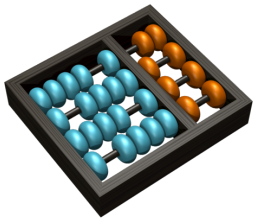


Atividade 1: Ferramentas e Sniffers

Aluno: Guilherme Luis Domingues

RA: 155619

Instituto de Computação

Universidade Estadual de Campinas

Campinas, 29 de Setembro de 2020.

Sumário

1 Comando ifconfig

1.1 Listar todas

Para listar todas as interfaces de rede deve-se utilizar o comando ***ifconfig -a***. A saída produzida por este comando é

```
[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::a481:e94a:c133:db4b  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:97:c7:94  txqueuelen 1000  (Ethernet)
        RX packets 21  bytes 3018 (2.9 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 41  bytes 5242 (5.1 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 4  bytes 240 (240.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 4  bytes 240 (240.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

1.2 Mostrar específica

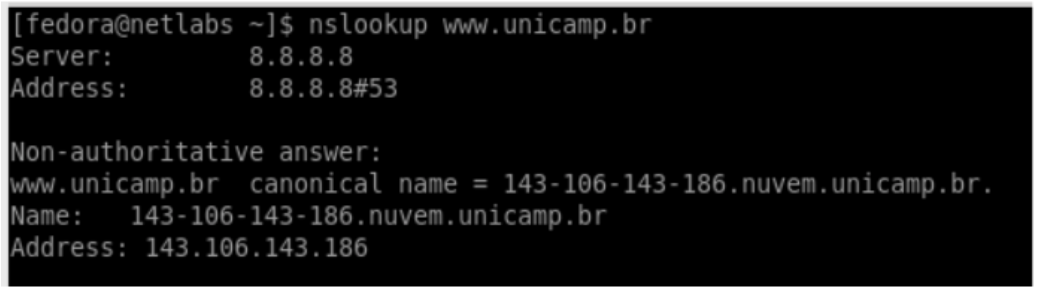
Para mostra apenas uma interface, é necessário utilizar o comando ***ifconfig <nome-interface>***. No caso abaixo, apenas a interface **inp0s3** foi solicitada.

```
[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::a481:e94a:c133:db4b  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:97:c7:94  txqueuelen 1000  (Ethernet)
        RX packets 36  bytes 4278 (4.1 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 56  bytes 6502 (6.3 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

2 Comando nslookup

endereços IP

Ao executar o comando ***nslookup www.unicamp.br*** era esperado retornar os endereços de IP utilizados pelo site www.unicamp.br. Porém, obtivemos apenas um endereço de IP, o 143.106.143.186. A figura abaixo mostra o retorno obtido.



2.1 Vantagens em multiplos endereços

Com a redundância de endereços, as conexões acabam ficando mais fluidas, visto que existem diversos servidores aptos a responderem as requisições feitas pelo usuário. E também, caso algum dos servidores falhe, essa falha acaba sendo transparente para o usuário.

3 Comando traceroute

3.1 Números de roteador

Ao executar o comando traceroute, espera-se encontrar o número de nós (ou roteadores) que estão fisicamente entre a fonte da requisição e o destinatário. Neste caso, o destinatário foi o endereço www.amazon.com.

Sendo o roteador da minha casa - endereço 192.168.15.1 a saída para a rede externa, temos então 6 roteadores entre minha estação e o servidor da Amazon. Levando em conta os endereços e, principalmente o tempo de resposta, ao todo são 3 servidores localizados o Brasil, sendo eles os roteadores referentes à minha provedora de internet.

A figura abaixo mostra a resposta obtida

4 Comando telnet

4.1 Conectar a um servidor

Sim, é possível.

De maneira geral, devemos utilizar o comando ***telnet <endereço> <porta>***.

Para se conectar no servidor da Amazon, utilizamos o comando ***telnet www.amazon.com 80***. Sendo 80 a porta padrão de servidores HTTP.

A figura ilustra o resultado obtido. Além do status de "connected", o servidor ainda espera por comandos

4.2 Servidor não escutando

Caso tente conectar à uma porta que não está sendo escutada por aquele servidor, recebemos um erro de time-out. Isto significa que o telnet tentou se conectar àquela porta porém não obteve nenhuma resposta.

4.3 Camada do telnet

O protocolo Telnet está na camada de Aplicação e utiliza-se do protocolo TCP para realizar o transporte entre o destino e a fonte.

5 Comando netstat

O comando ***netstat*** serve para exibir as conexões de rede para o protocolo TPC, tabelas de roteamento, além de interfaces de rede e estatísticas sobre a rede.
Abrindo o site da Dac no Mozila Firefox e executando o comando ***netstat*** em paralelo, obtivemos a saída apresentada na figura abaixo.

Na imagem é possível verificar que a conexão com o endereço 143-106-227-165.n foi estabelecida com sucesso. Além disso, mostra que a conexão foi do tipo HTTPS, o endereço local, bem como a porta que foi utilizada para a comunicação. Além disso, mostra que o protocolo utilizado para a transmissão foi o TPC.

```
[fedora@netlabs ~]$ netstat
Active Internet connections (w/o servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 netlabs:52616           104.18.21.226:http      ESTABLISHED
tcp        0      0 netlabs:57112           eze03s05-in-f227.:https ESTABLISHED
tcp        0      0 netlabs:46048           gru06s26-in-f4.1e:https TIME_WAIT
tcp        0      0 netlabs:32996           server-13-226-45-:https ESTABLISHED
tcp        0      0 netlabs:38212           192.16.58.8:http        ESTABLISHED
tcp        0      0 netlabs:59182           a23-54-22-16.deplo:http ESTABLISHED
tcp        0      0 netlabs:44202           proxy-iad01.fedor:https ESTABLISHED
tcp        0      0 netlabs:37790           gru10s02-in-f163.1:http ESTABLISHED
tcp        0      0 netlabs:52618           104.18.21.226:http      ESTABLISHED
tcp        0      0 netlabs:41554           104.16.123.175:https    ESTABLISHED
tcp        0      0 netlabs:48252           gru10s10-in-f14.1:https ESTABLISHED
tcp        0      0 netlabs:35586           gru14s20-in-f10.1:https ESTABLISHED
tcp        0      0 netlabs:48350           143-106-227-165.n:https ESTABLISHED
tcp        0      0 netlabs:37792           gru10s02-in-f163.1:http ESTABLISHED
tcp        0      0 netlabs:35020           gru14s20-in-f3.1e:https TIME_WAIT
tcp        0      0 netlabs:37794           gru10s02-in-f163.1:http ESTABLISHED
tcp        0      0 netlabs:35014           gru14s20-in-f3.1e:https ESTABLISHED
tcp        0      0 netlabs:56250           gru06s31-in-f10.1:https TIME_WAIT
tcp        0      0 netlabs:32998           server-13-226-45-:https TIME_WAIT
tcp        0      0 netlabs:47988           gru14s19-in-f8.1e:https ESTABLISHED
tcp        0      0 netlabs:37796           gru10s02-in-f163.1:http ESTABLISHED
tcp        0      0 netlabs:37786           gru10s02-in-f163.1:http ESTABLISHED
tcp        0      0 netlabs:46054           gru06s26-in-f4.1e:https TIME_WAIT
tcp        0      0 netlabs:35296           gru14s06-in-f3.1e1:http ESTABLISHED
```

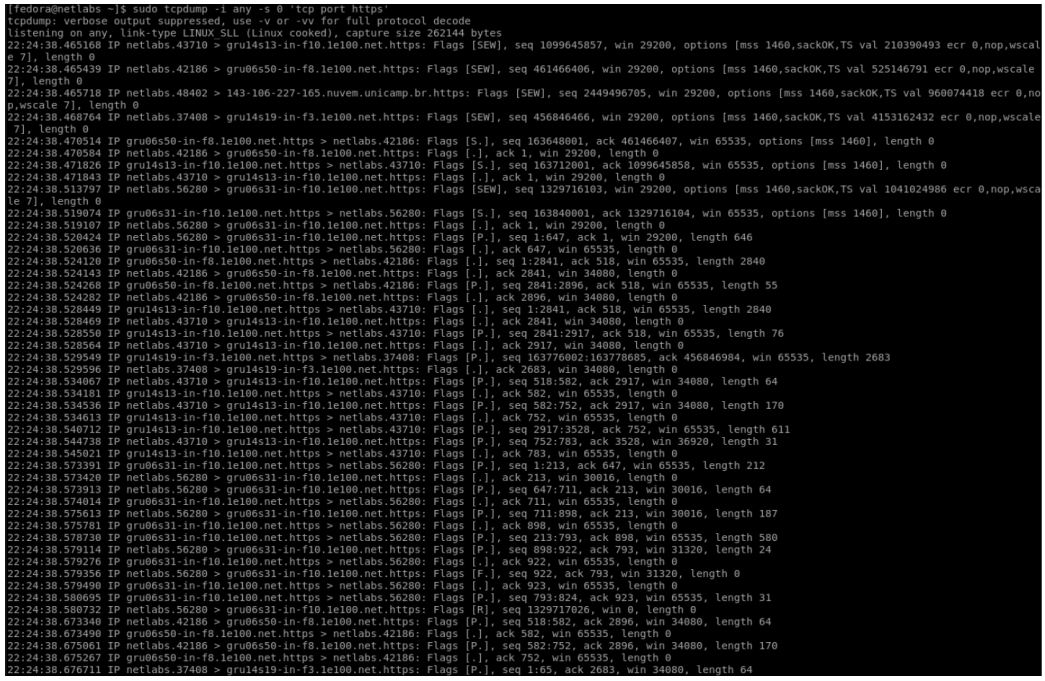
6 Ferramenta TCPDUMP

6.1 Filtro HTTPS

Sim, é possível.

O comando para filtrar apenas o tráfego do tipo https é: ***sudo tcpdum -i <inteface> -s <0-65535> <filtro>***. Para filtrar apenas as requisições do tipo HTTP, utilizamos o seguinte comando: ***sudo tcpdum -i any -s 0 'tcp port https'***. Nele, pegamos de todas as interfaces da máquina, além de usar o Snapshot Lenght como 0 para pegar o pacote por completo e, por fim, filtrar apenas chamadas HTTPS feitas com o TCP.

A imagem abaixo ilustra a resposta



6.2 Filtro por tamanho de pacote

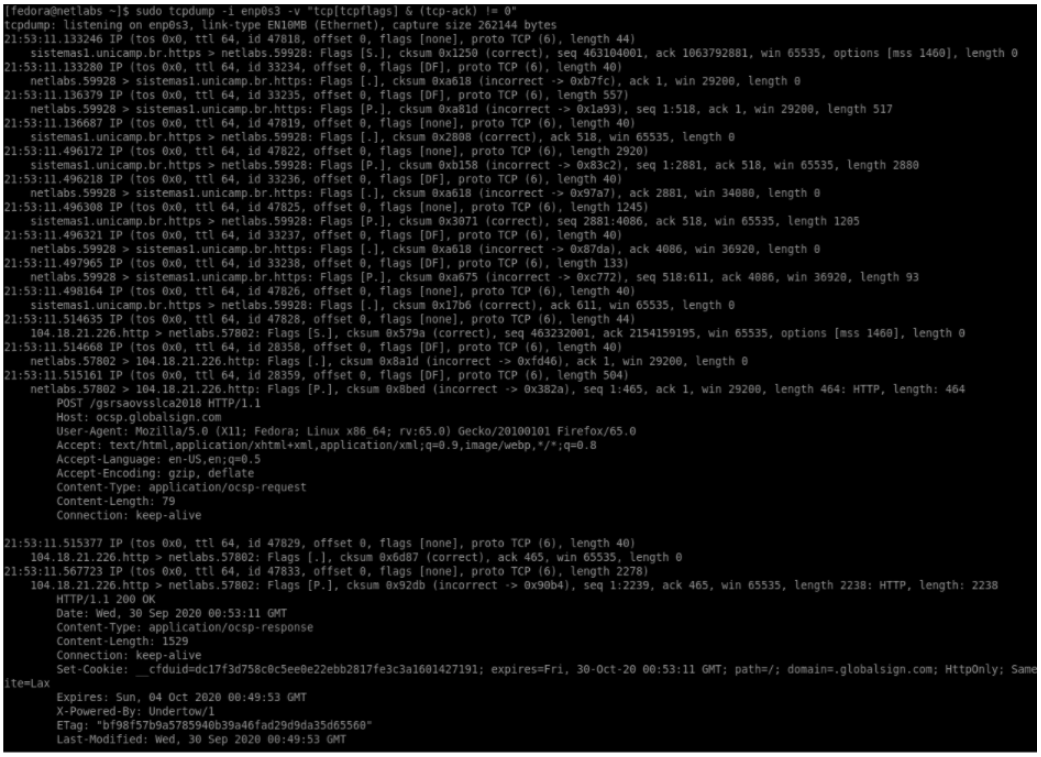
Para filtrar os pacotes com tamanho maior que 64 bits é necessário utilizar o seguinte comando: ***sudo tcpdump -n -i <interface> greater 64*** onde interface é obtido através do ifconfig.

No nosso caso, utilizamos ***sudo tcpdump -n -i enp0s3 greater 64***

A imagem a seguir mostra o retorno

6.3 Filtro por flag ACK

Para filtrar os resultado que tiveram a flag ACK, precisamos utilizar o comando ***sudo tcpdump -v "tcp[tcpflags] (tcp-ack) != 0"***. A flag -v é utilizada para mostrar o resultado de uma forma mais verbosa. Além disso, filtramos o trafego que utilizou o TCP, e que o TCP Acknowledgement recebido foi diferente de 0.



7 Ferramenta Wireshark

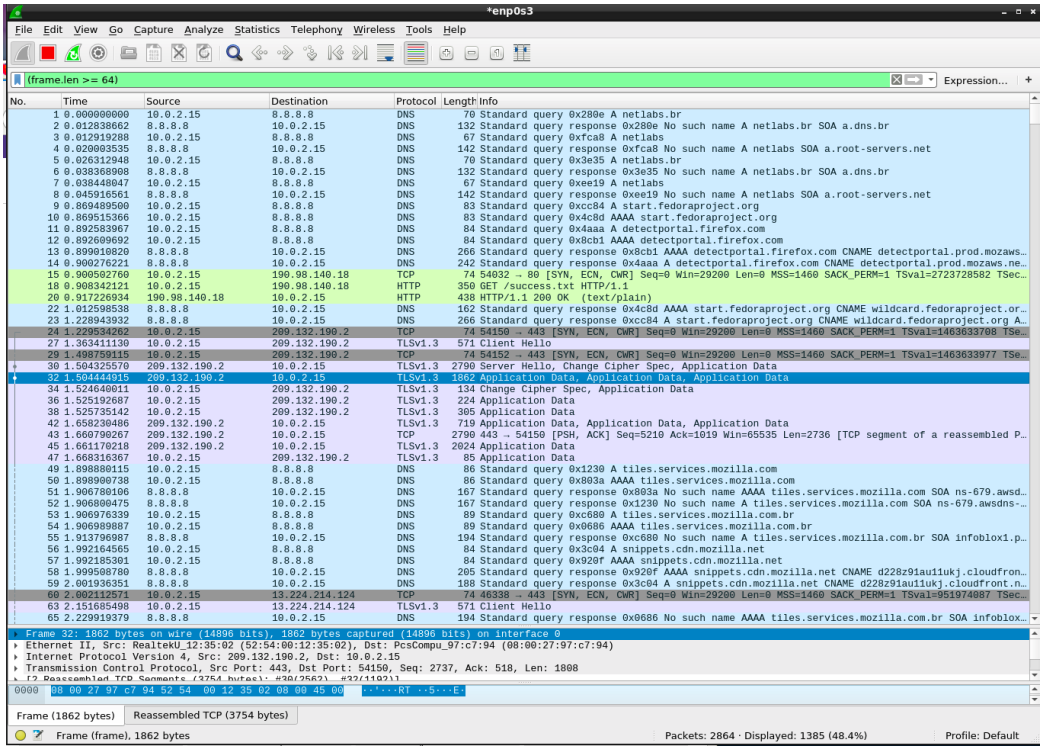
7.1 Wireshark vs ferramentas

Ao utilizar o Wireshark a primeira impressão é que é bem mais intuitiva do que os comandos do terminal. Com uma interface amigável, é possível ir navegando entre as janelas e explorando todos os pontos, filtros e comandos que a ferramenta permite. Já com os comandos e sniffers utilizados via terminal, esta exploração fica mais complicada.
Como comparativo, utilizamos o filtro dos pacotes cujo tamanho é maior que 64 bits.

Nesta imagem, encontramos a saída obtida através do comando tcp-

dump.

Nesta figura temos a interface do Wireshark, a qual mostra um campo dedicado aos filtros. Além de escrever, existe a possibilidade de aplicar algum outro tipo de filtro clicando com o botão direito em cima do campo de interesse.



7.2 Monitoramento de processos

Sim, é possível.

Para monitorar um processo específico, poderíamos começar sabendo qual é seu IP destino, através do nslookup. Uma vez conhecendo isso, podemos então utilizar o Wireshark, por ser uma ferramenta intuitiva e de fácil compreensão.

Nela é possível filtrar os endereços de IP destino ou fonte da requisição desejada. Sendo assim, essa é uma ótima ferramenta para acompanhar de perto um processo.