Wireshark Lab: HTTP

Versão 1.1

2008 BATISTA, O. M. N. Tradução e adaptação para Wireshark.

Tendo molhado os nossos pés com o Wireshark no laboratório introdutório, agora estamos prontos para utilizar o Wireshark para investigar protocolos em operação. Neste laboratório, exploraremos vários aspectos do protocolo HTTP: a interação básica GET/resposta do HTTP, formatos de mensagens HTTP, baixando arquivos grandes em HTML, baixando arquivos em HTML com objetos incluídos, e autenticação e segurança HTTP. Antes de iniciar este laboratório, você deve reler a seção 2.2 do livro.

1. A Interação Básica GET/Resposta do HTTP

Vamos iniciar a nossa exploração do HTTP baixando um arquivo em HTML simples - bastante pequeno, que não contém objetos incluídos. Faça o seguinte:

- inicie o navegador;
- inicie o Wireshark, como descrito no laboratório introdutório (mas não inicie a captura de pacotes ainda). Digite “http.request or http.response” (somente as letras, sem as aspas) na caixa de texto de especificação do filtro de exibição, de tal forma que apenas as mensagens HTTP capturadas serão exibidas na janela de listagem de pacotes. (Só estamos interessados em HTTP desta vez, e não desejamos ver todos os pacotes capturados);
- inicie a captura de pacotes
- digite o seguinte URL no navegador (figura 1)

  http://gaia.cs.umass.edu/ethereal-labs/HTTP-ethereal-file1.html
- pare a captura de pacotes.

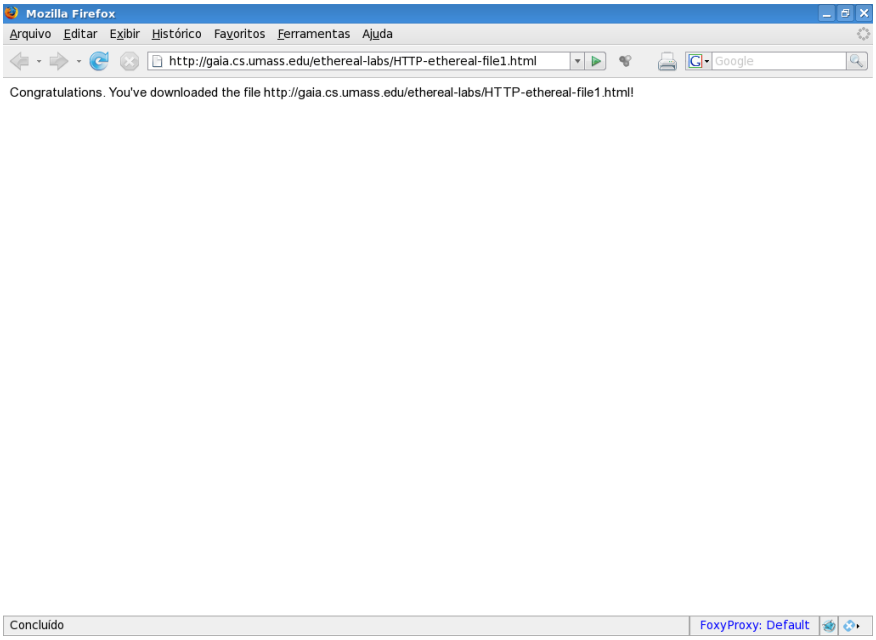

Figura 1. Janela do navegador.

A janela do Wireshark deve estar parecida com a janela exibida na figura 2.

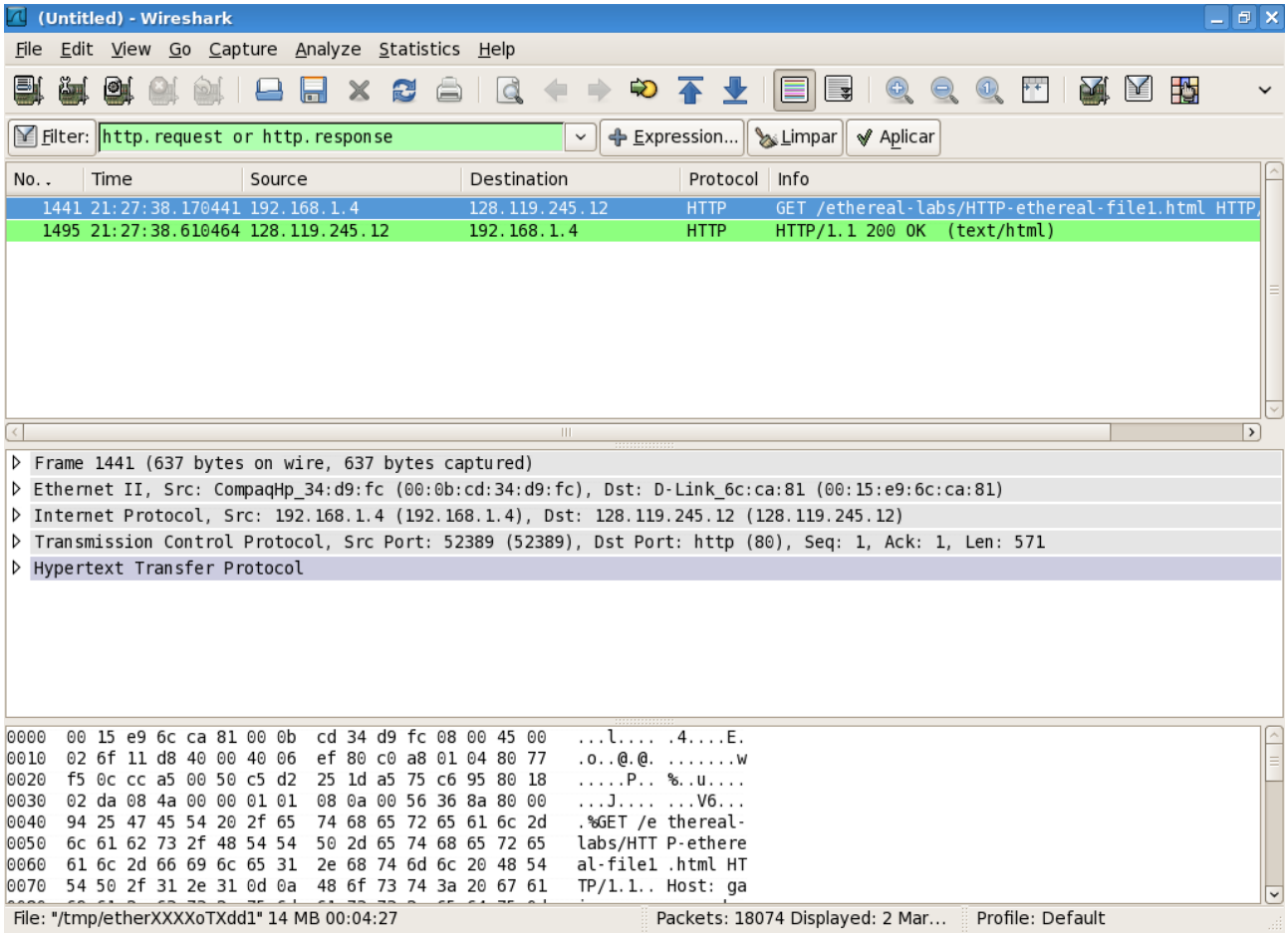

Figura 2. Requisição e Resposta HTTP.

O exemplo da figura 2 mostra na janela de listagem de pacotes duas mensagens HTTP

capturadas: a mensagem GET (do seu navegador para o servidor *web* gaia.cs.umass.edu.web) e a mensagem de resposta do servidor para o seu navegador. A janela de conteúdos de pacotes mostra detalhes da mensagem selecionada (neste caso a mensagem HTTP GET, que está em destaque na janela de listagem de pacotes). Lembre-se de que a mensagem HTTP é transportada em um segmento TCP, que é carregado em um datagrama IP, que é levado em um quadro Ethernet. O Wireshark exibe informações sobre o quadro, IP, TCP e HTTP. Você deve expandir as informações do HTTP clicando na seta ao lado esquerdo de "Hypertext Transfer Protocol".
Observando as informações das mensagens HTTP GET e de resposta, responda às seguintes perguntas. Quando responder às questões, você deve imprimir as mensagens GET e a resposta (veja a aula introdutória para saber como fazer isso) e indicar em que parte da mensagem você encontrou a informação que responde às questões.

1. O seu navegador executa HTTP 1.0 ou 1.1? Qual a versão de HTTP do servidor?
2. Quais linguagens (se alguma) o seu navegador indica que pode aceitar ao servidor?
3. Qual o endereço IP do seu computador? E do servidor gaia.cs.umass.edu?
4. Qual o código de status retornado do servidor para o seu navegador?
5. Quando o arquivo em HTML que você baixou foi modificado no servidor pela última vez?
6. Quantos bytes de conteúdo são retornados ao seu navegador?
7. Inspecionando os dados na janela de conteúdo do pacote, você vê algum cabeçalho dentro dos dados que não são exibidos na janela de listagem de pacotes? Caso a resposta seja afirmativa, indique um.

Na sua resposta à questão 5 acima, você deve ter se surpreendido em descobrir que o documento que você recebeu foi modificado pela última vez cerca de um minuto antes de você baixá-lo. Isso ocorre porque (para este arquivo particular), o servidor gaia.cs.umass.edu está atribuindo a hora de última modificação do arquivo para a hora atual, e faz isso uma vez por minuto. Assim, se você aguardar um minuto entre os acessos, o arquivo aparecerá como modificado recentemente, e desta forma o seu navegador baixará uma nova cópia do documento.

2. A Interação HTTP GET Condicional/Resposta

Lembre-se da seção 2.2.5 do livro que a maioria dos navegadores *web* tem um cache e, desta forma, realizam GET condicional quando baixam um objeto HTTP. Antes de realizar

os passos a seguir, apague o conteúdo do cache do seu navegador:

- inicie o navegador *web*, e certifique-se de que o cache seja apagado;
- inicie o Wireshark;
- digite o URL no navegador

  http://gaia.cs.umass.edu/ethereal-labs/HTTP-ethereal-file2

  seu navegador deve exibir um arquivo em HTML muito simples com cinco linhas;
- pressione o botão “refresh” no navegador (ou digite o URL novamente);
- pare a captura de pacotes, e digite “http” na caixa de texto de especificação de filtro, para que apenas as mensagens HTTP sejam apresentadas na janela de listagem de pacotes.

Responda às seguintes questões:

8. Inspecione o conteúdo da primeira mensagem HTTP GET do seu navegador para o servidor. Você vê uma linha “IF-MODIFIED-SINCE”?
9. Inspecione o conteúdo da resposta do servidor. O servidor retornou explicitamente o conteúdo do arquivo? Como você pode dizer isso?
10. Agora inspecione o conteúdo da segunda mensagem HTTP GET do seu navegador para o servidor. Você vê uma linha “IF-MODIFIED-SINCE”? Caso a resposta seja afirmativa, qual informação segue o cabeçalho “IF-MODIFIED-SINCE”?
11. Qual é o código de status e a frase retornada do servidor na resposta à segunda mensagem HTTP GET? O servidor retornou explicitamente o conteúdo do arquivo? Explique.

3. Baixando Documentos Longos

Nos exemplos até agora, os documentos baixados foram simples e pequenos arquivos em HTML. Vamos ver o que acontece quando baixamos um arquivo em HTML grande. Faça o seguinte:

- inicie o navegador *web*, certifique-se de que o cache seja apagado;
- inicie o Wireshark;
- digite o URL no navegador

  http://gaia.cs.umass.edu/ethereal-labs/HTTP-ethereal-file3

  seu navegador deve exibir um documento bastante longo;

- pare a captura de pacotes, e digite “http” na caixa de texto de especificação de filtro, para que apenas as mensagens HTTP seja exibidas.

Na janela de listagem de pacotes, você deve ver a sua mensagem HTTP GET, seguida por uma reposta em vários pacotes. Esta resposta em vários pacotes merece uma explicação. Lembre-se da seção 2.2 do livro (veja a figura 2.9) que a mensagem de resposta HTTP consiste de uma linha de status, seguida por zero ou mais linhas de cabeçalhos, seguida por uma linha em branco, seguida pela carga útil. No caso da nossa HTTP GET, a carga útil na resposta é o arquivo HTTP completo. No nosso caso aqui, o arquivo em HTML é bastante longo, e a 4500 bytes é muito grande para caber em um segmento TCP. A resposta HTTP simples é então quebrada em vários pedaços pelo TCP, com cada pedaço sendo contido dentro de um segmento TCP separado (veja a figura 1.22 no livro). Cada segmento TCP é capturado em um pacote separado pelo Wireshark, e o fato de que uma simples resposta foi fragmentada em vários segmentos TCP é indicada pela palavra “Continuation” exibida no Wireshark. Vale salientar que não há uma mensagem “Continuation”em HTTP!

Responda às seguintes questões:

12. Quantas mensagens HTTP GET foram enviadas pelo seu navegador?
13. Quantos segmentos TCP foram necessários para carregar a resposta?
14. Qual é o código de status e a frase associada com a resposta à mensagem HTTP GET?
15. Há alguma linha de status HTTP nos dados transmitidos associados com um “Continuation” TCP?

4. Documentos HTML com Objetos Incluídos

Agora que vimos como o Wireshark mostra o tráfego capturado para arquivos em HTML grandes, nós podemos observar o que acontece quando o seu browser baixa um arquivo com objetos incluídos, no nosso exemplo, imagens que estão armazenadas em outros servidores. Faça o seguinte:

- inicie o navegador *web*, certifique-se de que o cache seja apagado;
- inicie o Wireshark;
- digite o URL no navegador

http://gaia.cs.umass.edu/ethereal-labs/HTTP-ethereal-file4

seu navegador deve exibir um arquivo pequeno em HTML com duas imagens incluídas. Estas duas imagens estão referenciadas no arquivo em HTML. Isto é, as imagens não estão no arquivo em HTML, ao invés disso, há um URL para cada imagem no arquivo em HTML. Como discutido no livro, seu navegador terá que baixar estas imagens dos locais correspondentes. A imagem com a logomarca da editora está em www.awl.com. A imagem com a capa do livro está em maniac.cs.umass.edu;

- pare a captura de pacotes, e digite “http” na caixa de texto de especificação de filtro, para que apenas as mensagens HTTP seja exibidas.

Responda às seguintes questões:

16. Quantas mensagens HTTP GET foram enviadas pelo seu navegador? Para quais endereços na Internet estas mensagens foram enviadas?
17. Você consegue dizer se o seu navegador baixou as duas imagens em seqüência, ou se foram baixadas dos dois locais distintos em paralelo? Explique.

5. Autenticação HTTP

Finalmente, vamos tentar visitar um local na *web* que é protegido por senha e examinar a seqüência de mensagens HTTP trocadas com este local. O URL http://gaia.cs.umass.edu/ethereal-labs/protected_pages/HTTP-ethereal-file5 é protegido por senha. O usuário é “eth-students” (sem as aspas), e a senha é “networks” (novamente, sem as aspas). Então vamos acessar o local protegido por senha. Faça o seguinte:

- inicie o navegador *web*, certifique-se de que o cache seja apagado;
- inicie o Wireshark;
- digite o URL no navegador

  http://gaia.cs.umass.edu/ethereal-labs/protected_pages/HTTP-ethereal-file5

  seu navegador deve exibir um documento bastante longo;
- pare a captura de pacotes, e digite “http” na caixa de texto de especificação de filtro, para que apenas as mensagens HTTP seja exibidas.

Agora vamos examinar a saída do Wireshark. Você pode querer primeiro ler sobre a

autenticação HTTP revisando o material fácil de ler (em inglês) “HTTP Access Authentication Framework” em http://frontier.userland.com/stories/storyReader$2159

Responda às seguintes questões:

18. Qual é a resposta do servidor (código de status e frase) para a primeiro mensagem HTTP GET do seu navegador?
19. Quando o seu navegador envia a mensagem HTTP GET pela segunda vez, qual o novo campo que está incluído na mensagem?

O nome de usuário (eth-students) e a senha (network) que você digitou foram codificados na cadeia de caracteres (ZXRoLXN0dWRlbnRzOm5ldHdvcmtz) após o cabeçalho “Authorization: Basic” na mensagem HTTP GET. Parece que o nome e senha estão criptografados, mas na verdade estão simplesmente codificados em um formato denominado Base64. O nome do usuário e a senha não estão criptografados! Para ver isso, vá para http://www.securitystats.com/tools/base64.php e digite o texto ZXRoLXN0dWRlbnRzOm5ldHdvcmtz e pressione decode. *Voilá*! Você traduziu de Base64 para ASCII, e desta forma consegue ver o nome de usuário e a senha! Sabendo que alguém pode baixar o Wireshark e capturar pacotes (não somente os próprios), e alguém pode traduzir de Base64 para ASCII (você acabou de fazê-lo!), deve estar claro para você que o uso de senhas apenas em locais na *web* não garantem segurança, a não ser que medidas adicionais sejam tomadas.
Não tema! Como veremos no capítulo 7, há meios de fazer o acesso WWW ser mais seguro. Contudo, nós claramente precisamos de algo que vá além do *framework* básico de autenticação HTTP!