Disciplina: COM241 – Administração e Gerência de Redes de Computadores

Professor: Bruno Guazzelli Batista

Email: [brunoguazzelli@unifei.edu.br](mailto:brunoguazzelli@unifei.edu.br)

Alunos: Ivan Leoni vilas Boas 2018009073

**Atividade 1. O Básico do GET do HTTP/interação de resposta**

Vamos começar explorando o HTTP fazendo o download de um arquivo HTML muito simples – um arquivo pequeno e que não contém referências para objetos. Faça o seguinte:

Inicie o navegador Web.

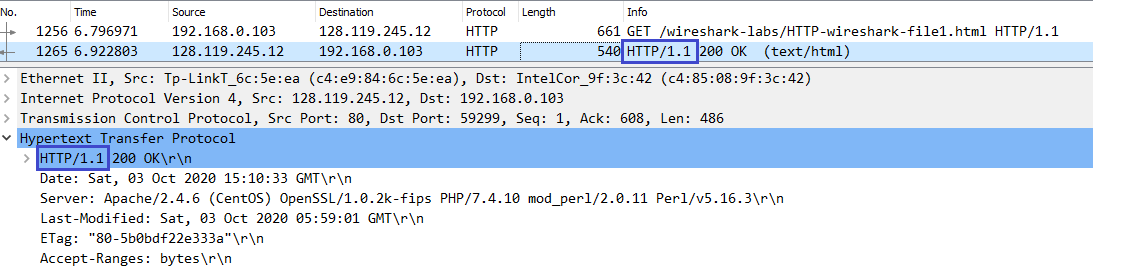
1. Inicie o sniffer de pacotes, Wireshark, mas não inicie a captura de pacotes. Digite “http” (somente as letras sem as aspas) na janela display-filter-specification, assim somente as mensagens HTTP serão mostradas pela ferramenta.
2. Espere um instante antes de iniciar a captura de pacotes.
3. Entre o endereço abaixo no navegador:
4. <http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file1.html>O navegador deve mostrar uma página bem simples com uma linha.
5. Pare a captura de pacotes do Wireshark.

Observando a informação do GET HTTP e a mensagem de resposta, responda as questões abaixo. Quando responder, inclua uma impressão das mensagens de GET e a resposta e indicar onde você encontrou as respostas.

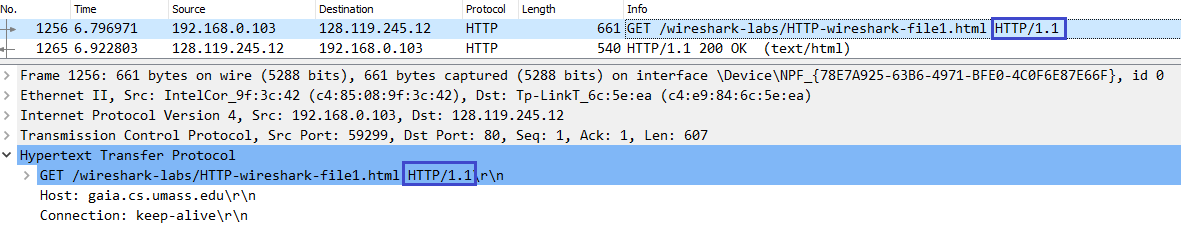
1. O seu navegador está rodando a versão 1.0 ou 1.1 do HTTP? Que versão do HTTP está sendo rodada no servidor?

**Sevidor e navegador usando HTTP 1.1.**

**Servidor:**

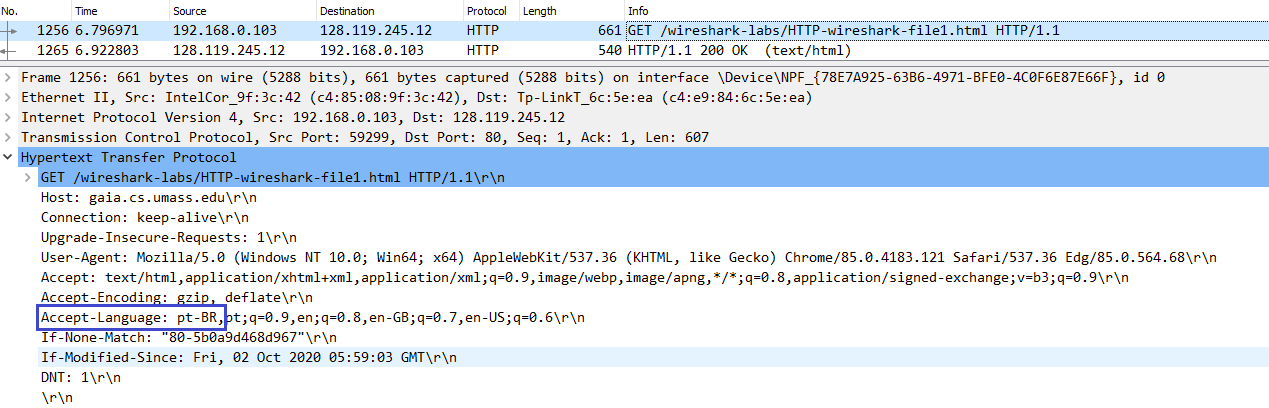
****

**Navegador:**



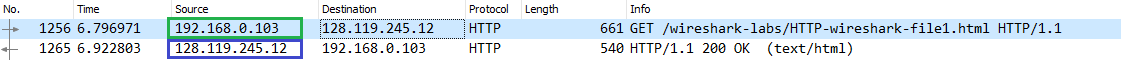
1. Que linguagem (se tiver) o seu navegador indica que ele pode aceitar do servidor?

**Português**



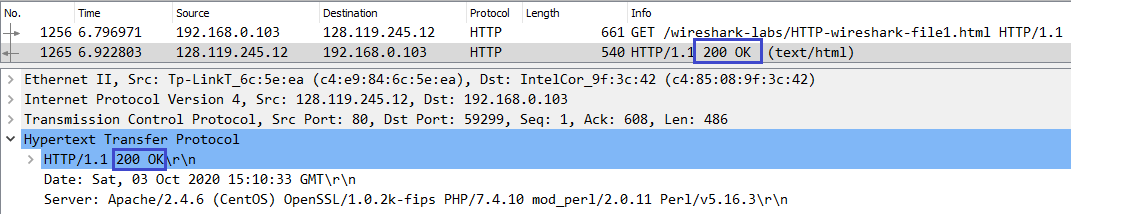
1. Qual é o endereço IP do seu computador? E do servidor gaia.cs.umass.edu?

**Do computador é o 192.168.0.103 e do Servidor é 128.119.245.12.**

****

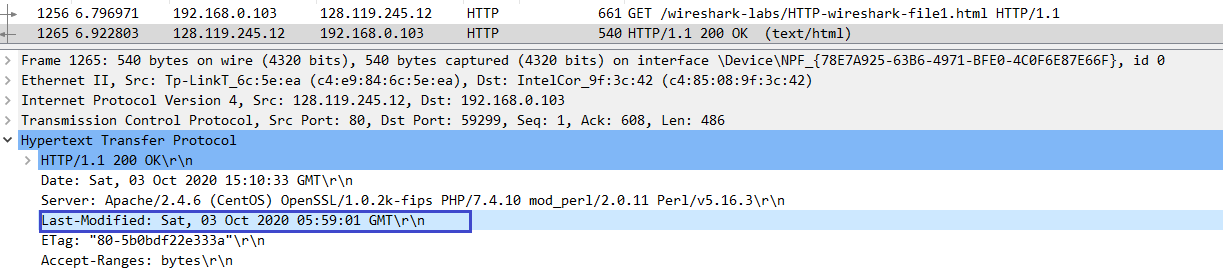
1. Qual é o código de estado retornado pelo servidor para o seu navegador?

**Código retornado é 200.**



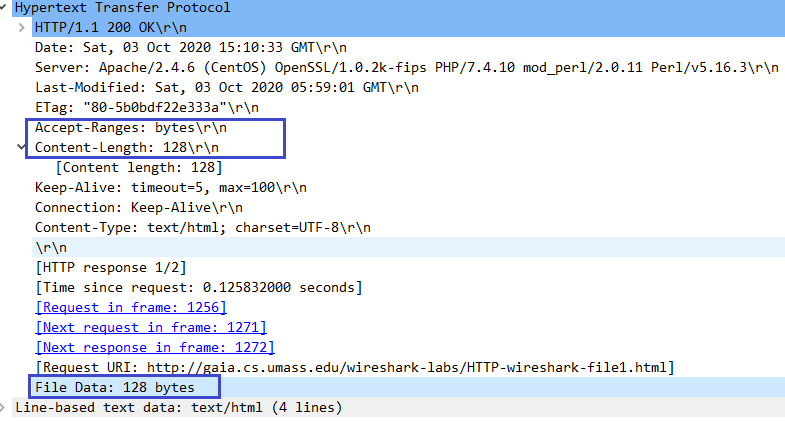
1. Quando foi a última alteração feita no arquivo HTML?

**Foi feita dia 03 de outubro de 2020 as 05:59:01 GMT**



1. Quantos bytes de conteúdo são retornados para o seu navegador?

**São retornados 128 bytes.**



1. Inspecionando o dado na janela de conteúdo do pacote, você viu algum cabeçalho com dados que não são mostrados na janela de listagem do Wireshark? Se sim, qual?

Não, todos os dados de cabeçalho disponíveis são mostrados na janela de listagem.

**Atividade 2. O GET HTTP CONDITIONAL/interação de resposta**

Um cache Web, também denominado servidor proxy, é uma entidade da rede que atende requisições HTTP em nome de um servidor Web de origem. Para isso, ele utiliza um GET condicional, um mecanismo que permite que um cache verifique se seus objetos estão atualizados.

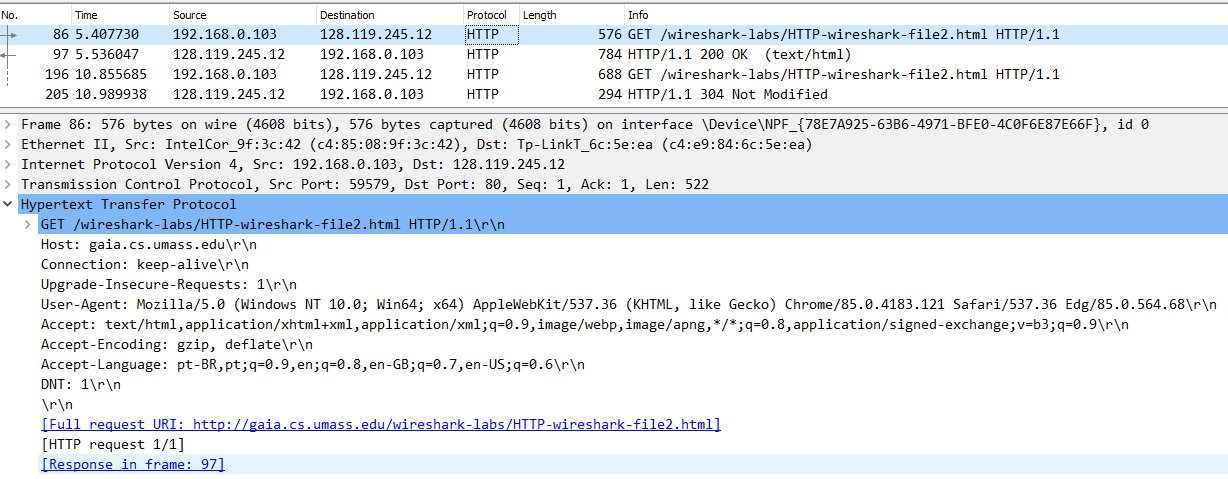
Antes de fazer esta atividade, certifique-se de que o cache do navegador esteja limpo (faça isso usando a opção para apagar o histórico do navegador). Agora, faça o seguinte:

1. Inicie o navegador e certifique-se de que o cache esteja limpo.
2. Inicie o Wireshark.
3. Entre com a URL no navegador <http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file2.html>
4. Seu navegador deve mostrar um arquivo bem simples HTML com cinco linhas.
5. Rapidamente entre com a URL no navegador novamente (ou simplesmente selecione o botão para recarregar a página).
6. Pare o Wireshark e entre com “http” na janela em que se especifica filtros para que somente as mensagens HTTP possam ser mostradas na janela de listagem de pacotes.

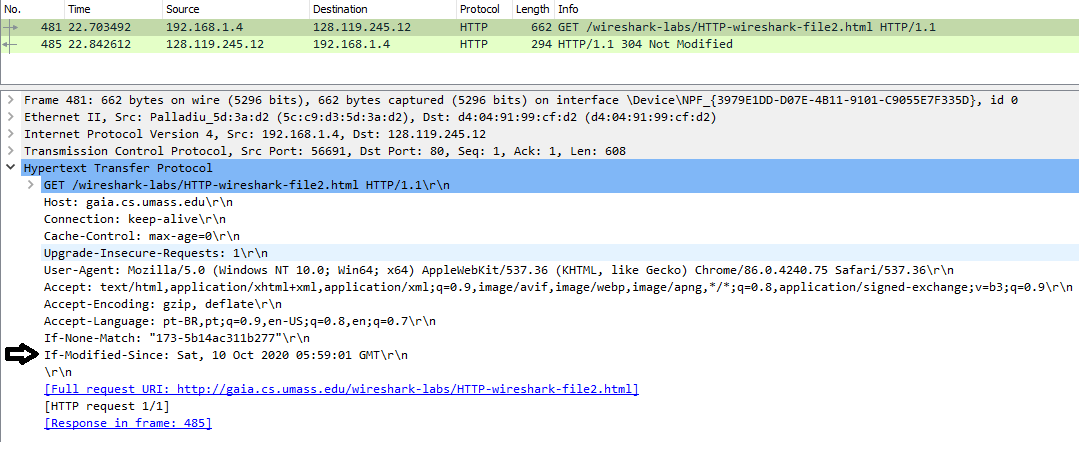
Responda às seguintes questões:

1. Inspecione o conteúdo da primeira requisição GET do HTTP de seu navegador para o servidor. Você viu uma linha com “IF-MODIFIED-SINCE” no GET do HTTP?

**Na primeira execução não achamos**

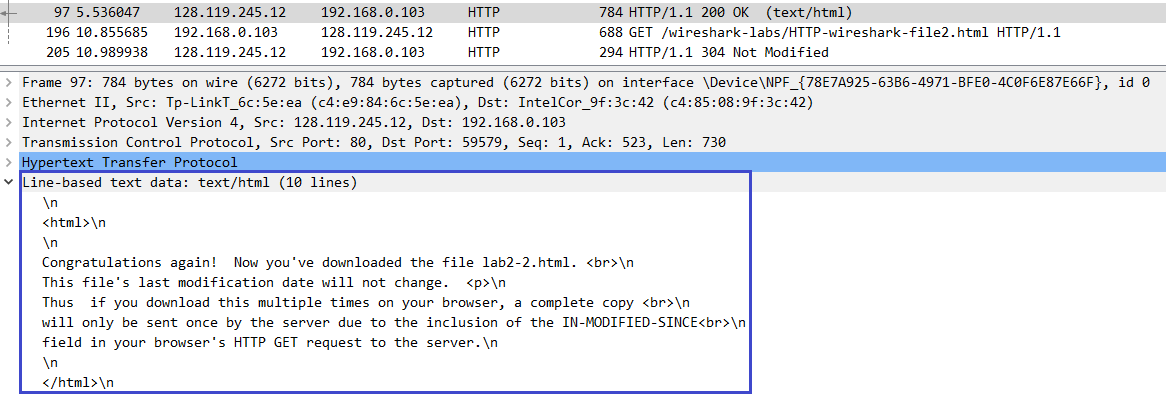
****

**Na segunda tentativa uma linha foi encontrada**

****

1. Inspecione o conteúdo da resposta do servidor. O servidor retornou o conteúdo do arquivo explicitamente? Como isso se observou?

**Explicitamente. No line-based text data**

****

1. Agora investigue o conteúdo da segunda requisição GET do HTTP de seu navegador para o servidor. Você viu uma linha com “IF-MODIFIED-SINCE” no GET do HTTP? Se sim, que informação segue o cabeçalho “IF-MODIFIED-SINCE:”?

**Sim achamos: If-Modified-Since: 03 de outubro de 2020**

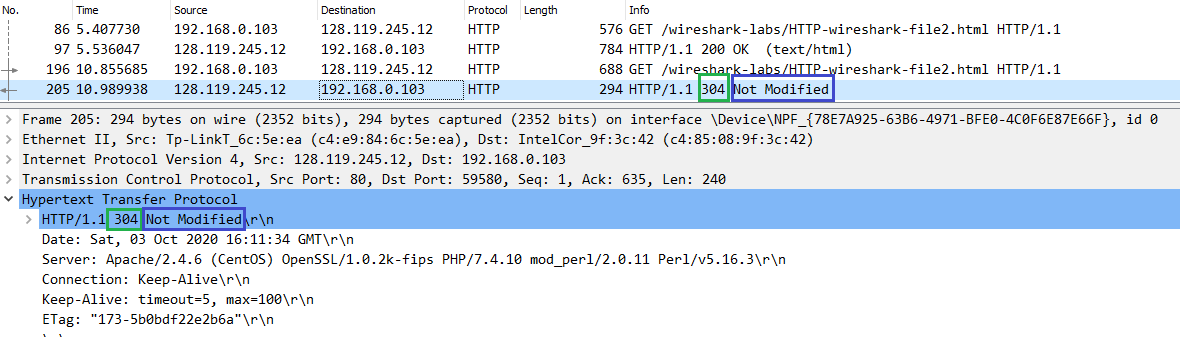
**Serve para que o navegador saiba se precisa atualizar a página ou usar os dados locais de acordo com as datas. Data da última alteração no servidor e do “arquivo” do navegador.**

****

1. Qual é o código de estado do HTTP e a frase retornada do servidor em resposta ao segundo GET do HTTP? O servidor explicitamente retornou o conteúdo do arquivo? Explique!

**Codigo 304 e frase: “Not Modified”.**

**O arquivo do servidor e o arquivo no navegador (a pagina** <http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file2.html>) **estão com a mesma data de modificação. Ou seja, não precisa ser alterado e o servidor por isso retorna o código 304 e a frase “Not Modified” dizendo que não foi modificado.**



**Atividade 3. Obtendo longos documentos**

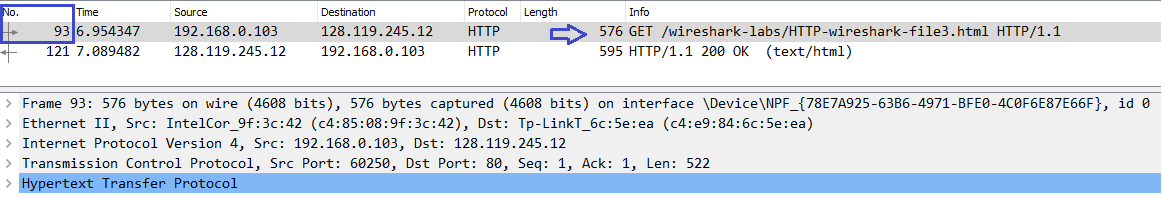
Em nossos exemplos até aqui, os documentos HTML obtidos eram simples e curtos. Vamos analisar o que acontece quando trabalharmos com arquivos HTML longos. Faça o seguinte:

1. Inicie o seu navegador e execute a limpeza do cache, como discutido acima.
2. Inicie o Wireshark.
3. Entre com a seguinte URL no navegador <http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file3.html>
4. Seu navegador deve mostrar o documento “US Bill of Rights”.
5. Pare a captura de pacotes pelo Wireshark e entre com “http” na janela em que se especifica filtros para que somente as mensagens HTTP possam ser mostradas na janela de listagem de pacotes.

Responda às seguintes questões:

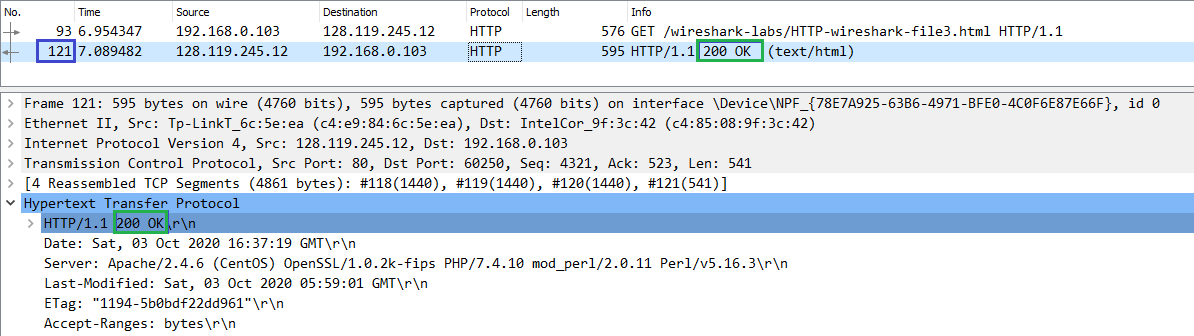
1. Quantas mensagens de requisição GET de HTTP o seu navegador enviou? Qual número de pacote contém a mensagem de GET do “Bill or Rights”?

**Apenas uma mensagem GET e o número do pacote é 93**



1. Qual número de pacote contém o código de estado e a frase associada com a resposta do pedido de GET de HTTP?

**Número do pacote: 121**



1. Qual é o código de estado e a frase de resposta?

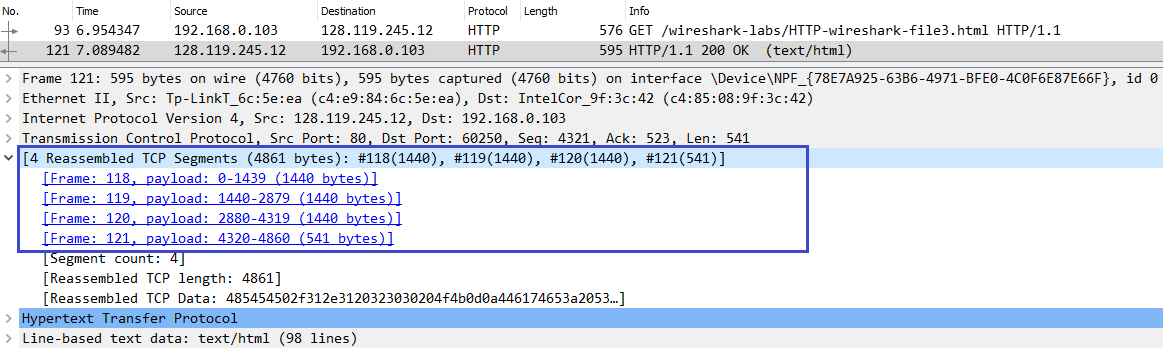
**Código de resposta:200**

**Frase de resposta: ok**

**C:\Users\IVAN LEONI\Desktop\17.png**

1. Quantos segmentos TCP contendo dados são necessários para uma simples resposta HTTP para o texto?

**Quatro segmentos**



**Atividade 4. Documentos HTML com objetos incluídos**

Agora que você viu com o Wireshark mostra o tráfego de pacotes capturados para arquivos HTML grandes, nós podemos ver o que acontece quando um navegador baixa um arquivo com objetos incluídos, isto é, um arquivo que inclui outros objetos (no exemplo abaixo, arquivos de imagens) que estão armazenados em outros servidores. Faça o seguinte:

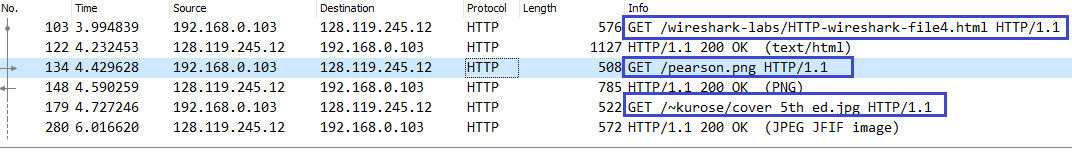
1. Inicie o seu navegador e execute a limpeza do cache, como discutido acima.
2. Inicie o Wireshark.
3. Entre com a seguinte URL no navegador <http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file4.html>
4. Seu navegador deve mostrar um pequeno arquivo HTML com duas imagens. Essas imagens são referenciadas no arquivo HTML base. Isto é, as imagens propriamente não estão contidas no HTML, ao invés disso, as URLs para as imagens estão contidas no arquivo HTML baixado.
5. Pare a captura de pacotes pelo Wireshark e entre com “http” na janela em que se especifica filtros para que somente as mensagens HTTP possam ser mostradas na janela de listagem de pacotes.

Responda às seguintes questões:

1. Quantas mensagens de requisição HTTP o seu navegador enviou? Para quais endereços da Internet foram enviados esses pedidos GET?

**Três mensagens de requisição HTTP.**

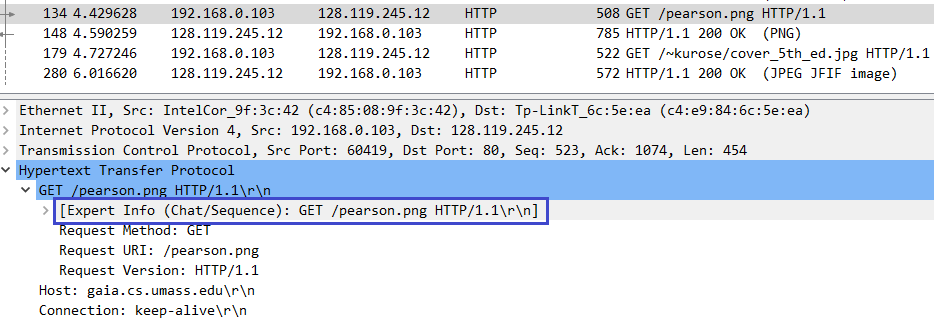
**Foram enviados para o endereço 128.119.245.12**



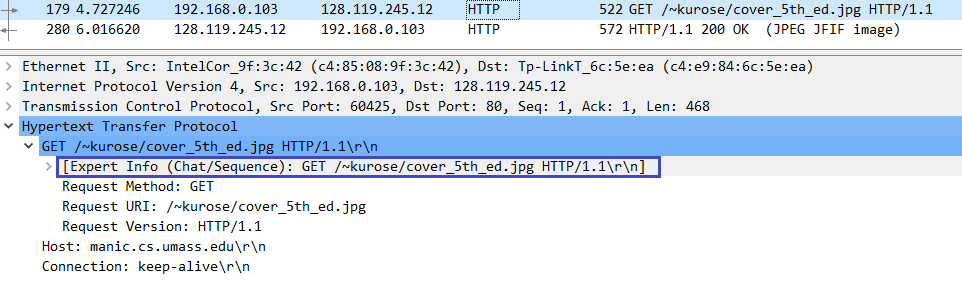
1. Você pode falar se o navegador baixou as duas imagens serialmente ou se elas foram baixadas dos dois sites paralelamente? Explique!

**Serialmente. Uma após a outra, como pode ser observado pelas imagens abaixo indicado no Expert Info e também pode ser observado a sequência pelo tempo de duração (TIME) de cada pacote onde é notório que o tempo da requisição seguinte e maior justamente porque esta começa sempre logo após que requisição anterior foi entendida.**

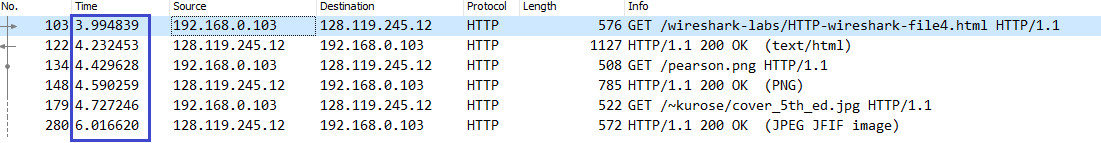
GET Imagem pearson.png:



GET Imagem cover\_5th\_ed.jpj:



Time progressivo e sequencial dos get’s:



**Atividade 5. Autenticação HTTP**

Finalmente, vamos visitar um site web protegido por senha e examinar a sequência de mensagens HTTP trocadas com esse site. A URL http://gaia.cs.umass.edu/wiresharklabs/protected\_pages/HTTP-wireshark-file5.html tem proteção por senha. O usuário é “wireshark-students” (sem as aspas) e a senha é “network” (novamente, sem as aspas). Então, vamos acessar este site “seguro”, protegido por senha. Faça o seguinte:

1. Inicie o seu navegador e execute a limpeza do cache, como discutido acima.
2. Inicie o Wireshark.
3. Entre com a seguinte URL no navegador <http://gaia.cs.umass.edu/wireshark-labs/protected_pages/HTTP-wireshark-file5.html>Digite o usuário e a senha nos campos apropriados.
4. Pare a captura de pacotes pelo Wireshark e entre com “http” na janela em que se especifica filtros para que somente as mensagens HTTP possam ser mostradas na janela de listagem de pacotes.

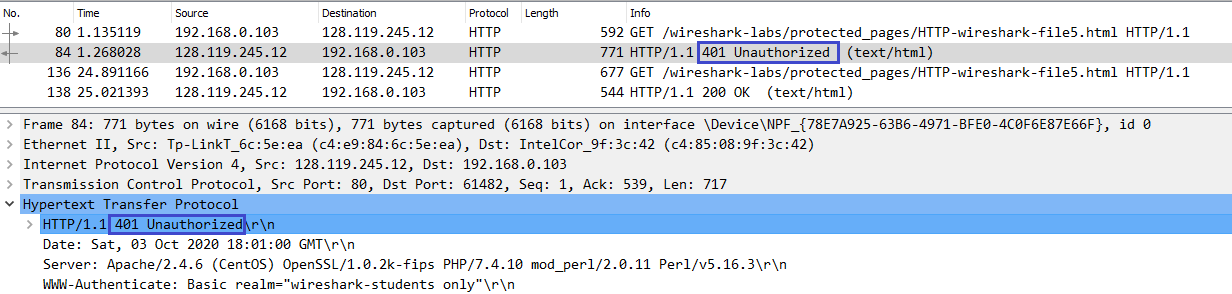
Agora, vamos examinar a saída do Wireshark. Você deve primeiramente querer entender melhor sobre autenticação HTTP. Você pode ver o material “HTTP Access Authentication Framework” em <http://frontier.userland.com/stories/storyReader$2159>

Responda às seguintes questões:

1. Qual é a resposta do servidor (código de estado e frase) na resposta ao GET de HTTP inicial do navegador?

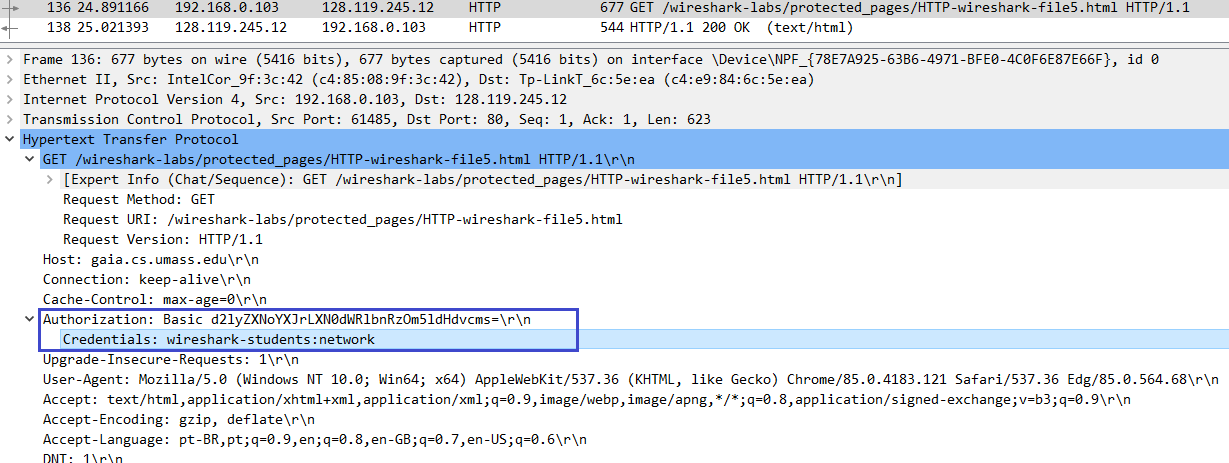
**Código de estado: 401**

**Frase de resposta: Unauthorized**

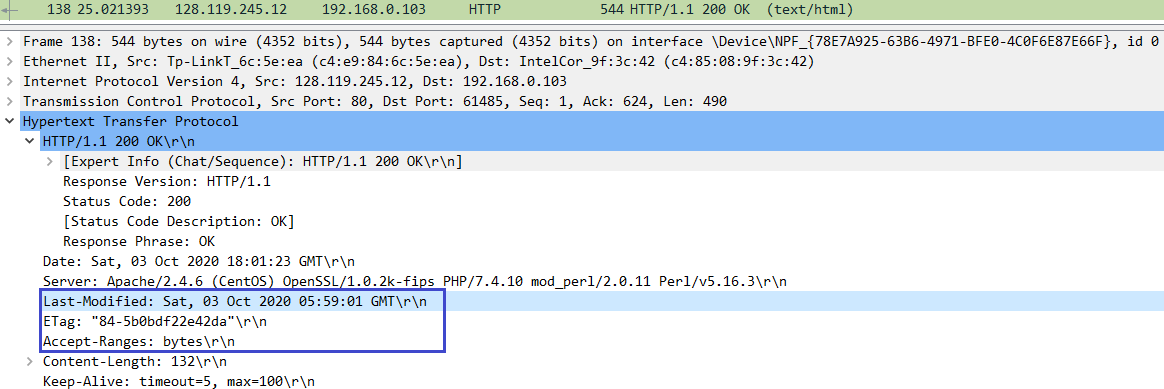


1. Quando seu navegador envia mensagem GET de HTTP pela segunda vez, que novos campos são incluídos na mensagem GET de HTTP?

**Na mensagem Get solicitada pelo cliente é inserido o campo de usuário e senha criptografados**.



**Na mensagem HTTP retornada pelo servidor são inseridos o Last-Modifield, o ETag e o Accept-Rangers.**



O usuário (wireshark-students) e a senha (network) que você entrou são codificados para a cadeia de caracteres (d2lyZXNoYXJrLXN0dWRlbnRzOm5ldHdvcms=) seguidos do cabeçalho “Authorization: Basic” na mensagem de GET do HTTP do cliente. Enquanto isso parece que o seu usuário e a sua senha estão criptografados, eles, na verdade, estão codificados no formato conhecido como formato Base64. O usuário e a senha não estão criptografados! Para ver isso vá para http://www.motobit.com/util/base64-decoder-encoder.asp e entre com a cadeia de caracteres codificadas na base64 d2lyZXNoYXJrLXN0dWRlbnRz e decodifique. Voila! Você acabou de traduzir uma codificação Base64 para codificação ASCII e o resultado deve ter sido o usuário! Para ver a senha, entre com o restante da cadeia Om5ldHdvcms= e pressione decodificar. Desde que qualquer um pode baixar uma ferramenta como o Wireshark e capturar pacotes (não somente os seus próprios pacotes) passando pela rede e qualquer um pode traduzir da Base64 para ASCII (você acabou de fazer isso!), está claro que senhas simples na WWW não são seguras a menos que medidas adicionais sejam tomadas.