Wireshark Lab: HTTP

1 As referências a figuras e seções são para a 6ª edição do texto, Redes de *Computadores, Uma Top-downdown*

*Abordagem, 6*  *ed., J.F.* *Kurose*  *e*  *K.W. Ross,*  *Addison-Wesley/Pearson, 2012.*

Protocolo HTTP: a interação básica get/resposta, formatos de mensagem HTTP

1. A interação básica http GET/resposta

Vamos começar nossa exploração de HTTP baixando um arquivo HTML muito simples e que não contém objetos incorporados . Faça o seguinte::

1. Inicie seu navegador web.

2. Inicie o farejador de pacotes Wireshark , como descrito no laboratório introdutório (mas ainda não comece a captura de pacotes). Digite "http" (apenas as letras, não as aspas) na janela display-filter-specification, de modo que apenas as mensagens HTTP capturadas serão exibidas mais tarde na janela de listagem de pacotes . (Só estamos interessados no HTTP, e não queremos ver a desordem de todos os pacotes capturados).

3. Espere um pouco mais de um minuto, e então comece a

captura de pacotes no Wireshark.

4. Digite o seguinte no seu navegador

http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file1.html

Seu navegador deve exibir o arquivo HTML de uma linha muito simples.

5. Pare a captura do pacote wireshark.

Sua janela wireshark deve ser semelhante à janela mostrada na Figura 1. (Veja o enunciado original, em inglês).

O exemplo na Figura 1 mostra na janela de listagem de pacotes que duas mensagens HTTP foram capturadas :

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. A janela de conteúdo do pacote mostra detalhes da mensagem selecionada (neste caso, a mensagem HTTP OK , que é destacada na janela de listagem de pacotes ). Lembre-se de que uma vez que a mensagem HTTP foi transportada dentro de um segmento TCP , que foi transportado dentro de um datagrama IP, que foi transportado dentro de um quadro Ethernet, o Wireshark exibe as informações do quadro Ethernet e dos protocolos IP e TCP também. Queremos minimizar a quantidade de dados não-HTTP exibidos (estamos interessado em HTTP aqui, e estaremos investigando esses outros protocolos em laboratórios posteriores), por isso certifique-se de que as caixas no extremo esquerdo do Quadro, Ethernet, IP e TCP têm um sinal de mais ou um ponto (o que significa que há informações ocultas, não listadas ), e a linha HTTP tem um sinal de menos ou um triângulo para baixo (o que significa que todas as informações sobre a mensagem HTTP são exibidas).

Olhando para as informações no HTTP GET e mensagens de resposta, responda a seguir

perguntas. Ao responder às seguintes perguntas, você deve imprimir o GET e responder

mensagens (veja o laboratório introdutório Wireshark para uma explicação de como fazer isso) e indicar onde na mensagem você encontrou as informações que respondem às seguintes perguntas. Quando você entregar sua tarefa, anotar a saída para que fique claro onde você está recebendo informações para sua resposta

1. Seu navegador está executando a versão HTTP 1.0 ou 1.1? Que versão do HTTP o servidor está executando?

2. Quais idiomas (se houver) o seu navegador indica que ele pode aceitar ao servidor?

3. Qual é o endereço IP do seu computador? E do servidor gaia.cs.umass.edu?

4. Qual é o código de status devolvido do servidor para o seu navegador?

5. Quando foi modificado, no servidor, o arquivo HTML que você está recuperando?

6. Quantos bytes de conteúdo estão sendo devolvidos ao seu navegador?

Em sua resposta à pergunta 5 acima, você pode ter ficado surpreso ao descobrir que o documento que você acabou de recuperar foi modificado pela última vez um minuto antes de baixar o documento. Isso é porque (para este arquivo em particular), o servidor gaia.cs.umass.edu está definindo o tempo modificado pela última vez do arquivo para ser o

tempo atual, e está fazendo isso uma vez por minuto. Assim, se você esperar um minuto entre os acessos, o arquivo parecerá ter sido recentemente modificado, e, portanto, o seu navegador vai baixar uma cópia "nova" de o documento.

2. A interação DE GET/resposta CONDICIONAL HTTP

Lembre-se de que, na Seção 2.2.6 do texto (6ª edição), que a maioria dos navegadores da Web executam cache de objetos e, portanto, executam um GET condicional ao recuperar um objeto HTTP . Antes de executar os passos abaixo, certifique-se que o cache do seu navegador está vazio. (Para fazer isso no Firefox, selecione  *Ferramentas->*Limpar *o*  *Histórico*  *Recente* e

verificar a caixa cache ou para o Internet Explorer, selecione  *Ferramentas->Opções de Internet*  *Options->Delete File;* estas ações removerão arquivos armazenados em cache do cache do seu navegador.) Agora faça o seguinte::

Inicie seu navegador web e certifique-se de que o cache do seu navegador esteja limpo, como discutido acima. Inicie o Wireshark . Digite a seguinte URL no seu navegador

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

Seu navegador deve exibir um arquivo HTML de cinco linhas muito simples. Digite a mesma URL no seu navegador novamente (ou simplesmente selecione o botão de atualização em seu navegador)

Pare a captura de pacotes e digite "http" na janela display-filter-specification, de modo

que apenas mensagens HTTP capturadas serão exibidas mais tarde na janela de listagem de pacotes .

Responda às seguintes perguntas:

8. Inspecione o conteúdo da primeira solicitação HTTP GET do seu navegador para o servidor. Você vê uma linha "IF-MODIFIED-SINCE" no HTTP GET?

9. Inspecione o conteúdo da resposta do servidor. O servidor devolveu explicitamente o conteúdo do arquivo? Como você pode dizer?

10. Agora inspecione o conteúdo da segunda solicitação HTTP GET do seu navegador para o

Servidor. Você vê uma linha "IF-MODIFIED-SINCE:" no HTTP GET? Se assim for, que

informações seguem o cabeçalho "IF-MODIFIED-SINCE:"?

11. Qual é o código de status HTTP e a frase devolvida do servidor em resposta ao

segundo HTTP GET? O servidor devolveu explicitamente o conteúdo do arquivo? Explique..

3. Recuperar documentos longos

Em nossos exemplos até agora, os documentos recuperados foram arquivos HTML simples e curtos. A seguir, veja o que acontece quando baixamos um arquivo HTML longo. Faça o seguinte:

Inicie seu navegador web e certifique-se de que o cache do seu navegador esteja limpo, como discutido acima.

Inicie o Wireshark.

Digite a seguinte URL no seu navegador

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

Seu navegador deve exibir um longo texto, a Declaração de Direitos dos EUA.

Pare a captura do pacotes e digite "http" na- janela display-filter-specification, de modo

que apenas mensagens HTTP capturadas serão exibidas.

Na janela de listagem de pacotes, você deve ver sua mensagem HTTP GET, seguida por múltiplos pacotes de resposta TCP à sua solicitação HTTP GET . Esta resposta de vários pacotes merece um pouco de explicação. Reveja a Seção 2.2 (Figura 2.9 no texto) que a mensagem de resposta HTTP

consiste em uma linha de status , seguida por linhas de cabeçalho, seguida por uma linha em branco, seguida pelo corpo da entidade. No caso do nosso HTTP GET, o corpo da mensagem de resposta é  *todo* o arquivo HTML solicitado. No nosso caso, o arquivo HTML é

bastante longo e 4500 bytes é muito grande para caber em um pacote TCP . A única mensagem de resposta HTTP é, portanto, quebrada em vários pedaços pelo TCP, com cada pedaço sendo contido dentro de um segmento TCP separado (veja Figura 1.24 no texto). Em versões recentes do Wireshark, são indicados cada

segmento TCP como um pacote separado, e o fato de que a única resposta HTTP foi fragmentada em vários pacotes TCP são indicados pelo "segmento TCP de uma PDU remontada" na coluna Informações “reassembled”. Versões anteriores do Wireshark usavam a frase "Continuação" para indicar que todo o conteúdo de uma mensagem HTTP foi quebrado em vários segmentos TCP .. Salientamos aqui que não há nenhuma mensagem "Continuação" em HTTP!

Responda às seguintes perguntas:

12. Quantas mensagens de solicitação HTTP GET seu navegador enviou? Qual o número do pacote que contém a mensagem GET para a Declaração ou Direitos?

13. Qual número do pacote contém o código de status e a frase associadas à resposta ao pedido HTTP GET ?

14. Qual é o código de status e a frase na resposta?

15. Quantos segmentos de TCP contendo dados foram necessários para transportar a única

Resposta HTTP e o texto da Declaração de Direitos?

4. Documentos HTML com objetos incorporados

Agora que vimos como o Wireshark exibe o tráfego de pacotes capturados para grandes arquivos HTML, podemos ver o que acontece quando seu navegador baixa um arquivo com objetos incorporados, ou seja, um arquivo que inclui outros objetos (no exemplo abaixo, arquivos de imagem) que são armazenados em outro servidor.

Faça o seguinte:

Inicie seu navegador web e certifique-se de que o cache do seu navegador esteja limpo, como discutido acima.

Inicie o Wireshark.

Entre com a seguinte URL em seu navegador: http://gaia.cs.umass .edu/wireshark-labs/HTTP-wireshark-file4.html

Seu navegador deve exibir um arquivo HTML curto com duas imagens. Estas duas imagens são

referenciadas no arquivo HTML base. Ou seja, as imagens em si não estão contidas no arquivo

HTML; em vez disso, os URLs para as imagens estão contidos no arquivo HTML baixado. Como discutido no livro, seu navegador terá que recuperar esses logotipos do sites Web indicados. O logotipo do editor do livro é recuperado do site www.aw-bc.com. A imagem

da capa para o nosso 5ª edição (uma de nossas capas favoritas) é armazenada no servidor manic.cs.umass.edu.

Pare a captura de pacotes e digite "http" no filtro da janela display-filter-specification, de modo

que apenas mensagens HTTP capturadas serão exibidas.

Responda às seguintes perguntas:

16. Quantas mensagens HTTP o seu navegador enviou? Para quais endereços da Internet essas mensagens 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.

5 Autenticação HTTP

Finalmente, vamos tentar visitar um site protegido por senha e examinar a sequência de mensagens HTTP trocadas por tal site. A URL http://gaia.cs.umass.edu/wiresharklabs/

protected\_pages/HTTP-wireshark-file5.html é protegida por senha . O nome de usuário é

"wireshark-students" (sem aspas), e a senha é "rede" (novamente, sem aspas).

Então vamos acessar este site protegido por senha. Faça o seguinte:

Certifique-se de que o cache do seu navegador esteja limpo, conforme discutido acima, e feche o seu navegador. Em seguida, inicie seu navegador

• Inicie o sniffer de pacotes Wireshark

• Digite o seguinte URL em seu navegador

<http://gaia.cs.umass.edu/wiresharklabs/protected_pages/HTTP-wireshark-file5.html>

Digite o nome de usuário e a senha solicitados na caixa pop-up.

Pare a captura de pacotes do Wireshark e digite “http” na janela display-filter-specification, então que apenas mensagens HTTP capturadas serão exibidas posteriormente na janela de listagem de pacotes.

Agora vamos examinar a saída do Wireshark. Responda as seguintes questões:

18. Qual é a resposta do servidor (código de status e frase) em resposta à primeira mensagem HTTP GET do seu navegador?

19. Quando o seu navegador envia a mensagem HTTP GET pela segunda vez, que novo campo está incluído na mensagem HTTP GET?

O nome de usuário (wireshark-students) e a senha (network) que você digitou são codificados no sequência de caracteres (d2lyZXNoYXJrLXN0dWRlbnRzOm5ldHdvcms=) seguindo o Cabeçalho “Authorization: Basic” na mensagem HTTP GET do cliente. Embora possa parecer que seu nome de usuário e senha são criptografados, eles são simplesmente codificados em um formato conhecido como formato Base64. O nome de usuário e a senha não são criptografados! Para ver isso, acesse http://www.motobit.com/util/base64-decoder-encoder.asp e insira a string codificada em base64 d2lyZXNoYXJrLXN0dWRlbnRz e decodificar. Voilá! Você traduziu da codificação Base64 para codificação ASCII e, portanto, deve ver seu nome de usuário! Para visualizar a senha, digite o restante da string Om5ldHdvcms= e pressione decode. Já que qualquer um pode baixar uma ferramenta como o Wireshark e sniff pacotes (não apenas seus próprios) passando por seu adaptador de rede, e qualquer um pode traduzir de Base64 para ASCII (você acabou de fazer isso!), deve ficar claro para você que senhas simples em sites WWW não são seguras a menos que medidas adicionais sejam tomadas. Não tenha medo! Existem maneiras de tornar o acesso à WWW mais seguro. No entanto, claramente precisaremos de algo que vá além da estrutura básica de autenticação HTTP