



Redes de Computadores – Laboratório 2 (14/03/2017) - Entrega: 03/06/2013

Este laboratório prático tem por objetivo usar o Wireshark para analisar o conteúdo dos pacotes de dois protocolos executados na camada de aplicação: HTTP e DNS.

1) DNS

O *Domain Name System* (DNS) traduz os nomes de máquinas em endereços IP. Nesta etapa do laboratório, vamos analisar de perto como o DNS funciona do ponto de vista do cliente. O papel do cliente no DNS é relativamente simples: envia uma busca (*query*) para o seu DNS local e recebe uma resposta (*response*). Por mais simples que parece ser, muita coisa acontece “nos bastidores”. Apesar disso, o DNS é uma parte indissociável do que chamamos de Internet.

2.1) O comando `nslookup`

Neste laboratório vamos utilizar o comando `nslookup`, disponível no Windows e Linux. Para executar o `nslookup`, você deve usar a linha de comando:

No Windows:

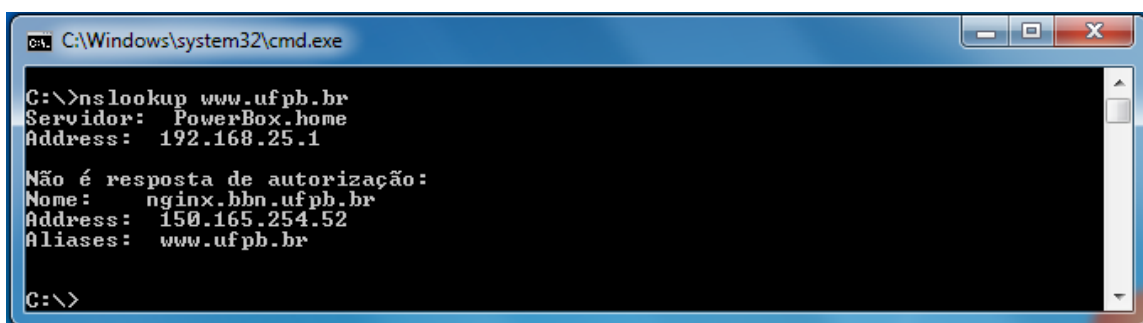
Menu Iniciar → Executar e logo em seguida digitar `cmd` e pressionar `ENTER` ou clicar em `OK`.

No Linux:

Abrir um `Terminal` (`shell`).

Para testar se os comandos estão funcionando, digite: `nslookup www.ufpb.br`

A resposta deve ser algo como a figura 1 (não necessariamente as mesmas saídas):



```
C:\Windows\system32\cmd.exe

C:\>nslookup www.ufpb.br
Servidor:  PowerBox.home
Address:  192.168.25.1

Não é resposta de autorização:
Nome:      nginx.bbn.ufpb.br
Address:   150.165.254.52
Aliases:   www.ufpb.br

C:\>
```

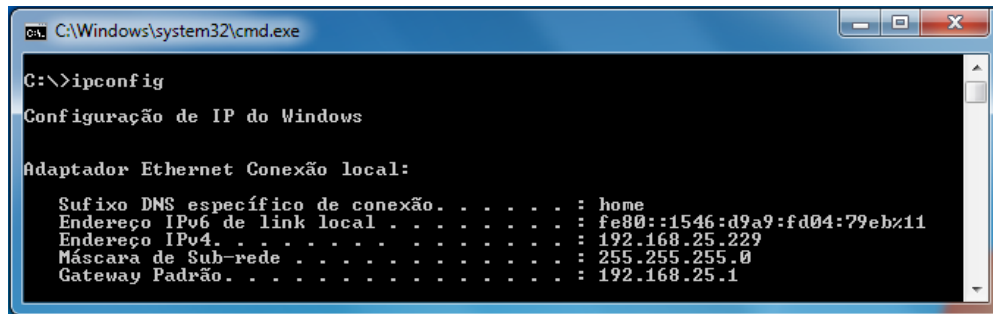
Figura 1: Saída do comando `nslookup`.

2.2) Rastreando DNS com o *Wireshark*

Aqui você deverá iniciar uma captura pelo *Wireshark* de uma atividade de comunicação de navegação na Web. Para isso:

- Apague o cache de DNS da sua máquina com o comando `ipconfig /flushdns`
- Abra o navegador (Firefox, Chrome, Internet Explorer, etc.) e apague todo o cache de navegação (configuração específica do seu navegador).
- Abra o *Wireshark* e digite `ip.addr == seu_endereço_IP`, onde `seu_endereço_IP` é o IP da sua máquina.

Dica: Para descobrir o endereço IP da sua máquina, digite no *Prompt de Comando* do Windows o comando `ipconfig`. Depois, na saída, procure o campo *Endereço IPv4*, onde você encontrará o endereço IP do seu computador (figura 2a e 2b). Já no Linux, use o comando `/sbin/ifconfig eth0` e procure pelo campo `inet end`.



```
C:\Windows\system32\cmd.exe
C:\>ipconfig

Configuração de IP do Windows

Adaptador Ethernet Conexão local:

    Sufixo DNS específico de conexão. . . . . : home
    Endereço IPv6 de link local . . . . . : fe80::1546:d9a9:fd04:79eb%11
    Endereço IPv4. . . . . : 192.168.25.229
    Máscara de Sub-rede . . . . . : 255.255.255.0
    Gateway Padrão. . . . . : 192.168.25.1
```

Figura 2a: Saída do comando `ipconfig` no Windows.



```
hacks@lab2-dcx-15:~$ /sbin/ifconfig eth0
eth0      Link encap:Ethernet  Endereço de HW 00:24:81:b1:ef:a3
          inet end.: 10.0.2.15  Bcast:10.0.255.255  Masc:255.255.0.0
          endereço inet6: fe80::224:81ff:feb1:efa3/64  Escopo:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Métrica:1
          pacotes RX:186307  erros:0  descartados:0  excesso:0  quadro:0
          Pacotes TX:5854  erros:0  descartados:0  excesso:0  portadora:0
          colisões:0  txqueuelen:1000
          RX bytes:20686044 (20.6 MB)  TX bytes:729828 (729.8 KB)
          IRQ:18

hacks@lab2-dcx-15:~$
```

Figura 2b: saída do comando `/sbin/ifconfig` no Linux.

- Inicie agora uma captura pelo *Wireshark*.
- Com o seu navegador, visite <http://www.ietf.org>
- Pare a captura de pacotes no *Wireshark*.

- a) Localize as mensagens de DNS de *query* e *response*. Estas mensagens foram enviadas via UDP ou TCP? (Dica: procure todas as mensagens em que na coluna *Protocol* aparece *DNS*).
- b) Qual é a porta de destino da mensagem de *query*?
- c) Para qual endereço IP a mensagem de *query* foi enviada?
- d) Examine a mensagem de DNS enviada. Qual é o “Tipo” (*Type*) da *query*? A mensagem de *query* contém alguma “resposta” (*Answer*)?
- e) Examine a mensagem de DNS resposta. Quantas “respostas” (*Answers*) foram fornecidas? O que contém cada uma dessas respostas?

Agora vamos trabalhar um pouco com o `nslookup`.

- Inicie uma captura de pacotes.
- Digite o comando `nslookup type=NS ufpb.br`
- Pare a captura de pacotes.

Você deve ver algo assim no *Wireshark* (figura 3):

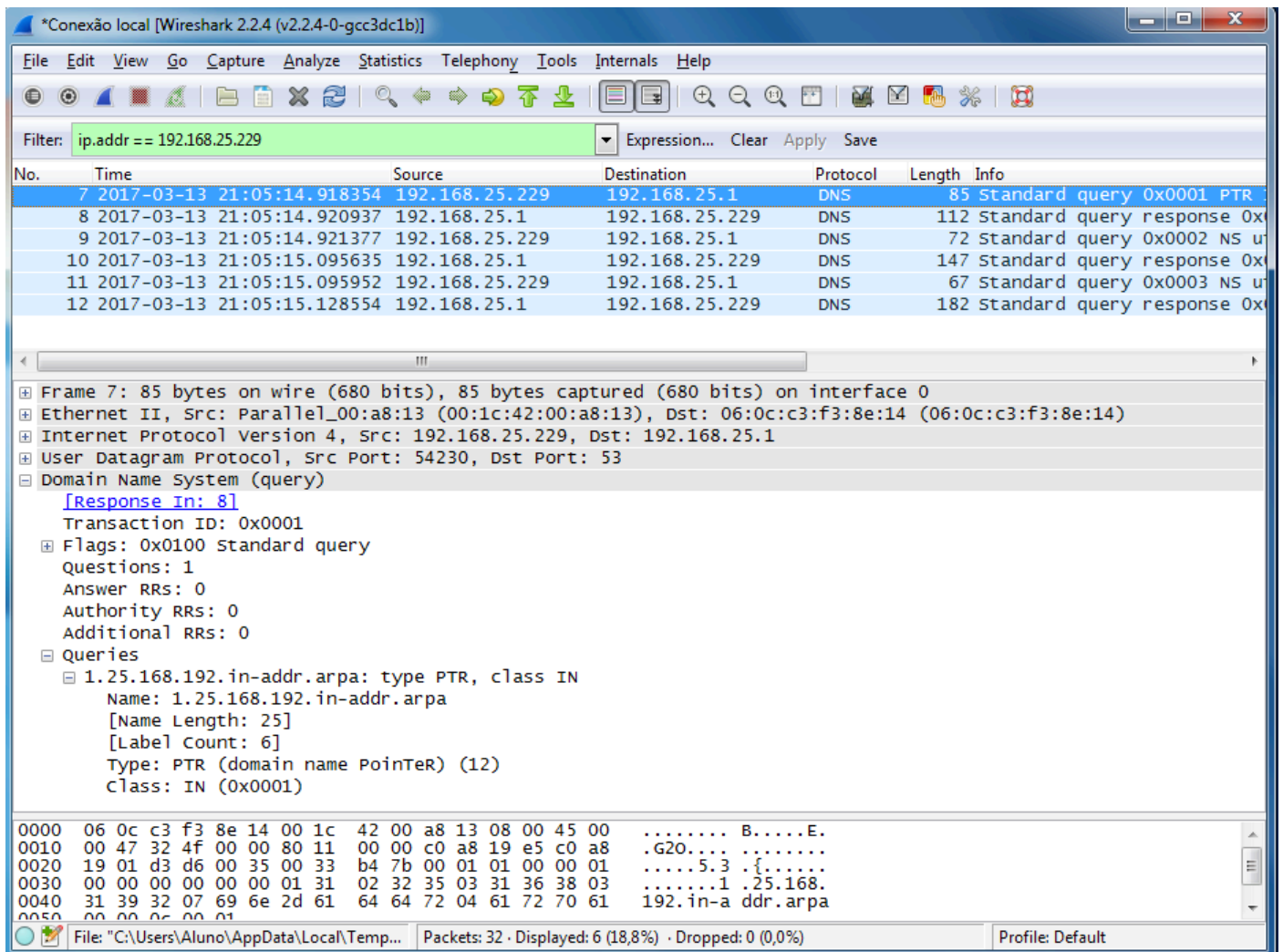


Figura 3: Captura das mensagens trocadas após execução do comando `nslookup`.

- f) Você verá mensagens de DNS *query* e DNS *response* no tráfego do Wireshark. Responda as questões c), d) e e) novamente utilizando esta nova captura de tráfego.

2) HTTP

Nesta seção vamos investigar a operação do protocolo de aplicação HTTP: a interação básica HTTP de GET/RESPONSE, formato da mensagem HTTP, requisição de grandes arquivos HTML, requisição de arquivos HTML com objetivos embutidos e segurança e autenticação em HTTP.

2.1) Interação básica HTTP GET/RESPONSE

Vamos iniciar a análise do HTTP fazendo o *download* de um arquivo HTML muito simples (este arquivo não contém objetos embutidos). Faça os seguintes procedimentos:

- Inicie um navegador.
- Inicie o *Wireshark* e crie um filtro para captura de pacotes HTTP. Isso é feito colocando-se a palavra *http* no campo *Filter*: do *Wireshark*.
- Inicie a captura de pacotes no *Wireshark*.
- Abra o navegador e digite: <http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file1.html>
Seu navegador deve mostrar uma página simples em HTML.
- Pare a captura de pacotes.

Observando a informação das mensagens HTTP GET e RESPONSE, responda as questões a seguir. Ao responder as questões abaixo, você deve imprimir as mensagens de GET

e RESPONSE e indicar na mensagem onde você encontrou a informação que responde as perguntas a seguir.

- a) Seu navegador está executando o protocolo HTTP 1.0 ou 1.1? Qual é a versão do HTTP rodando no servidor?
- b) Quais os idiomas (se existirem) seu navegador pode aceitar?
- c) Qual o endereço IP do seu computador? E do servidor `gaia.cs.umass.edu` server?
- d) Quando foi a última modificação no servidor do arquivo HTML recebido?
- e) Quantos *bytes* de conteúdo são retornados para o seu navegador?

2.2) Interação HTTP CONDICIONAL GET/RESPONSE

A maioria dos navegadores realiza *cache* de objetos e, conseqüentemente, também realiza um GET condicional quando requisita um objeto HTTP. Para os passos a seguir, você deverá esvaziar o *cache* do seu navegador (funcionalidade limpar cache). Em seguida faça:

- Inicie o navegador e tenha certeza que o *cache* está limpo (o procedimento de limpeza de *cache* varia de acordo com o navegador utilizado).
- Inicie o *Wireshark*.
- Entre o seguinte endereço no navegador:
<http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file2.html>
Seu navegador deve mostrar uma página simples em HTML
- Entre novamente o mesmo endereço no navegador (ou simplesmente recarregue a página – botão recarregar).
- Pare o *Wireshark* e configure um filtro *http* para mostrar somente o tráfego HTTP.

Responda as seguintes questões:

- f) Inspecione o conteúdo do primeiro pacote HTTP GET REQUEST do seu navegador para o servidor. Você consegue ver a linha “IF-MODIFIED-SINCE” no HTTP GET?
- g) Inspecione o conteúdo da resposta do servidor. O servidor explicitamente retorna o conteúdo do arquivo? O que você pode dizer sobre isso?
- h) Agora inspecione o conteúdo do segundo HTTP GET REQUEST (após recarregar a página) do seu navegador para o servidor. Você consegue ver a linha “IF-MODIFIED-SINCE” no HTTP GET? Se sim, qual a informação que se segue no cabeçalho “IF-MODIFIED-SINCE”?

2.3) Requisitando documentos grandes

Em nossos exemplos até agora, os documentos requisitados foram páginas HTML simples e curtas. Vamos ver o que acontece quando requisitamos um arquivo HTML grande. Faça o seguinte:

- Inicie o seu navegador e tenha certeza que o *cache* está vazio (realize o procedimento de limpar o *cache*).
- Inicie o *Wireshark*.
- Entre o seguinte endereço no navegador:
<http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file3.html>
Seu navegador deve mostrar um texto longo.
- Pare o *Wireshark* e configure um filtro “http”.

Feitos os procedimentos acima, você deve ver a mensagem de HTTP GET e HTTP RESPONSE no filtro do *Wireshark*. A mensagem GET RESPONSE dessa vez é diferente, pois contém um único arquivo HTML grande, de aproximadamente 4500 bytes, que como veremos, não cabe em um segment TCP (nível de transporte). Essa grande mensagem é quebrada em vários pedaços e colocados em diferentes pacotes TCP para então poder ser transmitida. Com isto podemos ver que mensagem HTTP existe somente uma, mas segments TCP existem vários, dependendo do tamanho da mensagem. Agora responda as seguintes questões:

- i) Quantas mensagens de HTTP GET REQUEST forem enviadas pelo seu navegador?
- j) Quantos segmentos TCP foram necessários para transmitir a mensagem de resposta HTTP?

2.4) Documentos HTML com Objetos Embutidos

Vimos até agora como o protocolo HTTP trabalha com documentos HTML simples (grandes e pequenos). Agora veremos o que acontece quando seu navegador requisita um arquivo HTML que contenha objetos embutidos, ou seja, arquivos que incluem referências para outros objetos (por exemplo, arquivos de imagens) que estão armazenados em outros servidores. Faça o seguinte:

- Inicie o seu navegador e tenha certeza que o *cache* está vazio (realize o procedimento de limpar o *cache*).
- Inicie o Wireshark.
- Entre o seguinte endereço no navegador:
<http://gaia.cs.umass.edu/wireshark-labs/HTTP-wireshark-file4.html>
Seu navegador deve mostrar um texto curto com duas imagens. Estas duas imagens são referenciadas pelo arquivo HTML inicial, ou seja, as imagens não estão contidas no arquivo HTML, são referências. Ao interpretar o HTML, o navegador requisita estas duas imagens dos endereços indicados no arquivo HTML.
- Pare o Wireshark e configure um filtro “http”.

Responda as seguintes questões:

- k) Quantas mensagens HTTP GET REQUEST foram enviadas pelo seu navegador? Para quais endereços na Internet essas mensagens de GET REQUEST foram enviadas?
- l) Você pode dizer se o seu navegador baixou as duas imagens de maneira serial (uma após a outra) ou foram baixadas em paralelo? Explique sua resposta.