MINISTÉRIO DA EDUCAÇÃO
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SANTA CATARINA
CURSO DE ENGENHARIA DE TELECOMUNICAÇÕES - CÂMPUS SÃO JOSÉ

RELATÓRIO TÉCNICO

LABORATÓRIO WIRESHARK E TCPDUMP

Arthur Cadore Matuella Barcella

TAREFA:

Ferramentas básicas: WireShark, encapsulamento e tcpdump

Objetivos

Após este laboratório o aluno deverá ser capaz de:

- utilizar a ferramenta wireshark para captura de pacote:
  - funções básicas de filtragem na captura e no display;
  - verificação de estruturas de pacotes;
- consolidar o conceito de protocolo e de camadas de protocolos através da análise de troca de pacotes com ping e traceroute usando:
  - as janelas com detalhes dos pacotes e encapsulamentos;
  - a opção de *flow graph* para visualizar as trocas de mensagens.

Sobre o analisador Wireshark

O analisador de pacotes exibe os conteúdos de todos os campos dentro de uma mensagem de protocolo. Para que isso seja feito, o analisador de pacotes deve "entender" a estrutura de todas as mensagens trocadas pelos protocolos.

Suponha que estamos interessados em mostrar os vários campos nas mensagens trocadas pelo protocolo HTTP na Figura 5. O analisador de pacotes entende o formato dos quadros Ethernet, e desta forma pode identificar o datagrama IP dentro de um quadro. Ele também entende o formato do datagrama IP, para que ele possa extrair o segmento TCP dentro do datagrama IP. Ele entende a estrutura do segmento TCP, para que possa extrair a mensagem HTTP contida no segmento. Finalmente, ele entende o protocolo HTTP e então, por exemplo, sabe que os primeiros bytes de uma mensagem HTTP contém a cadeia "GET", "POST" ou "HEAD".

Nós utilizaremos o *sniffer* Wireshark (http://www.wireshark.org) para estes laboratórios, o que nos permite exibir os conteúdos das mensagens sendo enviadas/recebidas de/por protocolos em diferentes camadas da pilha de protocolos. Tecnicamente falando, Wireshark é um analisador de pacotes que pode ser executado em computadores com Windows, Linux/UNIX e MAC.

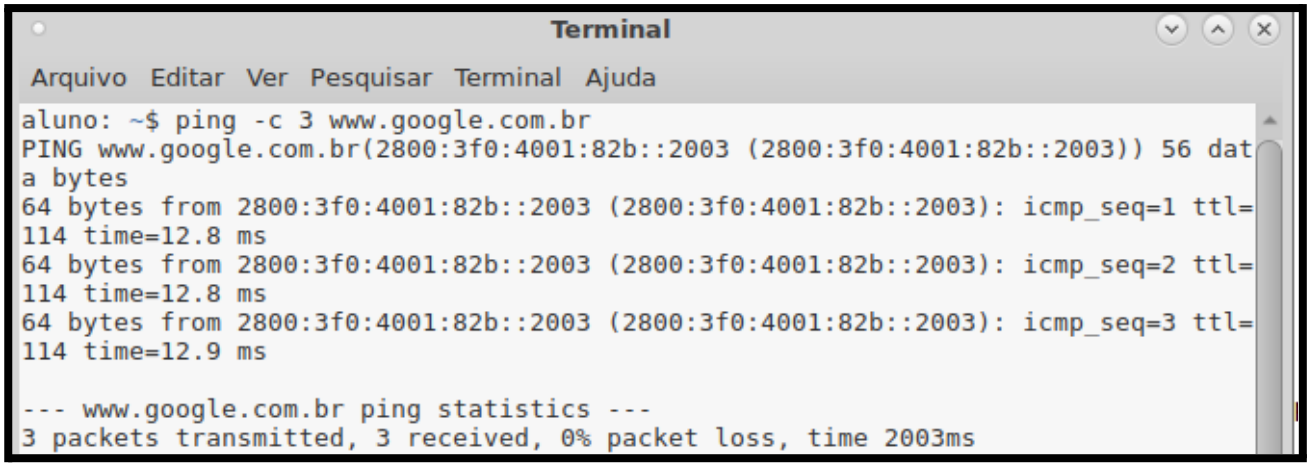
É um analisador de pacotes ideal para nossos laboratórios, pois é estável, tem uma grande base de usuários e é bem documentado incluindo um guia de usuário (http://www.wireshark.org/docs/wsug_html/), páginas de manual (http://www.wireshark.org/docs/man-pages/), e uma seção de FAQ detalhada (http://www.wireshark.org/faq.html), funcionalidade rica que inclui a capacidade de analisar mais que 500 protocolos, e uma interface com o usuário bem projetada. Ele funciona em computadores ligados a uma Ethernet para conectar-se à Internet, bem como protocolos ponto a ponto, tal como PPP.

OBS: Se o wireshark estiver instalado em sua máquina, para chamá-lo a partir de um terminal deve fazer:

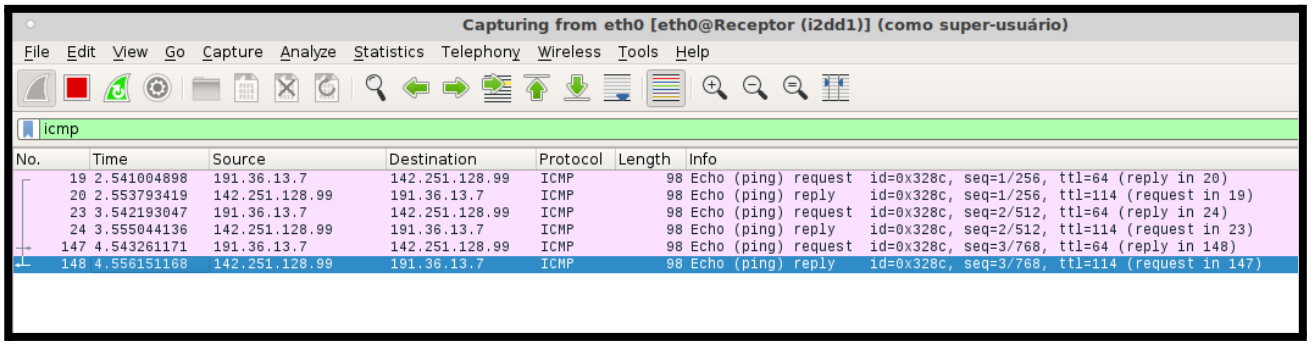
05/09/2022

DESENVOLVIMENTO

1. **Abra o Wireshark em modo captura.**
2. **Abra um terminal e faça um "ping -c 3" para um site conhecido (você pode usar o nome: www.ifsc.edu.br por exemplo).**



3. **Pare a captura de pacotes no Wireshark.**
4. **Aplique um filtro icmp no display. Recorte a tela observada e indique os pacotes ICMP ECHO REQUEST. Anote quem são os endereços IP e MAC que aparecem no pacote IP e Frame Ethernet.**



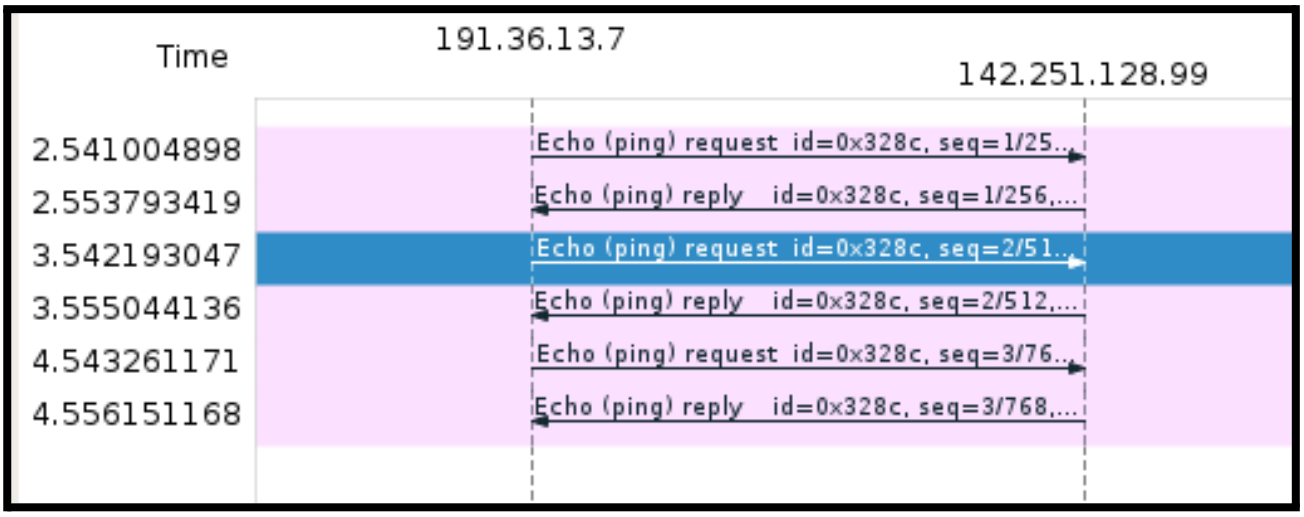
ICMP ECHO REQUEST:

IP de origem: 192.38.13.7 (IP da máquina)
IP de destino: 142.251.128.99 (IP do servidor externo)
MAC de origem: A8:A1:59:08:E0:AF (MAC da máquina)
MAC de destino: 00:AF:1F:8E:EB:78 (MAC do roteador da rede)

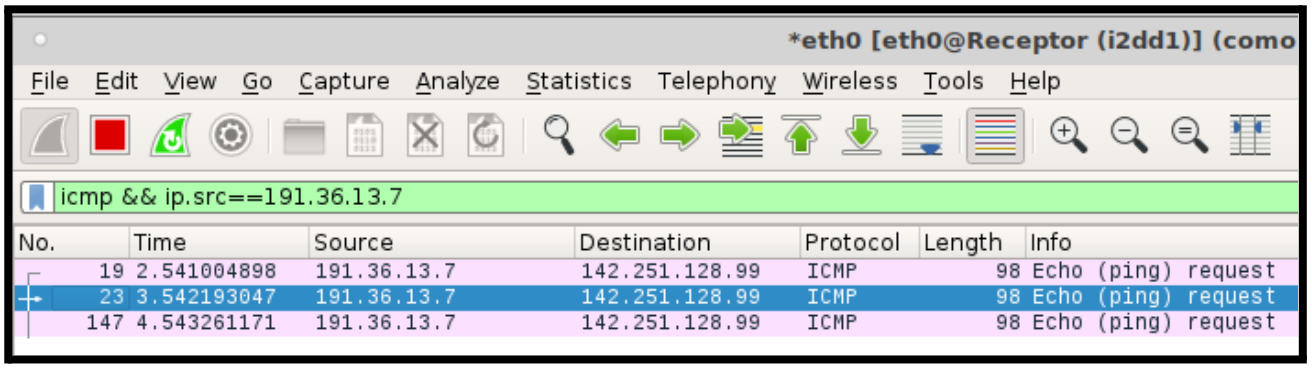
5. **Aplique um comando Flow Graph e mostre a troca de mensagens do ping através de um recorte da tela;**

   Statistics >> Flow Graph >> Abrirá uma nova janela com várias informações >> Aplique o filtro (Flow type:) ICMP Flows na base da janela. Salve esta tela no relatório.

   Feche esta janela.



6. **Crie um filtro para mostrar somente pacotes icmp que saem da sua máquina (ver filtro ip.src). Faça um recorte das telas de criação do filtro (mostrando o filtro).**



Leia atentamente o manual do tcpdump , principalmente os exemplos:

```
man tcpdump
```

```
aluno: ~$ sudo tcpdump host 142.251.128.99
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
10:44:15.799551 IP 191.36.13.7 > gru06s70-in-f3.1e100.net: ICMP echo request, id 16233, seq 47, length 64
10:44:15.812818 IP gru06s70-in-f3.1e100.net > 191.36.13.7: ICMP echo reply, id 16233, seq 47, length 64
10:44:16.800998 IP 191.36.13.7 > gru06s70-in-f3.1e100.net: ICMP echo request, id 16233, seq 48, length 64
10:44:16.820691 IP gru06s70-in-f3.1e100.net > 191.36.13.7: ICMP echo reply, id 16233, seq 48, length 64
10:44:17.802867 IP 191.36.13.7 > gru06s70-in-f3.1e100.net: ICMP echo request, id 16233, seq 49, length 64
10:44:17.815745 IP gru06s70-in-f3.1e100.net > 191.36.13.7: ICMP echo reply, id 16233, seq 49, length 64
10:44:18.803938 IP 191.36.13.7 > gru06s70-in-f3.1e100.net: ICMP echo request, id 16233, seq 50, length 64
10:44:18.816924 IP gru06s70-in-f3.1e100.net > 191.36.13.7: ICMP echo reply, id 16233, seq 50, length 64
10:44:19.805129 IP 191.36.13.7 > gru06s70-in-f3.1e100.net: ICMP echo request, id 16233, seq 51, length 64
10:44:19.817943 IP gru06s70-in-f3.1e100.net > 191.36.13.7: ICMP echo reply, id 16233, seq 51, length 64
^C
10 packets captured
10 packets received by filter
0 packets dropped by kernel
```

1. **Abra um terminal e faça um ping ifsc.edu.br e, com o uso de parâmetros apropriados, faça com que o tcpdump, aberto em outro terminal, armazene os em um arquivo denominado "pacotes_capturadosX.pcap" (um arquivo para cada item abaixo X):**

   Capture todos os pacotes oriundos e destinados à sua máquina.

```
aluno: ~$ sudo tcpdump -w pacotes_capturados1.pcap
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
^C185 packets captured
187 packets received by filter
0 packets dropped by kernel
```

Idem anterior com a *flag -vvv* ativa e, em seguida, a *flag* -n. Qual é a função dessas *flags*?

```
aluno: ~$ sudo tcpdump -vvv -n -w pacotes_capturados2.pcap
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
^C334 packets captured
340 packets received by filter
0 packets dropped by kernel
```

Capture somente os pacotes oriundos de sua máquina.Anote o comando utilizado.

```
aluno: ~$ sudo tcpdump -i eth0 src host 191.36.13.7
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
10:53:49.399989 IP 191.36.13.7 > 191.36.0.94: ICMP echo request, id 16674, seq 472, length 64
10:53:49.401007 IP 191.36.13.7.54028 > 191.36.8.2.domain: 54214+ PTR? 94.0.36.191.in-addr.arpa. (42)
10:53:49.403257 IP 191.36.13.7.58550 > 191.36.8.2.domain: 36130+ PTR? 7.13.36.191.in-addr.arpa. (42)
10:53:49.405039 IP 191.36.13.7.45049 > 191.36.8.2.domain: 63344+ PTR? 2.8.36.191.in-addr.arpa. (41)
10:53:50.423971 IP 191.36.13.7 > 191.36.0.94: ICMP echo request, id 16674, seq 473, length 64
10:53:51.451989 IP 191.36.13.7 > 191.36.0.94: ICMP echo request, id 16674, seq 474, length 64
10:53:52.457647 IP 191.36.13.7.53788 > 191.36.13.54.789: Flags [.], ack 468038499, win 63, options [nop,
10:53:52.457892 IP 191.36.13.7.33828 > 191.36.8.2.domain: 30620+ PTR? 54.13.36.191.in-addr.arpa. (43)
10:53:52.471900 IP 191.36.13.7 > 224.0.0.251: igmp v2 report 224.0.0.251
10:53:52.471914 IP 191.36.13.7 > 191.36.0.94: ICMP echo request, id 16674, seq 475, length 64
10:53:52.472006 IP 191.36.13.7.60936 > 191.36.8.2.domain: 18887+ PTR? 251.0.0.224.in-addr.arpa. (42)
```

Capture somente pacotes destinados à sua máquina. Anote o comando utilizado.

```
aluno: ~$ sudo tcpdump -i eth0 dst host 191.36.13.7
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
10:52:53.112590 IP 191.36.0.94 > 191.36.13.7: ICMP echo reply, id 16674, seq 417, length 64
10:52:53.115491 IP 191.36.8.2.domain > 191.36.13.7.48467: 18241 NXDomain 0/1/0 (102)
10:52:53.117246 IP 191.36.8.2.domain > 191.36.13.7.50526: 29979 NXDomain 0/1/0 (102)
10:52:53.118882 IP 191.36.8.2.domain > 191.36.13.7.36390: 23327 NXDomain 0/1/0 (101)
10:52:53.862071 IP 191.36.13.54.789 > 191.36.13.7.53768: Flags [P.], seq 2070768769:2070768860,
r 3719815348], length 91
10:52:53.862720 IP 191.36.13.54.789 > 191.36.13.7.53768: Flags [.], ack 82, win 64, options [nop
10:52:53.864214 IP 191.36.8.2.domain > 191.36.13.7.56186: 10582 NXDomain 0/1/0 (103)
10:52:54.136529 IP 191.36.0.94 > 191.36.13.7: ICMP echo reply, id 16674, seq 418, length 64
10:52:55.160499 IP 191.36.0.94 > 191.36.13.7: ICMP echo reply, id 16674, seq 419, length 64
10:52:55.467488 IP 191.36.13.54.789 > 191.36.13.7.53826: Flags [P.], seq 3407890173:3407890264,
r 3719816953], length 91
```

Procure um dos arquivos salvos, com o navegador de arquivos de sua máquina, dê um duplo clique sobre o mesmo. Com qual programa foi aberto o arquivo?

```
aluno: ~$ ls -l | grep pacotes
-rw-r--r-- 1 root  root  464088 set  5 10:49 pacotes_capturados1.pcap
-rw-r--r-- 1 root  root  333324 set  5 10:49 pacotes_capturados2.pcap
-rw-r--r-- 1 root  root   27210 set  5 10:54 pacotes_capturados3.pcap
-rw-r--r-- 1 root  root  156293 set  5 10:54 pacotes_capturados4.pcap
```

Os arquivos podem ser abertos pelo wireshark para visualização e análise.