Laboratório 4.5.2: Protocolos da Camada de Transporte TCP/IP, TCP e UDP

Diagrama de Topologia

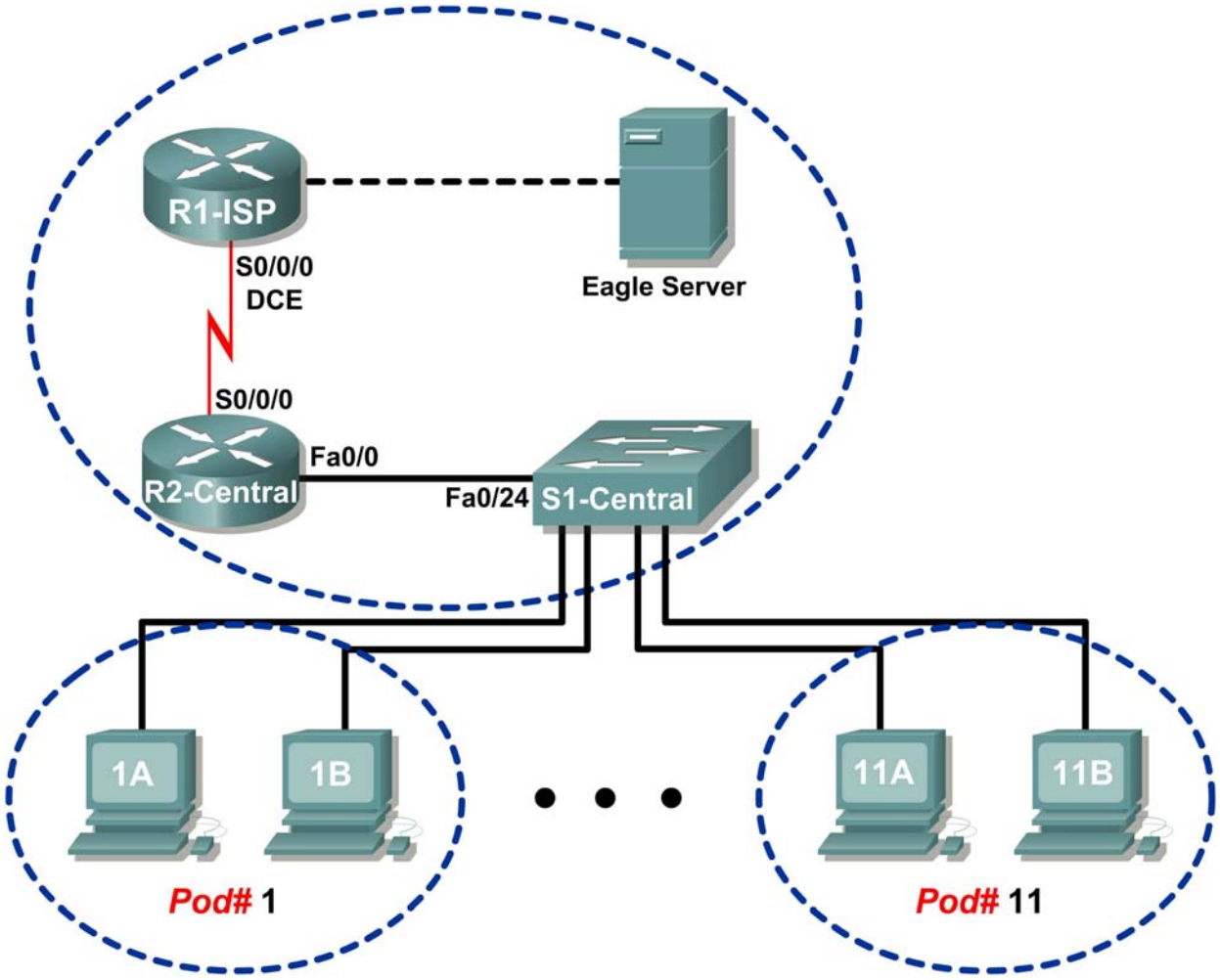

Tabela de Endereçamento

Dispositivo	Interface	Endereço IP	Máscara de Sub-Rede	Gateway Padrão
R1-ISP	**S0/0/0**	**10.10.10.6**	**255.255.255.252**	*N/A*
	Fa0/0	**192.168.254.253**	**255.255.255.0**	*N/A*
R2-Central	**S0/0/0**	**10.10.10.5**	**255.255.255.252**	*N/A*
	Fa0/0	**172.16.255.254**	**255.255.0.0**	*N/A*
Eagle Server	*N/A*	**192.168.254.254**	**255.255.255.0**	**192.168.254.253**
	N/A	**172.31.24.254**	**255.255.255.0**	*N/A*
host*Pod#***A**	*N/A*	**172.16.***Pod#***.1**	**255.255.0.0**	**172.16.255.254**
host*Pod#***B**	*N/A*	**172.16.***Pod#***.2**	**255.255.0.0**	**172.16.255.254**
S1-Central	*N/A*	**172.16.254.1**	**255.255.0.0**	**172.16.255.254**

Objetivos

- Identificar campos do cabeçalho TCP e sua operação usando uma captura de sessão FTP do Wireshark.
- Identificando campos do cabeçalho UDP e sua operação usando a captura de uma sessão TFTP Wireshark.

Contexto

Os dois protocolos na Camada de Transporte TCP/IP são o Transmission Control Protocol (TCP), definido no RFC 761, Janeiro de 1980, e o User Datagram Protocol (UDP), definido no RFC 768, Agosto de 1980. Ambos os protocolos suportam comunicação de protocolo 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 Domain Name Services (DNS) e para o Trivial File Transfer Protocol (TFTP), entre outros.

A capacidade de se entender os campos dos cabeçalhos TCP e UDP e sua operação é uma habilidade crucial para engenheiros de rede.

Cenário

Usando a captura Wireshark, analise os campos do cabeçalho dos protocolos TCP e UDP para transferências de arquivos entre o computador e o Eagle Server. Se o Wireshark não tiver instalado no computador, você pode fazer o download da URL ftp://eagle-server.example.com/pub/eagle_labs/eagle1/chapter4/, arquivo `wireshark-setup-0.99.4.exe`.

Os utilitários de linha de comando do Windows `ftp` e `tftp` serão usados para se conectar ao Eagle Server e fazer o download de arquivos.

Tarefa 1: Identificando Campos do Cabeçalho TCP e sua Operação usando uma Captura de Sessão FTP do Wireshark

Passo 1: Capturar uma sessão FTP.

Sessões TCP são bem controladas e gerenciadas por informações trocadas nos campos do cabeçalho TCP. Nesta tarefa, uma sessão FTP será feita ao Eagle Server. Ao finalizar, a captura da sessão será analisada. Computadores Windows usam o cliente FTP, **`ftp`**, para se conectar ao servidor FTP. Uma janela de linha de comando iniciará a sessão FTP e será feito o download do arquivo de configuração de texto do S1-central do Eagle Server, `/pub/eagle_labs/eagle1/chapter4/s1-central`, para o computador.

Abra uma janela de linha de comando clicando em Iniciar | Executar, digite **`cmd`** e pressione OK.

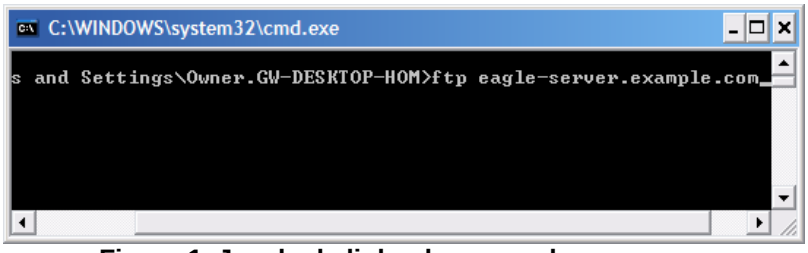

Figura 1. Janela de linha de comando

Uma janela similar à Figura 1 deverá ser aberta.

Inicie uma captura Wireshark na interface que possui o endereço IP `172.16.`*`Pod#`*`.[1-2]`.

Inicie uma conexão FTP ao Eagle Server. Digite o comando:

```
>ftp eagle-server.example.com
```

Quando solicitado um ID de usuário, digite **`anônimo`**. Quando solicitada uma senha, pressione **`<ENTER>`**.

Altere o diretório FTP para `/pub/eagle_labs/eagle1/chapter4/`:

```
ftp> cd /pub/eagle_labs/eagle1/chapter4/
```

Faça o download do arquivo `s1-central`:

```
ftp> get s1-central
```

Quando finalizado, finalize as sessões FTP em cada janela de linha de comando com o comando **`quit`**:

```
ftp> parar
```

Feche a janela de linha de comando com o comando **`exit`**:

```
> exit
```

Pare a captura do Wireshark.

Passo 2: Analisar os campos TCP.

No.	Time	Source	Destination	Protocol	Info
1	0.000000	172.16.1.1	192.168.254.254	TCP	1052 > ftp [SYN] Seq=0 Len=0 MSS=1460
2	0.000568	192.168.254.254	172.16.1.1	TCP	ftp > 1052 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1460
3	0.000610	172.16.1.1	192.168.254.254	TCP	1052 > ftp [ACK] Seq=1 Ack=1 Win=64240 Len=0
4	0.004818	192.168.254.254	172.16.1.1	FTP	Response: 220 Welcome to the eagle-server FTP service.
5	0.115430	172.16.1.1	192.168.254.254	TCP	1052 > ftp [ACK] Seq=1 Ack=47 Win=64194 Len=0
6	8.223541	172.16.1.1	192.168.254.254	FTP	Request: USER anonymous
7	8.224089	192.168.254.254	172.16.1.1	TCP	ftp > 1052 [ACK] Seq=47 Ack=17 Win=5840 Len=0
8	8.224126	192.168.254.254	172.16.1.1	FTP	Response: 331 Please specify the password.
9	8.327214	172.16.1.1	192.168.254.254	TCP	1052 > ftp [ACK] Seq=17 Ack=81 Win=64160 Len=0
10	9.517629	172.16.1.1	192.168.254.254	FTP	Request: PASS
11	9.519135	192.168.254.254	172.16.1.1	FTP	Response: 230 Login successful.
12	9.629097	172.16.1.1	192.168.254.254	TCP	1052 > ftp [ACK] Seq=24 Ack=104 Win=64137 Len=0
13	32.365752	172.16.1.1	192.168.254.254	FTP	Request: CWD /pub/eagle_labs/eagle1/chapter4
14	32.366375	192.168.254.254	172.16.1.1	FTP	Response: 250 Directory successfully changed.
15	32.376653	172.16.1.1	192.168.254.254	FTP	Request: PORT 172,16,1,1,4,33
16	32.377165	192.168.254.254	172.16.1.1	FTP	Response: 200 PORT command successful. Consider using PASV.
17	32.381726	172.16.1.1	192.168.254.254	FTP	Request: RETR s1-central
18	32.382337	192.168.254.254	172.16.1.1	TCP	ftp-data > 1057 [SYN] Seq=0 Len=0 MSS=1460 TSV=4755496 TSER=0 WS=2
19	32.382398	172.16.1.1	192.168.254.254	TCP	1057 > ftp-data [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 WS=0 TSV=0 TSER=0
20	32.382777	192.168.254.254	172.16.1.1	TCP	ftp-data > 1057 [ACK] Seq=1 Ack=1 Win=5840 Len=0 TSV=4755496 TSER=0
21	32.382891	192.168.254.254	172.16.1.1	FTP	Response: 150 Opening BINARY mode data connection for s1-central (3100 bytes).
22	32.383528	192.168.254.254	172.16.1.1	FTP-DATA	FTP Data: 1448 bytes
23	32.383589	192.168.254.254	172.16.1.1	FTP-DATA	FTP Data: 1448 bytes
24	32.383631	172.16.1.1	192.168.254.254	TCP	1057 > ftp-data [ACK] Seq=1 Ack=2897 Win=64240 Len=0 TSV=36854 TSER=4755496
25	32.383736	192.168.254.254	172.16.1.1	FTP-DATA	FTP Data: 204 bytes
26	32.383753	192.168.254.254	172.16.1.1	FTP	Response: 226 File send OK.
27	32.383773	172.16.1.1	192.168.254.254	TCP	1052 > ftp [ACK] Seq=100 Ack=281 Win=63960 Len=0
28	32.383779	192.168.254.254	172.16.1.1	TCP	ftp-data > 1057 [FIN, ACK] Seq=3101 Ack=1 Win=5840 Len=0 TSV=4755496 TSER=0
29	32.383805	172.16.1.1	192.168.254.254	TCP	1057 > ftp-data [ACK] Seq=1 Ack=3102 Win=64036 Len=0 TSV=36854 TSER=4755496
30	32.389457	172.16.1.1	192.168.254.254	TCP	1057 > ftp-data [FIN, ACK] Seq=1 Ack=3102 Win=64036 Len=0 TSV=36854 TSER=4755496
31	32.389845	192.168.254.254	172.16.1.1	TCP	ftp-data > 1057 [ACK] Seq=3102 Ack=2 Win=5840 Len=0 TSV=4755503 TSER=36854
32	34.438952	172.16.1.1	192.168.254.254	FTP	Request: QUIT
33	34.439532	192.168.254.254	172.16.1.1	FTP	Response: 221 Goodbye.
34	34.439893	192.168.254.254	172.16.1.1	TCP	ftp > 1052 [FIN, ACK] Seq=295 Ack=106 Win=5840 Len=0
35	34.439934	172.16.1.1	192.168.254.254	TCP	1052 > ftp [ACK] Seq=106 Ack=296 Win=63946 Len=0
36	34.442705	172.16.1.1	192.168.254.254	TCP	1052 > ftp [FIN, ACK] Seq=106 Ack=296 Win=63946 Len=0
37	34.443144	192.168.254.254	172.16.1.1	TCP	ftp > 1052 [ACK] Seq=296 Ack=107 Win=5840 Len=0

Figura 2. Captura FTP

Alterne as janelas de captura do Wireshark. A janela superior contém informações resumidas para cada registro capturado. A captura do aluno deve ser similar à captura exibida na Figura 2. Antes de se aprofundar nos detalhes do pacote TCP, é necessária uma explicação das informações resumidas. Quando o cliente FTP estiver conectado ao servidor FTP, o protocolo TCP de Camada de Transporte criou uma sessão confiável. O TCP é comumente usado durante uma sessão para controlar a entrega de datagrama, verificar a chegada de datagrama e gerenciar o tamanho da janela. Para 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. Finalmente, quando a sessão FTP é finalizada, o TCP desempenha um desligamento e término ordenadamente.

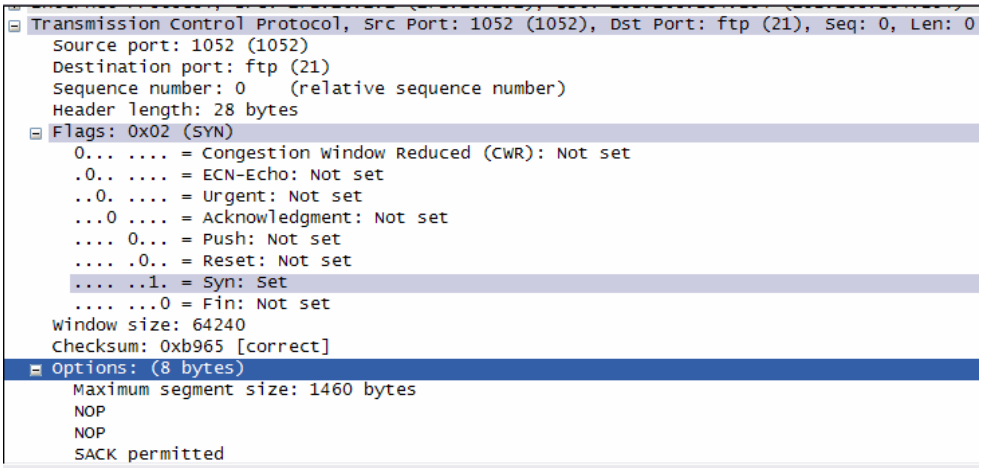

Figura 3. Captura Wireshark de um datagrama TCP

No Wireshark, informações TCP detalhadas estão disponíveis na janela do meio. Destaque o primeiro datagrama TCP do computador e mova a seta do mouse para a janela do meio. Pode ser necessário ajustar a janela do meio e expandir o registro TCP clicando na caixa de expansão de protocolo. O datagrama TCP expandido deve parecer com a Figura 3.

Como é identificado o primeiro datagrama em uma sessão TCP?

_____

_____

SEGMENTO TCP

0	4	10	16	24	31
NÚMERO DE PORTA DE ORIGEM TCP			NÚMERO DE PORTA DE DESTINO TCP		
NÚMERO DE SEQUENCIA					
NÚMERO DE CONFIRMAÇÃO					
TAMANHO DO CABEÇALHO			RESERVADO	BITS DE CÓDIGO	TAMANHO DA JANELA
TCP CHECKSUM			PONTEIRO DE URGÊNCIA		
OPÇÕES (SE HOUVER ALGUMA)			PREENCHIMENTO		
DADOS					
DADOS...					

BITS DE CÓDIGO	URG	ACK	RST	PSH	SYN	FIN

Figura 4. Campos do pacote TCP

Consulte a Figura 4, um diagrama de datagrama TCP. Fornecemos uma explicação de cada campo para refrescar a memória do aluno:

- **O Número de porta de Origem TCP** pertence ao host da sessão TCP que abriu uma conexão. O valor é geralmente um valor aleatório acima de 1023.
- **O Número de porta de destino** é usado para identificar o protocolo de camada superior ou aplicação no site remoto. Os valores no intervalo 0–1023 representam as chamadas “portas conhecidas” e estão associados a serviços e aplicações populares (conforme descrito no RFC 1700, tais como Telnet, File Transfer Protocol (FTP), HyperText Transfer Protocol (HTTP), etc). A combinação quádrupla do campo (Endereço IP de Origem, Porta de Origem, Endereço IP de Destino, Porta de Destino) identifica unicamente a sessão para o remetente e o receptor.
- **O Número de seqüência** especifica o número do último octeto em um segmento.
- **O Número de confirmação** especifica o próximo octeto esperado pelo receptor.
- **Os Bits de Código** possuem um significado especial em gerenciamento de sessão e no tratamento de segmentos. Entre valores interessantes estão:
  - ACK (Confirmação do recebimento de um segmento),
  - SYN (Sincronizar, somente estabelecido quando uma nova sessão TCP é negociada durante o handshake triplo do TCP).
  - FIN (Finalizar, solicitação para fechar a sessão TCP).
- **O Tamanho da janela** é o valor da janela – quantos octetos podem ser enviados antes de se esperar por uma confirmação.
- **O Ponteiro de Urgência** só é usado com uma flag URG (Urgente) – quando o remetente precisa enviar dados urgentes ao receptor.
- **Opções**: A única opção atualmente definida é o tamanho máximo de segmento TCP (valor opcional).

Usando a captura Wireshark da primeira inicialização da sessão TCP (bit SYN definido para 1), preencha as informações sobre o cabeçalho TCP:

Do computador ao Eagle Server (somente o bit SYN é definido para 1):

Endereço IP de Origem: 172.16.___.___	
Endereço IP de Destino: _____	
Número de porta de origem:_____ _____	
Número de porta de destino: _____	
Número de seqüência: _____	
Número de confirmação: _____	
Tamanho do cabeçalho: _____	
Tamanho da janela: _____	

Do Eagle Server ao computador (somente bits SYN e ACK são definidos para 1):

Endereço IP de Origem: _____	
Endereço IP de Destino: 172.16.___.___	
Número de porta de origem:_____ _____	
Número de porta de destino: _____	
Número de seqüência: _____	
Número de confirmação: _____	
Tamanho do cabeçalho: _____	
Tamanho da janela: _____	

Do computador ao Eagle Server (somente bit ACK é definido para 1):

Endereço IP de Origem: 172.16.___.___	
Endereço IP de Destino: _____	
Número de porta de origem:_____ _____	
Número de porta de destino: _____	
Número de seqüência: _____	
Número de confirmação: _____	
Tamanho do cabeçalho: _____	
Tamanho da janela: _____	

Ignorando a sessão TCP iniciada quando houve uma transferência de dados, quantos outros datagramas TCP continham um bit SYN?

_____

_____

Atacantes tiram vantagem do handshake triplo iniciando uma conexão “meio aberta”. Nesta seqüência, a sessão de abertura TCP envia um datagrama TCP com o bit SYN definido e o receptor envia um datagrama TCP relacionado com os bits SYN ACK definidos. Um bit final ACK nunca é enviado para finalizar o handshake TCP. Ao invés disso, uma nova conexão TCP é iniciada de modo meio aberto. Com sessões TCP suficientes no estado meio aberto, o computador receptor pode esgotar recursos e bloquear. Um bloqueio poderia envolver uma perda de serviços de rede ou o sistema operacional corrompido. Em quaisquer desses casos, o atacante venceu, o serviço de rede foi parado no receptor. Este é um exemplo de um ataque de negação de serviço (DoS).

Figura 5. Gerenciamento da sessão TCP

O cliente e o servidor FTP se comunicam um entre o outro, alheios e sem se preocupar que o TCP possui o controle e o gerenciamento sobre a sessão. Quando o servidor FTP envia uma Resposta: 220 ao cliente FTP, a sessão TCP no cliente FTP envia uma confirmação à sessão TCP no Eagle Server. Essa seqüência é exibida na Figura 5 e é visível na captura Wireshark.

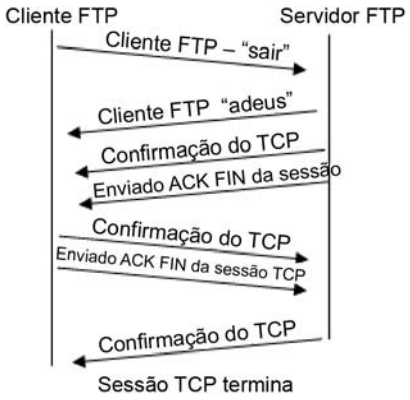

Figura 6. Término ordenado da sessão TCP

Com a finalização da sessão FTP, o cliente FTP envia um comando **`sair`**. O servidor FTP confirma o término FTP com uma `Resposta :221 Adeus`. 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. Esta seqüência é exibida na Figura 6 e é visível na captura Wireshark.

Sem um término ordenado, tal como quando a conexão é quebrada, as sessões TCP esperarão certo período de tempo até o fechamento. O valor de limite de tempo padrão varia, mas é geralmente de 5 minutos.

Tarefa 2: Identificando campos do cabeçalho UDP e sua operação usando a captura de uma sessão TFTP Wireshark

Passo 1: Capturar uma sessão TFTP.

Após o procedimento da Tarefa 1 acima, abra uma janela de linha de comando. O comando TFTP possui uma sintaxe diferente do FTP. Por exemplo, não há autenticação. Além disso, existem somente dois comandos, **`get`**, para recuperar um arquivo, e **`put`**, para enviar um arquivo.

```
>tftp –help

Transfere arquivos para e de um computador remoto executando o serviço
TFTP.

TFTP [-i] host [GET | PUT] source [destination]

              -i Especifica o modo de transferência de imagem binário
                 (também chamado octeto). O arquivo é movido em modo
                 binário de imagem literalmente, byte por byte. Use
                 este modo ao transferir arquivos binários.
            host Especifica o host local ou remoto.
             GET Transfere o destino (destination) do arquivo no host
                 remoto para a fonte (source) do arquivo no host local.
             PUT Transfere a fonte (source) do arquivo no host local
                 para o destino (destination) do arquivo no host
                 remoto.
          source Especifica o arquivo a ser transferido.
     destination Especifica onde transferir o arquivo.
```

Tabela 1. Sintaxe TFTP para um cliente TFTP Windows

A Tabela 1 contém a sintaxe do cliente TFTP Windows. O servidor TFTP possui seu próprio diretório no Eagle Server, `/tftpboot`, que é diferente da estrutura de diretório suportada pelo servidor FTP. Não é suportada autenticação.

Inicie uma captura Wireshark, então, faça o download do arquivo de configuração `s1-central` do Eagle Server com o cliente TFTP Windows. O comando e a sintaxe para executar isso são exibidos abaixo:

```
>tftp eagle-server.example.com get s1-central
```

Passo 2: Analisar os campos UDP.

No.	Time	Source	Destination	Protocol	Info
1	0.000000	172.16.1.1	192.168.254.254	TFTP	Read Request, File: s1-central, Transfer type: netascii
2	0.003171	192.168.254.254	172.16.1.1	TFTP	Data Packet, Block: 1
3	0.003314	172.16.1.1	192.168.254.254	TFTP	Acknowledgement, Block: 1
4	0.003962	192.168.254.254	172.16.1.1	TFTP	Data Packet, Block: 2
5	0.004021	172.16.1.1	192.168.254.254	TFTP	Acknowledgement, Block: 2
6	0.004615	192.168.254.254	172.16.1.1	TFTP	Data Packet, Block: 3
7	0.004673	172.16.1.1	192.168.254.254	TFTP	Acknowledgement, Block: 3
8	0.005274	192.168.254.254	172.16.1.1	TFTP	Data Packet, Block: 4
9	0.005332	172.16.1.1	192.168.254.254	TFTP	Acknowledgement, Block: 4
10	0.005930	192.168.254.254	172.16.1.1	TFTP	Data Packet, Block: 5
11	0.005989	172.16.1.1	192.168.254.254	TFTP	Acknowledgement, Block: 5
12	0.006588	192.168.254.254	172.16.1.1	TFTP	Data Packet, Block: 6
13	0.006644	172.16.1.1	192.168.254.254	TFTP	Acknowledgement, Block: 6
14	0.007078	192.168.254.254	172.16.1.1	TFTP	Data Packet, Block: 7 (last)
15	0.007131	172.16.1.1	192.168.254.254	TFTP	Acknowledgement, Block: 7

Figura 7. Captura resumida de uma sessão UDP

Alterne as janelas de captura do Wireshark. A captura do aluno deve ser similar à captura exibida na Figura 7. Uma transferência TFTP será usada para analisar a operação UDP na Camada de Transporte.

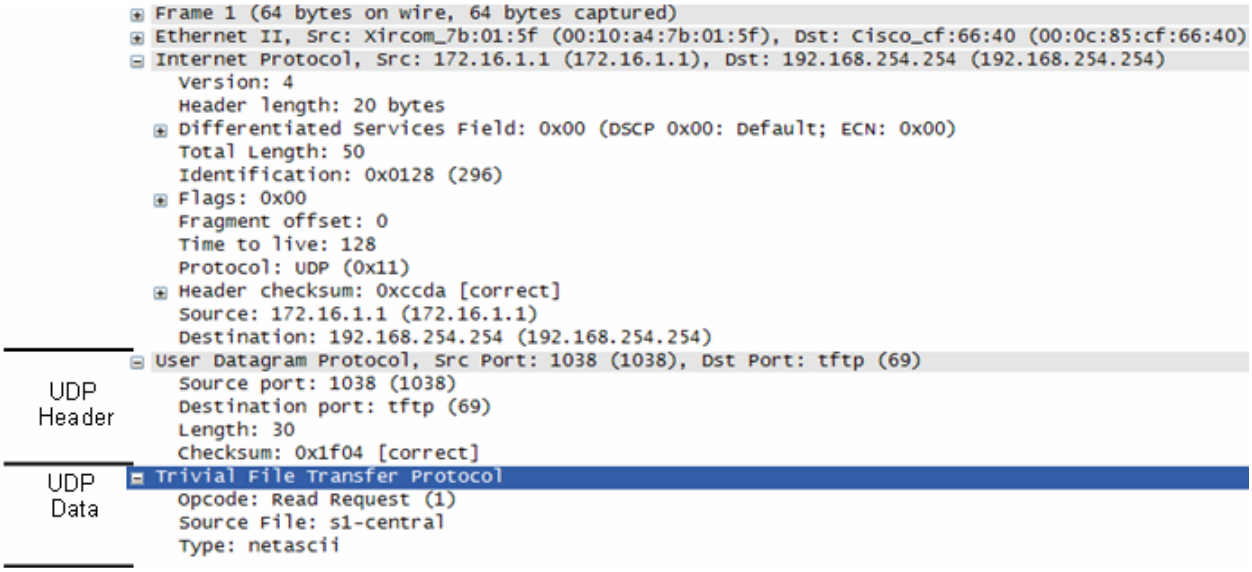

Figura 8. Captura Wireshark de um datagrama UDP

No Wireshark, informações UDP detalhadas estão disponíveis na janela do meio. Destaque o primeiro datagrama UDP do computador e mova a seta do mouse para a janela do meio. Pode ser necessário ajustar a janela do meio e expandir o registro UDP clicando na caixa de expansão de protocolo. O datagrama UDP expandido deve ser parecido com a Figura 8.

Figura 9. Formato UDP

Consulte a Figura 9, um diagrama de datagrama UDP. As informações de cabeçalho são escassas, em comparação ao datagrama TCP. No entanto, existem similaridades. 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:

Endereço IP de Origem: 172.16.___.___	
Endereço IP de Destino: _____	
Número de porta de origem:_____ _____	
Número de porta de destino: _____	
Tamanho da mensagem UDP:	

_____	
Checksum UDP: _____	

Como o UDP verifica a integridade do datagrama?

_____

Examine o primeiro pacote devolvido do Eagle Server. Preencha as informações sobre o cabeçalho UDP:

Endereço IP de Origem:	
Endereço IP de Destino: 172.16.___.___	
Número de porta de origem:_____ _____	
Número de porta de destino: _____	
Tamanho da mensagem UDP: _____	
Checksum UDP: 0x _____	

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. Como não há conexão confiável, somente a porta original de origem usada para iniciar a sessão TFTP é usada para manter a transferência TFTP.

Tarefa 5: Reflexão

Este laboratório proporcionou aos alunos a oportunidade de analisar operações de protocolos TCP e UDP das sessões FTP e TFTP capturadas. O TCP gerencia a comunicação de maneira bastante diferente do UDP, mas a confiabilidade e a entrega garantida exigem controle adicional sobre o canal de comunicação. O UDP possui menos overhead e controle e o protocolo de camada superior deve fornecer algum tipo de controle de confirmação. Ambos os protocolos, no entanto, transportam dados entre clientes e servidores usando protocolos de Camada de Aplicação e são adequados para o protocolo de camada superior que cada um suporta.

Tarefa 6: Desafio

Como nem o FTP nem o TFTP são protocolos seguros, todos os dados transferidos são enviados em texto claro. Isso inclui quaisquer IDs de usuários, senhas ou conteúdo de arquivo de texto claro. Analisar a sessão FTP de camada superior identificará rapidamente o ID do usuário, a senha e senhas de arquivos de configuração. Exame de dados TFTP de camada superior é um pouco mais complicado, mas o campo de dados pode ser examinado e o ID do usuário de configuração e as informações de senha podem ser extraídos.

Tarefa 7: Limpeza

Durante este laboratório, vários arquivos foram transferidos ao computador e devem ser removidos.

A menos que não solicitado pelo instrutor, desligue os computadores. Remova qualquer coisa que tenha sido trazida ao laboratório e deixe a sala pronta para a próxima aula.

