Wireshark Lab: HTTP

Versão 1.2

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

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. Lembre-se do conteúdo apresentado na disciplina por seu professor e se necessário releia o conteúdo no livro texto sobre WWW.

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/wireshark-labs/HTTP-wireshark-file1.html
- pare a captura de pacotes.

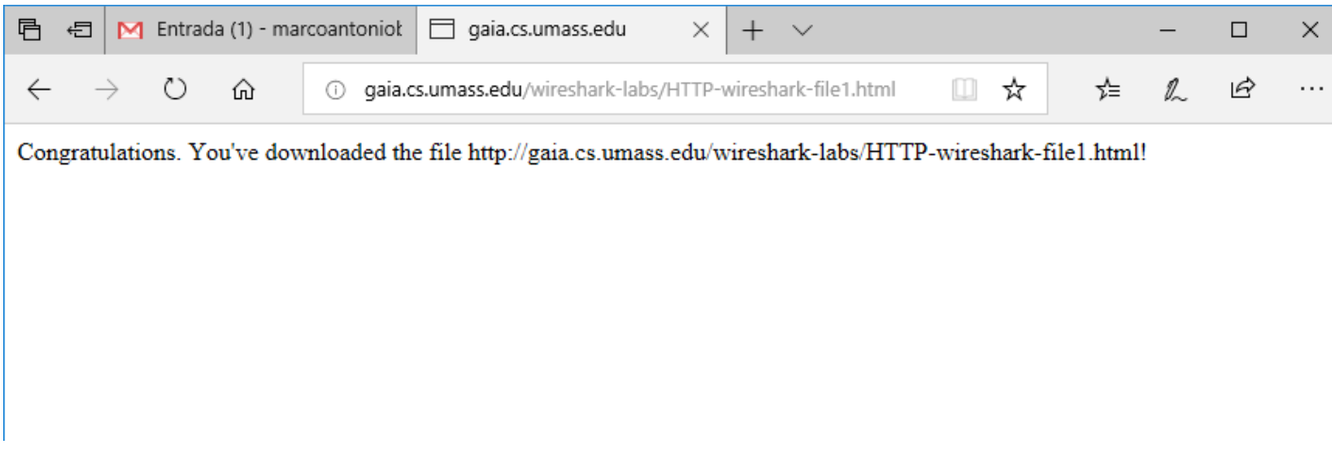

Figura 1. Janela do navegador.

A figura 1 mostra como deve estar a tela de seu navegador, uma página simples com um texto de parabenização. A janela do Wireshark deve estar parecida com a janela exibida na figura 2. Perceba que mesmo aplicando o filtro, podem aparecer vários pacotes (linhas) com o filtro aplicado, isto porque sua máquina está em constante

comunicação com a Internet por causa de várias aplicações distintas rodando em paralelo, muitas delas usam o HTTP para transferir conteúdos).

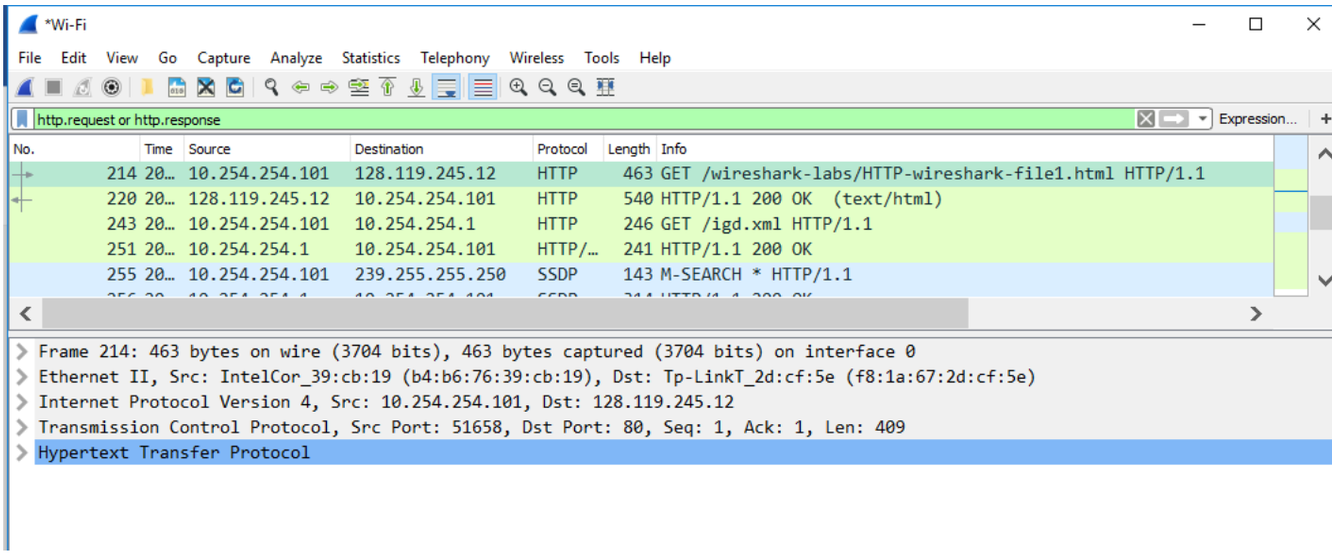

Figura 2. Requisição e Resposta HTTP capturadas pelo Wireshark.

O exemplo da figura 2 mostra na janela de listagem vários pacotes, em particular o pacote de número 214 (primeira coluna) mostra a requisição saindo do computador 10.254.254.101 para o servidor gaia.cs.umass.edu ip 128.119.245.12. Este pacote capturado: a mensagem GET (do seu navegador para o servidor web gaia.cs.umass.edu) e a mensagem de resposta do servidor para o seu navegador (no exemplo o pacote de número 220). 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 **tirar um print** da tela do Wireshark, circulando o ponto onde você extraiu as respostas!

1. O seu navegador executa HTTP 1.0, 1.1 ou 2.0? 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.

Salve sua captura com o nome HTTP1a7.pcap.

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 do livro onde se discute 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 Wreshark;
- digite o URL no navegador

o http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file2.html

seu navegador deve exibir um arquivo em HTML muito simples com cinco linhas;

- pressione o botão "refresh" ou F5 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.

Salve sua captura com o nome HTTP8a11.pcap.

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/wireshark-labs/HTTP-wireshark-file3.html
  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 sejam exibidas. Para ajudar ainda mais coloque “ip.addr == ip.servidor.gaia” Vc deve ter descoberto este ip na pergunta de número 3.

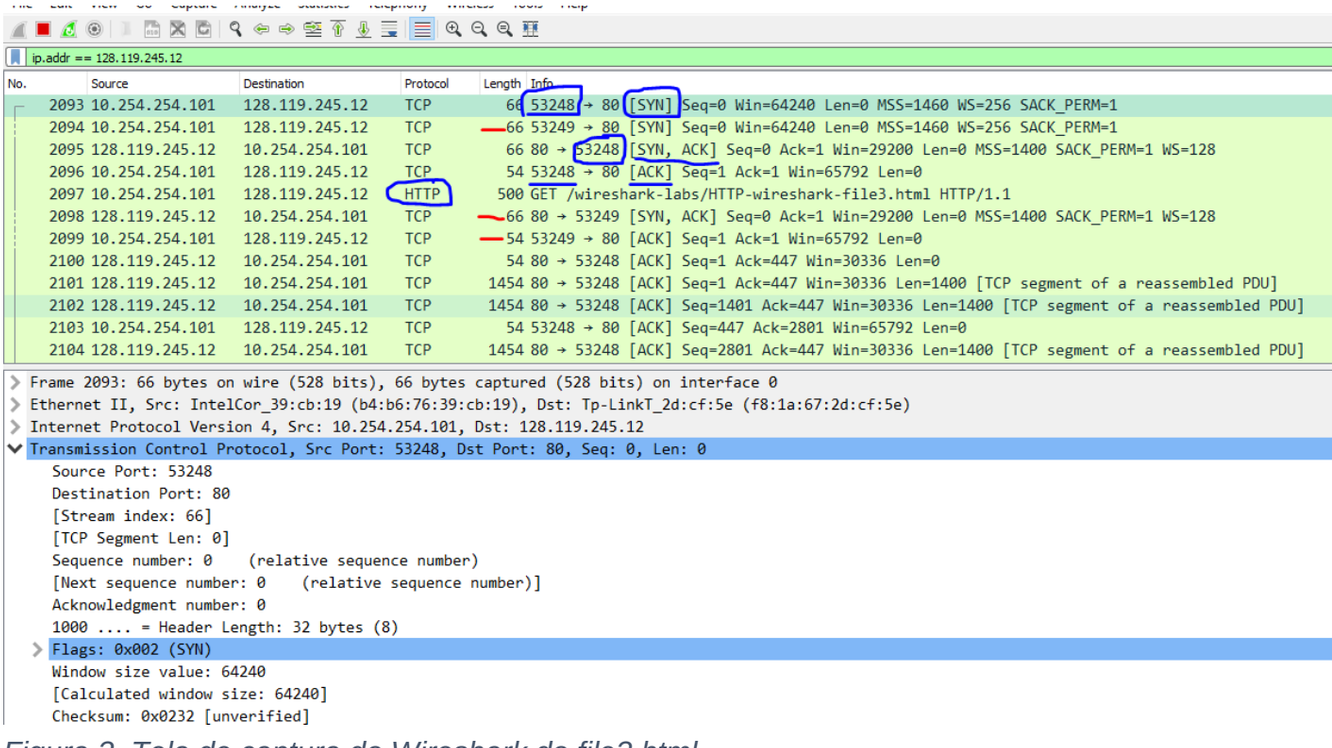

Figura 3. Tela de captura do Wireshark do file3.html

Na janela de listagem de pacotes, figura 3, você deve identificar dois estabelecimentos de conexões. A primeira, noc aso de nosso exemplo da figura 3, está identificada com riscos na cor azul, da porta 53248 para a porta 80. O trafego da segunda conexão em vermelho é feita por alguns navegadores com o objetivo de obter o favonico.ico que é aquela imagem que é colocada ao lado da URL no navegador. Concentre sua atenção para a primeira conexão, em nosso exemplo aquela que usa a porta 53248 e que logo após estabelecimento da conexão manda um HTTP GET. Você poderá perceber que alguns segmentos TCP serão encaminhados do servidor para você e que algumas confirmações serão devolvidas. Isto indica que foi necessário dividir a resposta em vários segmentos. Se você continuar seguindo os pacotes dessa conexão você achará um pacote HTTP 200 OK, exemplificado na figura 4.

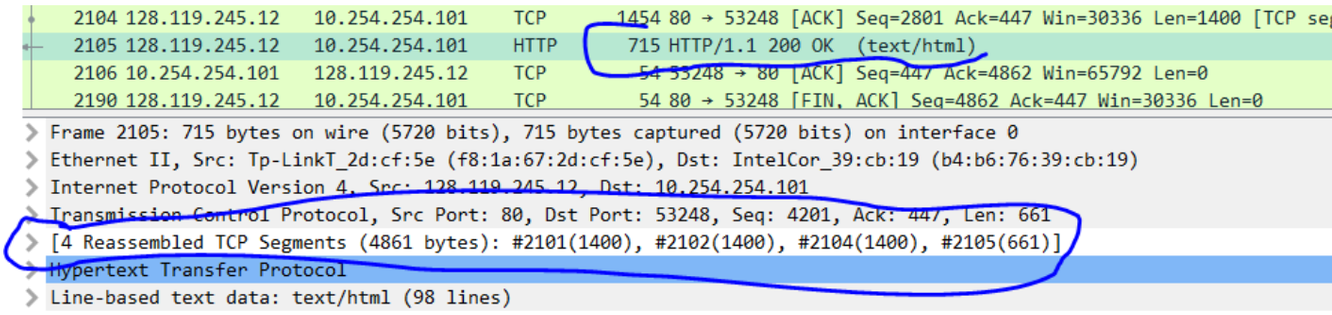

Figura 4. Captura Wireshark que mostra HTTP 200 OK.

Dentro dele é possível ver quantos segmentos TCP foram necessários, para o exemplo, os segmentos 2101, 2102, 2104 e 2105.

Lembre-se 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 da carga útil. No caso de nossa HTTP GET, a carga útil na resposta é o arquivo HTTP completo. No nosso caso aqui, o arquivo em HTML é bastante longo, um total de 4500 bytes que não cabem em um único pacote ou segmento TCP.

Observado a sua captura, 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 segmentos TCP associados a esta transferência?

Salve sua captura com o nome HTTP12a15.pcap.

4. Documentos HTML com Objetos Incluídos

Agora que vimos como o Wireshark mostra tráfego 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 armazenados em outros servidores. Faça o seguinte:

- inicie o navegador certifique-se o cache apagado;
- inicie o Wireshark;
- digite c URL no navegador
  http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file4.html
  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 no mesmo servidor que nos entregou o HTML solicitado. 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 sejam 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 sequência, ou se foram baixadas dos dois locais distintos em paralelo? Explique.

Salve sua captura com o nome HTTP16a17.pcap.

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/wireshark-labs/protected_pages/HTTP-wireshark-file5.html é protegido por senha. O usuário é " wireshark-students " (sem as aspas), e a senha é "network" (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/wireshark-labs/protected_pages/HTTP-wireshark-file5.html
  seu navegador deve solicitar usuário e senha, digite as informações passadas acima;
- pare a captura de pacotes, e digite "ip.addr == ip_servidor_gaia" na caixa de texto de especificação de filtro, para que apenas as mensagens para o servidor destino de nosso acesso.

Agora vamos examinar a saída do Wireshark. Você pode querer primeiro ler sobre a autenticação HTTP revisando o material 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 o primeira 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 (wireshark-students) e a senha (network) que você digitou foram codificados na cadeia de caracteres (d2lyZXNoYXJrLXN0dWRlbnRzOm5ldHdvcms=) 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 senha não estão criptografados! Para ver isto, vá em http://www.motobit.com/util/base64-decoder-encoder.asp e digite o texto d2lyZXNoYXJrLXN0dWRlbnRzOm5ldHdvcms= e pressione decode. Voilá! Você traduziu de Base64 para ASCII e desta forma consegue ver o usuário e senha.