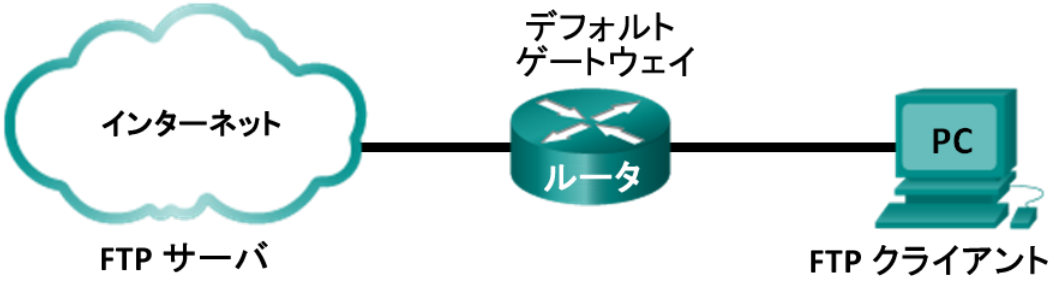
実習 - Wireshark を使用して FTP および TFTP キャプチャを調べる

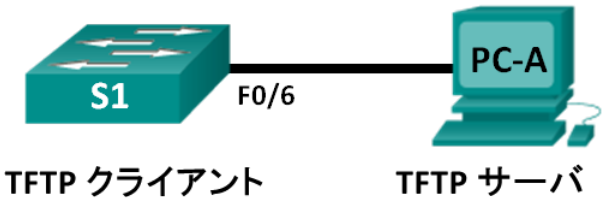
トポロジ - パート 1（FTP）

パート 1 では、FTP セッションの TCP キャプチャに焦点を合わせます。このトポロジは、インターネットにアクセスできる PC から成ります。



トポロジ - パート 2（TFTP）

パート 2 では、TFTP セッションの UDP キャプチャに焦点を合わせます。この PC には、スイッチ S1 へのコンソール接続とイーサネット接続の両方がなければなりません。



アドレッシング テーブル（パート 2）

デバイス	インターフェイス	IP アドレス	サブネット マスク	デフォルト ゲートウェイ
S1	VLAN 1	192.168.1.1	255.255.255.0	該当なし
PC-A	NIC	192.168.1.3	255.255.255.0	192.168.1.1

目的

パート 1：Wireshark の FTP セッション キャプチャを使用して TCP ヘッダー フィールドと動作を識別する

パート 2：Wireshark の TFTP セッション キャプチャを使用して UDP ヘッダー フィールドと動作を識別する

背景/シナリオ

TCP/IP トランスポート層の 2 つのプロトコルは、RFC 761 で定義されている TCP と、RFC 768 で定義されている UDP です。どちらのプロトコルも上位層プロトコルの通信をサポートしています。たとえば、TCP は HTTP（HyperText Transfer Protocol）や FTP プロトコルに対してトランスポート層サポートを提供します。UDP は DNS（Domain Name System）や TFTP に対してトランスポート層サポートを提供します。

注:TCP ヘッダーと UDP ヘッダーの役割と動作を理解することは、ネットワーク エンジニアにとって重要なスキルです。

この実習のパート 1 では、Wireshark オープン ソース ツールを使用して、ホスト コンピュータと anonymous FTP サーバとの間での FTP ファイル転送に関係する TCP プロトコルのヘッダー フィールドをキャプチャして分析します。anonymous FTP サーバへの接続とファイルのダウンロードには、Windows のコマンド ライン ユーティリティを使用します。この実習のパート 2 では、Wireshark を使用して、ホスト コンピュータとスイッチ S1 との間での TFTP ファイル転送に関係する UDP プロトコルのヘッダー フィールドをキャプチャして分析します。

注:使用するスイッチは、Cisco IOS リリース 15.0(2)(lanbasek9 イメージ)を搭載した Cisco catalyst 2960 です。他のスイッチおよび Cisco IOS バージョンを使用できます。モデルと Cisco IOS バージョンによっては、使用できるコマンドと生成される出力が、実習で表示されるものと異なる場合があります。

注:スイッチがクリアされて、スタートアップ コンフィギュレーションが存在しないことを確認してください。不明な場合は、インストラクタに相談してください。

注:パート 1 では、PC がインターネットにアクセスできることを前提にしています。また、Netlab を使用してパート 1 を実行することはできません。パート 2 は Netlab 互換です。

実習に必要なリソースや機器 - パート 1(FTP)

PC 1 台(インターネットとコマンド プロンプトを利用できて Wireshark がインストールされている Windows 7、Vista、または XP 搭載 PC)

実習に必要なリソースや機器 - パート 2(TFTP)

- スイッチ 1 台(Cisco IOS リリース 15.0(2) の lanbasek9 イメージを搭載した Cisco 2960 または同等機器)
- PC 1 台(Wireshark および tftpd32 などの TFTP サーバがインストールされている Windows 7、Vista、または XP 搭載 PC)
- コンソール ポート経由で Cisco IOS デバイスを設定するためのコンソール ケーブル
- トポロジに示すようなイーサネット ケーブル

パート 1: Wireshark の FTP セッション キャプチャを使用した TCP ヘッダー フィールドと動作の識別

パート 1 では、Wireshark を使用して FTP セッションをキャプチャし、TCP ヘッダー フィールドを詳しく調べます。

手順 1: Wireshark のキャプチャを開始します。

a. Wireshark のキャプチャ時におけるトラフィックの量を抑えるために、Web ブラウザなどの不要なネットワーク トラフィックをすべて閉じます。

b. Wireshark のキャプチャを開始します。

手順 2: Readme ファイルをダウンロードします。

a. コマンド プロンプトで「**ftp ftp.cdc.gov**」と入力します。

b. Centers for Disease Control and Prevention(CDC; 米国疾病管理予防センター)の FTP サイトにログインします。ただし、ユーザ名として **anonymous** を使用し、パスワードは使用しません。

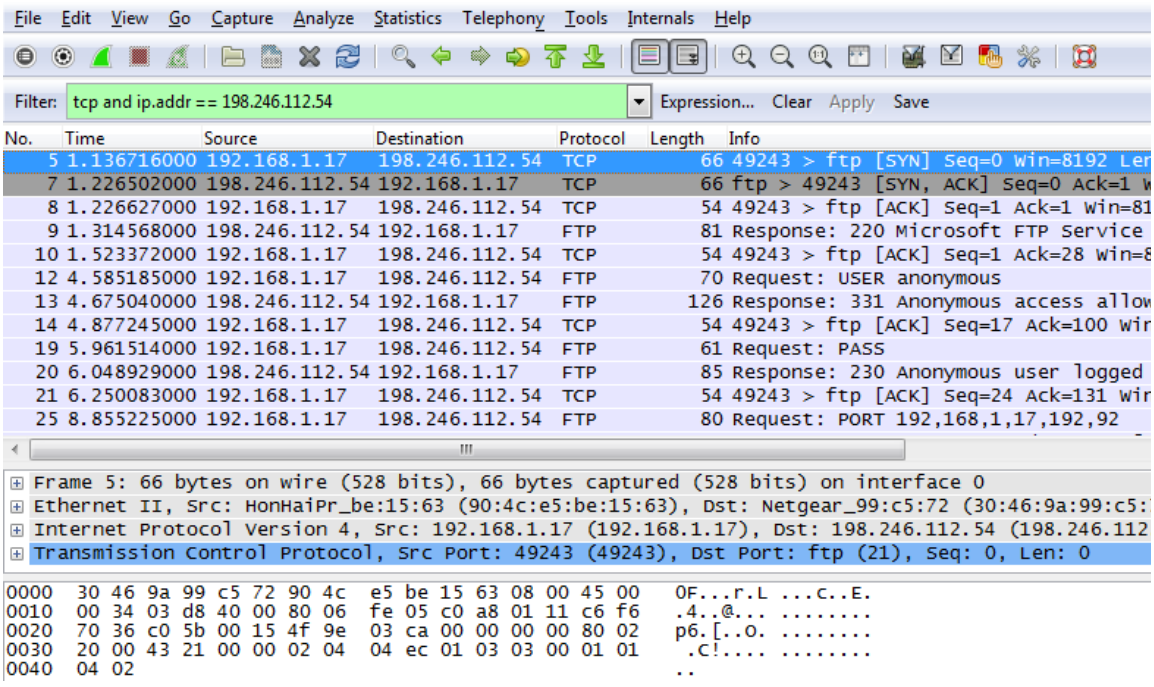
c. Readme ファイルを見つけてダウンロードします。

```
C:\Users\user1>ftp ftp.cdc.gov
Connected to ftp.cdc.gov.
220 Microsoft FTP Service
User (ftp.cdc.gov:(none)): anonymous
331 Anonymous access allowed, send identity (e-mail name) as password.
Password:
230 Anonymous user logged in.
ftp> ls
200 PORT command successful.
150 Opening ASCII mode data connection for file list.
aspnet_client
pub
Readme
Siteinfo
up.htm
w3c
web.config
welcome.msg
226 Transfer complete.
ftp: 76 bytes received in 0.00Seconds 19.00Kbytes/sec.
ftp> get Readme
200 PORT command successful.
150 Opening ASCII mode data connection for Readme(1428 bytes).
226 Transfer complete.
ftp: 1428 bytes received in 0.01Seconds 204.00Kbytes/sec.
ftp> quit
221
```

手順 3:　Wireshark のキャプチャを停止します。

手順 4:　Wireshark のメイン ウィンドウを表示します。

Wireshark は ftp.cdc.gov に対する FTP セッション中に多数のパケットをキャプチャしています。分析用にデータ量を制限するため、[**Filter:**] の入力領域に「**tcp and ip.addr == 198.246.112.54**」と入力し、[**Apply**] をクリックします。IP アドレス 198.246.112.54 は、ftp.cdc.gov のアドレスです。



手順 5:　TCP のフィールドを分析します。

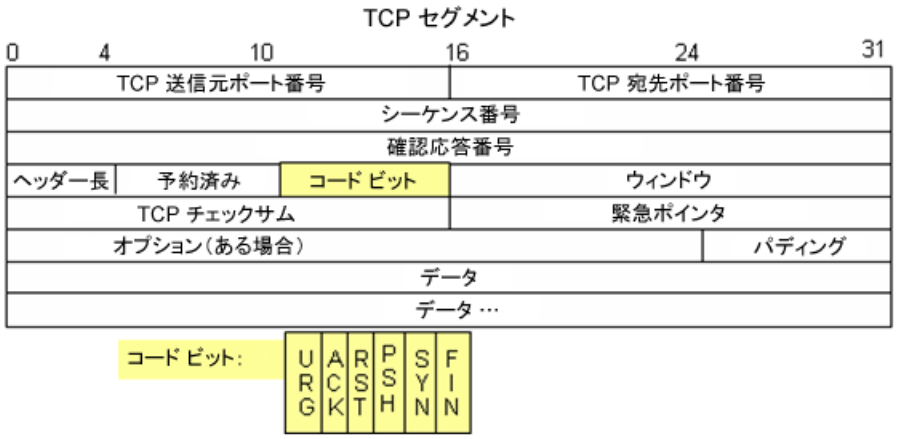
TCP フィルタが適用された後、パケット リスト ペイン（上部のセクション）の最初の 3 つのフレームには、信頼性の高いセッションを作成する TCP トランスポート層プロトコルが表示されます。[SYN]、[SYN, ACK]、[ACK] というシーケンスは、3 ウェイ ハンドシェイクを示しています。

```
    5 1.136716000  192.168.1.17    198.246.112.54   TCP        66 49243 > ftp [SYN] Seq=0 Win=8192 Le
    7 1.226502000  198.246.112.54  192.168.1.17     TCP        66 ftp > 49243 [SYN, ACK] Seq=0 Ack=1
    8 1.226627000  192.168.1.17    198.246.112.54   TCP        54 49243 > ftp [ACK] Seq=1 Ack=1 Win=8
```

セッション中にデータグラムの配信を制御し、データグラムの到達を確認し、ウィンドウ サイズを管理するために、決まって TCP が使用されます。FTP クライアントと FTP サーバの間でのデータ交換のたびに、新しい TCP セッションが開始されます。データ転送が終わると、TCP セッションが閉じられます。そして FTP セッションの終了時に、TCP が整然としたやり方でシャットダウンと終了を行います。

Wireshark では、パケット詳細ペイン（中央のセクション）に詳細な TCP 情報が表示されます。ホスト コンピュータからの最初の TCP データグラムを強調表示にし、TCP レコードを展開してください。展開されたデータグラムは、次に示すパケット詳細ペインと同じような状態になります。

```
⊞ 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
```



上の図は TCP データグラム図です。以下は各フィールドの説明です

- **TCP source port number**（TCP 送信元ポート番号）は、接続を開いた TCP セッションのホストに属します。通常、この値は 1,023 より大きいランダムな値です。
- **TCP destination port number**（TCP 宛先ポート番号）は、リモート サイトの上位層プロトコルまたはアプリケーションを識別するために使用されます。0 ～ 1,023 の範囲の値は「既知の（well-known）ポート」を表し、よく知られたサービスやアプリケーションと関連付けられています（これらは RFC 1700 で規定されており、Telnet、FTP、HTTP などがあります）。送信元 IP アドレス、送信元ポート、宛先 IP アドレス、宛先ポートの組み合わせは、送信側と受信側の両方へのセッションを一意に識別します。

注: 下の Wireshark キャプチャでは、宛先ポートが 21 になっています。これは FTP です。FTP サーバは、ポート 21 で FTP クライアント接続をリッスンします。

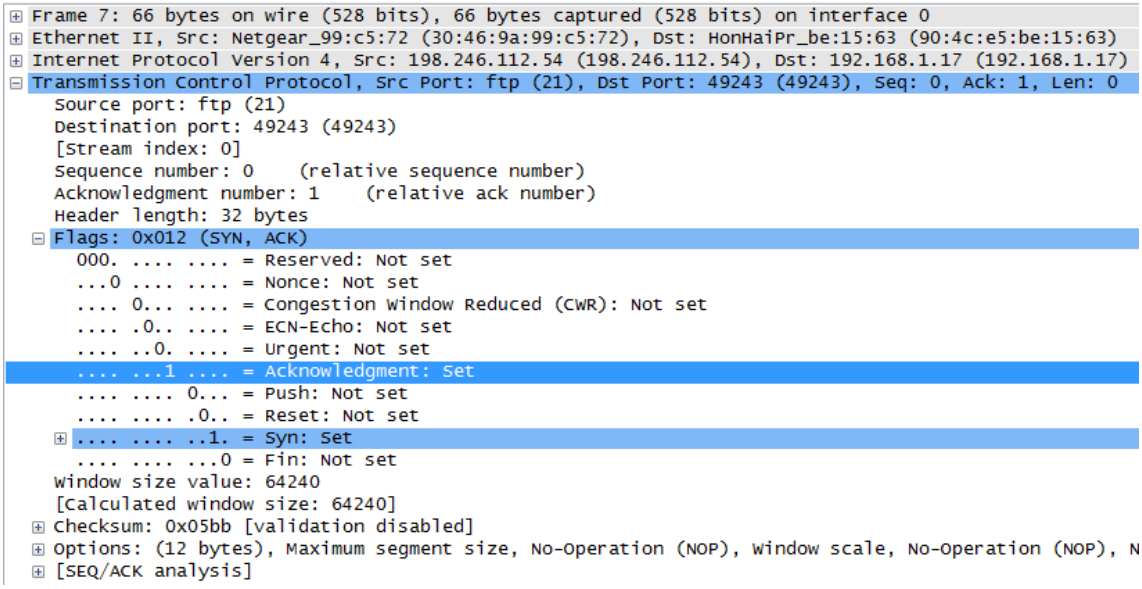
- **Sequence number**（シーケンス番号）は、セグメント内の最後のオクテットの番号を表します。
- **Acknowledgment number**（確認応答番号）は、受信側が期待する次のオクテットを表します。
- **Code bits**（コード ビット）は、セッション管理とセグメントの処理に関して特別な意味を持っています。それぞれのビットは、たとえば次のものを表します。
  - ACK - セグメントを受信したという確認応答。
  - SYN - 同期。TCP 3 ウェイ ハンドシェイクで新しい TCP セッションをネゴシエートするときにのみセットされます。
  - FIN - 終了。TCP セッションの終了を要求します。
- **Window size**（ウィンドウ サイズ）は、スライディング ウィンドウの値です。確認応答が返ってくる前にオクテットを何個まで送信できるかを決定するものです。
- **Urgent pointer**（緊急ポインタ）は、送信側が受信側に緊急データを送信する必要があるときにのみ緊急（URG）フラグとともに使用されます。
- **Options**（オプション）としては、現在のところ 1 つのオプションしかありません。これは最大 TCP セグメント サイズ（省略可能な値）として定義されています。

最初の TCP セッション スタートアップ（SYN ビットを 1 に設定）の Wireshark キャプチャを使用して、TCP ヘッダーに関する情報を記入してください。

PC から CDC サーバ（SYN ビットのみ 1 に設定）:

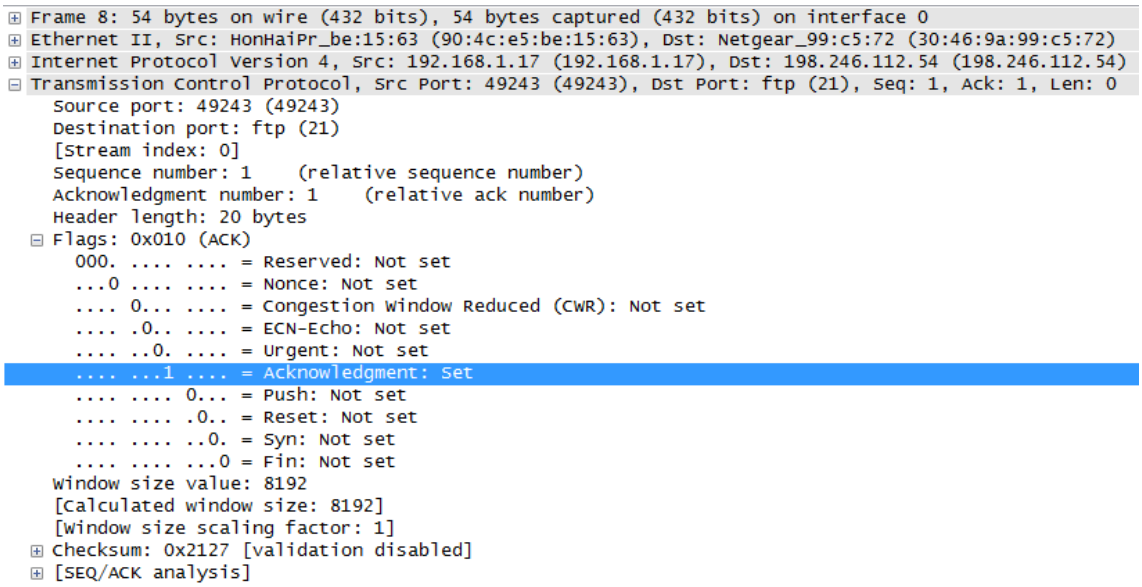
送信元 IP アドレス:	
宛先 IP アドレス:	
送信元ポート番号:	
宛先ポート番号:	
シーケンス番号:	
確認応答番号:	
ヘッダー長:	
ウィンドウ サイズ:	

フィルタリングされた 2 番目の Wireshark キャプチャでは、CDC FTP サーバが PC からの要求に対して確認応答を返しています。SYN ビットと ACK ビットの値を書き留めてください。



SYN-ACK メッセージに関して次の情報を記入してください。

送信元 IP アドレス:	
宛先 IP アドレス:	
送信元ポート番号:	
宛先ポート番号:	
シーケンス番号:	
確認応答番号:	
ヘッダー長:	
ウィンドウ サイズ:	



ACK メッセージに関して次の情報を記入してください。

送信元 IP アドレス:	
宛先 IP アドレス:	
送信元ポート番号:	
宛先ポート番号:	
シーケンス番号:	
確認応答番号:	
ヘッダー長:	
ウィンドウ サイズ:	

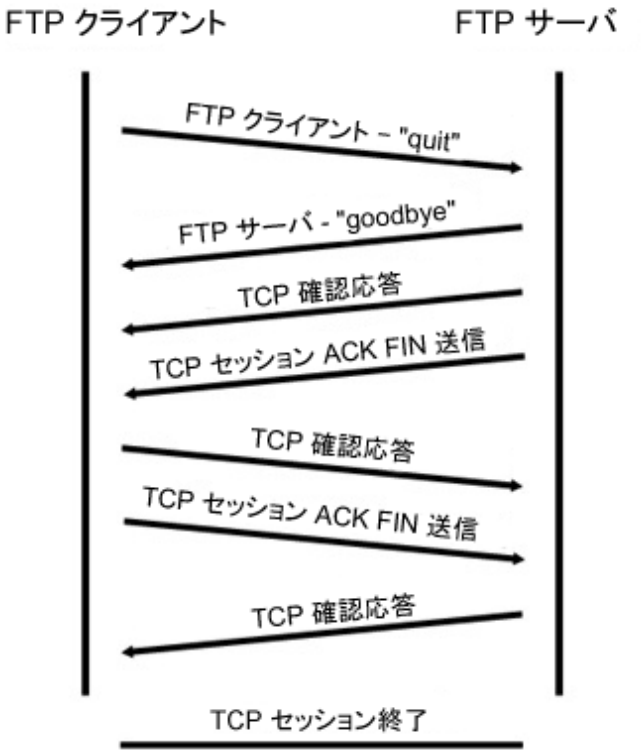
SYN ビットが含まれている TCP データグラムは他にいくつありますか。

TCP セッションが確立されると、PC と FTP サーバの間での FTP トラフィックが可能になります。FTP クライアントと FTP サーバは、TCP がセッションを制御し管理していることを意識せずに、相互に通信を行います。FTP サーバが FTP クライアントに Response: 220 を送信すると、FTP クライアント側の TCP セッションがサーバ側の TCP セッションに確認応答を送信します。このシーケンスは、次に示す Wireshark キャプチャで視覚化されています。

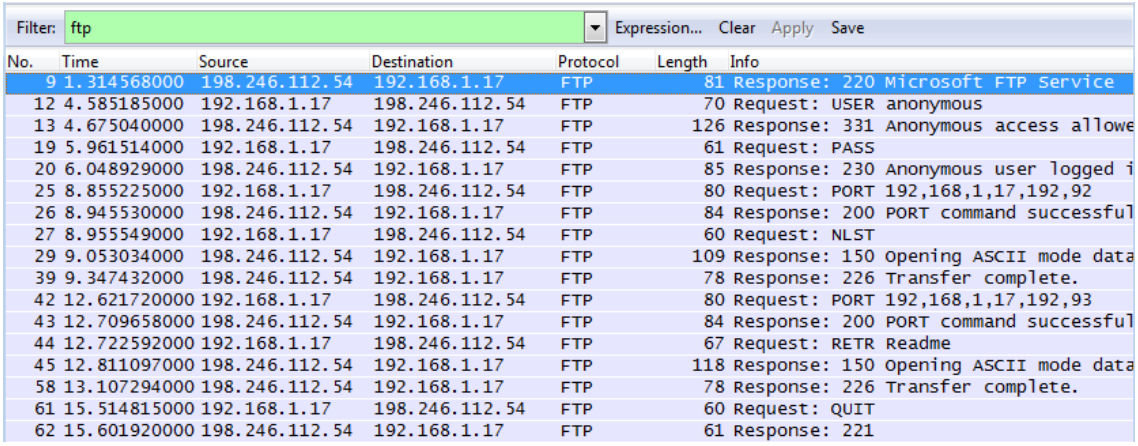
```
  9 1.314568000  198.246.112.54  192.168.1.17     FTP          81 Response: 220 Microsoft FTP Service
 10 1.523372000  192.168.1.17    198.246.112.54   TCP          54 49243 > ftp [ACK] Seq=1 Ack=28 Win=
 12 4.585185000  192.168.1.17    198.246.112.54   FTP          70 Request: USER anonymous
 13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

FTP セッションが終了すると、FTP クライアントは終了（quit）を意味するコマンドを送信します。それに対し、FTP サーバは Response: 221（goodbye）で確認応答します。この時点で、FTP サーバの TCP セッションが FTP クライアントに TCP データグラムを送信して、TCP セッションの終了を知らせます。FTP クライアントの TCP セッションは、終了データグラムを受け取った旨の確認応答を送信してから、自分自身の TCP セッションの終了を送信します。TCP 終了の発信元である FTP サーバが重複する終了を受信すると、その終了に対する確認応答として ACK データグラムが送信され、TCP セッションが閉じられます。このシーケンスは、次の図とキャプチャで視覚化されています。



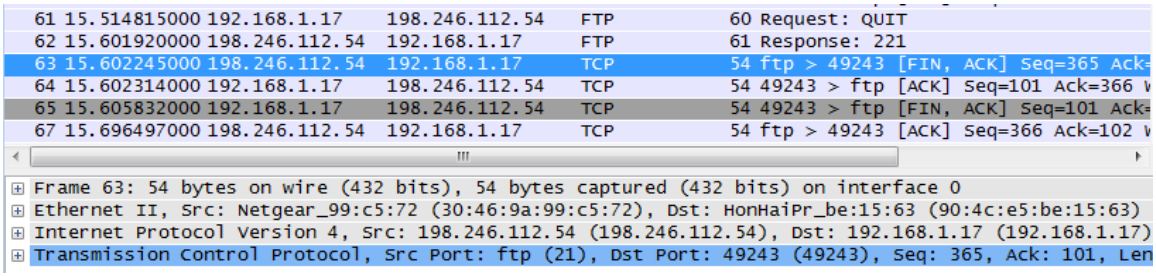
ftp フィルタを適用すると、Wireshark で FTP トラフィックの全シーケンスを調べることができます。この FTP セッション中に発生したイベントのシーケンスに注目してください。Readme ファイルの取得に anonymous というユーザ名が使用されています。ファイル転送が完了した後、ユーザは FTP セッションを終了しています。



TCP セッションの終了を調べるために、Wireshark で再び TCP フィルタを適用します。TCP セッションを終了するために、4 つのパケットが伝送されます。TCP 接続は全二重なので、各方向で別々に終了を行う必要があります。送信元アドレスと宛先アドレスを確認してください。

この例では、FTP サーバにはそのストリームで送信するデータがもうありません。FTP サーバは FIN フラグをセットしたセグメントをフレーム 63 で送信しています。PC はサーバからクライアントへのセッションを終了するための FIN に対する確認応答として ACK をフレーム 64 で送信しています。

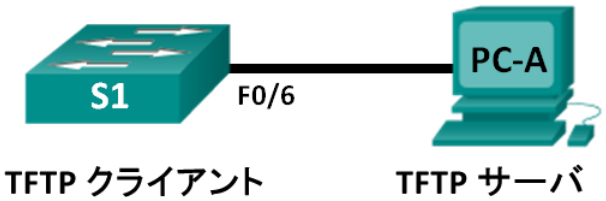
フレーム 65 では、PC が TCP セッションを終了するための FIN を FTP サーバに送信しています。FTP サーバは PC からの FIN に対する確認応答として ACK をフレーム 67 で送信しています。これで FTP サーバと PC の間の TCP セッションが終了したことになります。



パート 2: Wireshark の TFTP セッション キャプチャを使用した UDP ヘッダー フィールドと動作の識別

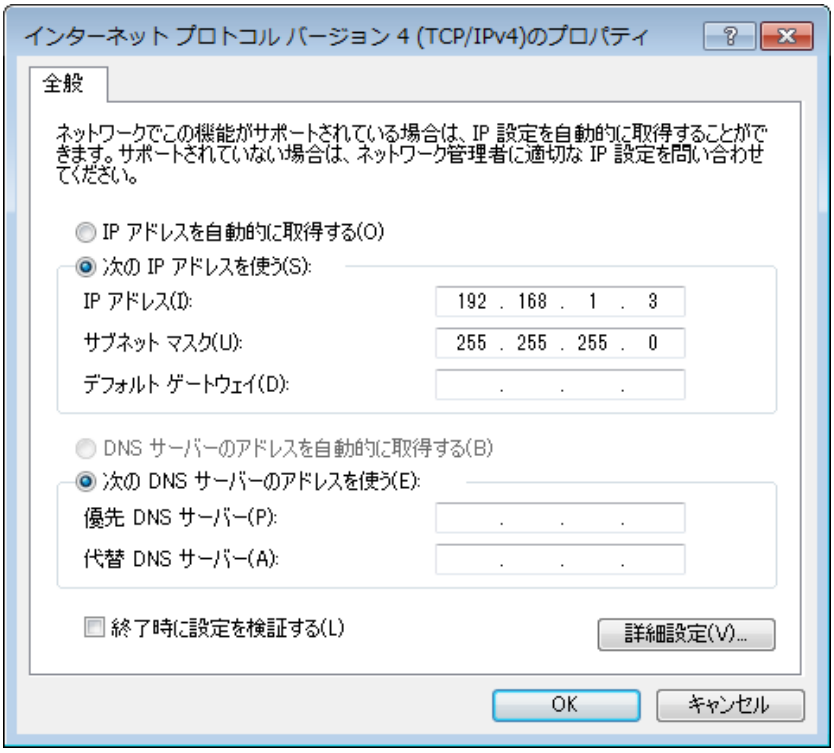
パート 2 では、Wireshark を使用して TFTP セッションをキャプチャし、UDP ヘッダー フィールドを詳しく調べます。

手順 1: この物理トポロジをセットアップし、TFTP キャプチャの準備をします。



a. PC-A とスイッチ S1 の間でコンソールおよびイーサネットの接続を確立します。

b. まだ設定していなければ、手動で PC の IP アドレスを 192.168.1.3 に 設定します。デフォルト ゲートウェイを設定する必要はありません。



c. スイッチを設定します。VLAN 1 に IP アドレス 192.168.1.1 を割り当てます。192.168.1.3 に ping して、PC との接続を確認します。必要に応じて、トラブルシューティングを行います。

```
Switch> enable
Switch# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# host S1
S1(config)# interface vlan 1
S1(config-if)# ip address 192.168.1.1 255.255.255.0
S1(config-if)# no shut
*Mar  1 00:37:50.166: %LINK-3-UPDOWN: Interface Vlan1, changed state to up
*Mar  1 00:37:50.175: %LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan1,
changed state to up
S1(config-if)# end
S1# ping 192.168.1.3
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.3, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/203/1007 ms
```

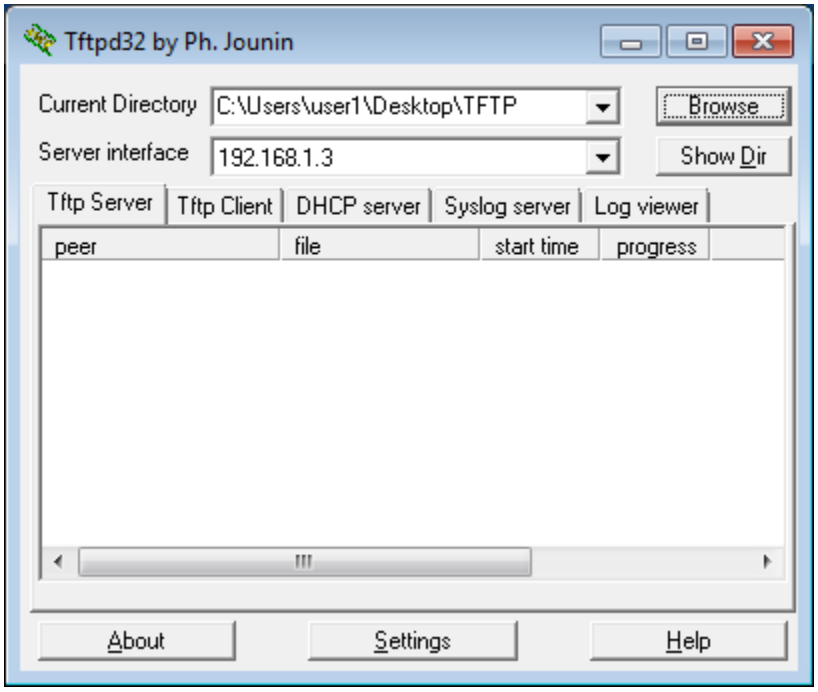
手順 2:　PC で TFTP サーバを準備します。

a. まだ存在しない場合は、PC のデスクトップ上に **TFTP** というフォルダを作成します。スイッチからのファイルがこの場所にコピーされます。

b. PC で **tftpd32** を起動します。

c. [**Browse**] をクリックし、現在のディレクトリを **C:\Users\user1\Desktop\TFTP** に変更します。ただし、user1 を自分のユーザ名に置き換えてください。

TFTP サーバは次のようになります。



現在のディレクトリ（Current Directory）にユーザが示されていて、サーバ（PC-A）インターフェイス（Server interface）として **192.168.1.3** という IP アドレス が示されていることに注目してください。

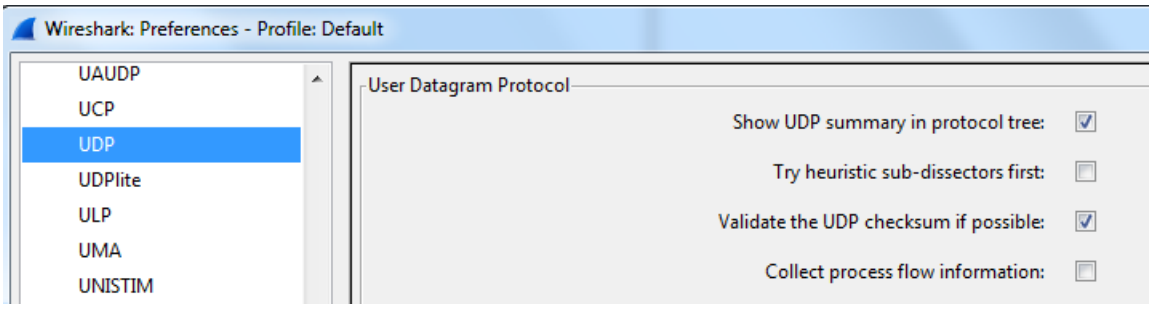
d. TFTP を使用してスイッチから PC に ファイルをコピーする機能をテストします。必要に応じて、トラブルシューティングを行います。

```
S1# copy start tftp
Address or name of remote host []? 192.168.1.3
Destination filename [s1-confg]?
!!
1638 bytes copied in 0.026 secs (63000 bytes/sec)
```

上記の出力のようにファイルがコピーされたことが確認できた場合は、次のステップに進むことができます。そうでない場合は、トラブルシューティングを行います。「`%Error opening tftp(Permission denied)`」エラーが表示された場合は、まずファイアウォールで TFTP がブロックされていないか確認し、自分のユーザ名に十分なアクセス許可が与えられている場所（デスクトップなど）にコピーしているか確認します。

手順 3:　Wireshark で TFTP セッションをキャプチャします。

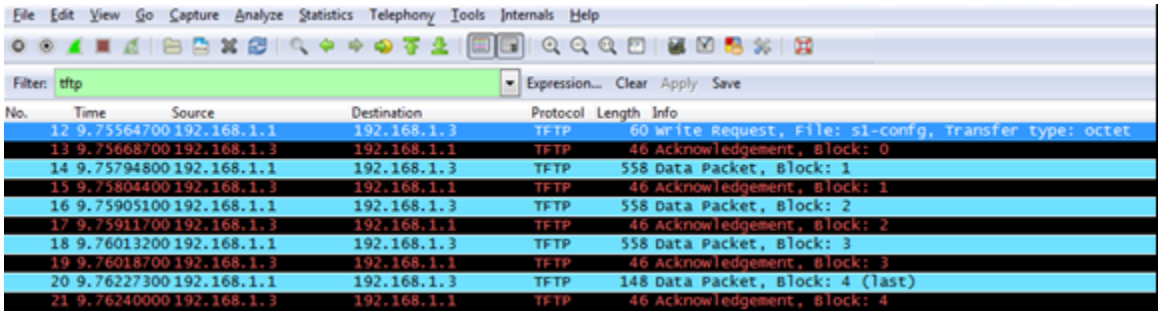
a. Wireshark を開きます。[**Edit**] メニューから [**Preferences**] を選択し、プラス（+）記号をクリックして、[**Protocols**] を展開します。下方にスクロールし、[**UDP**] を選択します。[**Validate the UDP checksum if possible**] チェック ボックスをクリックし、[**Apply**] をクリックします。[**OK**] をクリックします。



b. Wireshark のキャプチャを開始します。

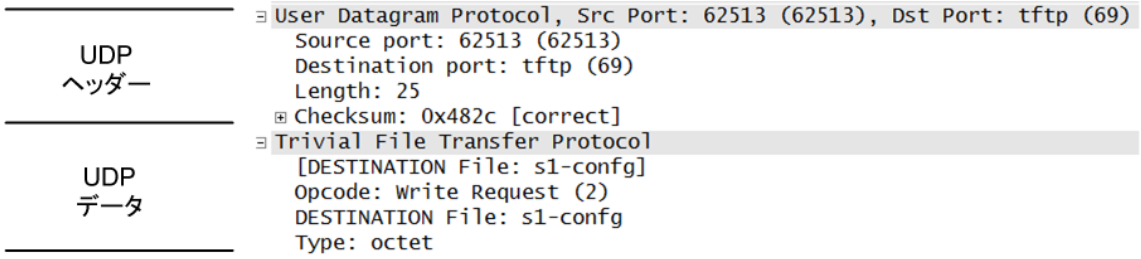
c. スイッチで **`copy start tftp`** コマンドを実行します。

d. Wireshark のキャプチャを停止します。

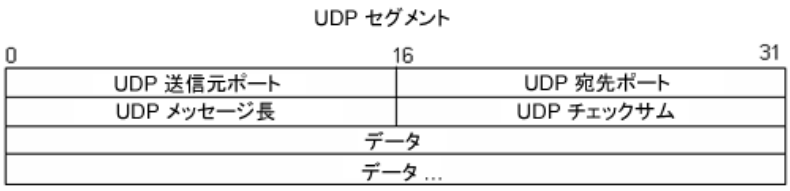


e. フィルタを **tftp** に設定します。出力は上に示した出力と類似したものになるはずです。この TFTP 転送を使用して、トランスポート層 UDP の動作を分析します。

Wireshark では、パケット詳細ペインに詳細な UDP 情報が表示されます。ホスト コンピュータからの最初の UDP データグラムを強調表示にし、マウス ポインタをパケット詳細ペインに移動してください。パケット詳細ペインを調整し、プロトコル展開ボックスをクリックして UDP レコードを展開することが必要になる場合もあります。展開した UDP データグラムは下の図と類似したものになるはずです。



下の図は UDP データグラム図です。TCP データグラムに比べて、ヘッダー情報が少ないことがわかるでしょう。TCP と同様、各 UDP データグラムは UDP 送信元ポートと UDP 宛先ポートによって識別されます。



最初の UDP データグラムの Wireshark キャプチャを使用して、UDP ヘッダーに関する情報を記入してください。チェックサム値は 16 進数（底値 16）の値です。16 進数は先頭に 0x コードが付きます。

送信元 IP アドレス:	
宛先 IP アドレス:	
送信元ポート番号:	
宛先ポート番号:	
UDP メッセージの長さ:	
UDP チェックサム:	

UDP はデータグラムの完全性をどのようにして確認しますか。

tftpd サーバから返された最初のフレームを調べます。UDP ヘッダーに関する情報を記入してください。

送信元 IP アドレス:	
宛先 IP アドレス:	
送信元ポート番号:	
宛先ポート番号:	
UDP メッセージの長さ:	
UDP チェックサム:	

```
⊟ User Datagram Protocol, Src Port: 58565 (58565), Dst Port: 62513 (62513)
    Source port: 58565 (58565)
    Destination port: 62513 (62513)
    Length: 12
  ⊞ Checksum: 0x8372 [incorrect, should be 0xa385 (maybe caused by "UDP checksum offload"?)]
⊟ Trivial File Transfer Protocol
    [DESTINATION File: s1-confg]
    Opcode: Acknowledgement (4)
    Block: 0
```

返される UDP データグラムの UDP 送信元ポートは異なりますが、その TFTP 転送の残りの部分では、この送信元ポートが使用されることに注意してください。信頼性の高い接続がないので、TFTP 転送を維持するために、TFTP セッションを開始するために使用された送信元ポートだけが使用されるのです。

UDP チェックサムが正しくないことにも注意してください。おそらく UDP チェックサム オフロードが原因でしょう。「UDP チェックサム オフロード」を検索すれば、これが発生する理由について詳しい情報が得られるはずです。

復習

この実習では、FTP セッションと TFTP セッションのキャプチャから TCP プロトコルと UDP プロトコルの動作を分析しました。TCP による通信の管理は、UDP とどのように異なりますか。

チャレンジ

FTP も TFTP も安全なプロトコルではないので、転送データはすべてクリア テキストで送信されます。これには、ユーザ ID、パスワード、またはクリアテキスト ファイル コンテンツが含まれます。上位層 FTP セッションを分析すると、ユーザ ID、パスワード、およびコンフィギュレーション ファイルのパスワードがすぐにわかります。上位層 TFTP データの調査はもう少し複雑ですが、データ フィールドを調べて、コンフィギュレーションのユーザ ID とパスワード情報を抽出することができます。

クリーンアップ

インストラクタから別の指示がない限り、次のことを行ってください。

1) PC にコピーされたファイルを削除します。
2) スイッチ **S1** の設定を消去します。
3) PC から手動による IP アドレスを削除し、インターネット接続を復元します。