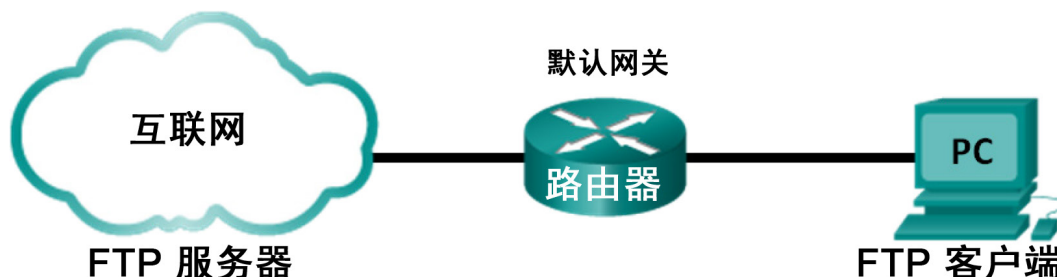


实验 - 使用 Wireshark 检查 TCP 和 UDP 捕获

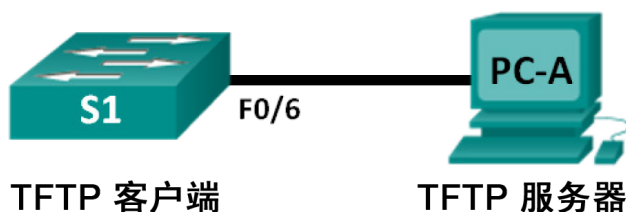
拓扑 - 第 1 部分 (FTP)

第 1 部分重点介绍 FTP 会话的 TCP 捕获。此拓扑包括一台能够访问互联网的 PC。



拓扑 - 第 2 部分 (TFTP)

第 2 部分重点介绍 TFTP 会话的 UDP 捕获。PC 必须具有到交换机 S1 的以太网连接和控制台连接。



地址分配表（第 2 部分）

设备	接口	IP 地址	子网掩码	默认网关
S1	VLAN 1	192.168.1.1	255.255.255.0	N/A
PC-A	网卡	192.168.1.3	255.255.255.0	192.168.1.1

目标

第 1 部分：使用 Wireshark 捕获 FTP 会话，了解 TCP 报头的字段和运行方式。

第 2 部分：使用 Wireshark 捕获 TFTP 会话，了解 UDP 报头的字段和运行方式

背景/场景

TCP/IP 传输层的两种协议为 TCP（在 RFC 761 中定义）和 UDP（在 RFC 768 中定义）。两个协议都支持上层协议通信。例如，TCP 为超文本传输协议 (HTTP) 和 FTP 等协议提供传输层支持；UDP 为域名系统 (DNS) 和 TFTP 诸如此类的操作提供传输层支持。

注意：理解 TCP 和 UDP 报头各部分及其运行方式是网络工程师应该掌握的关键技能。

本实验第 1 部分，您将使用 Wireshark 开源工具来捕获和分析 TCP 协议报头字段，以便在主机计算机与匿名 FTP 服务器之间进行 FTP 文件传输。Windows 命令行实用程序用于连接到匿名 FTP 服务器并下载文件。本实验第 2 部分，您将使用 Wireshark 来捕获和分析 UDP 报头字段，以便在主机计算机与 S1 之间进行 TFTP 文件传输。

注意：所用的交换机是采用 Cisco IOS Release 15.0(2) (Irbasek9 映像) 的 Cisco Catalyst 2960 系列。也可使用其他交换机以及 Cisco IOS 版本。根据型号以及 Cisco IOS 版本的不同，可用命令和产生的输出可能与实验显示的不一样。

注意：确保交换机的启动配置已经清除。如果不确定，请联系教师。

注意：第 1 部分假设 PC 可访问互联网，不能使用 Netlab 进行实验。第 2 部分可以使用 Netlab。

所需资源 - 第 1 部分 (FTP)

1 台 PC (采用 Windows 7 或 8 且可以访问命令提示符和互联网，并且已安装 Wireshark)

所需资源 - 第 2 部分 (TFTP)

- 1 台交换机 (支持 Cisco IOS 15.0(2) Irbasek9 版映像的 Cisco 2960 或同类交换机)
- 1 台 PC (采用 Windows 7 或 8 并且已安装 Wireshark 和 TFTP 服务器，例如 tftpd32)
- 用于通过控制台端口配置 Cisco IOS 设备的控制台电缆
- 如拓扑图所示的以太网电缆

第 1 部分：使用 Wireshark 捕获 FTP 会话，了解 TCP 报头的字段和运行方式

在第 1 部分，使用 Wireshark 捕获 FTP 会话并检查 TCP 报头字段。

第 1 步：开始 Wireshark 捕获。

- a. 在 Wireshark 捕获期间，关闭所有不必要的网络流量 (例如 Web 浏览器)，以限制数据流量。
- b. 开始 Wireshark 捕获。

第 2 步：下载 Readme 文件。

- a. 在命令提示符下输入 `ftp ftp.cdc.gov`。
- b. 使用用户 **anonymous** 且不用密码，登录疾病预防控制中心 (CDC) 的 FTP 站点。

```
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.
```

- c. 键入 **ls** 命令列出文件，以找到 Readme 文件并下载。

```
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.
```

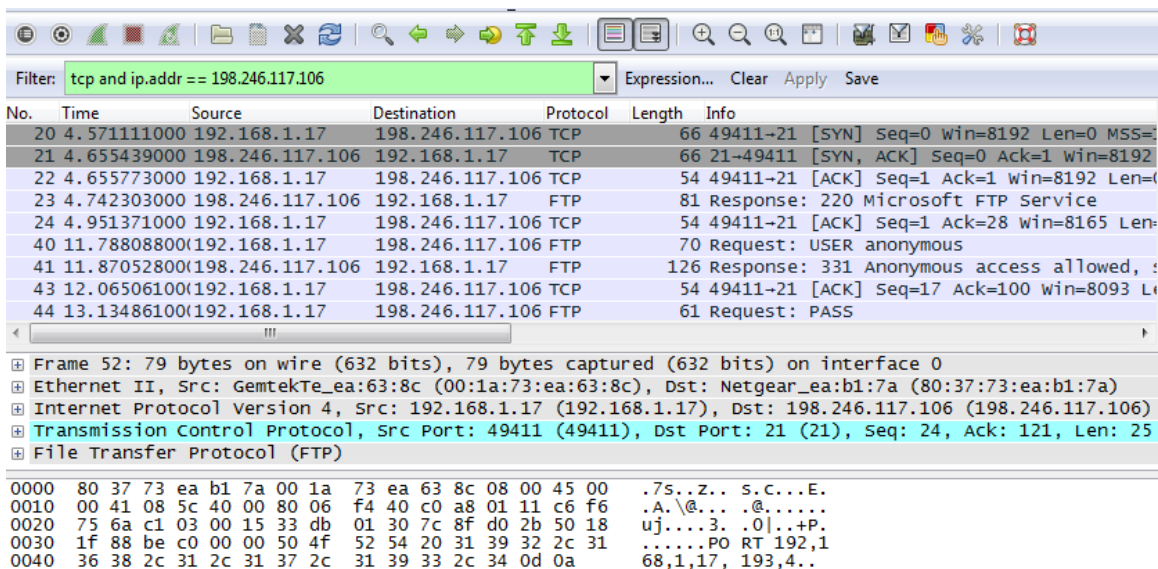
- d. 键入 **get Readme** 命令以下载该文件。当文件下载完成时，输入 **quit** 命令退出。

```
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
```

第 3 步：停止 Wireshark 捕获。

第 4 步：查看 Wireshark 主窗口。

Wireshark 在与 ftp.cdc.gov 执行 FTP 会话期间捕获了许多数据包。要限制分析的数据量，则在 **Filter: entry** (过滤: 条目) 区域键入 **tcp and ip.addr == 198.246.117.106**，然后单击 **Apply** (应用)。此时，IP 地址 198.246.117.106 是 ftp.cdc.gov 的地址。



The image shows the Wireshark network protocol analyzer interface. The top toolbar contains various icons for file operations, zooming, and packet analysis. Below the toolbar, the 'Filter' field is set to 'tcp and ip.addr == 198.246.117.106'. The main pane displays a list of captured packets with columns for No., Time, Source, Destination, Protocol, Length, and Info. The packets are filtered to show only those related to the specified IP address. The bottom pane shows the details of the selected packet (Frame 52), including Ethernet II, Internet Protocol Version 4, and Transmission Control Protocol (TCP) fields.

No.	Time	Source	Destination	Protocol	Length	Info
20	4.571111000	192.168.1.17	198.246.117.106	TCP	66	49411->21 [SYN] Seq=0 win=8192 Len=0 MSS=
21	4.655439000	198.246.117.106	192.168.1.17	TCP	66	21->49411 [SYN, ACK] Seq=0 Ack=1 win=8192
22	4.655773000	192.168.1.17	198.246.117.106	TCP	54	49411->21 [ACK] Seq=1 Ack=1 win=8192 Len=
23	4.742303000	198.246.117.106	192.168.1.17	FTP	81	Response: 220 Microsoft FTP Service
24	4.951371000	192.168.1.17	198.246.117.106	TCP	54	49411->21 [ACK] Seq=1 Ack=28 win=8165 Len=
40	11.788088000	192.168.1.17	198.246.117.106	FTP	70	Request: USER anonymous
41	11.870528000	198.246.117.106	192.168.1.17	FTP	126	Response: 331 Anonymous access allowed, s
43	12.065061000	192.168.1.17	198.246.117.106	TCP	54	49411->21 [ACK] Seq=17 Ack=100 win=8093 L
44	13.134861000	192.168.1.17	198.246.117.106	FTP	61	Request: PASS

Frame 52: 79 bytes on wire (632 bits), 79 bytes captured (632 bits) on interface 0

Ethernet II, Src: GemtekTe_ea:63:8c (00:1a:73:ea:63:8c), Dst: Netgear_ea:b1:7a (80:37:73:ea:b1:7a)

Internet Protocol Version 4, Src: 192.168.1.17 (192.168.1.17), Dst: 198.246.117.106 (198.246.117.106)

Transmission Control Protocol, Src Port: 49411 (49411), Dst Port: 21 (21), Seq: 24, Ack: 121, Len: 25

File Transfer Protocol (FTP)

0000 80 37 73 ea b1 7a 00 1a 73 ea 63 8c 08 00 45 00 .7s..z..S.C...E.

0010 00 41 08 5c 40 00 80 06 f4 40 c0 a8 01 11 c6 f6 .A.\@... .@.....

0020 75 6a c1 03 00 15 33 db 01 30 7c 8f d0 2b 50 18 uj....3. .0|...+P.

0030 1f 88 be c0 00 00 50 4f 52 54 20 31 39 32 2c 31PO RT 192,1

0040 36 38 2c 31 2c 31 37 2c 31 39 33 2c 34 0d 0a 68,1,17, 193,4..

第 5 步：分析 TCP 字段。

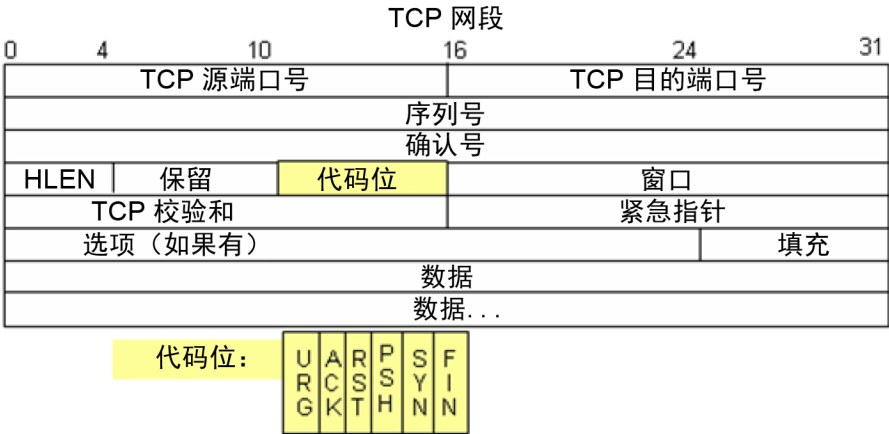
应用 TCP 过滤器后，数据包列表窗格（顶部）中的前三个帧显示创建可靠会话的传输层协议 TCP。[SYN]、[SYN, ACK] 和 [ACK] 的顺序说明了三次握手。

20	4.571111000	192.168.1.17	198.246.117.106	TCP	66	49411-21	[SYN]	Seq=0	win=8192	Len=0	MSS=
21	4.655439000	198.246.117.106	192.168.1.17	TCP	66	21-49411	[SYN, ACK]	Seq=0	Ack=1	win=8192	
22	4.655773000	192.168.1.17	198.246.117.106	TCP	54	49411-21	[ACK]	Seq=1	Ack=1	win=8192	Len=

TCP 经常用于控制数据报传送、确认数据报抵达和管理窗口大小等会话中。对于 FTP 客户端与 FTP 服务器之间的每次数据交换，都会启动一个新的 TCP 会话。在数据传输结束时，TCP 会话即会关闭。当 FTP 会话结束时，TCP 将会按顺序执行关闭和终止。

在 Wireshark 中，数据包详细信息窗格（中间部分）提供 TCP 详细信息。突出显示主机计算机的第一个 TCP 数据报，并将其展开。展开的 TCP 数据报与下方所示的数据包详细信息窗格类似。

Frame 20: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface 0	
Ethernet II, Src: GemtekTe_ea:63:8c (00:1a:73:ea:63:8c), Dst: Netgear_ea:b1:7a (80:37:73:ea:b1:7a)	
Internet Protocol Version 4, Src: 192.168.1.17 (192.168.1.17), Dst: 198.246.117.106 (198.246.117.106)	
Transmission Control Protocol, Src Port: 49411 (49411), Dst Port: 21 (21), Seq: 0, Len: 0	
Source Port: 49411 (49411)	
Destination Port: 21 (21)	
[Stream index: 1]	
[TCP Segment Len: 0]	
Sequence number: 0 (relative sequence number)	
Acknowledgment number: 0	
Header Length: 32 bytes	
... 0000 0000 0010 = 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: 0x5bba [validation disabled]	
Urgent pointer: 0	
Options: (12 bytes), Maximum segment size, No-Operation (NOP), window scale, No-Operation (NOP), No-O	



上图是 TCP 数据报图。每个字段的说明可供参考：

- **TCP Source port number**（TCP 源端口号）属于一个打开了连接的 TCP 会话主机，其值通常是大于 1,023 的随机值。
- **TCP destination port number**（TCP 目的端口号）用于标识远程站点的上层协议或应用程序。0-1,023 范围内的值代表“公认端口”，与常用的服务和应用（如 RFC 1700 所述，例如 Telnet、FTP、HTTP 等）关联。源 IP 地址、源端口、目的 IP 地址及目的端口的组合用于唯一标识到发送方和接收方的会话。

注意：在下面的 Wireshark 捕获中，目的端口是 21，即 FTP。FTP 服务器侦听端口 21 上的 FTP 客户端连接。

- **Sequence number**（序列号）指定数据段中最后一个二进制八位数的编号。
- **Acknowledgment number**（确认号）指定接收方预期的下一个二进制八位数。
- **Code Bits**（代码位）在会话管理中以及数据段的处理中具有特殊的含义。需要关注的值包括：
 - ACK — 数据段接收确认。
 - SYN — 同步，仅在 TCP 三向握手期间协商新的 TCP 会话时才设置。
 - FIN — 完成，请求关闭 TCP 会话。
- **Window size**（窗口大小）是滑动窗口的值。该值确定等待确认消息之前可以发送的二进制八位数的数量。
- **Urgent pointer**（紧急指针）只用于 URG（紧急）标志 - 当发送方需要发送紧急数据到接收方时。
- **Options**（选项）现在只有一个选项，它定义为最大 TCP 数据段大小（可选值）。

使用第一个 TCP 会话启动时的 Wireshark 捕获（SYN 位设置为 1），填写 TCP 报头的相关信息：

从 PC 到 CDC 服务器（仅 SYN 位设置为 1）：

源 IP 地址	
目的 IP 地址	
源端口号	
目的端口号	
序列号	
确认号	
报头长度	
窗口大小	

在第二个 Wireshark 过滤的捕获中，CDC FTP 服务器确认来自 PC 的请求。注意 SYN 和 ACK 位的值。

```

+ Frame 21: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface 0
+ Ethernet II, Src: Netgear_ea:b1:7a (80:37:73:ea:b1:7a), Dst: GemtekTe_ea:63:8c (00:1a:73:ea:63:8c)
+ Internet Protocol Version 4, Src: 198.246.117.106 (198.246.117.106), Dst: 192.168.1.17 (192.168.1.17)
- Transmission Control Protocol, Src Port: 21 (21), Dst Port: 49411 (49411), Seq: 0, Ack: 1, Len: 0
  Source Port: 21 (21)
  Destination Port: 49411 (49411)
  [Stream index: 1]
  [TCP Segment Len: 0]
  Sequence number: 0 (relative sequence number)
  Acknowledgment number: 1 (relative ack number)
  Header Length: 32 bytes
- .... 0000 0001 0010 = Flags: 0x012 (SYN, ACK)
  000. .... = Reserved: Not set
  ...0 .... = Nonce: Not set
  .... 0... = Congestion Window Reduced (CWR): Not set
  .... .0.. = ECN-Echo: Not set
  .... ..0. = Urgent: Not set
  .... ...1 .... = Acknowledgment: 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: 0x0ee7 [validation disabled]
  urgent pointer: 0
+ Options: (12 bytes), Maximum segment size, No-operation (NOP), window scale, No-operation (NOP), No
+ [SEQ/ACK analysis]
```

填写有关 SYN-ACK 消息的以下信息。

源 IP 地址	
目的 IP 地址	
源端口号	
目的端口号	
序列号	
确认号	
报头长度	
窗口大小	

在协商建立通信的最后阶段，PC 向服务器发送确认消息。注意只有 ACK 位设置为 1，并且序列号已增加到 1。

```

+ Frame 22: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface 0
+ Ethernet II, Src: GemtekTe_ea:63:8c (00:1a:73:ea:63:8c), Dst: Netgear_ea:b1:7a (80:37:73:ea:b1:7a)
+ Internet Protocol Version 4, Src: 192.168.1.17 (192.168.1.17), Dst: 198.246.117.106 (198.246.117.106)
- Transmission Control Protocol, Src Port: 49411 (49411), Dst Port: 21 (21), Seq: 1, Ack: 1, Len: 0
  Source Port: 49411 (49411)
  Destination Port: 21 (21)
  [Stream index: 1]
  [TCP Segment Len: 0]
  Sequence number: 1 (relative sequence number)
  Acknowledgment number: 1 (relative ack number)
  Header Length: 20 bytes
  - ... 0000 0001 0000 = Flags: 0x010 (ACK)
    000. .... = Reserved: Not set
    ...0 .... = Nonce: Not set
    .... 0... = Congestion window Reduced (CWR): Not set
    .... .0.. = ECN-Echo: Not set
    .... ..0. = Urgent: Not set
    .... ...1 .... = Acknowledgment: Set
    .... .... 0... = Push: Not set
    .... .... .0.. = Reset: Not set
    .... .... ..0. = Syn: Not set
    .... .... ...0 = Fin: Not set
  window size value: 8192
  [Calculated window size: 8192]
  [window size scaling factor: 1]
  + Checksum: 0x4f6a [validation disabled]
  Urgent pointer: 0
  + [SEQ/ACK analysis]
```

填写有关 ACK 消息的以下信息。

源 IP 地址	
目的 IP 地址	
源端口号	
目的端口号	
序列号	
确认号	
报头长度	
窗口大小	

一个 SYN 位包含多少个其他 TCP 数据报？

在 TCP 会话建立后，PC 和 FTP 服务器之间会产生 FTP 流量。FTP 客户端与服务器相互通信，不了解 TCP 对会话的控制和管理。当 FTP 服务器发送 *Response:220* 到 FTP 客户端时，FTP 客户端上的 TCP 会话将会发送确认到服务器上的 TCP 会话。从下面的 Wireshark 捕获中可以看到序列。

23	4.742303000	198.246.117.106	192.168.1.17	FTP	81	Response: 220 Microsoft FTP Service
24	4.951371000	192.168.1.17	198.246.117.106	TCP	54	49411→21 [ACK] Seq=1 Ack=28 win=8165 Len=
40	11.788088000	192.168.1.17	198.246.117.106	FTP	70	Request: USER anonymous
41	11.870528000	198.246.117.106	192.168.1.17	FTP	126	Response: 331 Anonymous access allowed,

Frame 23: 81 bytes on wire (648 bits), 81 bytes captured (648 bits) on interface 0

Ethernet II, Src: Netgear_ea:b1:7a (80:37:73:ea:b1:7a), Dst: GemtekTe_ea:63:8c (00:1a:73:ea:63:8c)

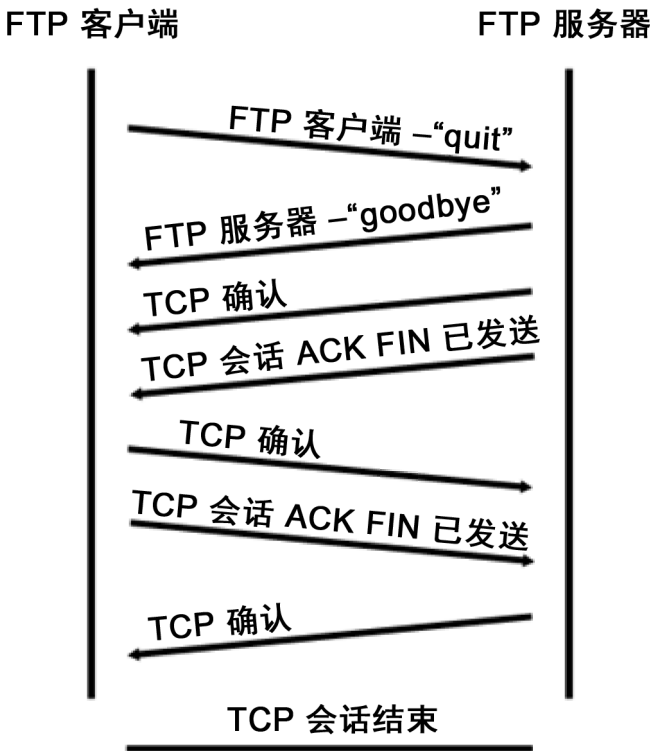
Internet Protocol Version 4, Src: 198.246.117.106 (198.246.117.106), Dst: 192.168.1.17 (192.168.1.17)

Transmission Control Protocol, Src Port: 21 (21), Dst Port: 49411 (49411), Seq: 1, Ack: 1, Len: 27

File Transfer Protocol (FTP)

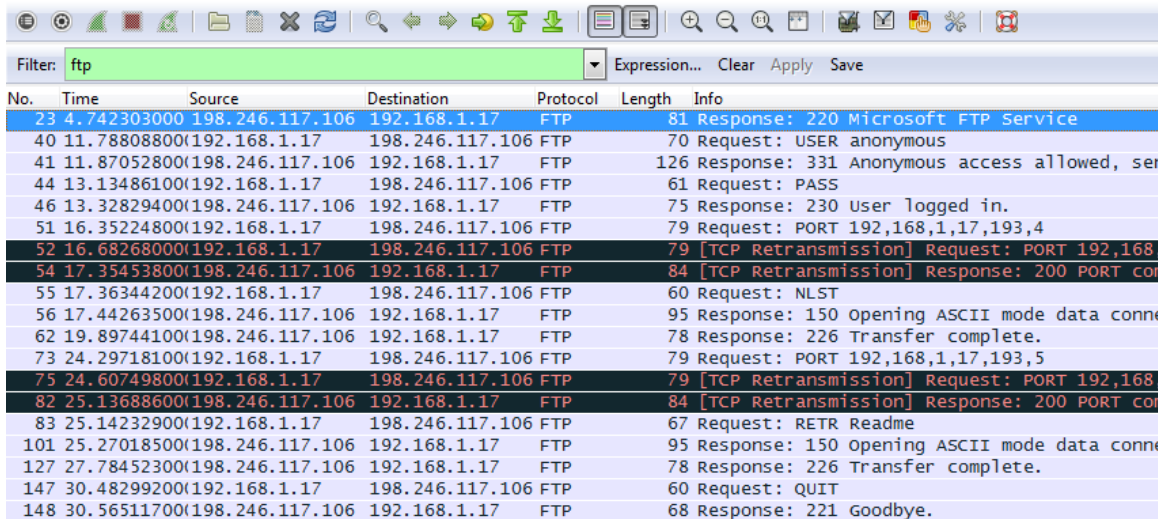
220 Microsoft FTP Service\r\nResponse code: Service ready for new user (220)Response arg: Microsoft FTP Service

当 FTP 会话完成时，FTP 客户端将发送命令以便“退出”。FTP 服务器以 *Response: 221 Goodbye* 确认 FTP 终止。此时 FTP 服务器 TCP 会话发送 TCP 数据报到 FTP 客户端，宣告 TCP 会话终止。FTP 客户端 TCP 会话确认收到终止数据报，然后发送自己的 TCP 会话终止。当 TCP 终止的发起者（FTP 服务器）收到重复的终止时，将发送 ACK 数据报确认终止，然后 TCP 会话关闭。从下面的图和捕获中可以看到序列。



实验 - 使用 Wireshark 检查 TCP 和 UDP 捕获

通过应用 **ftp** 过滤器，可以在 Wireshark 中检查 FTP 流量的整个序列。注意 FTP 会话期间的事件序列。用户名 **anonymous** 用于检索 Readme 文件。在完成文件传输后，用户将关闭 FTP 会话。

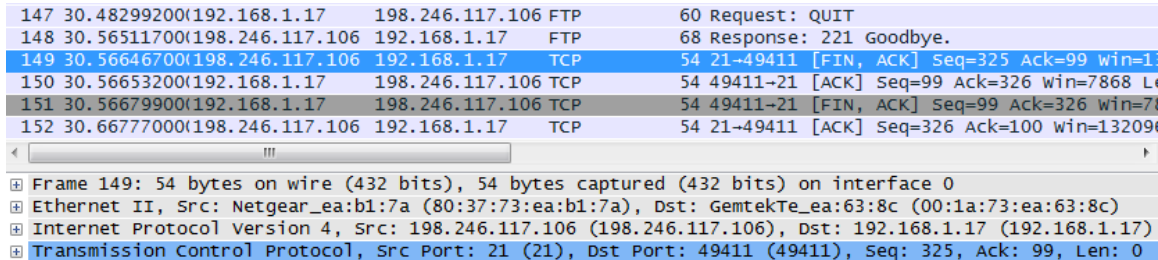


No.	Time	Source	Destination	Protocol	Length	Info
23	4.742303000	198.246.117.106	192.168.1.17	FTP	81	Response: 220 Microsoft FTP Service
40	11.788088000	192.168.1.17	198.246.117.106	FTP	70	Request: USER anonymous
41	11.870528000	198.246.117.106	192.168.1.17	FTP	126	Response: 331 Anonymous access allowed, send
44	13.134861000	192.168.1.17	198.246.117.106	FTP	61	Request: PASS
46	13.328294000	198.246.117.106	192.168.1.17	FTP	75	Response: 230 User logged in.
51	16.352248000	192.168.1.17	198.246.117.106	FTP	79	Request: PORT 192,168,1,17,193,4
52	16.682680000	192.168.1.17	198.246.117.106	FTP	79	[TCP Retransmission] Request: PORT 192,168
54	17.354538000	198.246.117.106	192.168.1.17	FTP	84	[TCP Retransmission] Response: 200 PORT com
55	17.363442000	192.168.1.17	198.246.117.106	FTP	60	Request: NLST
56	17.442635000	198.246.117.106	192.168.1.17	FTP	95	Response: 150 Opening ASCII mode data conn
62	19.897441000	198.246.117.106	192.168.1.17	FTP	78	Response: 226 Transfer complete.
73	24.297181000	192.168.1.17	198.246.117.106	FTP	79	Request: PORT 192,168,1,17,193,5
75	24.607498000	192.168.1.17	198.246.117.106	FTP	79	[TCP Retransmission] Request: PORT 192,168
82	25.136886000	198.246.117.106	192.168.1.17	FTP	84	[TCP Retransmission] Response: 200 PORT com
83	25.142329000	192.168.1.17	198.246.117.106	FTP	67	Request: RETR Readme
101	25.270185000	198.246.117.106	192.168.1.17	FTP	95	Response: 150 Opening ASCII mode data conn
127	27.784523000	198.246.117.106	192.168.1.17	FTP	78	Response: 226 Transfer complete.
147	30.482992000	192.168.1.17	198.246.117.106	FTP	60	Request: QUIT
148	30.565117000	198.246.117.106	192.168.1.17	FTP	68	Response: 221 Goodbye.

再次在 Wireshark 中应用 TCP 过滤器，检查 TCP 会话是否终止。终止 TCP 会话需要发送四个数据包。由于 TCP 连接是全双工，每个方向必须分别终止。检查源地址和目的地址。

在本例中，FTP 服务器没有其他要发送的数据流。在第 149 帧中，它将发送带 FIN 标志设置的数据段。在第 150 帧中，PC 发送 ACK 消息来确认收到 FIN，以终止从服务器到客户端的会话。

在第 151 帧中，PC 向 FTP 服务器发送 FIN 以终止 TCP 会话。在第 152 帧中，FTP 服务器响应 ACK 以确认来自 PC 的 FIN。现在 FTP 服务器和 PC 之间的 TCP 会话即可终止。

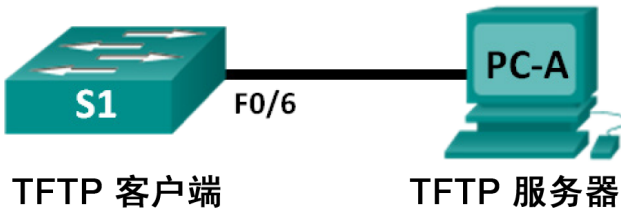


147	30.482992000	192.168.1.17	198.246.117.106	FTP	60	Request: QUIT
148	30.565117000	198.246.117.106	192.168.1.17	FTP	68	Response: 221 Goodbye.
149	30.566467000	198.246.117.106	192.168.1.17	TCP	54	21-49411 [FIN, ACK] Seq=325 Ack=99 win=1
150	30.566532000	192.168.1.17	198.246.117.106	TCP	54	49411-21 [ACK] Seq=99 Ack=326 win=7868 L
151	30.566799000	192.168.1.17	198.246.117.106	TCP	54	49411-21 [FIN, ACK] Seq=99 Ack=326 win=7
152	30.667770000	198.246.117.106	192.168.1.17	TCP	54	21-49411 [ACK] Seq=326 Ack=100 win=13209

第 2 部分：使用 Wireshark 捕获 TFTP 会话，了解 UDP 报头的字段和运行方式

在第 2 部分，使用 Wireshark 捕获 TFTP 会话并检查 UDP 报头字段。

第 1 步：建立此物理拓扑并为 TFTP 捕获做好准备。



- 建立 PC-A 与 S1 之间的控制台和以太网连接。
- 将 PC 的 IP 地址手动更改为 192.168.1.3。它不需要设置默认网关。
- 配置交换机。为 VLAN 1 分配 IP 地址 192.168.1.1。通过对 192.168.1.3 执行 ping 操作检验与 PC 的连接。根据情况进行故障排除。

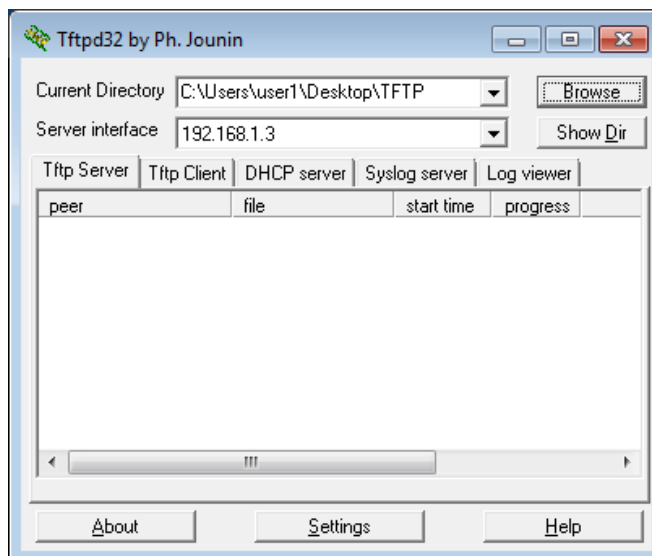
```
Switch> enable
Switch# conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Switch(config)# host S1
S1(config)# interface vlan 1
S1(config-if)# ip address 192.168.1.1 255.255.255.0
S1(config-if)# no shut
*Mar  1 00:37:50.166: %LINK-3-UPDOWN: Interface Vlan1, changed state to up
*Mar  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
```

- 将运行配置保存到 NVRAM。
S1# copy run start

第 2 步：准备好在 PC 上使用 TFTP 服务器。

- 如果尚不存在，请在 PC 桌面上创建名为 **TFTP** 的文件夹。来自交换机的文件将被复制到此位置。
- 在 PC 上启动 **tftpd32**。
- 单击 **Browse**（浏览）并将当前目录更改为 **C:\Users\user1\Desktop\TFTP**，用您的用户名替换 user1。

TFTP 服务器如下所示：



注意在 Current Directory（当前目录）中会列出用户，还会列出服务器 (PC-A) 接口的 IP 地址 **192.168.1.3**。

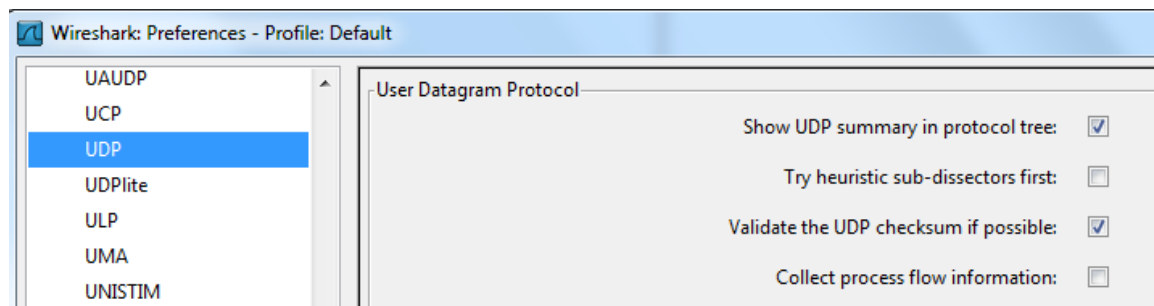
- d. 测试使用 TFTP 将文件从交换机复制到 PC 的功能。根据情况进行故障排除。

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

如果您看到文件已复制，则您可以继续下一步。如果此文件尚未复制，进行必要的故障排除。如果遇到 **%Error opening tftp (Permission denied)** 错误，请确定您的防火墙是否拦截 TFTP，并确定是否将文件复制到您的用户名有足够权限的位置，例如桌面。

第 3 步：使用 Wireshark 捕获 TFTP 会话

- a. 打开 Wireshark。从 **Edit**（编辑）菜单中，选择 **Preferences**（首选项），然后单击 (+) 号展开 **Protocols**（协议）。向下滚动并选择 **UDP**。单击 **Validate the UDP checksum if possible**（尽可能验证 UDP 校验和）复选框并单击 **Apply**（应用）。然后单击 **OK**（确定）。



- b. 开始 Wireshark 捕获。
c. 在交换机上运行 **copy start tftp** 命令。
d. 停止 Wireshark 捕获。

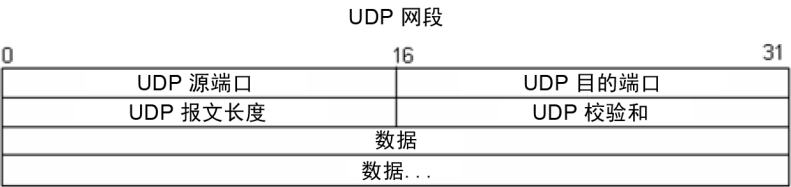
No.	Time	Source	Destination	Protocol	Length	Info
12	9.75564700	192.168.1.1	192.168.1.3	TFTP	60	write Request, File: s1-config, Transfer type: octet
13	9.75668700	192.168.1.3	192.168.1.1	TFTP	46	Acknowledgement, Block: 0
14	9.75794800	192.168.1.1	192.168.1.3	TFTP	558	Data Packet, Block: 1
15	9.75804400	192.168.1.3	192.168.1.1	TFTP	46	Acknowledgement, Block: 1
16	9.75905100	192.168.1.1	192.168.1.3	TFTP	558	Data Packet, Block: 2
17	9.75911700	192.168.1.3	192.168.1.1	TFTP	46	Acknowledgement, Block: 2
18	9.76013200	192.168.1.1	192.168.1.3	TFTP	558	Data Packet, Block: 3
19	9.76018700	192.168.1.3	192.168.1.1	TFTP	46	Acknowledgement, Block: 3
20	9.76227300	192.168.1.1	192.168.1.3	TFTP	148	Data Packet, Block: 4 (last)
21	9.76240000	192.168.1.3	192.168.1.1	TFTP	46	Acknowledgement, Block: 4

- e. 将过滤器设置为 **tftp**。屏幕上会显示如上所示的类似输出。此 TFTP 传输用于分析传输层 UDP 操作。

Wireshark 数据包详细信息窗格提供 UDP 详细信息。突出显示来自主机计算机的第一个 UDP 数据报，然后将鼠标指针移到数据包详细信息窗格。可能必须调整数据包详细信息窗格，并且单击协议扩展框来展开 UDP 记录。展开的 UDP 数据报应类似于下图。

UDP 报头	<div>⊟ User Datagram Protocol, Src Port: 62513 (62513), Dst Port: tftp (69)</div> <div>Source port: 62513 (62513)</div> <div>Destination port: tftp (69)</div> <div>Length: 25</div>
UDP 数据	<div>⊟ Checksum: 0x482c [correct]</div> <div>⊟ Trivial File Transfer Protocol</div> <div>[DESTINATION File: s1-config]</div> <div>Opcode: Write Request (2)</div> <div>DESTINATION File: s1-config</div> <div>Type: octet</div>

下图是 UDP 数据报图。与 TCP 数据报相比，UDP 数据报的报头信息较少。与 TCP 类似，每个 UDP 数据报都由 UDP 源端口和 UDP 目的端口标识。



使用第一个 UDP 数据报的 Wireshark 捕获，填写 UDP 报头的相关信息。校验和值是一个用前导 0x 代码表示的十六进制（以 16 为基数）值：

源 IP 地址	
目的 IP 地址	
源端口号	
目的端口号	
UDP 报文长度	
UDP 校验和	

UDP 如何检验数据报的完整性？

检查从 tftpd 服务器返回的第一个帧。填写 UDP 报头的相关信息：

源 IP 地址	
目的 IP 地址	
源端口号	
目的端口号	
UDP 报文长度	
UDP 校验和	

```

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-config]
  Opcode: Acknowledgement (4)
  Block: 0

```

请注意，返回的 UDP 报文具有不同的 UDP 源端口，而这个源端口用于剩下的 TFTP 传输。由于没有可靠的连接，因此只使用开始 TFTP 会话时所用的原始源端口来保持 TFTP 传输。

另外注意 UDP 校验和是错误的。这可能是由于 UDP 校验和卸载引起的。您可以搜索“UDP 校验和卸载”来了解更多详细信息。

思考

本实验让学生从捕获的 FTP 和 TFTP 会话来分析 TCP 及 UDP 协议的工作原理。TCP 与 UDP 管理通信的方式有何不同？

练习

FTP 和 TFTP 都不是安全协议，因此所有传输的数据都以明文形式发送。包括用户 ID、密码或明文文件内容。分析上层 FTP 会话即可迅速识别用户 ID、密码和配置文件密码。上层 TFTP 数据研究要复杂一些，但可以研究数据字段来提取配置用户 ID 和密码信息。

课后清理

除非教师另有指示。

- 1) 删除复制到 PC 上的文件。
- 2) 清除 S1 上的配置。
- 3) 从 PC 手动删除 IP 地址并恢复互联网连接。