Instituto de Computação - Universidade Estadual de Campinas

MC833A

Programação de Redes de Computadores

Atividade 1 - Ferramentas e Sniffers

Luana Felipe de Barros, RA:201705

1. Considere para esta questão o comando `ifconfig`.
   a. Qual opção deve ser usada para exibir informações sobre todas as interfaces de rede?

      `ifconfig -a`

```
[fedora@netlabs ~]$ ifconfig -a
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.2.15  netmask 255.255.255.0  broadcast 10.0.2.255
        inet6 fe80::3b0c:9686:1bc0:d7df  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:31:27:41  txqueuelen 1000  (Ethernet)
        RX packets 194  bytes 55905 (54.5 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 198  bytes 20531 (20.0 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 24  bytes 1440 (1.4 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 24  bytes 1440 (1.4 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

   b. O que deve ser feito para exibir somente informações de uma interface específica?

      `ifconfig <nome_da_interface>`

      No exemplo abaixo, exibimos informações da interface de rede enp0s3.

```
[fedora@netlabs ~]$ ifconfig enp0s3
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.2.15  netmask 255.255.255.0  broadcast 10.0.2.255
        inet6 fe80::3b0c:9686:1bc0:d7df  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:31:27:41  txqueuelen 1000  (Ethernet)
        RX packets 217  bytes 57795 (56.4 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 221  bytes 22421 (21.8 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

2. Através da execução do comando `nslookup` seguido dos parâmetros adequados, responda às seguintes questões:

a. Quais são os endereços IP do host www.unicamp.br?

   Para encontrar todos os endereços IP associados a este domínio, utilizamos:

   `nslookup www.unicamp.br`

   E obtemos o seguinte resultado:

```
(base) luanabarros@guitarra:~$ nslookup www.unicamp.br
Server:         143.106.219.8
Address:        143.106.219.8#53

Non-authoritative answer:
www.unicamp.br  canonical name = 143-106-143-186.nuvem.unicamp.br.
Name:   143-106-143-186.nuvem.unicamp.br
Address: 143.106.143.186
```

b. Há alguma vantagem em haver mais de um endereço IP?

   Sim, pois uma corporação pode ter servidores distribuídos ao redor do mundo e utilizar apenas um domínio na Internet. Desta forma, a resolução dos nomes de domínio associados ao endereço IP correto é feita por servidores DNS. Além disso, a distribuição de IPs ajuda a diminuir o tráfego na rede para acessar um servidor específico.

3. Através da execução do comando `traceroute` seguido dos parâmetros adequados, responda à seguinte questão:
   a. Quantos roteadores estão entre a sua estação e o host www.amazon.com? Pelos nomes dos roteadores, quantos deles estão localizados no Brasil?

      Pela figura, é possível ver que o traceroute fez o caminho até um dos servidores da amazon por 17 roteadores. Adicionei a opção -I no comando, pois sem ela muitos roteadores não respondiam - ou por timeout, ou por segurança, ou por alguma configuração pra não responder pacotes udp. Além disso, é possível notar que o sexto roteador na rota é o último que consigo ver que está no Brasil. Logo, os anteriores também estão. No total, assumo que existem 6 roteadores no Brasil nesse caminho.

```
[fedora@netlabs ~]$ sudo traceroute  www.amazon.com -I
traceroute to www.amazon.com (13.227.106.126), 30 hops max, 60 byte packets
 1  _gateway (10.0.2.2)  0.280 ms  0.238 ms  0.213 ms
 2  10.0.0.1 (10.0.0.1)  3.471 ms  6.595 ms  6.614 ms
 3  bras-fibra1.netnz.com.br (201.130.20.3)  9.325 ms  9.360 ms  9.356 ms
 4  100.127.29.5 (100.127.29.5)  10.025 ms  10.095 ms  10.951 ms
 5  backbone.netnz.com.br (201.130.20.1)  13.070 ms  13.188 ms  13.193 ms
 6  as16509.saopaulo.sp.ix.br (187.16.217.163)  15.534 ms  9.973 ms  10.036 ms
 7  52.93.146.197 (52.93.146.197)  10.519 ms  12.928 ms  12.988 ms
 8  54.240.244.165 (54.240.244.165)  9.452 ms  12.402 ms  12.855 ms
 9  * * *
10  52.93.44.52 (52.93.44.52)  20.175 ms  20.148 ms  20.322 ms
11  150.222.70.47 (150.222.70.47)  18.413 ms  18.422 ms  10.246 ms
12  * * *
13  * * *
14  * * *
15  * * *
16  * * *
17  server-13-227-106-126.gru50.r.cloudfront.net (13.227.106.126)  16.861 ms  10.732 ms  10.727 ms
```

Para testar, utilizei uma máquina diferente e realizei o mesmo comando, mostrada abaixo. Fiquei surpresa pela rapidez da execução, e principalmente pelo domínio www.amazon.com nesta máquina estar associado ao IP: 72.246.131.124. Na figura anterior, dizia que este mesmo domínio está mapeado ao endereço IP: 13.227.106.126. Logo, o servidor DNS associado a cada máquina é diferente, e as rotas até o endereço IP final é diferente. Neste caso, temos 5 roteadores entre minha máquina e o roteador final. Dentre os quais, é possível ver que 3 estão no Brasil.

```
(base) luanabarros@guitarra:~$ sudo traceroute www.amazon.com -I
traceroute to www.amazon.com (72.246.131.124), 30 hops max, 60 byte packets
 1  nics-gw.nics.unicamp.br (143.106.219.129)  0.477 ms  0.473 ms  0.442 ms
 2  area4-gw.unicamp.br (143.106.1.193)  0.774 ms  0.834 ms  0.980 ms
 3  ptp-nct-ndg.unicamp.br (143.106.199.17)  0.749 ms  0.788 ms  0.771 ms
 4  * * *
 5  a72-246-131-124.deploy.static.akamaitechnologies.com (72.246.131.124)  3.235 ms  3.253 ms  3.251 ms
```

4. Através da execução do comando `telnet`, seguido dos parâmetros adequados, responda às seguintes questões:
    a. É possível conectar-se com este comando em um servidor HTTP? Se sim, como deve-se executar o comando para conectar-se no host www.amazon.com na porta padrão do HTTP?

    Sim, o comando telnet é usado na comunicação de portas entre hosts, enviando mensagens definidas pelo protocolo Telnet. É possível se conectar a um servidor HTTP na porta 80 (padrão) da seguinte maneira:

    ```
    telnet www.amazon.com 80
    ```

    Desta forma, vemos na figura que nos conectamos ao servidor nesta porta. Agora, o servidor espera uma requisição. Tentei fazer um GET, mas não

funcionou.

```
(base) luanabarros@guitarra:~$ telnet www.amazon.com 80
Trying 72.246.131.124...
Connected to e15316.e22.akamaiedge.net.
Escape character is '^]'.
GET index.html HTTP/1.0
Host: www.amazon.com
HTTP/1.0 408 Request Time-out
Server: AkamaiGHost
Mime-Version: 1.0
Date: Mon, 12 Oct 2020 18:27:03 GMT
Content-Type: text/html
Content-Length: 218
Expires: Mon, 12 Oct 2020 18:27:03 GMT

<HTML><HEAD>
<TITLE>Request Timeout</TITLE>
</HEAD><BODY>
<H1>Request Timeout</H1>
The server timed out while waiting for the browser's request.<P>
Reference&#32;&#35;2&#46;dcbc1002&#46;1602527223&#46;0
</BODY></HTML>
Connection closed by foreign host.
```

b. Caso não haja um servidor escutando na porta passada pelo comando telnet, o que ocorre? Justifique.

```
(base) luanabarros@guitarra:~$ telnet google.com 10
Trying 172.217.173.78...
Trying 2800:3f0:4001:819::200e...
telnet: Unable to connect to remote host: Network is unreachable
(base) luanabarros@guitarra:~$
```

c. A qual a camada da rede o telnet pertence?

A aplicação telnet implementa o protocolo Telnet, que faz parte da camada de aplicação, visto que conecta com outras aplicações através de portas.

5. Acesse o site da DAC (https://www.dac.unicamp.br/) e, em paralelo em um terminal, verifique a saída do comando `netstat`. Quais são as informações fornecidas a respeito da conexão ao site da DAC?

```
[fedora@netlabs ~]$ nslookup www.dac.unicamp.br
Server:         10.0.0.1
Address:        10.0.0.1#53

Name:   www.dac.unicamp.br
Address: 143.106.227.165
www.dac.unicamp.br      canonical name = 143-106-227-165.nuvem.unicamp.br.

[fedora@netlabs ~]$ sudo netstat | grep 143.106.227.165
tcp        0      0 netlabs:60258           143-106-227-165.n:https ESTABLISHED
[fedora@netlabs ~]$
```

Podemos ver que o comando nos diz que foi utilizada uma conexão TCP para se conectar ao servidor da DAC, através de um socket. Neste socket, o endereço e porta do emissor são netlabs:60258 e do emissor o endereço é 143.106.227.165, e além disso podemos ver que a conexão está estável.

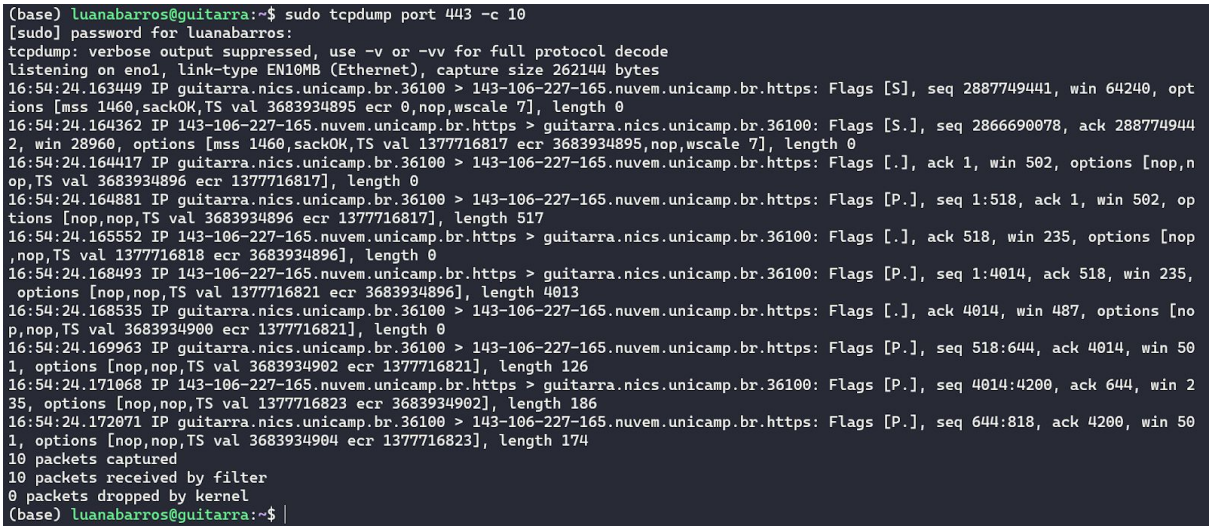
6. Considere a ferramenta `TCPDUMP`, e responda às seguintes questões:

   a. Utilizando o TCPDUMP corretamente com os filtros é possível somente capturar o tráfego HTTPS? Se sim, execute o comando junto com os filtros e anexe uma figura que comprove sua resposta no relatório. Se sua resposta foi não, então justifique-a.

   Sim, isso é possível através do comando:

   ```
   sudo tcpdump port 443
   ```

   No print abaixo, coloquei um limite de 10 pacotes para que fosse possível capturar a tela. Utilizamos a porta 443, responsável pelas aplicações HTTPS.



   b. Utilizando o comando TCPDUMP seguido dos parâmetros corretos imprima somente os pacotes superiores a 64 bits. Indique qual foi a sequência de comandos utilizada.

   ```
   sudo tcpdump greater 8 -c 10
   ```

   A opção greater é uma condição de filtro dos pacotes maiores que um número em bytes. No caso, 64 bits são 8 bytes. E novamente, limitei a quantidade de pacotes capturados por 10.

```
(base) luanabarros@guitarra:~$ sudo tcpdump greater 8 -c 10
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eno1, link-type EN10MB (Ethernet), capture size 262144 bytes
18:21:55.586367 IP guitarra.nics.unicamp.br.5678 > vpn-143-106-203-112.home.unicamp.br.50325: Flags [P.], seq 1195327235:1195327423,
ack 1740149261, win 501, length 188
18:21:55.587055 IP guitarra.nics.unicamp.br.51164 > tomtom.nics.unicamp.br.domain: 62678+ PTR? 112.203.106.143.in-addr.arpa. (46)
18:21:55.588489 IP tomtom.nics.unicamp.br.domain > guitarra.nics.unicamp.br.51164: 62678 1/2/0 PTR vpn-143-106-203-112.home.unicamp.b
r. (131)
18:21:55.588782 IP guitarra.nics.unicamp.br.37152 > tomtom.nics.unicamp.br.domain: 9109+ PTR? 8.219.106.143.in-addr.arpa. (44)
18:21:55.588805 IP guitarra.nics.unicamp.br.5678 > vpn-143-106-203-112.home.unicamp.br.50325: Flags [P.], seq 188:392, ack 1, win 501
, length 204
18:21:55.589868 IP tomtom.nics.unicamp.br.domain > guitarra.nics.unicamp.br.37152: 9109* 1/3/2 PTR tomtom.nics.unicamp.br. (164)
18:21:55.590124 IP guitarra.nics.unicamp.br.5678 > vpn-143-106-203-112.home.unicamp.br.50325: Flags [P.], seq 392:828, ack 1, win 501
, length 436
18:21:55.590174 IP guitarra.nics.unicamp.br.5678 > vpn-143-106-203-112.home.unicamp.br.50325: Flags [P.], seq 828:1144, ack 1, win 50
1, length 316
18:21:55.590224 IP guitarra.nics.unicamp.br.5678 > vpn-143-106-203-112.home.unicamp.br.50325: Flags [P.], seq 1144:1332, ack 1, win 5
01, length 188
18:21:55.590244 IP guitarra.nics.unicamp.br.5678 > vpn-143-106-203-112.home.unicamp.br.50325: Flags [P.], seq 1332:1520, ack 1, win 5
01, length 188
10 packets captured
11 packets received by filter
0 packets dropped by kernel
```

c. Utilizando o TCPDUMP seguido de filtros, imprima somente os resultados que tiverem a flag ‘ACK’. Insira o comando seguido dos filtros e uma figura no seu relatório para comprovar o sucesso.

```
sudo tcpdump 'tcp[tcpflags] & (tcp-ack) != 0' -c 5
```

```
(base) luanabarros@guitarra:~$ sudo tcpdump 'tcp[tcpflags] & (tcp-ack) != 0' -c 5
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eno1, link-type EN10MB (Ethernet), capture size 262144 bytes
18:58:04.826585 IP guitarra.nics.unicamp.br.5678 > vpn-143-106-203-112.home.unicamp.br.50325: Flags [P.], seq 1196560695:1196560883,
ack 1740158185, win 501, length 188
18:58:04.828828 IP guitarra.nics.unicamp.br.5678 > vpn-143-106-203-112.home.unicamp.br.50325: Flags [P.], seq 188:392, ack 1, win 501
, length 204
18:58:04.829097 IP guitarra.nics.unicamp.br.5678 > vpn-143-106-203-112.home.unicamp.br.50325: Flags [P.], seq 392:580, ack 1, win 501
, length 188
18:58:04.829366 IP guitarra.nics.unicamp.br.5678 > vpn-143-106-203-112.home.unicamp.br.50325: Flags [P.], seq 580:768, ack 1, win 501
, length 188
18:58:04.829548 IP guitarra.nics.unicamp.br.5678 > vpn-143-106-203-112.home.unicamp.br.50325: Flags [P.], seq 768:956, ack 1, win 501
, length 188
5 packets captured
6 packets received by filter
0 packets dropped by kernel
```

7. Considere a ferramenta Wireshark para responder às questões a seguir:
   a. Comparado às demais ferramentas apresentadas na aula de MC833 descreva quais são principais diferenças e vantagens de usar o Wireshark? Escolha pelo menos uma ferramenta/sniffer e elabore uma tabela comparativa para responder a questão.

      Utilizei para comparação a ferramenta tcpdump, visto que elas tem muitas funcionalidades em comum.

      Tabela de Comparação entre os Sniffers - [1]

Tcpdump	Wireshark
Consegue mostrar os pacotes no formato cru, de modo que os usuários consigam analisar de forma clara e entender melhor os conceitos de TCP.	Tem uma interface amigável para analisar os pacotes e consegue detectar ataques em Firewalls, e brechas na rede
Consegue capturar os pacotes baseados em filtros: strings, números	Este por sua vez, captura todos os pacotes e depois aplica o filtro.

ou até mesmo um fragmento de programa em C, com as opções -dd ou -ddd.	
Consegue capturar e armazenar em um único arquivo Não possui nenhuma funcionalidade estatística	Pode capturar os pacotes em vários arquivos utilizando o comando –b Consegue coletar e mostrar diferentes tipos de estatísticas sobre os pacotes que pode ser mostrada na tela em tempo real
É eficiente e com maior poder de processamento do que o wireshark	Utiliza muito mais memória do que o tcpdump, mas tem maior velocidade na captura dos pacotes

b. Em uma rede com vários processos acontecendo ao mesmo tempo é possível gerenciar de forma isolada um único processo específico na rede utilizando ferramentas/sniffers apresentados nesta disciplina? Se sim, quais ferramentas e/ou sniffers você usaria? Justifique sua resposta.

   Sim. Os processos são aplicações de rede que comunicam-se entre si utilizando as camadas inferiores. A camada de transporte é responsável pelo serviço de multiplexação e demultiplexação, e faz isso utilizando identificadores de número de protocolo para identificar o protocolo de transporte correto, e estes utilizam número de porta para identificar cada processo.[2] Então, por exemplo, se queremos gerenciar o processo servidor WEB, devemos primeiramente descobrir o protocolo da camada de transporte e a porta utilizada. Isso é possível utilizando a ferramenta netstat. Tendo esse conhecimento, podemos utilizar o sniffer Wireshark para filtrar por esses dois parâmetros. Então, sim, é possível, mas com a combinação destas duas ferramentas.

Referências

[1] Goyal, P. and Anurag Goyal. “Comparative study of two most popular packet sniffing tools-Tcpdump and Wireshark.” 2017 9th International Conference on Computational Intelligence and Communication Networks (CICN) (2017): 77-81.

[2] https://www.cs.ait.ac.th/~on/O/oreilly/tcpip/tcpip/ch02_07.htm