LABORATÓRIO WIRESHARK
HTTP

Tradução: Marjorie R. S. Rosa

2014

WIRESHARK - HTTP

Esse manual de laboratório é baseado em "Wireshark Lab: HTTP", versão 2.0 (setembro de 2009), de J.F. Kurose, K.W. Ross, disponível aqui. Isso foi preparado por Farrokh Ghani Zadegan e Niklas Carlsson, e sua última modificação foi em Janeiro de 2013.

Para esse laboratório, você precisa ter lido primeiro "Wireshark Lab: Getting Started". Esse documento e os exercícios vão ter te preparado para esse laboratório. Contudo, note que você não precisa apresentar os resultados (mencionados no pdf "Wireshark getting started"), como esses não são os resultados para essa tarefa (e aqueles exercícios devem apenas ser usados como prática).

Em segundo lugar, você vai ser questionado a responder e/ou discutir um número de perguntas. Para poupar tempo, é importante que você cuidadosamente leia as instruções de forma que você possa fornecer respostas em formato adequado.

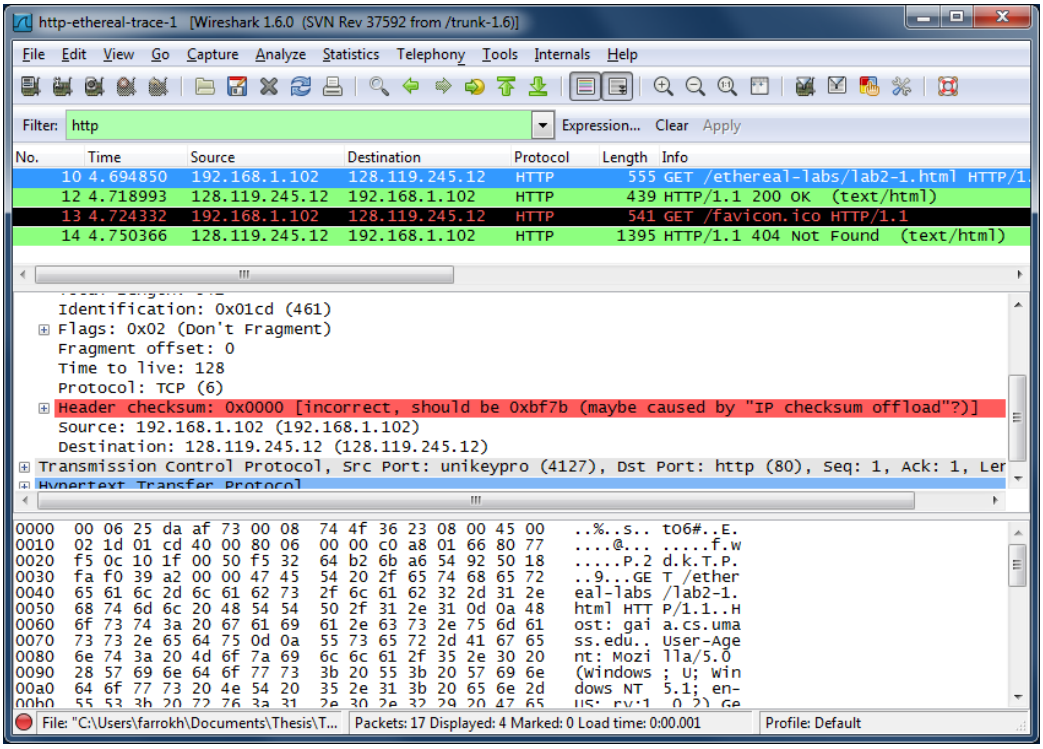
Além disso, por favor sinta-se livre para instalar o Wireshark no seu próprio computador e fazer suas próprias capturas de tráfego para analisar. Importante:

Traces adicionais de HTTP: Se você quer traços adicionais de HTTP que você quer tentar investigar (e usar engenharia reversa) para saber o que se passa, você também pode olhar em alguns outros traces de HTTP no arquivo zip acima.

Tendo estudado o laboratório introdutório de captura de pacotes acima, agora nós estamos prontos para usar o Wireshark para investigar protocolos em operação. Nesse laboratório, nós vamos explorar vários aspectos do protocolo HTTP: A interação básica de GET/Response, formatos de mensagem HTTP, recuperação de arquivos HTML, recuperação de arquivos HTML com objetos indexados e autentificação HTTP e segurança. Antes de começar esses laboratórios, você pode querer rever a seção 2.2 do texto.

Antes de começar, por favor considere o seguinte: A informação que aparece entre colchetes [ ] no Wireshark é do próprio Wireshark e não é parte dos protocolos.

Baseado nas configurações de rede da plataforma onde você está rodando o Wireshark, você pode observar que todos os pacotes de saída estão marcados pelo Wireshark como sendo erros de checksum (soma de verificação). (Veja a figura 1). Isso, como o sugerido pelo Wireshark (veja o painel de detalhes do pacote na figura 1), pode ser devido ao offloading do checksum, uma configuração que alivia a CPU de gerar valores de checksum para pacotes de saída e deixa esse trabalho ser feito pelo adaptador de rede. Como o Wireshark captura os pacotes antes que eles alcancem o adaptador de rede, a soma de verificação (checksum) para todos os pacotes capturados é zero. Se você encontrar esse código de cores distrativo ou incômodo, você pode simplesmente desabilitar os erros de checksum colorindo a regra do menu de itens. View > Coloring Rules...



Erros de verificação do Wireshak

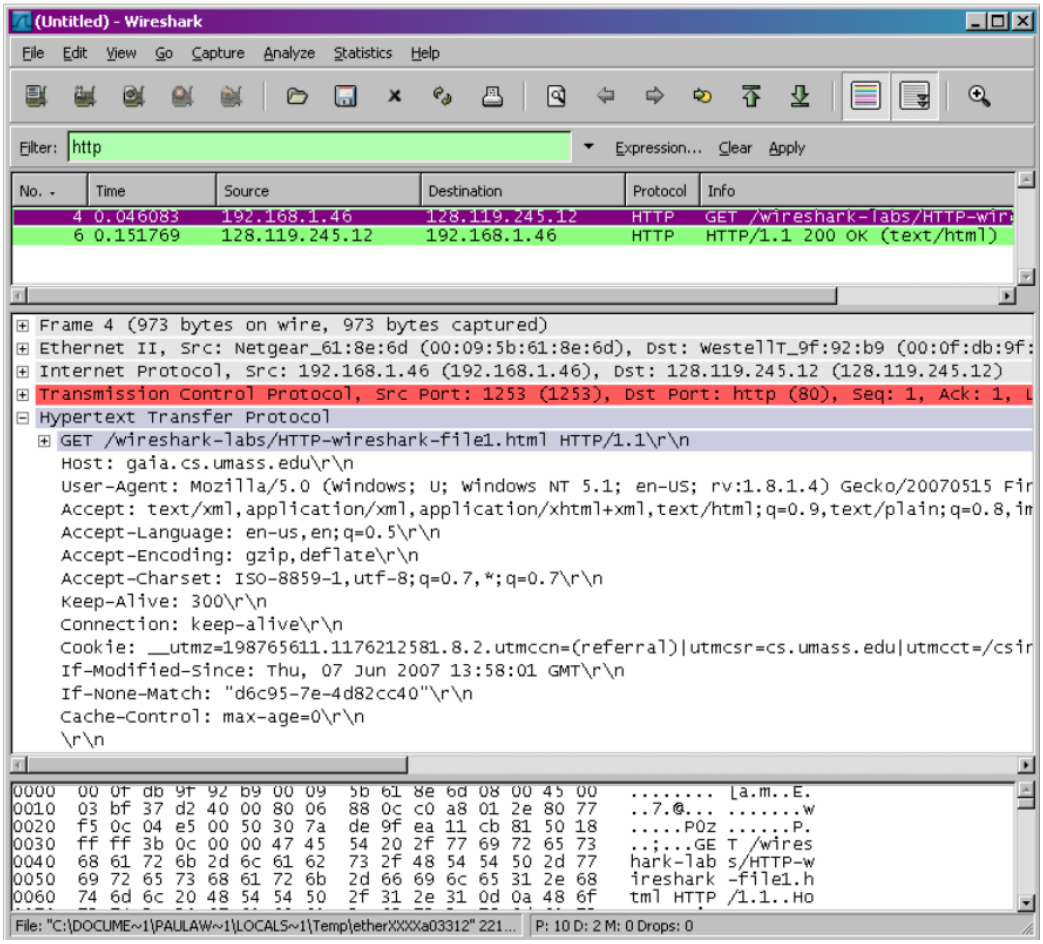
INTERAÇÃO BÁSICA DE HTTP GET/ RESPONSE

Vamos começar nossa exploração do HTTP baixando um simples arquivo HTTP – Ele é muito pequeno e contém objetos indexados. Faça o

seguinte:

1. Inicie seu navegador.
2. Inicie o Wireshark, como descrito no laboratório introdutório (mas ainda não comece a captura de pacotes). Digite “http” (apenas as letras, sem aspas) na janela do filtro de especificações, assim apenas as mensagens HTTP capturadas vão ser mostradas na lista de pacotes.
3. Espere pouco mais de um minuto (veremos porque tão pouco tempo), e então comece a captura de pacotes do Wireshark.
4. Entre no seguinte link pelo seu navegador: link. Seu navegador deve mostrar um arquivo HTML simples, de uma linha.
5. Pare a captura de pacotes do Wireshark.

Sua janela do Wireshark vai ser similar a da janela mostrada na figura 2. Se você não pode executar o Wireshark em uma conexão de rede em tempo real, você pode baixar um pacote de rotas que foi criado quando os passos acima foram seguidos. Para fazer isso, baixe o arquivo zip link e extraia o arquivo http-ethereal-trace-1. As rotas desse arquivo zip foram coletadas pelo Wireshark quando foi executado em um dos computadores do autor, enquanto realizava os passos indicados no laboratório do Wireshark. Uma vez que você tenha baixado o arquivo, você pode carregá-lo dentro do Wireshark e ver as rotas usando o menu “File pull down”, escolhendo “abrir” (Open) e selecionando o arquivo http-ethereal-trace-1. A tela resultante deve se parecer como a figura 2 após aplicar o filtro “http”.



Tela do Wireshak após o arquivo 1 ter sido recuperado do seu navegador

O exemplo mostrado na figura 2 mostra a janela de lista de pacotes onde duas mensagens HTTP foram capturadas: A mensagem GET (de seu navegador para o servidor gaia.cs.umass.edu) e a mensagem de resposta do servidor para o seu navegador. A janela de conteúdo dos pacotes mostra detalhes das mensagens selecionadas (nesse caso mensagens HTTP GET, que é destacado na janela de listagem de pacotes). Lembre que a mensagem HTTP está dentro de um segmento TCP, que foi carregada dentro de um datagrama IP, que foi carregado dentro de um quadro Ethernet, o Wireshark mostrará informações do quadro, Ethernet, IP e TCP. Nós queremos minimizar a quantidade de dados não HTTP mostrados (nós estamos interessados em HTTP, e investigaremos outros protocolos em capítulos seguintes), então tenha certeza de que as caixas no canto esquerdo que sinalizam informações de quadro, Ethernet, IP e TCP tem um sinal de mais (+), que significa que a informação está oculta, e a linha de HTTP tem um sinal de menos (-) que significa que toda a

informação HTTP está sendo mostrada.

Tarefa A: Olhando a informação das mensagens de HTTP GET/ response, responda as questões a seguir. Quando estiver respondendo as questões, você deve imprimir as mensagens GET/ response (para incluir as informações de pacote no seu relatório, você deve exportar os pacotes de dados selecionados como um arquivo de texto. Para isso, use a janela "File -> Export -> File ...", selecione o tipo como "Plain Text" e escolha "Selected Packet", indicando onde as mensagens que respondam as questões foram encontradas (Perceba que o procedimento de exportação deve diferir de acordo com a plataforma onde você está executando o Wireshark, como Linux, Windows etc). Para todas as questões é importante que você indique claramente qual é sua resposta, como você obteve a resposta e (se aplicável) discuta as implicações/ideias que enriqueçam suas respostas. Por exemplo, na questão a seguinte, você pode elaborar o motivo de você ter observado o que observou?

1. O seu navegador utiliza a versão de HTTP 1.0 ou 1.1? Que versão o servidor usa?
2. Que linguagens (se existe alguma) seu navegador diz que pode ser aceita pelo servidor? Na sessão de captura, que outra informação (se existe alguma) o navegador provê para o servidor a respeito do utilizador/ navegador?
3. Qual o endereço de IP do seu computador? E o do servidor gaia.cs.umass.edu?
4. Qual o código de status retornado do servidor para seu navegador?
5. Quando foi a última vez que o arquivo HTML que você está usando foi modificado no servidor?
6. Quantos bytes de conteúdo foram enviados ao seu navegador?
7. Inspecionando os dados brutos no painel "packet bytes", você vê algum cabeçalho HTTP com dados que não foram mostrados no painel "packet details"? Se sim, cite um.

Nas suas cinco respostas acima, você deve estar surpreso por notar que o documento que você recuperou foi modificado menos de um minuto antes de você baixá-lo. Isso é porque (para esse arquivo em particular), o servidor gaia.cs.umass.edu está configurado para que a última modificação do arquivo seja a atual, e atualiza isso uma vez por minuto. Assim, se você quiser esperar um minuto entre os acessos, o arquivo vai ter sido recentemente modificado e consequentemente seu navegador vai baixar uma "nova" cópia do arquivo.

INTERAÇÃO CONDICIONAL DE HTTP GET/ RESPONSE

Recordando a seção 2.2.6 do texto, que mostra que a maioria dos navegadores guardam objetos em cache e assim fazem a condicional GET quando recuperam um objeto HTTP. Antes de realizar os passos a seguir, tenha certeza que o cache do seu navegador está vazio. (Para fazer isso no Firefox, selecione Tools -> Clear Private Data. Essas ações vão remover os arquivos de cache do seu navegador. Agora faça o seguinte:

1. Inicie seu navegador e certifique-se de que o cache está limpo, como discutido acima.
2. Comece o Wireshark packet sniffer
3. Entre na seguinte URL pelo seu navegador: link. Seu navegador deve mostrar uma mensagem HTTP simples de uma linha
4. Rapidamente entre na mesma URL pelo seu navegador novamente (ou simplesmente atualize a página de seu navegador)
5. Pare a captura de pacotes do Wireshark e digite “http” no filtro de especificações, assim apenas pacotes HTTP vão ser mostradas depois na lista de pacotes

(Nota: Se você não pode executar o Wireshark em uma conexão de tempo real, você pode usar o pacote http-ethereal-trace-2 para responder as questões abaixo, veja aqui link. Esse arquivo foi montado seguindo os passos acima).

Questões práticas:

8. Inspecione o conteúdo da primeira requisição de HTTP GET do seu navegador ao servidor. Você vê a linha “IF-MODIFIED-SINCE” no HTTP GET?

9. Inspecione o conteúdo de resposta do servidor. O servidor retornou explicitamente o conteúdo do arquivo? Como você pode afirmar?

10. Agora observe o conteúdo da segunda requisição HTTP GET do seu navegador ao servidor. Você vê a linha “IF-MODIFIED-SINCE” no HTTP GET? Se sim, qual informação segue o cabeçalho “IF-MODIFIED-SINCE”?

11. Qual é o status HTTP e a frase retornados pelo servidor em resposta ao segundo HTTP GET? O servidor explicitamente retornou os conteúdos do arquivo? Explique.

Tarefa B: Para as questões de 8 à 11, primeiro escreva uma resposta breve porém precisa para cada questão acima, então escreva um parágrafo explicando suas observações das questões práticas. Note que sua resposta pode se beneficiar da explicação e/ou referências de algumas de suas observações explícitas.

RECUPERAÇÃO DE DOCUMENTOS LONGOS

Nos nossos exemplos, até agora, os documentos recuperados foram arquivos HTML simples e curtos. Agora veremos o que acontece quando você baixa um arquivo HTML grande. Faça o seguinte:

1. Inicie seu navegador e tenha certeza de que o cache está limpo, como discutido acima.

2. Comece o Wireshark packet sniffer.

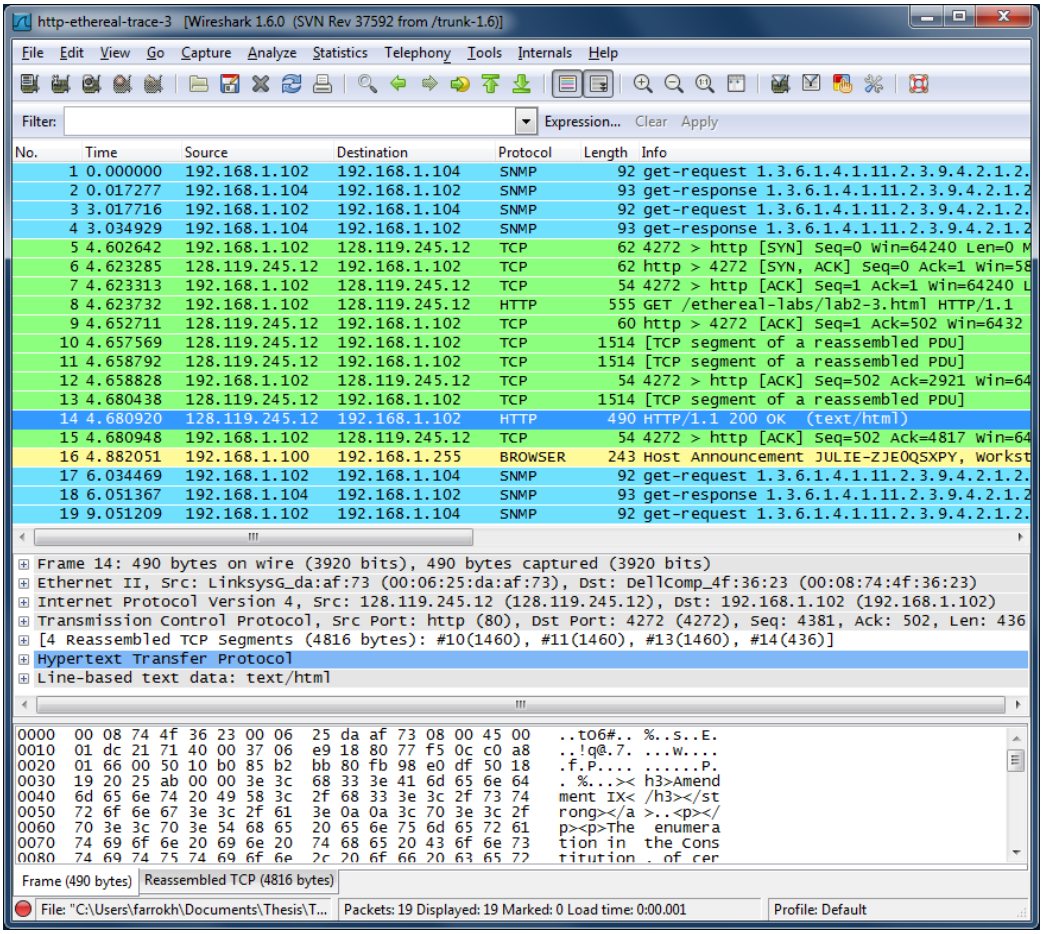
3.Entre com a seguinte URL no seu navegador link. Seu navegador deve mostrar a carta de direitos dos Estados Unidos.

4. Pare a captura de pacotes do Wireshark e digite “http” no filtro de especificações, assim apenas as mensagens HTTP capturadas serão mostradas.

Nota: Se você não pode executar o Wireshark em uma conexão de tempo real, você pode usar o pacote http-ethereal-trace-3 para responder as questões abaixo, veja aqui link. Esse arquivo foi montado seguindo os passos acima).

Na janela de listagem de pacotes, você deve ver sua mensagem HTTP GET, seguida de múltiplos pacotes de resposta para sua requisição HTTP GET. Esses múltiplos pacotes merecem uma pequena explicação. Recordando a seção 2.2 (veja no texto), aquela mensagem de resposta HTTP consistia em uma linha de status, seguida por linhas de cabeçalho, seguida por uma linha em branco, seguida pelo corpo da mensagem. No

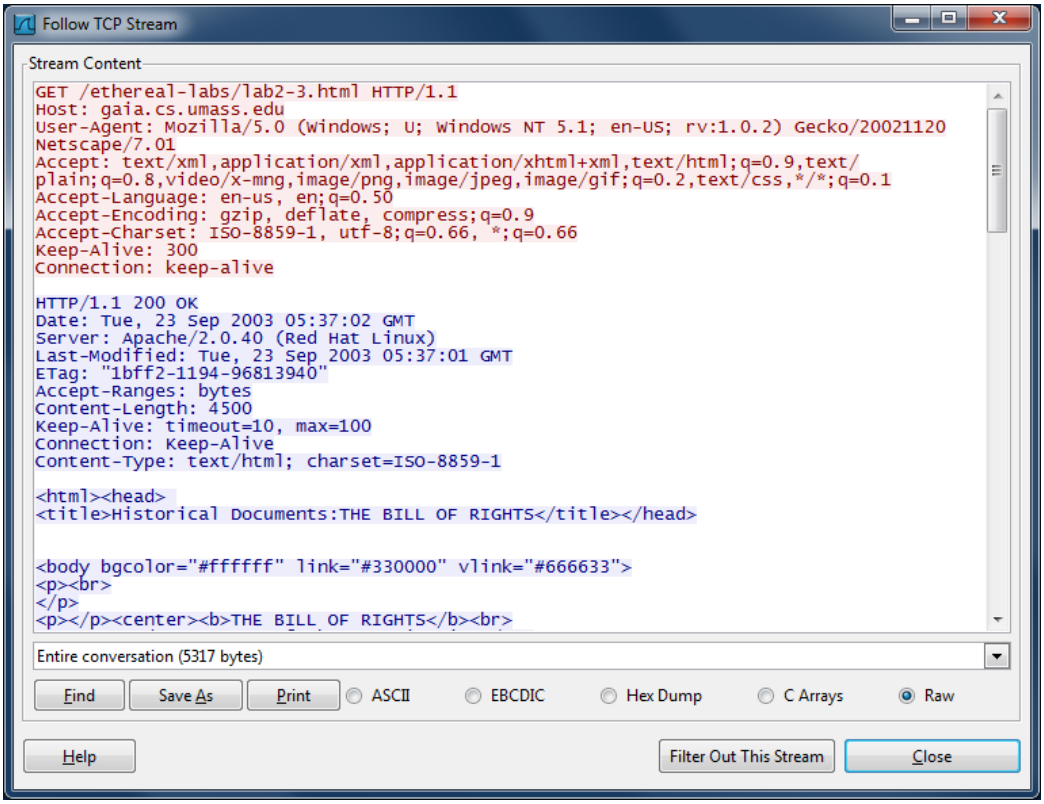
caso de nossa HTTP GET, o corpo da mensagem é o arquivo HTML de requisição inteiro. No caso de agora, o arquivo de HTML é muito grande e os 4500 bytes são grandes demais para caber em um único pacote TCP. Uma única mensagem de resposta HTTP é então quebrada em alguns pedaços pelo TCP, e cada pedaço contém separadamente um segmento TCP (Veja no texto). Cada segmento TCP é gravado pelo Wireshark como um pedaço separado, e o fato de que uma única resposta HTTP foi fragmentada em múltiplos pacotes TCP é indicado pela frase “TCP segment of a reassembled PDU”, mostrada na tela do Wireshak. Ressaltamos aqui que não existe uma mensagem de “TCP segment of a reassembled PDU” em HTTP! Nesse sentido, a figura 3 mostra uma captura de tela do Wireshark mostrando a rota de um http-ethereal-trace-3. Na listagem de pacotes capturados, o pacote número 8 mostra uma requisição HTTP GET e o pacote número 14 mostra a resposta HTTP correspondente. Pode ser visto que os pacotes número 10, 11 e 13 são nomeados com “TCP segment of a ressaembled PDU”. Clicando na resposta HTTP, como no pacote 14, o painel de detalhes do pacote mostra [4 Reassembled TCP Segments (4816 bytes): 10(1460), 11(1460), 13(1460), 14(436)] (veja a figura 3). Adicionalmente, o painel de pacotes de bytes mostra uma noma tabela entitulada Reassembled TCP que mostra a resposta HTTP inteira.



Tela do Wireshak com a rota do terceiro pacote

Uma maneira mais conveniente de ver todos os dados (todos as requisições HTTP e as respostas transportadas pelo TCP) é usando uma característica extra do Wireshark chamada “Following TCP Streams”. Clicando com o botão direito do mouse em qualquer dos pacotes TCP associados com um determinado fluxo TCP e selecionado o “Follow TCP Stream” no menu, uma nova janela abrirá, contendo a troca de dados. A figura 4 mostra a janela “Follow TCP Stream” para a requisição GET /ethereal-labs/lab2-3.html HTTP/1.1 e sua resposta associada. Nessa janela, caracteres não demonstráveis são substituídos por pontos. De qualquer modo, a escolha de Raw ou ASCII feita nessa janela afeta completamente o modo como você poderá salvar seu fluxo de dados. Isso é, se Raw é selecionado, o fluxo de dados será salvo como arquivo binário, preservando os caracteres não demonstráveis. Já no caso de ASCII, o fluxo de dados é salvo como um arquivo de texto onde os caracteres não demonstráveis são substituídos por pontos. Note como o Wireshark

mudou (e aplicou) as mudanças no filtro que mostra apenas os pacotes selecionados.



A janela do Wireshark durante o fluxo TCP

Questões práticas:

12. Quantas mensagens de requisição HTTP GET foram enviadas pelo seu navegador?

13. Quantos segmentos TCP contendo dados são necessários para carregar essa única resposta HTTP?

14. Qual o status e a frase associada com a requisição HTTP GET?

15. Existe alguma informação de cabeçalho HTTP nos dados associados com a segmentação TCP?

Tarefa C: Para as questões de 12 à 15, primeiro escreva uma resposta breve porém precisa para cada questão acima, então escreva um pará-

grafo explicando suas observações das questões práticas. Note que sua resposta pode se beneficiar da explicação e/ou referências de algumas de suas observações explícitas.

DOCUMENTOS HTML COM OBJETOS EMBUTIDOS

Agora que vimos como o Wireshark mostra o tráfego de captura de pacotes para arquivos HTML grandes, podemos olhar o que acontece quando seu navegador baixa um arquivo com objetos embutidos, como um arquivo que incluem outros objetos (no exemplo a seguir, imagens) que estão armazenados em outro servidor. Faça o seguinte:

1.Inicie seu navegador e tenha certeza de que o cache está limpo, como discutido acima.
2.Comece o Wireshark packet sniffer.
3.Entre com a seguinte URL no seu navegador link Seu navegador deve mostrar um arquivo HTML curto com duas imagens. Essas duas imagens são referenciadas na base do arquivo HTML. Isso é, as imahens não estão contidas no HTML; no lugar disso URLs para o conteúdo das imagens são baixadas no arquivo HTML. Como discutido no livro, seu navegador vai recuperar esses logos de sites indicados. Nosso logo é recuperado do link www.aw-bc.com. A imagem da capa do nosso livro é armazenada no servidor manic.cs.umass.edu.
4.Pare a captura de pacotes do Wireshark e digite “http” no filtro de especificações, assim apenas as mensagens HTTP capturadas serão mostradas. Nota: Se você não pode executar o Wireshark em uma conexão de tempo real, você pode usar o pacote http-ethereal-trace-4 para responder as questões abaixo, veja aqui link Esse arquivo foi montado seguindo os passos acima).

Questões práticas:

16. Quantas requisições HTTP GET foram enviadas pelo seu navegador? Para quais endereços essas requisições GET foram enviadas?

17. Você pode dizer se o seu navegador baixou as duas imagens em série ou se elas foram baixadas dos dois sites em paralelo? Explique.
Tarefa D: Para as questões de 16 à 17, primeiro escreva uma resposta

breve porém precisa para cada questão acima, então escreva um parágrafo explicando suas observações das questões práticas. Note que sua resposta pode se beneficiar da explicação e/ou referências de algumas de suas observações explícitas.

AUTENTICAÇÃO HTTP

Finalmente, vamos visitar um site protegido por senha e examinar a sequência de mensagens HTTP trocadas com esse site. A URL link é protegida por senha. O username é “wireshark-students” (sem as aspas) e a senha é “network” (novamente, sem as aspas). Então vamos acessar esse site protegido por senha “seguro”. Faça o seguinte:

1.Inicie seu navegador e tenha certeza de que o cache está limpo, como discutido acima.
2.Comece o Wireshark packet sniffer.
3.Entre com a seguinte URL no seu navegador link Digite o username e o password na janela pop up.
4.Pare a captura de pacotes do Wireshark e digite “http” no filtro de especificações, assim apenas as mensagens HTTP capturadas serão mostradas.
Nota: Se você não pode executar o Wireshark em uma conexão de tempo real, você pode usar o pacote http-ethereal-trace-5 para responder as questões abaixo, veja aqui link Esse arquivo foi montado seguindo os passos acima).
Agora vamos examinar a saída de dados do Wireshark. Você pode querer ler primeiro sobre autentificação HTTP revendo o material em “HTTP Acess Athentication Framework” em http://frontier.userland.com/stories/storyReader
Questões práticas (não precisam ser respondidas explicitamente):
18. Qual é a resposta do servidor (status e frase) em resposta a mensagem inicial de HTTP GET do seu navegador?
19. Quando seu navegador envia uma mensagem HTTP GET pela segunda vez, qual é o novo campo incluído na mensagem HTTP GET?
Você não precisa relatar as questões 18 e 19; de qualquer modo, sinta-se livre para redigir um parágrafo explicando e discutindo suas observações sobre as questões práticas acima. O username e a senha que você forneceu são codificadas na string de caracteres seguindo o cabeçalho “Authorization: Basic” na mensagem HTTP do cliente. Enquanto pode parecer que o seu username e senha estão e criptografados, eles estão simplesmente codificados em um formato conhecido como Base64. O

username e a senha não estão criptografados! Para ver isso, vá para http://gtools.org/tool/base64-encode-decode/ e enyte com a string de codificação base64 d2lyZXNoYXJrLXN0dWRlbnRz dentro da caixa “Decode from base64” e pressione “Go”. Voila! Você traduziu de base64 para codificação ASCII, e então pode ver seu username! Para ver a senha, entre com a string Om5ldHdvcms= e pressione “decode”. Já que qualquer um pode baixar ferramentas como o Wireshark e rastrear pacotes (não só os seus próprios) passando pelo adaptador de rede, qualquer um pode traduzir Base64 para ASCII (como você acabou de fazer!). Deve estar claro que senhas simples em sites WWW não são seguros, a não ser que medidas adicionais sejam tomadas.
Não tema! Como veremos no capítulo 7, existem maneiras de fazer que acessos WWW sejam mais seguros. De qualquer modo, nós claramente precisamos de algo que vá além da autentificação básica de HTTP!

QUESTÕES PREPARATÓRIAS PARA A SEGUNDA TAREFA

20. O que o cabeçalho “Connection: close” e “Connection: Keep-alive” implicam no protocolo HTTP? Quando uma delas deve ser usada ao invés da outra?

Tarefa E: Para a questão 20, primeiro escreva um breve porém preciso parágrafo que responda a questão acima, então escreva um parágrafo explicando e discutindo como essas observações podem ser úteis para a próxima tarefa.

Demonstração e relatórios Para essa tarefa você vai precisar escrever um relatório que cuidadosamente responda as questões 1-17 e 20, e escreva um parágrafo discutindo cada grupo de questões 1-7 (Tarefa A), 8-11 (Tarefa B), 12-15 (Tarefa C), 16-17 (Tarefa D), e 20 (Tarefa E). Note que cada grupo de questões tem um tema e você deve convencer o leitor que seu relatório (incluindo você se você ler seus documentos meses/semanas depois) que você entende esses aspectos do HTTP. Por favor estruture seu relatório de modo que suas respostas estejam claramente indicadas para cada questão (e sessão). Questões e respostas correspondentes devem estar claramente indicadas. Estruture seu relatório; Além disso, suas respostas devem ser explicadas e apoiadas usando informações adicionais, quando aplicável. Durante a demonstração podem ser feitas questões similares para obtermos a certeza de que você

compreendeu todo esse laboratório. Você deve explicar claramente suas respostas. Como os relatórios são feitos em duplas, ambos os membros responderão perguntas.

Instruções adicionais podem ser encontradas aqui:link