



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	2
1.1	Listar todas	2
1.2	Mostrar específica	2
2	Comando nslookup	3
2.1	Vantagens em multiplos endereços	3
3	Comando traceroute	3
3.1	Números de roteador	3
4	Comando telnet	4
4.1	Conectar a um servidor	4
4.2	Servidor não escutando	5
4.3	Camada do telnet	5
5	Comando netstat	5
6	Ferramenta TCPDUMP	6
6.1	Filtro HTTPS	6
6.2	Filtro por tamanho de pacote	7
6.3	Filtro por flag ACK	8
7	Ferramenta Wireshark	9
7.1	Wireshark vs ferramentas	9
7.2	Monitoramento de processos	11

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.

```
[fedora@netlabs ~]$ nslookup www.unicamp.br
Server:      8.8.8.8
Address:     8.8.8.8#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
```

2.1 Vantagens em múltiplos 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

```
[fedora@netlabs ~]$ traceroute www.amazon.com
traceroute to www.amazon.com (13.33.130.223), 30 hops max, 60 byte packets
 1  gateway (10.0.2.2)  0.192 ms  0.154 ms  0.138 ms
 2  192.168.15.1 (192.168.15.1)  1.157 ms  1.522 ms  1.646 ms
 3  * * *
 4  152-255-155-172.user.vivozap.com.br (152.255.155.172)  6.698 ms 152-255-155-174.user.vivozap.com.br (152.255.155.174)  6.697 ms 152-255-155-178.user.vivozap.com.br (152.255.155.178)  6.914 ms
 5  152-255-178-56.user.vivozap.com.br (152.255.178.56)  7.641 ms 152-255-178-54.user.vivozap.com.br (152.255.178.54)  7.035 ms 152-255-178-56.user.vivozap.com.br (152.255.178.56)  7.209 ms
 6  187-100-83-6.dsl.telesp.net.br (187.100.83.6)  9.870 ms 9.293 ms 152-255-176-177.user.vivozap.com.br (152.255.176.177)  9.663 ms
 7  ge-3-0-2-3606-graliml4.net.telefonicaglobalsolutions.com (213.140.50.198)  9.262 ms 9.961 ms 84.16.8.22 (84.16.8.22)  20.520 ms
 8  213.140.50.193 (213.140.50.193)  22.621 ms 22.583 ms *
 9  150.222.12.5 (150.222.12.5)  23.573 ms amazon-be50-grtriotw2.net.telefonicaglobalsolutions.com (213.140.50.195)  23.222 ms 150.222.12.5 (150.222.12.5)  23.726 ms
10  52.93.67.195 (52.93.67.195)  22.388 ms 150.222.12.29 (150.222.12.29)  28.653 ms 150.222.12.31 (150.222.12.31)  20.069 ms
11  * * 52.93.67.201 (52.93.67.201)  19.364 ms
12  * * *
13  * * *
14  * * *
15  * * *
16  server-13-33-130-223.gig51.r.cloudfront.net (13.33.130.223)  17.604 ms 17.952 ms *
[fedora@netlabs ~]$
```

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

```
[fedora@netlabs ~]$ telnet www.amazon.com 80
Trying 23.76.255.112...
Connected to www.amazon.com.
Escape character is '^]'.

```

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.

```
[fedora@netlabs ~]$ telnet www.amazon.com 99
Trying 13.227.106.126...
telnet: connect to address 13.227.106.126: Connection timed out
```

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 TCP, tabelas de roteamento, além de interfaces de rede e estatísticas sobre a rede.

Abrindo o site da Dac no Mozilla 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 TCP.

```
[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 tcpdump -i <interface> -s <0-65535> <filtro>***. Para filtrar apenas as requisições do tipo HTTP, utilizamos o seguinte comando: ***sudo tcpdump -i any -s 0 'tcp port https'***. Nele, pegamos de todas as interfaces da máquina, além de usar o Snapshot Length 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

```

[ifedora@netlabs ~]$ sudo tcpdump -i any -s 0 'tcp port https'
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on any, link-type LINUX_SLL (Linux cooked), capture size 262144 bytes
22:24:38.485160 IP netlabs.43710 > gru14513-in-f10.1e100.net.https: Flags [SEW], seq 1099645857, win 29200, options [mss 1460,sackOK,TS val 210390493 ecr 0,nop,wscale 7], length 0
22:24:38.465439 IP netlabs.42186 > gru06550-in-f8.1e100.net.https: Flags [SEW], seq 461466406, win 29200, options [mss 1460,sackOK,TS val 525146791 ecr 0,nop,wscale 7], length 0
22:24:38.405718 IP netlabs.48402 > 143-106-227-165.nuvem.unicamp.br.https: Flags [SEW], seq 2449496705, win 29200, options [mss 1460,sackOK,TS val 960074418 ecr 0,nop,wscale 7], length 0
22:24:38.468764 IP netlabs.37408 > gru14519-in-f3.1e100.net.https: Flags [SEW], seq 456846466, win 29200, options [mss 1460,sackOK,TS val 4153162432 ecr 0,nop,wscale 7], length 0
22:24:38.470514 IP gru06550-in-f8.1e100.net.https > netlabs.42186: Flags [S.], seq 163648001, ack 461466407, win 65535, options [mss 1460], length 0
22:24:38.470584 IP netlabs.42186 > gru06550-in-f8.1e100.net.https: Flags [.], ack 1, win 29200, length 0
22:24:38.471826 IP gru14513-in-f10.1e100.net.https > netlabs.43710: Flags [S.], seq 163712001, ack 1099645858, win 65535, options [mss 1460], length 0
22:24:38.471843 IP netlabs.43710 > gru14513-in-f10.1e100.net.https: Flags [.], ack 1, win 29200, length 0
22:24:38.513797 IP netlabs.56280 > gru06531-in-f10.1e100.net.https: Flags [SEW], seq 1329710103, win 29200, options [mss 1460,sackOK,TS val 1041024986 ecr 0,nop,wscale 7], length 0
22:24:38.519074 IP gru06531-in-f10.1e100.net.https > netlabs.56280: Flags [S.], seq 163840001, ack 1329710104, win 65535, options [mss 1460], length 0
22:24:38.519107 IP netlabs.56280 > gru06531-in-f10.1e100.net.https: Flags [.], ack 1, win 29200, length 0
22:24:38.520424 IP netlabs.56280 > gru06531-in-f10.1e100.net.https: Flags [P.], seq 1:647, ack 1, win 29200, length 646
22:24:38.520636 IP gru06531-in-f10.1e100.net.https > netlabs.56280: Flags [.], ack 647, win 65535, length 0
22:24:38.524120 IP gru06550-in-f8.1e100.net.https > netlabs.42186: Flags [.], seq 1:2841, ack 518, win 65535, length 2840
22:24:38.524143 IP netlabs.42186 > gru06550-in-f8.1e100.net.https: Flags [.], ack 2841, win 34080, length 0
22:24:38.524260 IP gru06550-in-f8.1e100.net.https > netlabs.42186: Flags [P.], seq 2841:2896, ack 518, win 65535, length 55
22:24:38.524282 IP netlabs.42186 > gru06550-in-f8.1e100.net.https: Flags [.], ack 2896, win 34080, length 0
22:24:38.528449 IP gru14513-in-f10.1e100.net.https > netlabs.43710: Flags [.], seq 1:2841, ack 518, win 65535, length 2840
22:24:38.528469 IP netlabs.43710 > gru14513-in-f10.1e100.net.https: Flags [.], ack 2841, win 34080, length 0
22:24:38.528550 IP gru14513-in-f10.1e100.net.https > netlabs.43710: Flags [P.], seq 2841:2917, ack 518, win 65535, length 76
22:24:38.528564 IP netlabs.43710 > gru14513-in-f10.1e100.net.https: Flags [.], ack 2917, win 34080, length 0
22:24:38.529549 IP gru14519-in-f3.1e100.net.https > netlabs.37408: Flags [P.], seq 163776002:163778605, ack 456846984, win 65535, length 2683
22:24:38.529596 IP netlabs.37408 > gru14519-in-f3.1e100.net.https: Flags [.], ack 2683, win 34080, length 0
22:24:38.534007 IP netlabs.43710 > gru14513-in-f10.1e100.net.https: Flags [P.], seq 518:582, ack 2917, win 34080, length 64
22:24:38.534181 IP gru14513-in-f10.1e100.net.https > netlabs.43710: Flags [.], ack 582, win 65535, length 0
22:24:38.534536 IP netlabs.43710 > gru14513-in-f10.1e100.net.https: Flags [P.], seq 582:752, ack 2917, win 34080, length 170
22:24:38.534612 IP gru14513-in-f10.1e100.net.https > netlabs.43710: Flags [.], ack 752, win 65535, length 0
22:24:38.540712 IP gru14513-in-f10.1e100.net.https > netlabs.43710: Flags [P.], seq 2917:3528, ack 752, win 65535, length 611
22:24:38.544730 IP netlabs.43710 > gru14513-in-f10.1e100.net.https: Flags [P.], seq 752:783, ack 3528, win 36920, length 31
22:24:38.545921 IP gru14513-in-f10.1e100.net.https > netlabs.43710: Flags [.], ack 783, win 65535, length 0
22:24:38.572391 IP gru06531-in-f10.1e100.net.https > netlabs.56280: Flags [P.], seq 1:213, ack 647, win 65535, length 212
22:24:38.572420 IP netlabs.56280 > gru06531-in-f10.1e100.net.https: Flags [.], ack 213, win 30016, length 0
22:24:38.573913 IP netlabs.56280 > gru06531-in-f10.1e100.net.https: Flags [P.], seq 647:711, ack 213, win 30016, length 64
22:24:38.574014 IP gru06531-in-f10.1e100.net.https > netlabs.56280: Flags [.], ack 711, win 65535, length 0
22:24:38.575613 IP netlabs.56280 > gru06531-in-f10.1e100.net.https: Flags [P.], seq 711:898, ack 213, win 30016, length 187
22:24:38.575701 IP gru06531-in-f10.1e100.net.https > netlabs.56280: Flags [.], ack 898, win 65535, length 0
22:24:38.578730 IP gru06531-in-f10.1e100.net.https > netlabs.56280: Flags [P.], seq 213:793, ack 898, win 65535, length 580
22:24:38.579114 IP netlabs.56280 > gru06531-in-f10.1e100.net.https: Flags [P.], seq 898:922, ack 793, win 31320, length 24
22:24:38.579276 IP gru06531-in-f10.1e100.net.https > netlabs.56280: Flags [.], ack 922, win 65535, length 0
22:24:38.579350 IP netlabs.56280 > gru06531-in-f10.1e100.net.https: Flags [P.], seq 922, ack 793, win 31320, length 0
22:24:38.579490 IP gru06531-in-f10.1e100.net.https > netlabs.56280: Flags [.], ack 923, win 65535, length 0
22:24:38.580695 IP gru06531-in-f10.1e100.net.https > netlabs.56280: Flags [P.], seq 793:824, ack 923, win 65535, length 31
22:24:38.580732 IP netlabs.56280 > gru06531-in-f10.1e100.net.https: Flags [R], seq 1329717026, win 0, length 0
22:24:38.673340 IP netlabs.42186 > gru06550-in-f8.1e100.net.https: Flags [P.], seq 518:582, ack 2896, win 34080, length 64
22:24:38.673490 IP gru06550-in-f8.1e100.net.https > netlabs.42186: Flags [.], ack 582, win 65535, length 0
22:24:38.675961 IP netlabs.42186 > gru06550-in-f8.1e100.net.https: Flags [P.], seq 582:752, ack 2896, win 34080, length 170
22:24:38.675267 IP gru06550-in-f8.1e100.net.https > netlabs.42186: Flags [.], ack 752, win 65535, length 0
22:24:38.676711 IP netlabs.37408 > gru14519-in-f3.1e100.net.https: Flags [P.], seq 1:65, ack 2683, win 34080, length 64

```

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


```

[rfedor@netlabs ~]$ sudo tcpdump -n -i enp0s3 greater 64
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on enp0s3, link-type EN10MB (Ethernet), capture size 262144 bytes
21:45:00.644325 IP 10.0.2.15.58572 > 8.8.8.8.domain: 56570+ A? www.dac.unicamp.br. (36)
21:45:00.644797 IP 10.0.2.15.58572 > 8.8.8.8.domain: 45569+ AAAA? www.dac.unicamp.br. (36)
21:45:00.645563 IP 10.0.2.15.57750 > 143.186.227.165.https: Flags [P.], seq 372717431:372717545, ack 398047567, win 65535, length 114
21:45:00.650411 IP 8.8.8.8.domain > 10.0.2.15.58572: 45569 1/1/0 CNAME 143-186-227-165.nuvem.unicamp.br. (137)
21:45:00.650439 IP 8.8.8.8.domain > 10.0.2.15.58572: 56570 2/0/0 CNAME 143-186-227-165.nuvem.unicamp.br., A 143.186.227.165 (88)
21:45:00.854596 IP 143.186.227.165.https > 10.0.2.15.57750: Flags [P.], seq 1:945, ack 114, win 65535, length 944
21:45:00.854661 IP 143.186.227.165.https > 10.0.2.15.57750: Flags [P.], seq 945:2385, ack 114, win 65535, length 1440
21:45:00.854790 IP 143.186.227.165.https > 10.0.2.15.57750: Flags [P.], seq 2385:5079, ack 114, win 65535, length 2694
21:45:00.854994 IP 143.186.227.165.https > 10.0.2.15.57750: Flags [P.], seq 5079:6519, ack 114, win 65535, length 1440
21:45:00.855152 IP 143.186.227.165.https > 10.0.2.15.57750: Flags [P.], seq 6519:8081, ack 114, win 65535, length 1562
21:45:00.901444 IP 10.0.2.15.46377 > 8.8.8.8.domain: 14903+ A? fonts.googleapis.com. (38)
21:45:00.901645 IP 10.0.2.15.46377 > 8.8.8.8.domain: 29245+ AAAA? fonts.googleapis.com. (38)
21:45:00.902271 IP 10.0.2.15.48260 > 172.217.28.138.https: Flags [P.], seq 3533960600:3533960696, ack 397447017, win 53960, length 96
21:45:00.906235 IP 10.0.2.15.48260 > 172.217.28.138.https: Flags [P.], seq 96:185, ack 1, win 53960, length 89
21:45:00.906999 IP 10.0.2.15.48260 > 172.217.28.138.https: Flags [P.], seq 185:266, ack 1, win 53960, length 81
21:45:00.907190 IP 8.8.8.8.domain > 10.0.2.15.46377: 14903 1/0/0 A 172.217.162.202 (54)
21:45:00.907223 IP 8.8.8.8.domain > 10.0.2.15.46377: 29245 1/0/0 AAAA 2800:3f0:4001:80f::200a (66)
21:45:00.907906 IP 10.0.2.15.54803 > 8.8.8.8.domain: 8906+ A? www.dac.unicamp.br. (36)
21:45:00.908060 IP 10.0.2.15.54803 > 8.8.8.8.domain: 57037+ AAAA? www.dac.unicamp.br. (36)
21:45:00.908898 IP 10.0.2.15.57750 > 143.186.227.165.https: Flags [P.], seq 114:1042, ack 8081, win 65535, length 928
21:45:00.910243 IP 10.0.2.15.57750 > 143.186.227.165.https: Flags [P.], seq 1042:1335, ack 8081, win 65535, length 293
21:45:00.913265 IP 10.0.2.15.57750 > 143.186.227.165.https: Flags [P.], seq 1335:1478, ack 8081, win 65535, length 143
21:45:00.913999 IP 8.8.8.8.domain > 10.0.2.15.54803: 8906 2/0/0 CNAME 143-186-227-165.nuvem.unicamp.br., A 143.186.227.165 (88)
21:45:00.914029 IP 10.0.2.15.46745 > 8.8.8.8.domain: 48646+ A? translate.google.com. (38)
21:45:00.914546 IP 10.0.2.15.46745 > 8.8.8.8.domain: 21004+ AAAA? translate.google.com. (38)
21:45:00.917624 IP 10.0.2.15.35304 > 172.217.162.110.https: Flags [P.], seq 108748367:108748466, ack 397570487, win 36747, length 99
21:45:00.918662 IP 10.0.2.15.38420 > 8.8.8.8.domain: 18623+ A? unpkg.com. (27)
21:45:00.918761 IP 10.0.2.15.38420 > 8.8.8.8.domain: 5316+ AAAA? unpkg.com. (27)
21:45:00.919519 IP 10.0.2.15.45494 > 104.16.126.175.https: Flags [P.], seq 628957575:628957648, ack 397633719, win 34304, length 73
21:45:00.921132 IP 8.8.8.8.domain > 10.0.2.15.46745: 48646 2/0/0 CNAME www3.l.google.com., A 216.58.202.206 (75)
21:45:00.921156 IP 8.8.8.8.domain > 10.0.2.15.46745: 21004 2/0/0 CNAME www3.l.google.com., AAAA 2800:3f0:4001:81c::200e (87)
21:45:00.922411 IP 10.0.2.15.57750 > 143.186.227.165.https: Flags [P.], seq 1478:1621, ack 8081, win 65535, length 143
21:45:00.923145 IP 10.0.2.15.57750 > 143.186.227.165.https: Flags [P.], seq 1621:1764, ack 8081, win 65535, length 143
21:45:00.925677 IP 8.8.8.8.domain > 10.0.2.15.38420: 18623 5/0/0 A 104.16.125.175, A 104.16.123.175, A 104.16.124.175, A 104.16.126.175, A 104.16.122.175 (107)
21:45:00.925700 IP 8.8.8.8.domain > 10.0.2.15.38420: 5316 5/0/0 AAAA 2606:4700::6810:7caf, AAAA 2606:4700::6810:7daf, AAAA 2606:4700::6810:7aaf, AAAA 2606:4700::6810:7baf, AAAA 2606:4700::6810:7eaf (167)
21:45:00.927666 IP 8.8.8.8.domain > 10.0.2.15.54803: 57037 1/1/0 CNAME 143-186-227-165.nuvem.unicamp.br. (137)
21:45:00.959868 IP 172.217.28.138.https > 10.0.2.15.48260: Flags [P.], seq 1:610, ack 266, win 65535, length 609
21:45:00.960552 IP 172.217.28.138.https > 10.0.2.15.48260: Flags [P.], seq 610:894, ack 266, win 65535, length 284

```

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.

```

[redora@netlabs ~]$ sudo tcpdump -i enp0s3 -v 'tcp[tcpflags] & (tcp-ack) != 0'
tcpdump: listening on enp0s3, link-type EN10MB (Ethernet), capture size 262144 bytes
21:53:11.133246 IP (tos 0x0, ttl 64, id 47818, offset 0, flags [none], proto TCP (6), length 44)
    sistemas1.unicamp.br.https > netlabs.59928: Flags [S.], cksum 0x1250 (correct), seq 463104001, ack 1063792881, win 65535, options [mss 1460], length 0
21:53:11.133280 IP (tos 0x0, ttl 64, id 33234, offset 0, flags [DF], proto TCP (6), length 40)
    netlabs.59928 > sistemas1.unicamp.br.https: Flags [.], cksum 0xa618 (incorrect -> 0xb7fc), ack 1, win 29200, length 0
21:53:11.136379 IP (tos 0x0, ttl 64, id 33235, offset 0, flags [DF], proto TCP (6), length 557)
    netlabs.59928 > sistemas1.unicamp.br.https: Flags [P.], cksum 0xa81d (incorrect -> 0x1e93), seq 1:518, ack 1, win 29200, length 517
21:53:11.136687 IP (tos 0x0, ttl 64, id 47819, offset 0, flags [none], proto TCP (6), length 40)
    sistemas1.unicamp.br.https > netlabs.59928: Flags [.], cksum 0x2808 (correct), ack 518, win 65535, length 0
21:53:11.496172 IP (tos 0x0, ttl 64, id 47822, offset 0, flags [none], proto TCP (6), length 2920)
    sistemas1.unicamp.br.https > netlabs.59928: Flags [P.], cksum 0xb158 (incorrect -> 0x83c2), seq 1:2881, ack 518, win 65535, length 2880
21:53:11.496218 IP (tos 0x0, ttl 64, id 33236, offset 0, flags [DF], proto TCP (6), length 40)
    netlabs.59928 > sistemas1.unicamp.br.https: Flags [.], cksum 0xa618 (incorrect -> 0x97a7), ack 2881, win 34080, length 0
21:53:11.496308 IP (tos 0x0, ttl 64, id 47825, offset 0, flags [none], proto TCP (6), length 1245)
    sistemas1.unicamp.br.https > netlabs.59928: Flags [P.], cksum 0x3071 (correct), seq 2881:4086, ack 518, win 65535, length 1205
21:53:11.496321 IP (tos 0x0, ttl 64, id 33237, offset 0, flags [DF], proto TCP (6), length 40)
    netlabs.59928 > sistemas1.unicamp.br.https: Flags [.], cksum 0xa618 (incorrect -> 0x87da), ack 4086, win 36920, length 0
21:53:11.497965 IP (tos 0x0, ttl 64, id 33238, offset 0, flags [DF], proto TCP (6), length 133)
    netlabs.59928 > sistemas1.unicamp.br.https: Flags [P.], cksum 0xa675 (incorrect -> 0xc772), seq 518:611, ack 4086, win 36920, length 93
21:53:11.498164 IP (tos 0x0, ttl 64, id 47826, offset 0, flags [none], proto TCP (6), length 40)
    sistemas1.unicamp.br.https > netlabs.59928: Flags [.], cksum 0x17b6 (correct), ack 611, win 65535, length 0
21:53:11.514635 IP (tos 0x0, ttl 64, id 47828, offset 0, flags [none], proto TCP (6), length 44)
    104.18.21.226.http > netlabs.57802: Flags [S.], cksum 0x579a (correct), seq 463222001, ack 2154159195, win 65535, options [mss 1460], length 0
21:53:11.514668 IP (tos 0x0, ttl 64, id 28358, offset 0, flags [DF], proto TCP (6), length 40)
    netlabs.57802 > 104.18.21.226.http: Flags [.], cksum 0x8a1d (incorrect -> 0xf046), ack 1, win 29200, length 0
21:53:11.515161 IP (tos 0x0, ttl 64, id 28359, offset 0, flags [DF], proto TCP (6), length 504)
    netlabs.57802 > 104.18.21.226.http: Flags [P.], cksum 0x8bed (incorrect -> 0x382a), seq 1:465, ack 1, win 29200, length 464: HTTP, length: 464
    POST /gsrsaovsslca2018 HTTP/1.1
    Host: ocsps.globalsign.com
    User-Agent: Mozilla/5.0 (X11; Fedora; Linux x86_64; rv:65.0) Gecko/20100101 Firefox/65.0
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
    Accept-Language: en-US,en;q=0.5
    Accept-Encoding: gzip, deflate
    Content-Type: application/ocsp-request
    Content-Length: 79
    Connection: keep-alive
21:53:11.515377 IP (tos 0x0, ttl 64, id 47829, offset 0, flags [none], proto TCP (6), length 40)
    104.18.21.226.http > netlabs.57802: Flags [.], cksum 0x6d87 (correct), ack 465, win 65535, length 0
21:53:11.567723 IP (tos 0x0, ttl 64, id 47833, offset 0, flags [none], proto TCP (6), length 2278)
    104.18.21.226.http > netlabs.57802: Flags [P.], cksum 0x92db (incorrect -> 0x90b4), seq 1:2239, ack 465, win 65535, length 2238: HTTP, length: 2238
    HTTP/1.1 200 OK
    Date: Wed, 30 Sep 2020 00:53:11 GMT
    Content-Type: application/ocsp-response
    Content-Length: 1529
    Connection: keep-alive
    Set-Cookie: _cfdid=dc17f3d758c0c5ee0e22ebb2817fe3c3a1601427191; expires=Fri, 30-Oct-20 00:53:11 GMT; path=/; domain=.globalsign.com; HttpOnly; SameSite=Lax
    Expires: Sun, 04 Oct 2020 00:49:53 GMT
    X-Powered-By: Undertow/1
    ETag: "bf98f57b9a5785940b39a46fad29d9da35d65560"
    Last-Modified: Wed, 30 Sep 2020 00:49:53 GMT

```

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.

```
(fedora@netlabs ~)$ sudo tcpdump -n -i enp0s3 greater 64
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on enp0s3, link-type EN10MB (Ethernet), capture size 262144 bytes
21:45:00.644325 IP 10.0.2.15.58572 > 8.8.8.8.domain: 56570+ A? www.dac.unicamp.br. (36)
21:45:00.644797 IP 10.0.2.15.58572 > 8.8.8.8.domain: 43560+ AAAA? www.dac.unicamp.br. (36)
21:45:00.645563 IP 10.0.2.15.57750 > 143.106.227.165.https: Flags [P.], seq 372717431:372717545, ack 398047567, win 65535, length 114
21:45:00.650411 IP 8.8.8.8.domain > 10.0.2.15.58572: 45569 1/1/0 CNAME 143-106-227-165.nuvem.unicamp.br. (137)
21:45:00.650439 IP 8.8.8.8.domain > 10.0.2.15.58572: 56570 2/0/0 CNAME 143-106-227-165.nuvem.unicamp.br., A 143.106.227.165 (88)
21:45:00.854596 IP 143.106.227.165.https > 10.0.2.15.57750: Flags [P.], seq 1:945, ack 114, win 65535, length 944
21:45:00.854661 IP 143.106.227.165.https > 10.0.2.15.57750: Flags [P.], seq 945:2385, ack 114, win 65535, length 1440
21:45:00.854790 IP 143.106.227.165.https > 10.0.2.15.57750: Flags [P.], seq 2385:5079, ack 114, win 65535, length 2694
21:45:00.854994 IP 143.106.227.165.https > 10.0.2.15.57750: Flags [P.], seq 5079:6519, ack 114, win 65535, length 1440
21:45:00.855152 IP 143.106.227.165.https > 10.0.2.15.57750: Flags [P.], seq 6519:8081, ack 114, win 65535, length 1562
21:45:00.901444 IP 10.0.2.15.46377 > 8.8.8.8.domain: 14903+ A? fonts.googleapis.com. (38)
21:45:00.901645 IP 10.0.2.15.46377 > 8.8.8.8.domain: 29245+ AAAA? fonts.googleapis.com. (38)
21:45:00.902271 IP 10.0.2.15.48260 > 172.217.28.138.https: Flags [P.], seq 3533960600:3533960696, ack 397447017, win 53960, length 96
21:45:00.906235 IP 10.0.2.15.48260 > 172.217.28.138.https: Flags [P.], seq 96:185, ack 1, win 53960, length 89
21:45:00.906999 IP 10.0.2.15.48260 > 172.217.28.138.https: Flags [P.], seq 185:266, ack 1, win 53960, length 81
21:45:00.907190 IP 8.8.8.8.domain > 10.0.2.15.46377: 14903 1/0/0 A 172.217.162.202 (54)
21:45:00.907223 IP 8.8.8.8.domain > 10.0.2.15.46377: 29245 1/0/0 AAAA 2800:3f0:4001:80f::200a (66)
21:45:00.907906 IP 10.0.2.15.54803 > 8.8.8.8.domain: 8906+ A? www.dac.unicamp.br. (36)
21:45:00.908060 IP 10.0.2.15.54803 > 8.8.8.8.domain: 57037+ AAAA? www.dac.unicamp.br. (36)
21:45:00.908898 IP 10.0.2.15.57750 > 143.106.227.165.https: Flags [P.], seq 114:1042, ack 8081, win 65535, length 928
21:45:00.910243 IP 10.0.2.15.57750 > 143.106.227.165.https: Flags [P.], seq 1042:1335, ack 8081, win 65535, length 293
21:45:00.913265 IP 10.0.2.15.57750 > 143.106.227.165.https: Flags [P.], seq 1335:1478, ack 8081, win 65535, length 143
21:45:00.913999 IP 8.8.8.8.domain > 10.0.2.15.54803: 8906 2/0/0 CNAME 143-106-227-165.nuvem.unicamp.br., A 143.106.227.165 (88)
21:45:00.914029 IP 10.0.2.15.46745 > 8.8.8.8.domain: 48646+ A? translate.google.com. (38)
21:45:00.914546 IP 10.0.2.15.46745 > 8.8.8.8.domain: 21084+ AAAA? translate.google.com. (38)
21:45:00.917624 IP 10.0.2.15.35304 > 172.217.162.110.https: Flags [P.], seq 108748367:108748466, ack 397570487, win 36747, length 99
21:45:00.918662 IP 10.0.2.15.38420 > 8.8.8.8.domain: 18623+ A? unpkg.com. (27)
21:45:00.918761 IP 10.0.2.15.38420 > 8.8.8.8.domain: 5316+ AAAA? unpkg.com. (27)
21:45:00.919519 IP 10.0.2.15.45494 > 104.16.126.175.https: Flags [P.], seq 628957575:628957648, ack 397633719, win 34304, length 73
21:45:00.921132 IP 8.8.8.8.domain > 10.0.2.15.46745: 48646 2/0/0 CNAME www3.l.google.com., A 216.58.202.206 (75)
21:45:00.921150 IP 8.8.8.8.domain > 10.0.2.15.46745: 21084 2/0/0 CNAME www3.l.google.com., AAAA 2800:3f0:4001:81c::200e (87)
21:45:00.922411 IP 10.0.2.15.57750 > 143.106.227.165.https: Flags [P.], seq 1478:1621, ack 8081, win 65535, length 143
21:45:00.923145 IP 10.0.2.15.57750 > 143.106.227.165.https: Flags [P.], seq 1621:1764, ack 8081, win 65535, length 143
21:45:00.925677 IP 8.8.8.8.domain > 10.0.2.15.38420: 18623 5/0/0 A 104.16.125.175, A 104.16.123.175, A 104.16.124.175, A 104.16.126.175, A 104.16.122.175 (107)
21:45:00.925700 IP 8.8.8.8.domain > 10.0.2.15.38420: 5316 5/0/0 AAAA 2606:4700::6810:7caf, AAAA 2606:4700::6810:7daf, AAAA 2606:4700::6810:7aaf, AAAA 2606:4700::6810:7baf, AAAA 2606:4700::6810:7eaf (167)
21:45:00.927666 IP 8.8.8.8.domain > 10.0.2.15.54803: 57037 1/1/0 CNAME 143-106-227-165.nuvem.unicamp.br. (137)
21:45:00.959808 IP 172.217.28.138.https > 10.0.2.15.48260: Flags [P.], seq 1:610, ack 266, win 65535, length 609
21:45:00.960552 IP 172.217.28.138.https > 10.0.2.15.48260: Flags [P.], seq 610:894, ack 266, win 65535, length 284
```

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.

The image shows a Wireshark packet capture window titled "enp0s3". The filter bar at the top shows "(frame.len >= 64)". The packet list on the left shows a series of packets, with packet 32 selected. The packet details pane on the right shows the structure of the selected packet, which is a TLSv1.3 packet. The packet bytes pane at the bottom shows the raw data of the selected packet.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	10.0.2.15	8.8.8.8	DNS	70	Standard query 0x280e A netlabs.br
2	0.01238662	8.8.8.8	10.0.2.15	DNS	132	Standard query response 0x280e No such name A netlabs.br SOA a.dns.br
3	0.012919288	10.0.2.15	8.8.8.8	DNS	67	Standard query 0xfc8a A netlabs
4	0.020003535	8.8.8.8	10.0.2.15	DNS	142	Standard query response 0xfc8a No such name A netlabs SOA a.root-servers.net
5	0.026312948	10.0.2.15	8.8.8.8	DNS	70	Standard query 0x3e35 A netlabs.br
6	0.038389008	8.8.8.8	10.0.2.15	DNS	132	Standard query response 0x3e35 No such name A netlabs.br SOA a.dns.br
7	0.038448047	10.0.2.15	8.8.8.8	DNS	67	Standard query 0xee19 A netlabs
8	0.045916561	8.8.8.8	10.0.2.15	DNS	142	Standard query response 0xee19 No such name A netlabs SOA a.root-servers.net
9	0.069409500	10.0.2.15	8.8.8.8	DNS	83	Standard query 0xc8b4 A start.fedoraproject.org
10	0.069515366	10.0.2.15	8.8.8.8	DNS	83	Standard query 0x4c8d AAAA start.fedoraproject.org
11	0.092583967	10.0.2.15	8.8.8.8	DNS	84	Standard query 0x4aaa A detectportal.firefox.com
12	0.092609692	10.0.2.15	8.8.8.8	DNS	84	Standard query 0x8cb1 AAAA detectportal.firefox.com
13	0.099118920	8.8.8.8	10.0.2.15	DNS	266	Standard query response 0x8cb1 AAAA detectportal.firefox.com CNAME detectportal.prod.mozaws-
14	0.090276221	8.8.8.8	10.0.2.15	DNS	242	Standard query response 0x4aaa A detectportal.firefox.com CNAME detectportal.prod.mozaws-
15	0.090502760	10.0.2.15	190.98.140.18	TCP	74	54932 -> 80 [SYN, ECN, CWR] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=2723728582 TSec-
16	0.090542121	10.0.2.15	190.98.140.18	HTTP	350	GET /success.txt HTTP/1.1
17	0.091722694	190.98.140.18	10.0.2.15	HTTP	438	HTTP/1.1 200 OK (text/plain)
22	1.012598538	8.8.8.8	10.0.2.15	DNS	162	Standard query response 0x4c8d AAAA start.fedoraproject.org CNAME wildcard.fedoraproject.or-
23	1.228943932	8.8.8.8	10.0.2.15	DNS	266	Standard query response 0xc8b4 A start.fedoraproject.org CNAME detectportal.prod.mozaws.ne-
24	1.229334962	10.0.2.15	209.132.190.2	TCP	74	54150 -> 443 [SYN, ECN, CWR] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=140363708 TSe-
27	1.363411130	10.0.2.15	209.132.190.2	TLSv1.3	571	Client Hello
29	1.498759115	10.0.2.15	209.132.190.2	TCP	74	54152 -> 443 [SYN, ECN, CWR] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=140363397 TSe-
30	1.504252570	209.132.190.2	10.0.2.15	TLSv1.3	2790	Server Hello, Change Cipher Spec, Application Data
32	1.504444915	209.132.190.2	10.0.2.15	TLSv1.3	1862	Application Data, Application Data, Application Data
34	1.524640611	10.0.2.15	209.132.190.2	TLSv1.3	134	Change Cipher Spec, Application Data
36	1.525192687	10.0.2.15	209.132.190.2	TLSv1.3	224	Application Data
38	1.525735142	10.0.2.15	209.132.190.2	TLSv1.3	305	Application Data
42	1.658238486	209.132.190.2	10.0.2.15	TLSv1.3	719	Application Data, Application Data, Application Data
43	1.660796267	209.132.190.2	10.0.2.15	TCP	2790	443 -> 54150 [PSH, ACK] Seq=5219 Ack=1019 Win=65535 Len=2736 [TCP segment of a reassembled P-
45	1.661170210	209.132.190.2	10.0.2.15	TLSv1.3	2024	Application Data
47	1.668316367	10.0.2.15	209.132.190.2	TLSv1.3	85	Application Data
49	1.898880115	10.0.2.15	8.8.8.8	DNS	86	Standard query 0x1230 A tiles.services.mozilla.com
50	1.898980738	10.0.2.15	8.8.8.8	DNS	86	Standard query 0x803a AAAA tiles.services.mozilla.com
51	1.906780106	8.8.8.8	10.0.2.15	DNS	167	Standard query response 0x803a No such name AAAA tiles.services.mozilla.com SOA ns-679.awsdl-
52	1.906800475	8.8.8.8	10.0.2.15	DNS	167	Standard query response 0x1230 No such name A tiles.services.mozilla.com SOA ns-679.awsdl-
53	1.906976339	10.0.2.15	8.8.8.8	DNS	89	Standard query 0xc689 A tiles.services.mozilla.com.br
54	1.906989887	10.0.2.15	8.8.8.8	DNS	89	Standard query 0x0686 AAAA tiles.services.mozilla.com.br
55	1.913769987	8.8.8.8	10.0.2.15	DNS	194	Standard query response 0xc689 No such name A tiles.services.mozilla.com.br SOA infoblox1.p-
56	1.992164565	10.0.2.15	8.8.8.8	DNS	84	Standard query 0x3c04 A snippets.cdn.mozilla.net
57	1.992185301	10.0.2.15	8.8.8.8	DNS	84	Standard query 0x920f AAAA snippets.cdn.mozilla.net
58	1.995080780	10.0.2.15	8.8.8.8	DNS	205	Standard query response 0x920f AAAA snippets.cdn.mozilla.net CNAME d228291au1ukj.cloudfront.n-
59	2.001936351	8.8.8.8	10.0.2.15	DNS	188	Standard query response 0x3c04 A snippets.cdn.mozilla.net CNAME d228291au1ukj.cloudfront.n-
60	2.002125574	10.0.2.15	13.224.214.124	TCP	74	46338 -> 443 [SYN, ECN, CWR] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=951974897 TSec-
63	2.151085490	10.0.2.15	13.224.214.124	TLSv1.3	571	Client Hello
65	2.22919370	8.8.8.8	10.0.2.15	DNS	194	Standard query response 0x0686 No such name AAAA tiles.services.mozilla.com.br SOA infoblox-

Frame 32: 1862 bytes on wire (14896 bits), 1862 bytes captured (14896 bits) on interface 0
 Ethernet II, Src: RealtekU_12:35:02 (52:54:00:12:35:02), Dst: PcsCompu_97:c7:94 (08:00:27:97:c7:94)
 Internet Protocol Version 4, Src: 209.132.190.2, Dst: 10.0.2.15
 Transmission Control Protocol, Src Port: 443, Dst Port: 54150, Seq: 2737, Ack: 518, Len: 1808
 [2] Reassembled TCP Ssegments / 3754 bytes (380/262) #32(11031)
 0000 00 00 27 97 c7 94 52 54 00 12 35 02 08 00 45 00 ...K...S...E
 Frame (1862 bytes) Reassembled TCP (3754 bytes)
 Frame (frame), 1862 bytes
 Packets: 2864 - Displayed: 1385 (48.4%) Profile: Default

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.