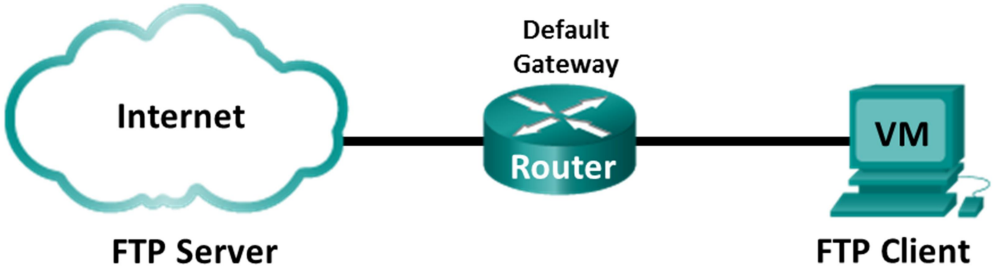
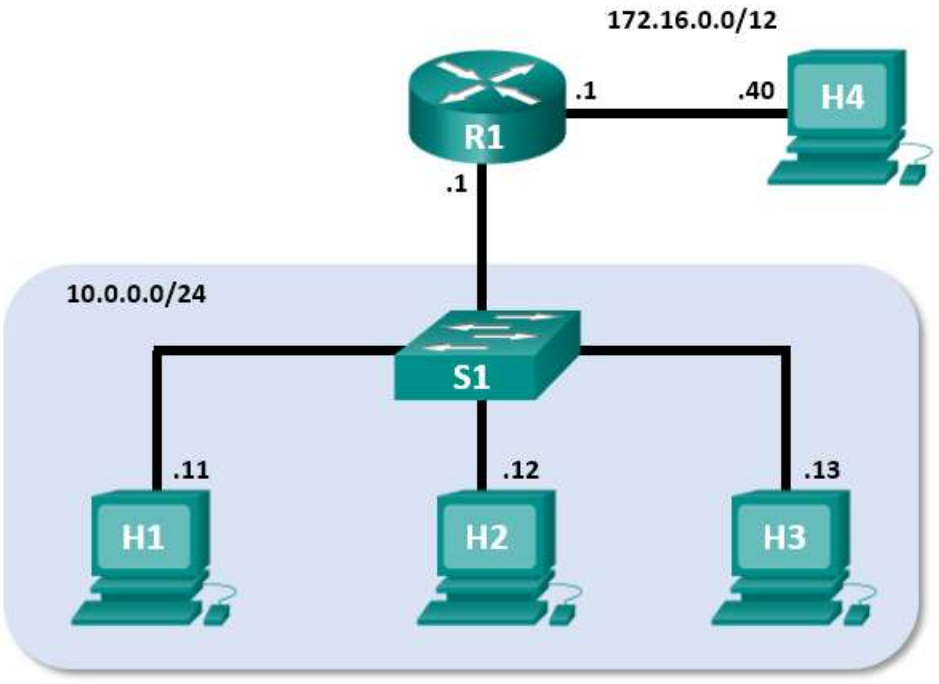
Laboratório - Usando o Wireshark para Examinar Capturas FTP e TFTP

Topologia - Parte 1 (FTP)



A parte 1 destacará uma captura TCP de uma sessão FTP. Esta topologia consiste na CyberOps Workstation VM com acesso à Internet.

Topologia Mininet - Parte 2 (TFTP)



Objetivos

Parte 1: Identificar os Campos do Cabeçalho e a Operação TCP Usando uma Captura de Sessão FTP do Wireshark

Parte 2: Identificar os Campos do Cabeçalho e a Operação UDP Usando uma Captura de Sessão TFTP do Wireshark

Histórico/Cenário

Dois protocolos na camada de transporte TCP/IP são TCP (definido na RFC 761), e UDP (definido na RFC 768). Ambos os protocolos suportam a comunicação de protocolos de camada superior. Por exemplo, o TCP é usado para fornecer suporte de camada de transporte para os protocolos HTTP e FTP, entre outros. O UDP fornece suporte de camada de transporte para o DNS e o TFTP, entre outros.

Na parte 1 deste laboratório, você usará a ferramenta de código aberto Wireshark para capturar e analisar os campos do cabeçalho do protocolo TCP para transferências de arquivos FTP entre o computador host e um servidor FTP anônimo. A linha de comando do terminal é usada para se conectar a um servidor FTP anônimo e baixar um arquivo. Na Parte 2 deste laboratório, você usará o Wireshark para capturar e analisar campos de cabeçalho UDP para transferências de arquivo TFTP entre dois computadores host Mininet.

Recursos Necessários

- VM do local de trabalho de CyberOps
- Acesso à Internet

Instruções

Parte 1: Identificar os Campos do Cabeçalho e a Operação TCP Usando uma Captura de Sessão FTP do Wireshark

Na Parte 1, use o Wireshark para capturar uma sessão FTP e inspecionar os campos do cabeçalho TCP.

Etapa 1: Iniciar uma captura do Wireshark.

a. Inicie e faça login na VM CyberOps Workstation. Abra uma janela de terminal e inicie o Wireshark. O e comercial (&) envia o processo para o plano de fundo e permite que você continue trabalhando no mesmo terminal.

```
[analyst@secOps ~]$ wireshark &
```

b. Inicie uma captura Wireshark para a interface **enp0s3**.

c. Abra outra janela de terminal para acessar um site ftp externo. Digite **ftp ftp.cdc.gov** no prompt. Efetue login no site FTP do Centro de Controle e Prevenção de Doenças (CDC) com o usuário **anonymous** e nenhuma senha.

```
[analyst@secOps ~]$ ftp ftp.cdc.gov
Connected to ftp.cdc.gov.
220 Microsoft FTP Service
Name (ftp.cdc.gov:analyst): anonymous
331 Anonymous access allowed, send identity (e-mail name) as password.
Senha:
230 User logged in.
O tipo de sistema remoto é Windows_nt.
ftp>
```

Etapa 2: Baixar o arquivo Readme (Leiame).

a. Digite o comando **ls** para listar os arquivos, em seguida, localize e baixe o arquivo Readme (Leiame).

```
ftp> ls
200 PORT command successful.
125 Data connection already open; Transfer starting.
-rwxrwxrwx 1 owner group 128 May 9 1995 .change.dir
-rwxrwxrwx 1 owner group 107 May 9 1995 .message
drwxrwxrwx 1 owner group 0 Feb 2 11:21 pub
-rwxrwxrwx 1 owner group 1428 May 13 1999 Readme
rwxrwxrwx 1 owner group 383 May 13 1999 Siteinfo
-rwxrwxrwx 1 owner group 0 May 17 2005 up.htm
drwxrwxrwx 1 owner group 0 May 20 2010 w3c
-rwxrwxrwx 1 owner group 202 Sep 22 1998 welcome.msg
226 Transfer complete.
```

Observação: Você pode receber as seguintes mensagens:

```
421 Service not available, remote server has closed connection
ftp: No control connection for command

501 Server cannot access argument
500 command not understood
ftp: bind: Address already in use
```

Se isso acontecer, então o servidor FTP está inativo no momento. No entanto, você pode prosseguir com o resto do laboratório analisando os pacotes que você foi capaz de capturar e ler junto para pacotes que você não capturou. Você também pode retornar ao laboratório mais tarde para ver se o servidor FTP está de backup.

b. Digite o comando **get Readme** para baixar o arquivo. Quando o download do arquivo estiver completo, digite o comando **quit** para sair. (**Observação**: se você não conseguir fazer o download do arquivo, você pode prosseguir com o resto do laboratório.)

```
ftp> get Readme
200 PORT command successful.
125 Data connection already open; Transfer starting.
AVISO! 36 bare linefeeds received in ASCII mode
O arquivo pode não ter sido transferido corretamente.
226 Transfer complete.
1428 bytes received in 0.056 seconds (24.9 kbytes/s)
```

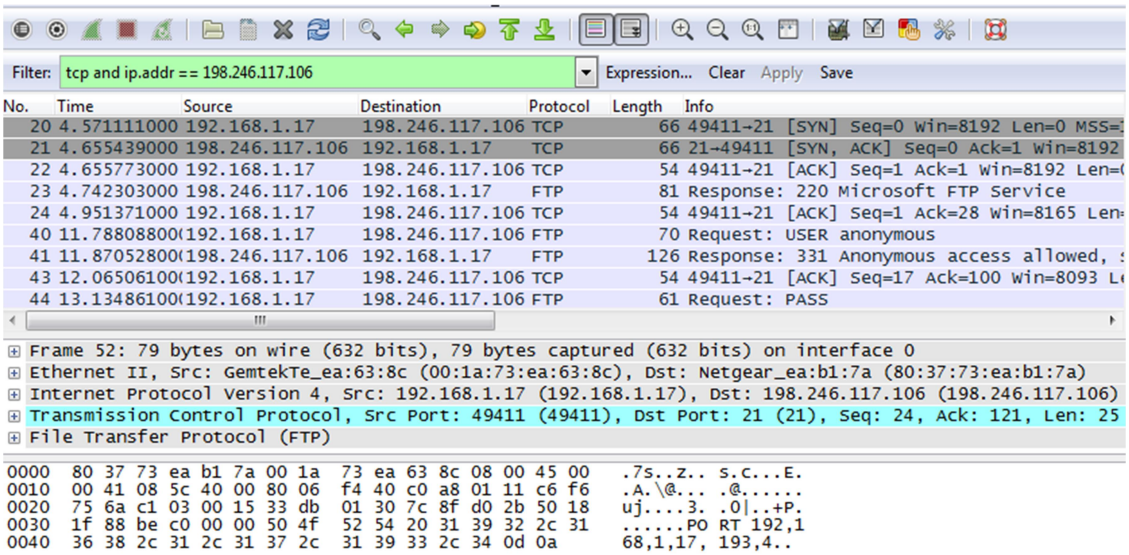
c. Após a conclusão da transferência, digite **quit**para sair do ftp.

Etapa 3: Parar a captura do Wireshark.

Etapa 4: Exibir a janela principal do Wireshark.

O Wireshark capturou muitos pacotes durante a sessão FTP com ftp.cdc.gov. Para limitar a quantidade de dados para análise, aplique o filtro **tcp e ip.addr == 198.246.117.106** e clique em **Apply**.

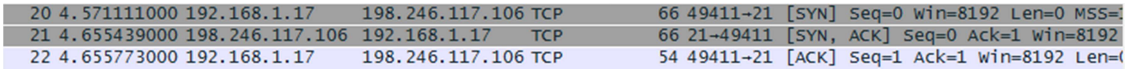
Nota: O endereço IP, 198.246.117.106, é o endereço de ftp.cdc.gov na época em que este laboratório foi criado. O endereço IP pode ser diferente para você. Se assim for, procure o primeiro pacote TCP que iniciou o handshake de 3 vias com ftp.cdc.gov. O endereço IP de destino é o endereço IP que você deve usar para o filtro.



Observação: sua interface Wireshark pode parecer um pouco diferente da imagem acima.

Etapa 5: Analisar os campos TCP.

Depois que o filtro TCP foi aplicado, os três primeiros pacotes (seção superior) exibem a sequência de [SYN], [SYN, ACK] e [ACK] que é o handshake de três vias TCP.

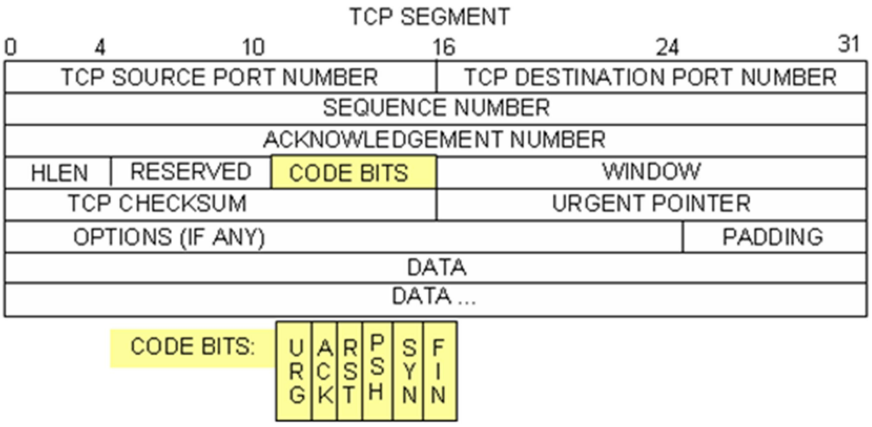


O TCP é comumente usado durante uma sessão para controlar a entrega de datagramas, verificar a chegada de datagramas e gerenciar o tamanho da janela. Em cada troca de dados entre o cliente FTP e o servidor FTP, uma nova sessão TCP é iniciada. Com a conclusão da transferência de dados, a sessão TCP é fechada. Quando a sessão FTP é finalizada, o TCP desempenha ordenadamente um fechamento e um término.

No Wireshark, as informações detalhadas do TCP estão disponíveis no painel de detalhes do pacote (seção do meio). Realce o primeiro datagrama TCP do computador host e expanda as partes do datagrama TCP, conforme mostrado abaixo.

```
⊞ 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
```

O datagrama TCP expandido é semelhante ao painel de detalhes do pacote, conforme mostrado abaixo.



A imagem acima é um diagrama de um datagrama TCP. Uma explicação de cada campo é fornecida para referência:

- O **TCP Source Port Number** (Número da porta TCP origem) pertence ao host da sessão TCP que abriu uma conexão. O valor é geralmente um valor aleatório acima de 1.023.
- O **TCP Destination Port Number** (Número da porta TCP destino) é usado para identificar o protocolo de camada superior ou aplicação no site remoto. Os valores no intervalo 0-1.023 representam as “portas bem conhecidas” e estão associados a serviços e aplicações populares (conforme descrito na RFC 1700, tais como Telnet, FTP e HTTP). A combinação do endereço IP de origem, porta de origem, endereço IP de destino e porta de destino identifica de forma exclusiva a sessão para o remetente e o destinatário.

Observação: na captura do Wireshark acima, a porta de destino é 21, que é FTP. Os servidores FTP ouvem a porta 21 para conexões de clientes FTP.

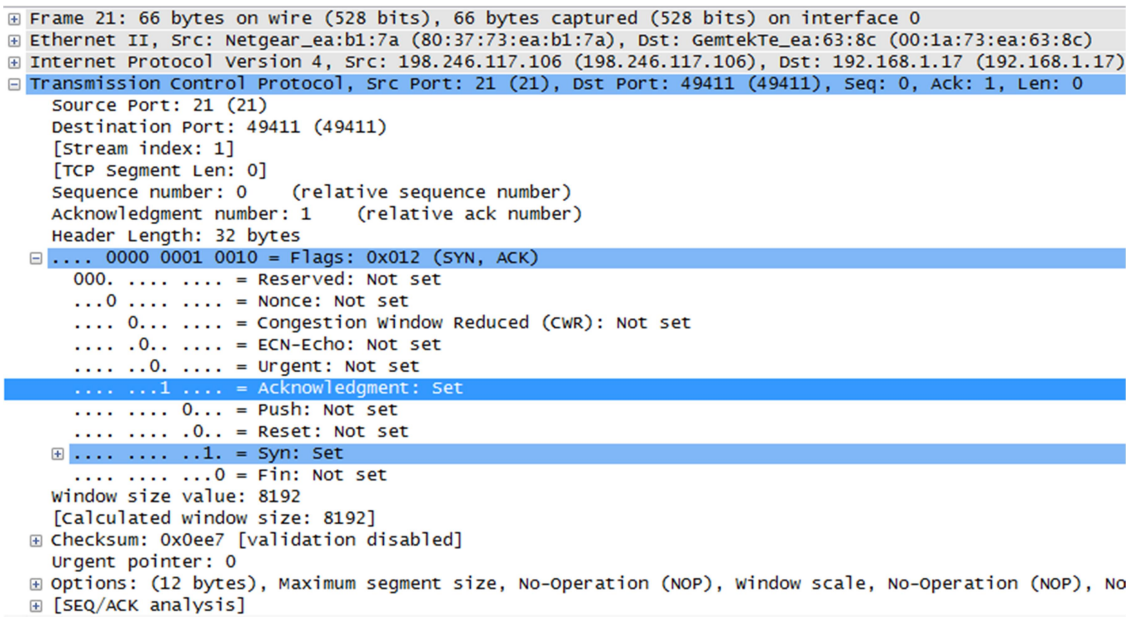
- O **Sequence number** (Número de sequência) especifica o número do último octeto em um segmento.
- O **Acknowledgment number** (Número de confirmação) especifica o próximo octeto esperado pelo destinatário.
- Os **Code bits** (Bits de Código) possuem um significado especial no gerenciamento de sessão e no tratamento de segmentos. Entre valores interessantes estão:
  - **ACK** — Confirmação de recebimento de segmento.
  - **SYN** — Sincronizar, apenas definido quando uma nova sessão TCP é negociada durante o handshake TCP de três vias.
  - **FIN** — Concluir, a solicitação para fechar a sessão TCP.
- O **Window size** (Tamanho da Janela) é o valor da janela deslizante. Ele determina quantos octetos podem ser enviados antes de se esperar por uma confirmação.
- O **Urgent pointer** (Ponteiro de Urgência) só é usado com um flag URG (Urgente) quando o remetente precisa enviar dados urgentes ao destinatário.
- As **Opções** têm apenas uma opção atualmente e estão definidas como o tamanho máximo de segmento TCP (valor opcional).

Usando a captura do Wireshark da primeira inicialização da sessão TCP (bit SYN em 1), preencha as informações sobre o cabeçalho TCP. Alguns campos podem não se aplicar a este pacote.

Da VM para o servidor CDC (apenas o bit SYN é definido como 1):

Descrição	Resultados Wireshark
Endereço IP origem	
Endereço IP destino	
Número da porta de origem	
Número da porta de destino	
Número de sequência	
Número de confirmação:	
Tamanho do cabeçalho	
Tamanho da janela	

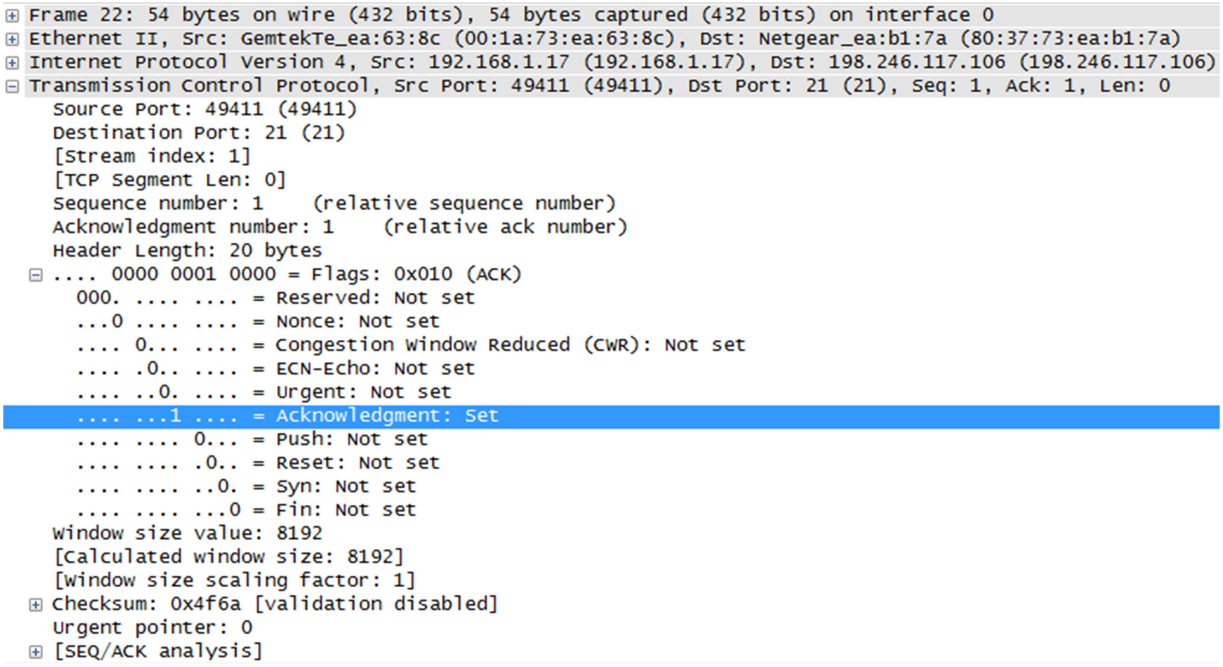
Na segunda captura filtrada do Wireshark, o servidor FTP do CDC reconhece a solicitação da VM. Observe os valores dos bits SYN e ACK.



Preencha as seguintes informações com relação à mensagem de SYN-ACK.

Descrição	Resultados Wireshark
Endereço IP origem	
Endereço IP destino	
Número da porta de origem	
Número da porta de destino	
Número de sequência	
Número de confirmação:	
Tamanho do cabeçalho	
Tamanho da janela	

No estágio final da negociação para estabelecer a comunicação, a VM envia uma mensagem de confirmação ao servidor. Observe que apenas o bit ACK está definido como 1 e o número de sequência foi incrementado para 1.

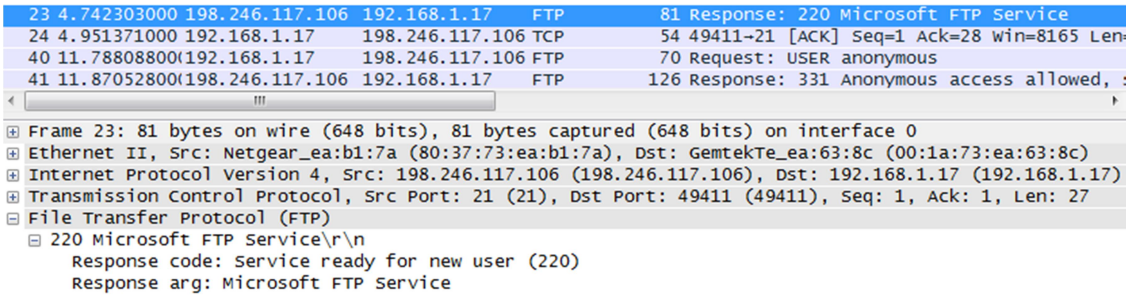


Preencha as seguintes informações com relação à mensagem de ACK.

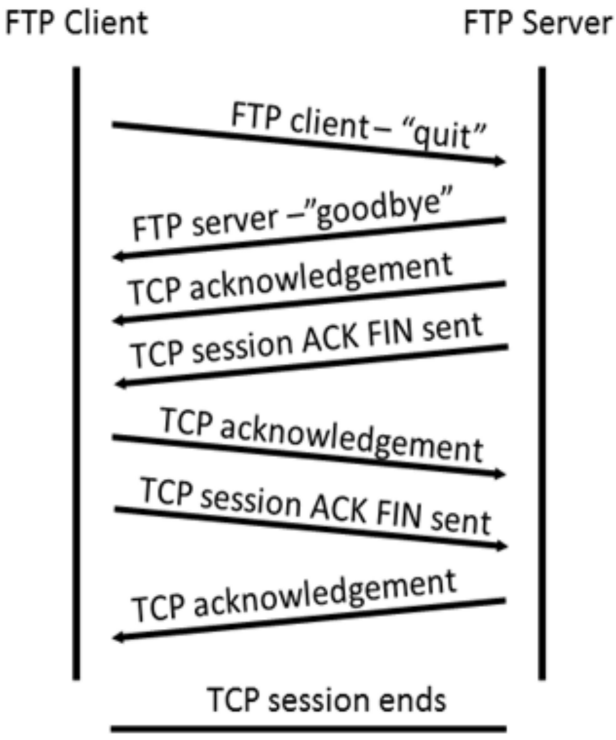
Descrição	Resultados Wireshark
Endereço IP origem	
Endereço IP destino	
Número da porta de origem	
Número da porta de destino	
Número de sequência	
Número de confirmação:	
Tamanho do cabeçalho	
Tamanho da janela	

Quantos outros datagramas TCP continham um bit SYN?

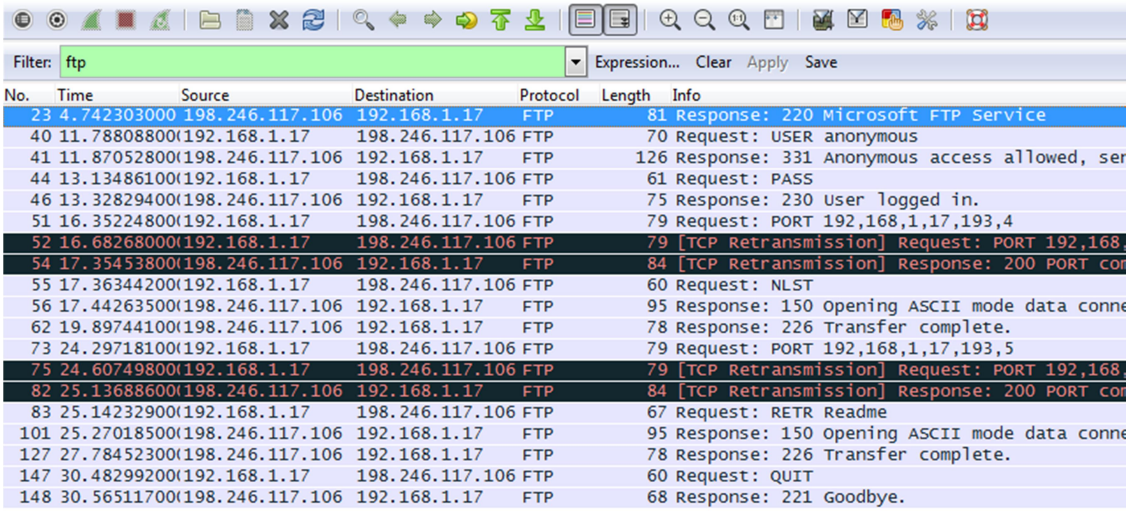
Após uma sessão TCP ser estabelecida, o tráfego FTP pode ocorrer entre o computador e o servidor FTP. O cliente e o servidor FTP se comunicam, sem saber que o TCP possui o controle e o gerenciamento sobre a sessão. Quando o servidor FTP envia uma *Response: 220* ao cliente FTP, a sessão TCP no cliente FTP envia uma confirmação à sessão TCP no servidor. Essa sequência é visível na captura do Wireshark abaixo.



Com a finalização da sessão FTP, o cliente FTP envia um comando “quit”. O servidor FTP confirma o término do FTP com *Response: 221 Goodbye*. Neste momento, a sessão TCP do servidor FTP envia um datagrama TCP ao cliente FTP, anunciando o término da sessão TCP. A sessão TCP do cliente FTP confirma o recebimento do datagrama de término, então, envia seu próprio término da sessão TCP. Quando o originador do término TCP (o servidor FTP) recebe um término duplicado, um datagrama ACK é enviado para confirmar o término e a sessão TCP é fechada. Essa sequência é visível na captura e no diagrama abaixo.



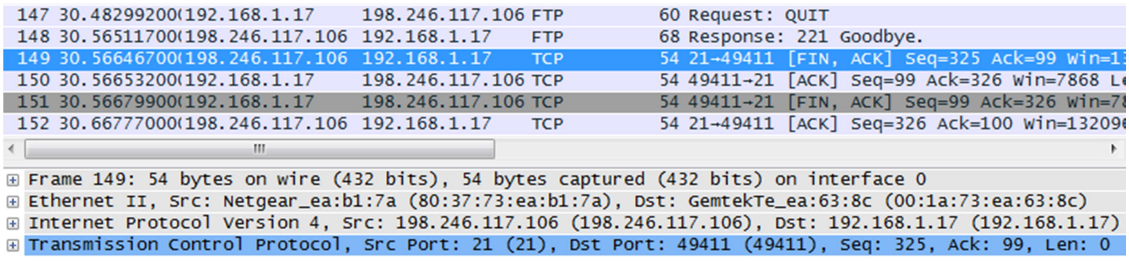
Ao aplicar um filtro **ftp**, toda a sequência do tráfego FTP pode ser examinada no Wireshark. Observe a sequência dos eventos durante esta sessão FTP. O nome de usuário **anonymous** foi usado para recuperar o arquivo Readme (Leia-me). Quando a transferência for concluída, o usuário terá concluído a sessão FTP.



Aplique o filtro TCP novamente no Wireshark para examinar o término da sessão TCP. Quatro pacotes são transmitidos para o encerramento da sessão TCP. Como a conexão TCP é full duplex, cada direção deve terminar de forma independente. Examine os endereços origem e destino.

Neste exemplo, o servidor FTP não tem mais dados para enviar na transmissão. Envia um segmento com o flag FIN ligado no quadro 149. O computador envia uma ACK para confirmar o recebimento do FIN para terminar a sessão do servidor com o cliente no quadro 150.

No quadro 151, o computador envia um FIN para o servidor FTP para terminar a sessão TCP. O servidor FTP responde com um ACK para confirmar o FIN do computador no quadro 152. Agora a sessão TCP é encerrada entre o servidor FTP e o PC.



Parte 2: Identificar os Campos do Cabeçalho e a Operação UDP Usando uma Captura de Sessão TFTP do Wireshark

Na Parte 2, use o Wireshark para capturar uma sessão TFTP e inspecionar os campos do cabeçalho UDP.

Etapa 1: Inicie o serviço Mininet e tftpd.

a. Inicie a Mininet. Digite **cyberops** como a senha quando solicitado.

```
[analyst@secOps ~]$ sudo lab.support.files/scripts/cyberops_topo.py
[sudo] password for analyst:
```

b. Inicie H1 e H2 no **mininet>**prompt.

```
*** Starting CLI:
```

```
mininet> xterm H1 H2
```

c. Na janela do terminal **H1**, inicie o servidor tftpd usando o script fornecido.

```
[root@secOps analyst]# /home/analyst/lab.support.files/scripts/start_tftpd.sh
[root@secOps analyst]#
```

Etapa 2: Crie um arquivo para transferência tftp.

a. Crie um arquivo de texto no prompt de terminal **H1** na pasta /srv/tftp/.

```
[root@secOps analyst]# echo "This file contains my tftp data." >
/srv/tftp/meu_tftp_data
```

b. Verifique se o arquivo foi criado com os dados desejados na pasta.

```
[root@secOps analyst]# cat /srv/tftp/my_tftp_data
Este arquivo contém meus dados tftp.
```

c. Por causa da medida de segurança para este servidor tftp específico, o nome do arquivo de recebimento já precisa existir. Em **H2**, crie um arquivo chamado **my_tftp_data**.

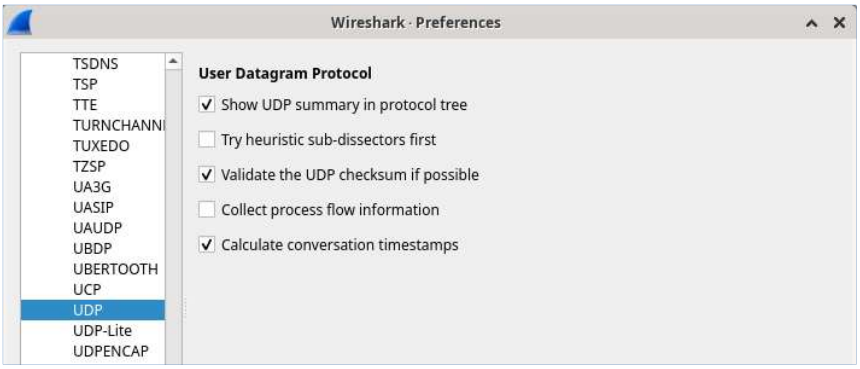
```
[root@secOps analyst]# touch my_tftp_data
```

Etapa 3: Capturar uma sessão TFTP no Wireshark

a. Inicie o Wireshark em **H1**.

```
[root@secOps analyst]# wireshark &
```

b. No menu **Edit**, escolha **Preferences** e clique na seta para expandir **Protocols**. Role para baixo e selecione **UDP**. Clique na **caixa de seleção Validar a soma de verificação UDP se possível** e clique em **OK**.

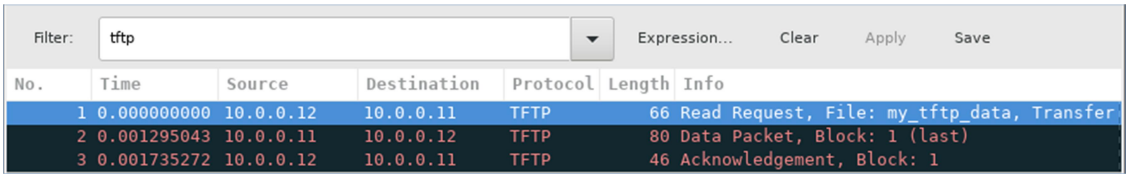


c. Inicie uma captura Wireshark na interface **H1-eth0**.

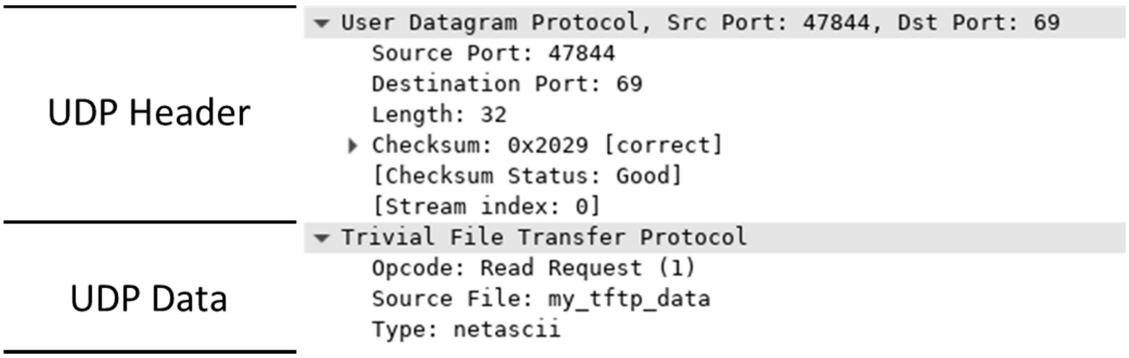
d. Inicie uma sessão tftp do **H2** para o servidor tftp em **H1** e obtenha o arquivo **my_tftp_data**.

```
[root@secOps analyst]# tftp 10.0.0.11 -c get my_tftp_data
```

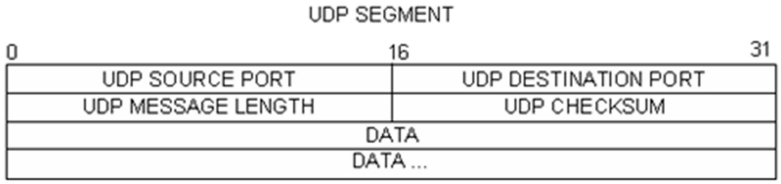
e. Parar a captura do Wireshark. Defina o filtro como **tftp** e clique em **Apply**. Use os três pacotes TFTP para preencher a tabela e responder às perguntas no resto do laboratório.



Informações detalhadas do UDP estão disponíveis no painel de detalhes do pacote do Wireshark. Destaque o primeiro datagrama UDP enviado pelo host e mova o cursor do mouse para o painel de detalhes do pacote. Pode ser necessário ajustar o painel de detalhes do pacote e expandir o registro UDP clicando na caixa de expansão do protocolo. O datagrama UDP expandido deve ser semelhante ao diagrama abaixo.



A figura a seguir é um diagrama do datagrama UDP. As informações de cabeçalho são escassas, em comparação ao datagrama TCP. Assim como o TCP, cada datagrama UDP é identificado pela porta de origem UDP e pela porta de destino UDP.



Usando a captura Wireshark do primeiro datagrama UDP, preencha as informações sobre o cabeçalho UDP. O valor de checksum é um valor hexadecimal (base 16), denotado pelo código precedente 0x:

Descrição	Resultados Wireshark
Endereço IP de origem	
Endereço IP de destino	
Número da porta de origem	
Número da porta de destino	
Tamanho da mensagem UDP	
Checksum UDP	

Como o UDP verifica a integridade do datagrama?

Examine o primeiro quadro retornado pelo servidor tftpd. Preencha as informações sobre o cabeçalho UDP.

Descrição	Resultados Wireshark
Endereço IP de origem	
Endereço IP de destino	
Número da porta de origem	
Número da porta de destino	
Tamanho da mensagem UDP	
Checksum UDP	

Note que o datagrama UDP de retorno tem uma porta de origem UDP diferente, mas esta porta de origem é usada para o restante da transferência TFTP. Por não haver conexão confiável, somente a porta de origem original usada para iniciar a sessão TFTP é usada para manter a transferência TFTP.

Observe também que o checksum UDP está incorreto. Isso é causado provavelmente pelo checksum offload UDP. Você pode saber mais sobre por que isso acontece pesquisando por " checksum offload UDP".

Etapa 4: Limpeza

Nesta etapa, você desligará e limpará a Mininet.

a. No terminal que iniciou o Mininet, digite **quit** no prompt.

```
mininet> quit
```

b. No prompt, digite **sudo mn -c** para limpar os processos iniciados pela Mininet.

```
[analyst@secOps ~]$ sudo mn -c
```

Perguntas para reflexão

Este laboratório proporcionou a oportunidade de analisar operações dos protocolos TCP e UDP das sessões FTP e TFTP capturadas. Como o TCP gerencia a comunicação de forma diferente do UDP?