu sposobu traktowania segmentów. Wśród nich wyróżniamy:

  - ACK — bit/flaga potwierdzenia otrzymania segmentu,

- SYN — bit/flaga synchronizacji, ustawiona tylko wtedy, gdy nowa sesja jest negocjowana podczas trójetapowego uzgadniania,
- FIN — bit/flaga zakończenia, która oznacza żądanie zamknięcia sesji.

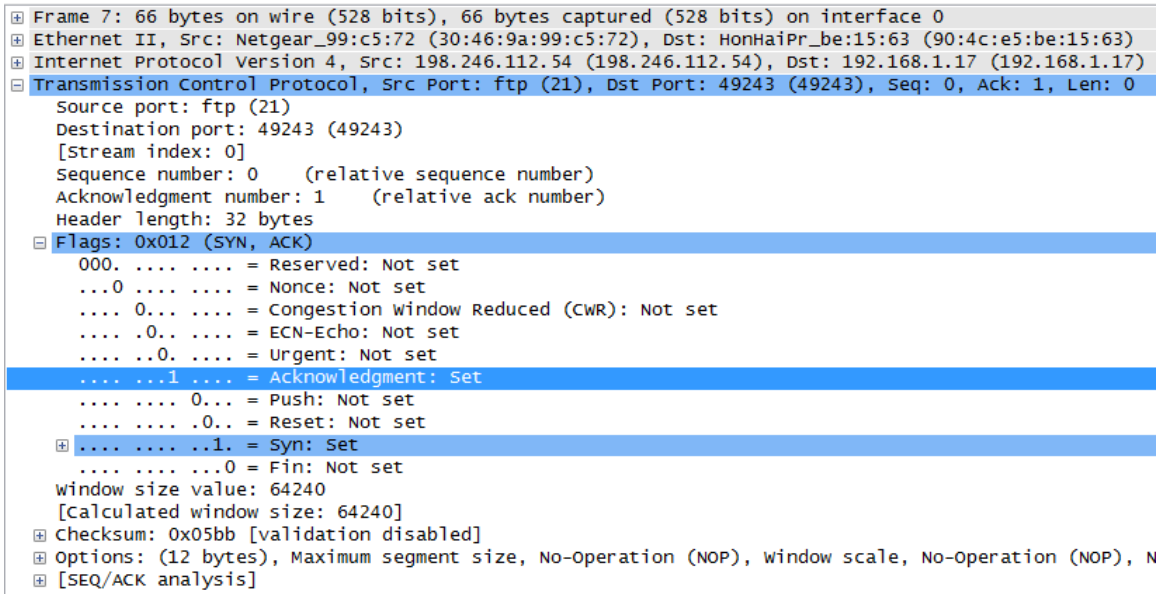
- **Rozmiar okna** to wartość rozmiaru okna przesuwnego, oznaczająca ile oktetów może być przesłanych zanim nadawca będzie musiał czekać na potwierdzenie.
- **Wskaźnik Urgent** jest używany tylko z flagą Urgent (URG), gdy nadawca musi wysłać pilne dane do odbiornika.
- **Options** ma obecnie tylko jedną możliwość i jest określona jako maksymalna wielkość segmentu TCP (wartość opcjonalna).

Używając programu Wireshark przechwyć pierwszą fazę w sesji TCP (flaga SYN ustawiona na 1) i wypełnij informację o nagłówku datagramu TCP:

Z komputera PC do serwera CDC (tylko flaga SYN jest ustawiona na 1):

Adres IP nadawcy:	
Adres docelowy IP:	
Numer portu źródłowego:	
Numer portu docelowego	
Numer sekwencyjny:	
Numer potwierdzenia:	
Długość nagłówka:	
Rozmiar okna:	

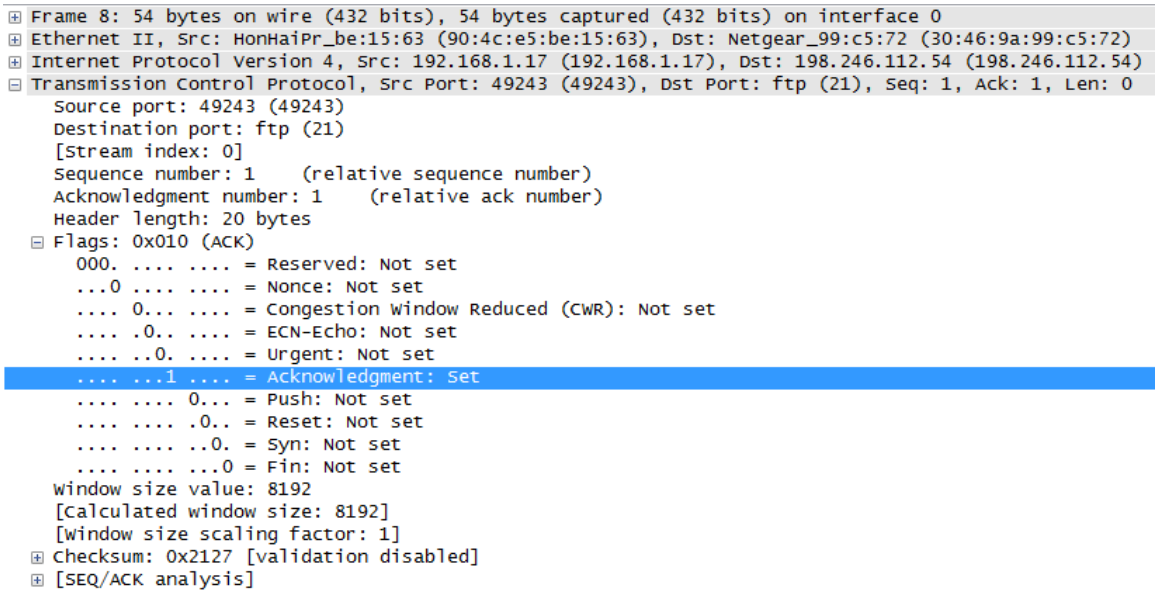
W drugim filtrowanym przechwytywaniu Wiresharka serwer CDC FTP potwierdza żądanie z komputera PC. Zanotuj wartości bitów SYN i ACK.



Wypełnij następujące informacje dotyczące wiadomości SYN-ACK.

Źródłowy adres IP:	
Adres docelowy IP:	
Numer portu źródłowego:	
Numer portu docelowego:	
Numer sekwencyjny:	
Numer potwierdzenia:	
Długość nagłówka:	
Rozmiar okna:	

W końcowej fazie negocjacji w celu nawiązania komunikacji, komputer wysyła do serwera komunikat potwierdzający. Zauważ, że tylko bit ACK jest ustawiony na 1, a numer sekwencyjny został zwiększony do 1.



Wypełnij następujące informacje dotyczące wiadomości ACK.

Źródłowy adres IP:	
Adres docelowy IP:	
Numer portu źródłowego:	
Numer portu docelowego:	
Numer sekwencyjny:	
Numer potwierdzenia:	
Długość nagłówka:	
Rozmiar okna:	

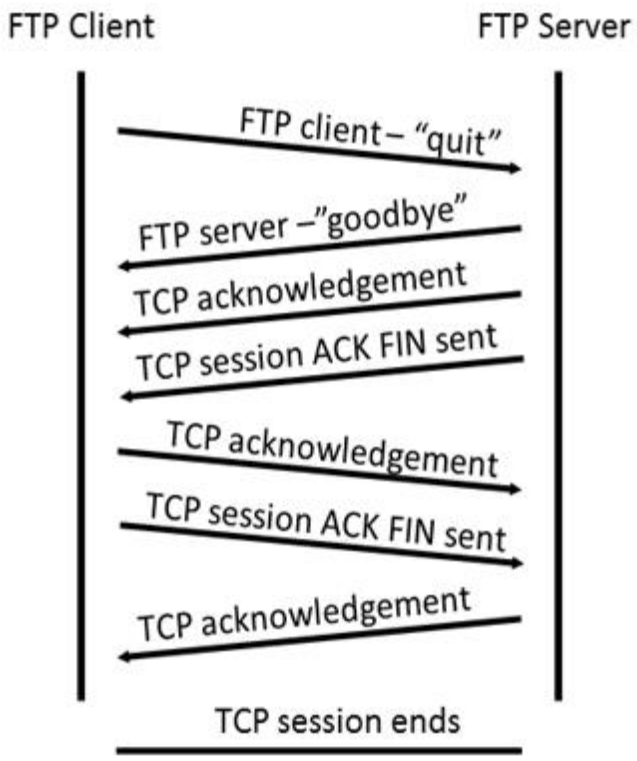
Ile innych datagramów TCP zawiera bit SYN?

Po ustanowieniu sesji TCP, może wystąpić ruch FTP pomiędzy komputerem PC i serwerem FTP. Klient i serwer FTP komunikują się ze sobą nie wiedząc, że TCP kontroluje i zarządza nawiązaną przez nich sesją. Gdy serwer FTP wysyła odpowiedź: 220 do klienta FTP, to sesja TCP w kliencie FTP wysyła potwierdzenie do sesji TCP na serwerze. Tą sekwencję można przechwycić i obejrzeć w programie Wireshark.

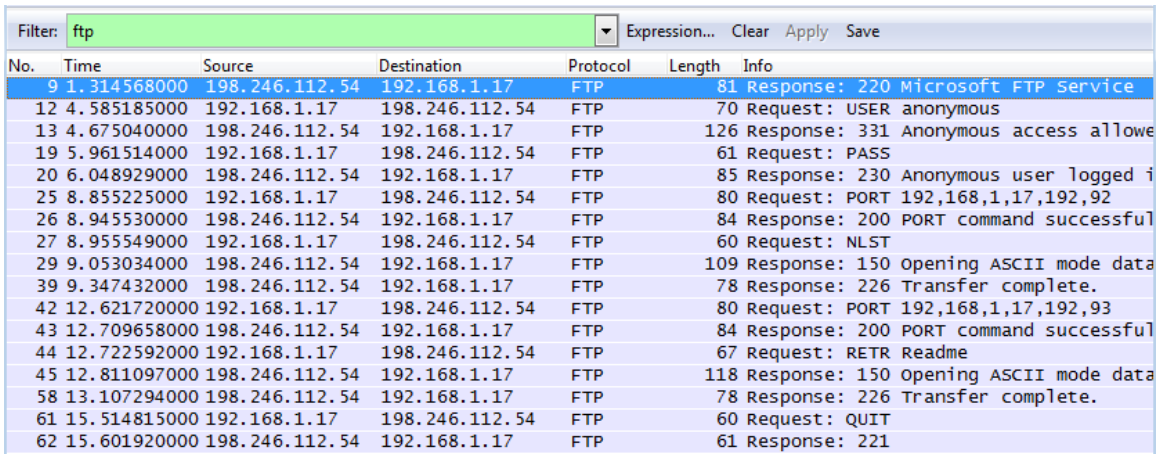
```
 9 1.314568000  198.246.112.54  192.168.1.17     FTP          81 Response: 220 Microsoft FTP Service
10 1.523372000  192.168.1.17    198.246.112.54   TCP          54 49243 > ftp [ACK] Seq=1 Ack=28 Win=
12 4.585185000  192.168.1.17    198.246.112.54   FTP          70 Request: USER anonymous
13 4.675040000  198.246.112.54  192.168.1.17     FTP         126 Response: 331 Anonymous access allo
```

```
⊞ Frame 9: 81 bytes on wire (648 bits), 81 bytes captured (648 bits) on interface 0
⊞ Ethernet II, Src: Netgear_99:c5:72 (30:46:9a:99:c5:72), Dst: HonHaiPr_be:15:63 (90:4c:e5:be:15:63)
⊞ Internet Protocol Version 4, Src: 198.246.112.54 (198.246.112.54), Dst: 192.168.1.17 (192.168.1.17)
⊞ Transmission Control Protocol, Src Port: ftp (21), Dst Port: 49243 (49243), Seq: 1, Ack: 1, Len: 27
⊟ File Transfer Protocol (FTP)
  ⊟ 220 Microsoft FTP Service\r\n
      Response code: Service ready for new user (220)
      Response arg: Microsoft FTP Service
```

Gdy połączenie FTP jest zakończone, to klient FTP wysyła komendę "quit". Serwer FTP potwierdza zakończenie połączenia FTP za pomocą odpowiedzi: 221 Goodbye. W tym momencie sesja TCP w serwerze FTP wysyła datagram TCP do klienta FTP, ogłaszający zakończenie sesji TCP. Sesja TCP na kliencie FTP potwierdza otrzymanie datagramu kończącego sesję i wysyła własny datagram TCP kończący sesję. Gdy źródło zakończenia sesji TCP, serwer FTP otrzyma podwójne zakończenie, datagram z ustawionym bitem ACK jest wysyłany aby potwierdzić zakończenie sesji TCP i sesja TCP jest zamknięta. Tą sekwencję można przechwycić i obejrzeć na diagramie.



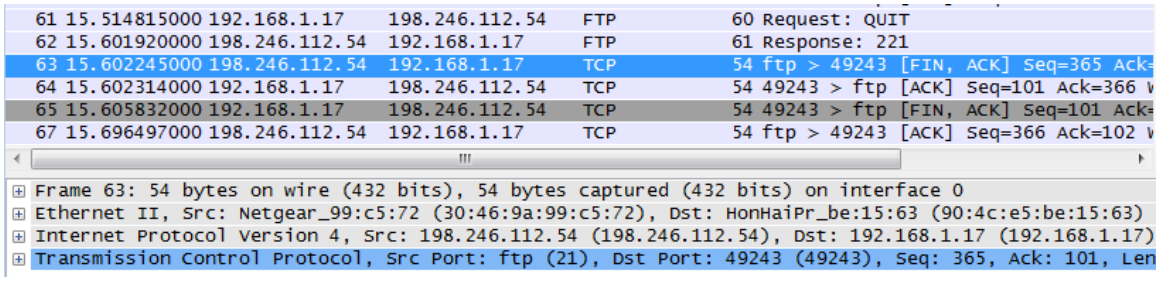
Dzięki zastosowaniu filtru **ftp**, cała sekwencja ruchu FTP może być badana w programie Wireshark. Zwróć uwagę na kolejność występowania zdarzeń podczas tej sesji FTP. Do pobrania pliku Readme użyto nazwy użytkownika anonymous. Po zakończeniu transferu plików użytkownik zakończył sesję FTP.



Zastosuj ponownie filtr TCP w Wireshark aby zbadać zakończenie sesji TCP. Cztery pakiety są transmitowane dla zakończenia sesji TCP. Ponieważ połączenie TCP jest typu full-duplex, to każda strona musi samodzielnie dokonać zakończenia. Sprawdź adresy źródłowe i docelowe.

W tym przykładzie serwer FTP nie ma już danych do wysłania w strumieniu; serwer wysyła segment z ustawioną flagą FIN w ramce 63. PC wysyła ACK, aby potwierdzić otrzymanie FIN w celu zakończenia sesji z serwera do klienta w ramce 64.

W ramce 65 komputer PC wysyła FIN do serwera FTP, aby zakończyć sesję TCP. Serwer FTP odpowiada za pomocą ACK, aby potwierdzić FIN przychodzący z komputera PC w ramce 67. Teraz sesja TCP między serwerem FTP i PC jest zakończona.



Wyzwanie

Ponieważ FTP nie są bezpiecznym protokołem, to wszystkie dane przesyłane za jego pomocą wysyłane są otwartym tekstem. Dotyczy to także ID użytkowników, haseł i zawartości plików tekstowych nieszyfrowanych. Analizując sesję FTP szybko odnajdziemy ID użytkownika, hasło a także hasła w plikach konfiguracyjnych.