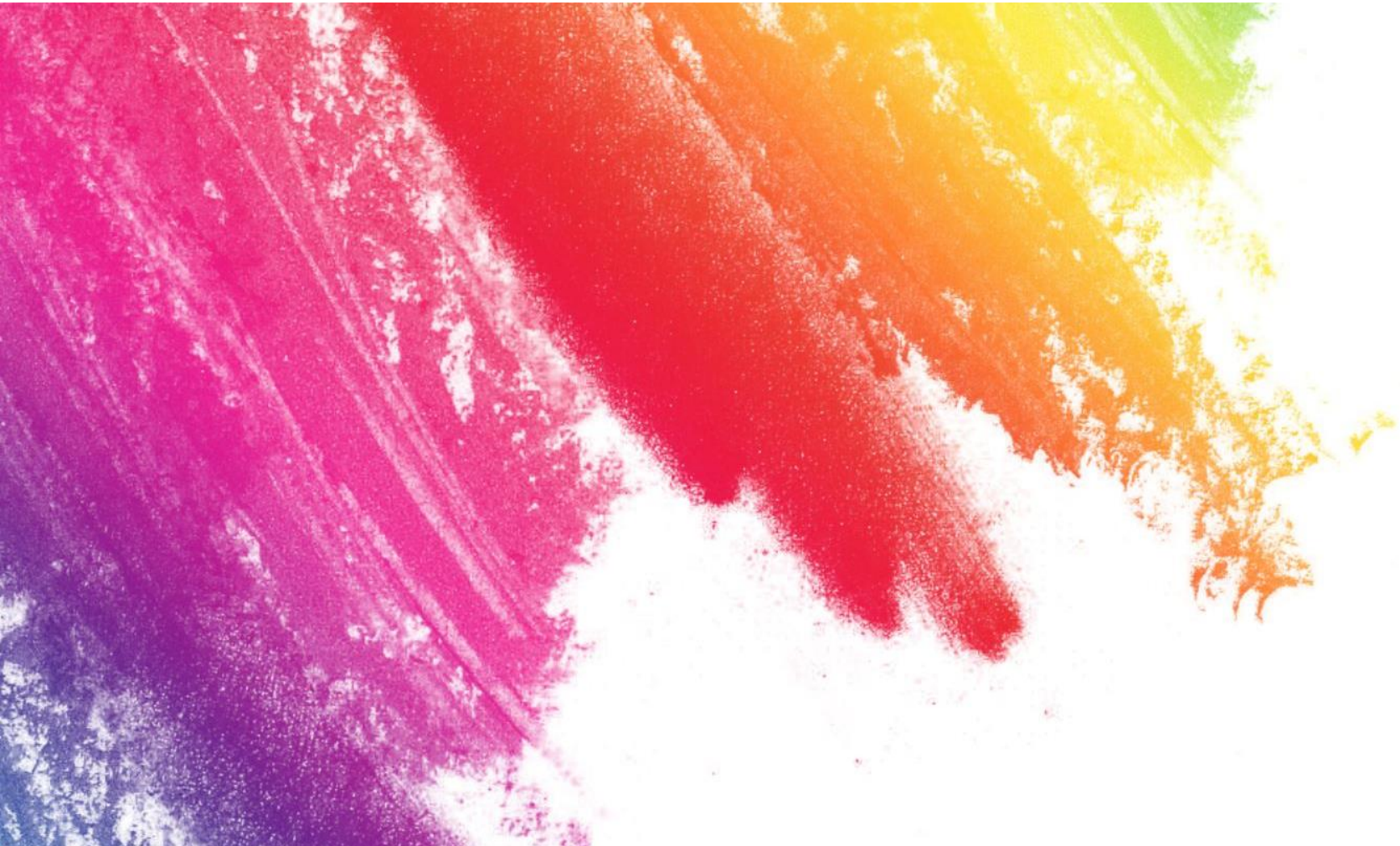
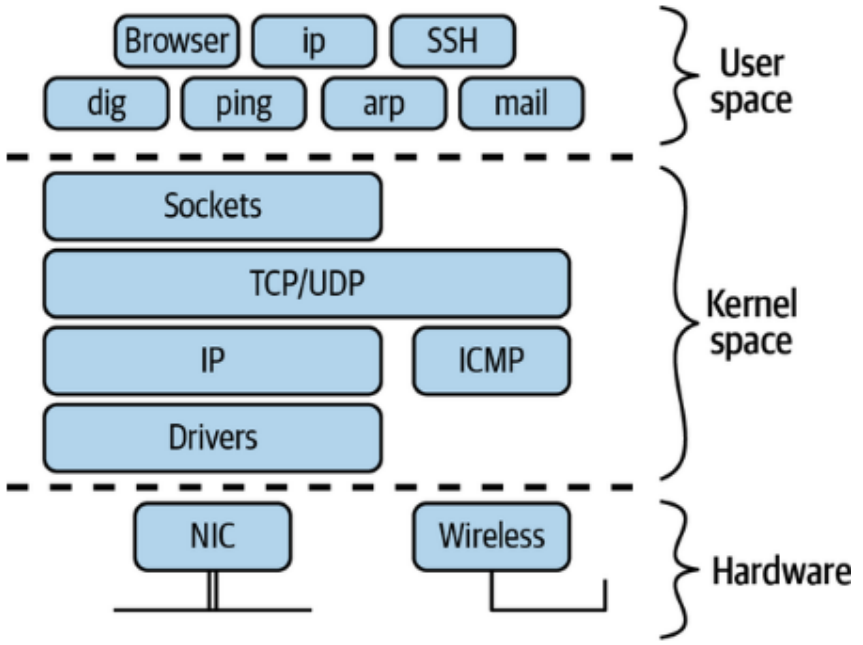
Administração Básica de Redes de Computadores

PROF. RICARDO MESQUITA

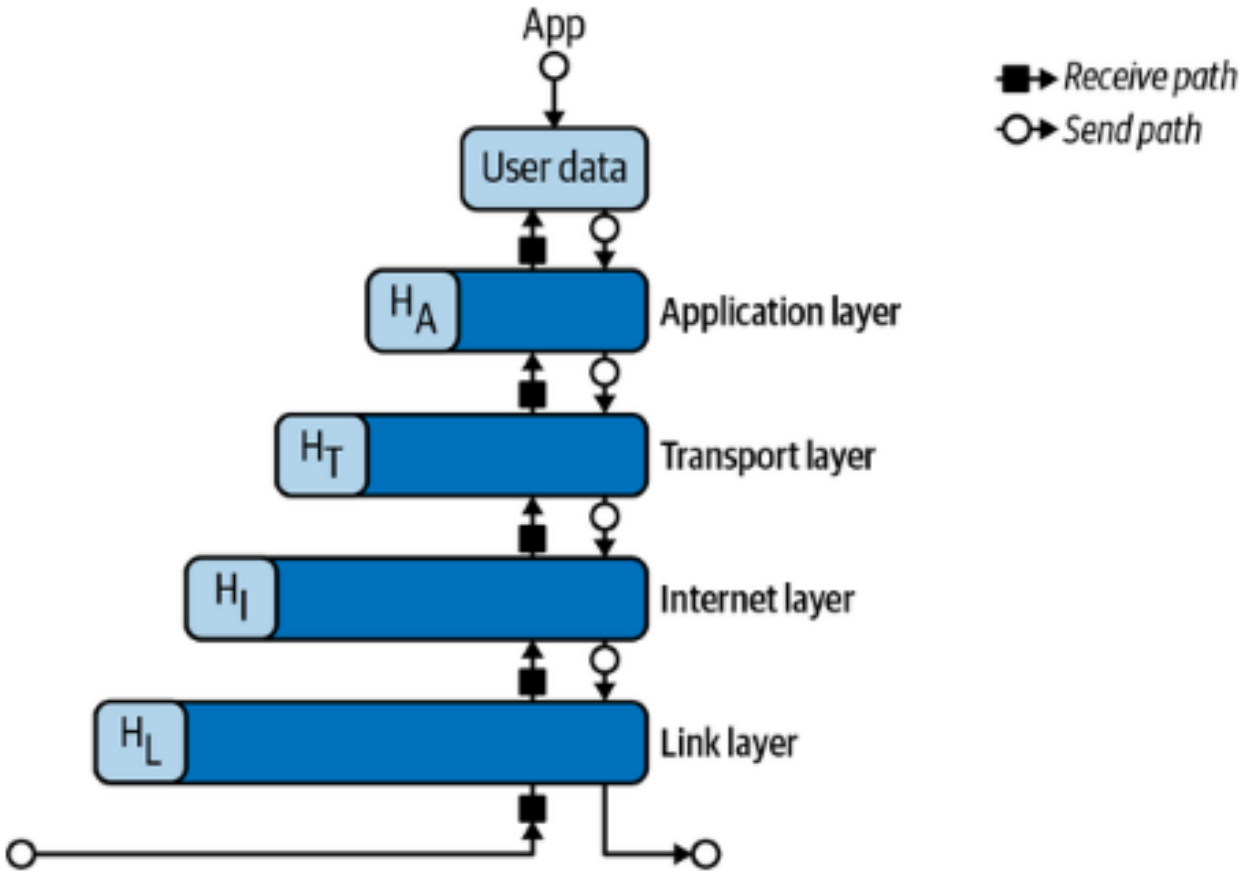


Redes com Linux

- A figura mostra como, em alto nível, a rede funciona no Linux:
  - Há algum tipo de hardware de rede, como Ethernet ou placas para acesso sem fio;
  - Existe série de componentes de nível de kernel, como a pilha TCP/IP; e
  - No espaço do usuário, há uma gama de ferramentas para configurar, consultar e usar a rede.



Lembrando da Pilha TCP/IP



Network Interface Controller (NIC)

Lembrando...

- Vamos ver um exemplo com **ifconfig** (obsoleto):

```
$ ifconfig
lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536 1
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 7218  bytes 677714 (677.7 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 7218  bytes 677714 (677.7 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

wlp1s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500 2
        inet 192.168.178.40  netmask 255.255.255.0  broadcast 192.168.178.255
        inet6 fe80::be87:e600:7de7:e08f  prefixlen 64  scopeid 0x20<link>
        ether 38:de:ad:37:32:0f  txqueuelen 1000  (Ethernet)
        RX packets 2398756  bytes 3003287387 (3.0 GB)
        RX errors 0  dropped 7  overruns 0  frame 0
        TX packets 504087  bytes 85467550 (85.4 MB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

- Interface de loopback com o endereço IP 127.0.0.1
- MTU de 65.536 bytes (tamanhos maiores significam maiores rendimentos)

- Interface NIC com seu endereço MAC (ether 38:de:ad:37:32:0f).
- Os sinalizadores (<UP,BROADCAST,RUNNING,MULTICAST>) sugerem estar operacional.

Network Interface Controller (NIC)

Lembrando...

- Para uma abordagem mais moderna de fazer a mesma coisa (consultar interfaces e verificar seu status), use o comando ip.

```
$ ip link show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue 1
   state UNKNOWN mode DEFAULT group default qlen 1000
   link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: wlp1s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue 2
   state UP mode DORMANT group default qlen 1000
    link/ether 38:de:ad:37:32:0f brd ff:ff:ff:ff:ff:ff
```

Observe que o nome (**wlp1s0**) aqui diz algo sobre a interface: é uma interface sem fio (**wl**) no barramento PCI 1 (**p1**) e no slot 0 (**s0**).

Address Resolution Protocol (ARP)

- Mapeia endereços MAC para endereços IP.
- Em certo sentido, ele faz a ponte entre a camada de enlace e a camada de rede.
- Use o comando **arp** (mais antigo) para mostrar o cache de mapeamento de endereços MAC para nomes de host ou endereços IP.
  - Você pode usar arp -n para impedir a resolução de nome de host e mostrar os endereços IP.

$ arp

Address	HWtype	HWaddress	Flags Mask	Iface
mh9-imac.fritz.box	ether	00:25:4b:9b:64:49	C	wlp1s0
fritz.box	ether	3c:a6:2f:8e:66:b3	C	wlp1s0

Address Resolution Protocol (ARP)

- A abordagem mais moderna para o **arp** é o comando **ip**:

$ ip neigh

192.168.178.34 dev wlp1s0 lladdr 00:25:4b:9b:64:49 STALE

192.168.178.1 dev wlp1s0 lladdr 3c:a6:2f:8e:66:b3 REACHABLE

Address Resolution Protocol (ARP)

- Para exibir, configurar e solucionar problemas de dispositivos sem fio, você pode usar o comando **iw**.
- Por exemplo, eu sei que minha NIC sem fio é chamada wlp1s0, então eu posso consultá-la:

```
$ iw dev wlp1s0 info
Interface wlp1s0
        ifindex
        wdev 0x1
        addr 38:de:ad:37:32:0f
        ssid FRITZ!Box 7530 QJ
        type managed
        wiphy 0
        channel 5 (2432 MHz), width: 20 MHz, center1: 2432 MHz
        txpower 20.00 dBm
```

Mostrar informações básicas sobre a interface sem fio wlp1s0.

Roteador ao qual a interface está conectada.

Exercício:
Identifique e faça a mesma consulta em sua máquina.

Address Resolution Protocol (ARP)

- Pode-se também coletar informações relacionadas com o tráfego.

```
$ iw dev wlp1s0 link
Connected to 74:42:7f:67:ca:b5 (on wlp1s0)
        SSID: FRITZ!Box 7530 QJ
        freq: 2432
        RX: 28003606 bytes (45821 packets)
        TX: 4993401 bytes (15605 packets)
        signal: -67 dBm
        tx bitrate: 65.0 MBit/s MCS 6 short GI

        bss flags:      short-preamble short-slot-time
        dtim period:    1
        beacon int:     100
```

Mostrar informações de conexão sobre a interface sem fio wlp1s0.

Estas linhas mostram estatísticas de envio (TX significa "transmitir") e recebimento (RX), ou seja, bytes e pacotes enviados e recebidos por meio desta interface.

IP

- Consultando IPs na máquina:

Listar endereços de todas as interfaces

IP de loopback

```
$ ip addr show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue
   state UNKNOWN group default qlen 1000
   link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
   inet 127.0.0.1/8 scope host lo
      valid_lft forever preferred_lft forever
   inet6 ::1/128 scope host
      valid_lft forever preferred_lft forever
2: wlp1s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc
   noqueue state UP group default qlen 1000
   link/ether 38:de:ad:37:32:0f brd ff:ff:ff:ff:ff:ff
   inet 192.168.178.40/24 brd 192.168.178.255 scope global dynamic
   noprefixroute wlp1s0
      valid_lft 863625sec preferred_lft 863625sec
   inet6 fe80::be87:e600:7de7:e08f/64 scope link noprefixroute
      valid_lft forever preferred_lft forever
```

- O endereço IP (privado) da NIC sem fio.
- Observe que este é o endereço IP local da LAN da máquina, que não é roteável publicamente, pois está no intervalo 192.168.0.0/16.

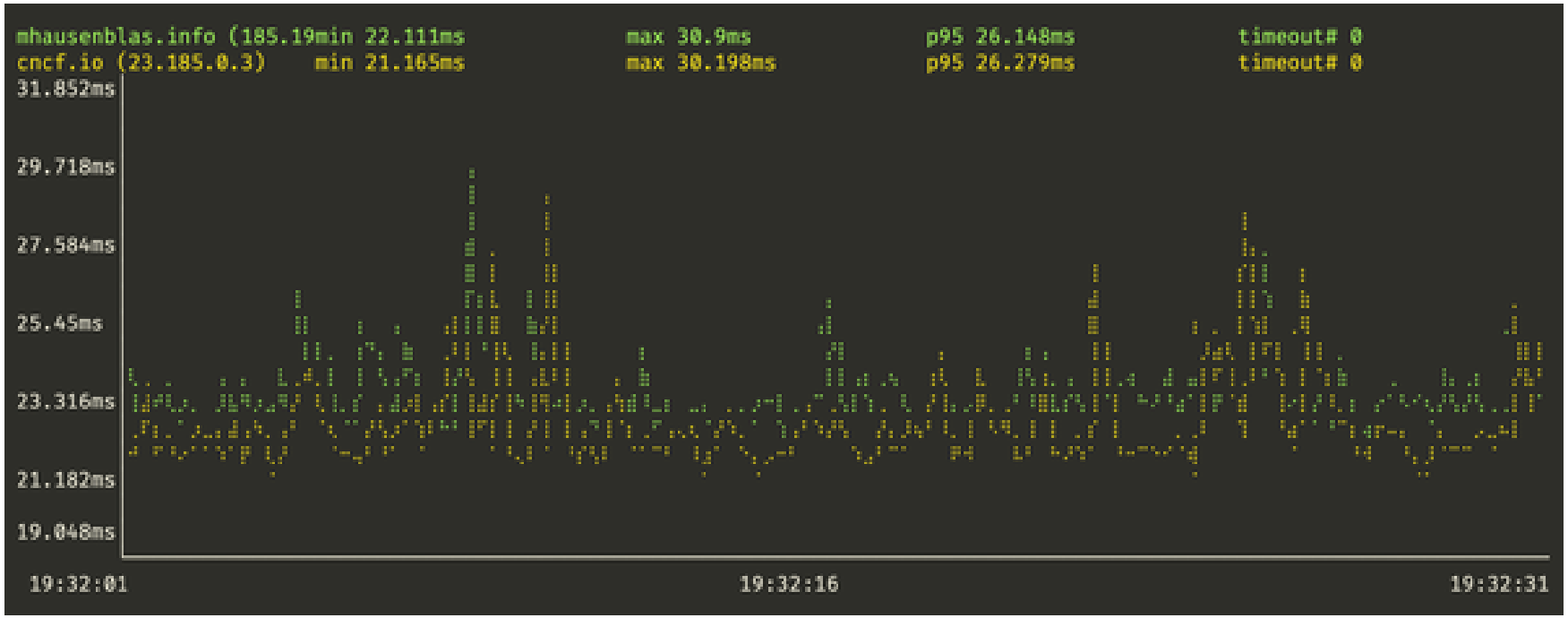
Internet Control Message Protocol (ICMP)

- O ICMP é usado para componentes de nível inferior para enviar mensagens de erro e informações operacionais como disponibilidade.
- Teste a acessibilidade de algum site usando ping

```
$ ping mhausenblas.info
PING mhausenblas.info (185.199.109.153): 56 data bytes
64 bytes from 185.199.109.153: icmp_seq=0 ttl=38 time=23.140 ms
64 bytes from 185.199.109.153: icmp_seq=1 ttl=38 time=23.237 ms
64 bytes from 185.199.109.153: icmp_seq=2 ttl=38 time=23.989 ms
64 bytes from 185.199.109.153: icmp_seq=3 ttl=38 time=24.028 ms
64 bytes from 185.199.109.153: icmp_seq=4 ttl=38 time=24.826 ms
64 bytes from 185.199.109.153: icmp_seq=5 ttl=38 time=23.579 ms
64 bytes from 185.199.109.153: icmp_seq=6 ttl=38 time=22.984 ms
^C
--- mhausenblas.info ping statistics ---
7 packets transmitted, 7 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 22.984/23.683/24.826/0.599 ms
```

Internet Control Message Protocol (ICMP)

- Alternativamente, você pode usar **gping**, que pode executar ping em vários alvos ao mesmo tempo e traçar um gráfico na linha de comando.
- Experimente também **ping6**.



Roteamento

- Consulte e exiba informações de roteamento:
  - O –n força a exibição numérica dos IPs.

```
$ sudo route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         192.168.178.1   0.0.0.0         UG    600    0        0 wlp1s0
169.254.0.0     0.0.0.0         255.255.0.0     U     1000   0        0 wlp1s0
192.168.178.0   0.0.0.0         255.255.255.0   U     600    0        0 wlp1s0
```

Custo associado à rota.

IPs de destino. 0.0.0.0 significa inespecífico ou desconhecido.

Para endereços fora da rede local.

Máscara de sub-rede usada

G significa que a rota usa um gateway. U significa que a interface a ser usada está 'up' (ativa e disponível).

A interface de rede que o pacote irá usar.

Roteamento

- Abordagem mais moderna:

$ sudo ip route

default via 192.168.178.1 dev wlp1s0 proto dhcp metric 600

169.254.0.0/16 dev wlp1s0 scope link metric 1000

192.168.178.0/24 dev wlp1s0 proto kernel scope link src 192.168.178.40 metric 600

Roteamento

- Não está acessando algo? Teste a conectividade da seguinte maneira:

$ traceroute mhausenblas.info

traceroute to mhausenblas.info (185.199.108.153), 30 hops max, 60 byte packets

1 _gateway (192.168.5.2) 1.350 ms 1.306 ms 1.293 ms

Transporte

- Conforme já mencionado, você pode ver as portas e o mapeamento em /etc/services e, além disso, há uma lista abrangente de números de portas TCP e UDP que você pode consultar se não tiver certeza.
- Se você quiser ver o que está em uso na sua máquina local (*não faça isso na máquina de outra pessoa/em um IP não local*):

Escaneie portas na máquina local

```
$ nmap -A localhost

Starting Nmap 7.60 ( https://nmap.org ) at 2021-09-19 14:53 IST
Nmap scan report for localhost (127.0.0.1)
Host is up (0.00025s latency).
Not shown: 999 closed ports
PORT    STATE SERVICE VERSION
631/tcp open  ipp     CUPS 2.2
| http-methods:
|_  Potentially risky methods: PUT
| http-robots.txt: 1 disallowed entry
|_/
|_http-server-header: CUPS/2.2 IPP/2.1
|_http-title: Home - CUPS 2.2.7

Service detection performed. Please report any incorrect results
at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 6.93 seconds
```

Encontrou uma porta aberta, 631, que é o Internet Printing Protocol (IPP).

Transporte

- Se você quiser fazer uma análise de tráfego de rede de baixo nível – ou seja, quiser ver exatamente os pacotes na pilha – você pode usar a ferramenta de linha de comando **tshark** ou sua versão baseada em GUI, **wireshark**.

```
$ sudo tshark -i wlp1s0 tcp
Running as user "root" and group "root". This could be dangerous.
Capturing on 'wlp1s0'
    1 0.000000000 192.168.178.40 → 34.196.251.55 TCP 66 47618 → 443
    [ACK] Seq=1 Ack=1 Win=501 Len=0 TSval=3796364053 TSecr=153122458
    2 0.111215098 34.196.251.55 → 192.168.178.40 TCP 66
    [TCP ACKed unseen segment] 443 → 47618 [ACK] Seq=1 Ack=2 Win=283
    Len=0 TSval=153167579 TSecr=3796227866
    ...
    8 7.712741925 192.168.178.40 → 185.199.109.153 HTTP 146 GET / HTTP/1.1
    9 7.776535946 185.199.109.153 → 192.168.178.40 TCP 66 80 → 42000 [ACK]
    Seq=1 Ack=81 Win=144896 Len=0 TSval=2759410860 TSecr=4258870662
   10 7.878721682 185.199.109.153 → 192.168.178.40 TCP 2946 HTTP/1.1 200 OK
   [TCP segment of a reassembled PDU]
   11 7.878722366 185.199.109.153 → 192.168.178.40 TCP 2946 80 → 42000
   [PSH, ACK] Seq=2881 Ack=81 Win=144896 Len=2880 TSval=2759410966 \
   TSecr=4258870662
   [TCP segment of a reassembled PDU]
   ...
```

Experimente também usar **tcpdump**.

Sockets

- Uma interface de comunicação de alto nível fornecida pelo Linux são os sockets.
- São como pontos finais de uma comunicação, com sua identidade distinta: são compostos pela porta TCP (ou UDP) e pelo endereço IP.
- Vamos usar o comando **ss** para exibir informações relacionadas ao soquete.
- Vamos supor que queremos obter uma visão geral dos soquetes TCP em uso no sistema:

Sockets

$ ss -s

Total: 913 (kernel 0)

TCP: 10 (estab 4, closed 1, orphaned 0, synrecv 0, timewait 1/0), ports 0

Transport	Total	IP	IPv6
*	0	-	-
RAW	1	0	1
UDP	10	8	2
TCP	9	8	1
INET	20	16	4
FRAG	0	0	0

Use o comando ss para consultar portas (com -s, pedimos um resumo).

O resumo do TCP; no geral, 10 soquetes em uso.

Uma visão geral mais detalhada, discriminada por tipo e versão IP.

Sockets

- O parâmetro -u restringe-se a sockets UDP, -l é para selecionar soquetes de escuta e -p também mostra as informações do processo (nenhuma, no exemplo).

$ ss -ulp

State	Recv-Q	Send-Q	Local Address:Port	Peer Address:Port
UNCONN	0	0	0.0.0.0:60360	0.0.0.0:*
UNCONN	0	0	127.0.0.53%lo:domain	0.0.0.0:*
UNCONN	0	0	0.0.0.0:bootpc	0.0.0.0:*
UNCONN	0	0	0.0.0.0:ipp	0.0.0.0:*
UNCONN	0	0	0.0.0.0:mdns	0.0.0.0:*
UNCONN	0	0	[::]:mdns	[::]:*
UNCONN	0	0	[::]:38359	[::]:*

Sockets

-c: selecionar especificamente um processo por nome.
-i: limitar ao UDP.
-5: exibir apenas as 5 primeiras linhas no pipe.

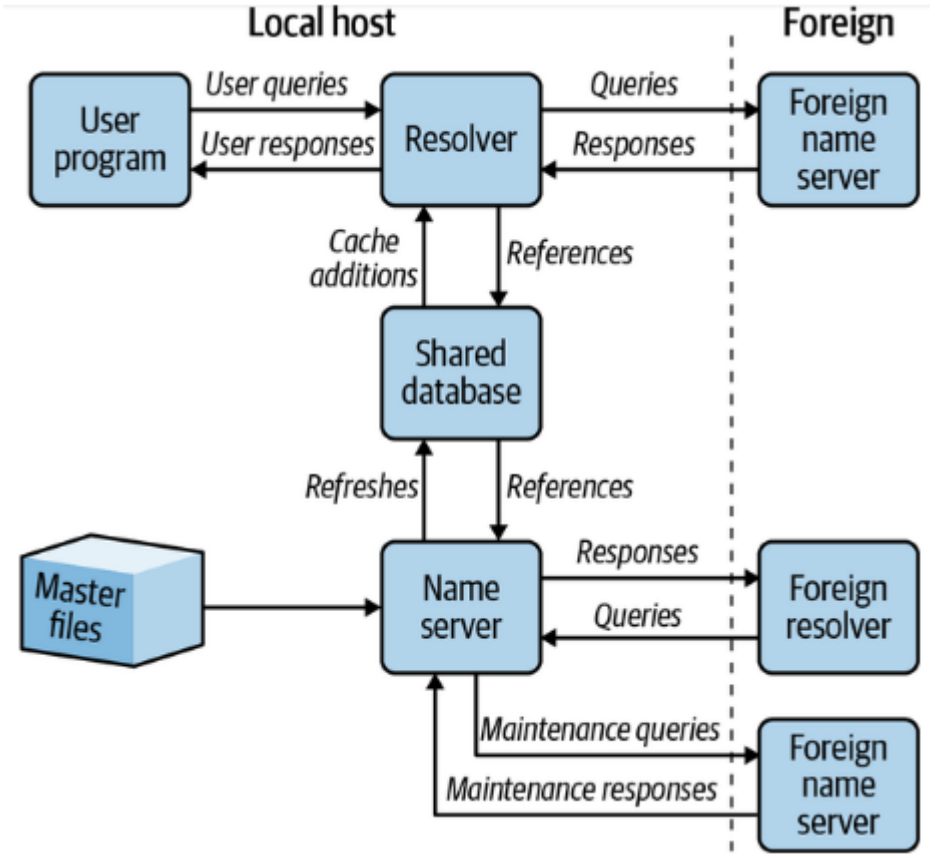
- Outra ferramenta que você pode achar útil neste contexto (soquetes e processos) é o lsof.
- Por exemplo, vamos ver quais sockets UDP o Chrome usa na máquina (saída editada):

```
$ lsof -c chrome -i udp | head -5
```

COMMAND	PID	USER	FD	TYPE	DEVICE	NODE	NAME
chrome	3131	mh9	cwd	DIR	0,5	265463	/proc/5321/fdinfo
chrome	3131	mh9	rtd	DIR	0,5	265463	/proc/5321/fdinfo
chrome	3131	mh9	txt	REG	253,0	3673554	/opt/google/chrome/chrome
chrome	3131	mh9	mem	REG	253,0	3673563	/opt/google/chrome/icudtl.dat
chrome	3131	mh9	mem	REG	253,0	12986737	/usr/lib/locale/locale-archive

DNS

- Configuração do DNS conforme a RFC 1035.



DNS

O início deste arquivo de zona no namespace.

Tempo de expiração padrão em segundos de todos os RRs que não definem seu próprio TTL.

```
$ORIGIN example.com.
$TTL 3600
@           SOA nse.example.com. nsmaster.example.com. (
                        1234567890 ; serial number
                        21600      ; refresh after 6 hours
                        3600       ; retry after 1 hour
                        604800     ; expire after 1 week
                        3600 )     ; minimum TTL of 1 hour
example.com.  IN  NS    nse
example.com.  IN  MX    10 mail.example.com.
example.com.  IN  A     1.2.3.4
nse           IN  A     5.6.7.8
www           IN  CNAME example.com.
mail          IN  A     9.0.0.9
```

O servidor de nomes para este domínio.

O servidor de e-mails para este domínio.

O endereço IPv4 deste domínio.

O endereço IPv4 do servidor de nomes.

O endereço IPv4 do servidor de email.

Torne www.example.com um alias para este domínio, ou seja, example.com.

DNS Lookups

Lookup local para endereços IP

```
$ host -a localhost
Trying "localhost.fritz.box"
Trying "localhost"
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 49150
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;localhost.                IN      ANY

;; ANSWER SECTION:
localhost.          0       IN      A       127.0.0.1
localhost.          0       IN      AAAA    ::1

Received 71 bytes from 127.0.0.53#53 in 0 ms
```

Lookup FQDN

```
$ host mhausenblas.info
mhausenblas.info has address 185.199.110.153
mhausenblas.info has address 185.199.109.153
mhausenblas.info has address 185.199.111.153
mhausenblas.info has address 185.199.108.153
```

Pesquisa reversa de endereço IP para encontrar um FQDN; se parece com o CDN do GitHub.

```
$ host 185.199.110.153
153.110.199.185.in-addr.arpa domain name pointer cdn-185-199-110-153.github.com.
```

DNS Lookups

- Uma maneira mais poderosa de consultar os registros DNS é usar o comando **dig**:

```
$ dig mhausenblas.info
; <<>> DiG 9.10.6 <<>> mhausenblas.info
;; global options: +cmd
...
;; ANSWER SECTION:
mhausenblas.info.    1799    IN    A    185.199.111.153
mhausenblas.info.    1799    IN    A    185.199.108.153
mhausenblas.info.    1799    IN    A    185.199.109.153
mhausenblas.info.    1799    IN    A    185.199.110.153

;; AUTHORITY SECTION:
mhausenblas.info.    1800    IN    NS    dns1.registrar-servers.com.
mhausenblas.info.    1800    IN    NS    dns2.registrar-servers.com.
...
```

Registros DNS

Servidor de nomes

DNS Lookups

- Digamos que queremos saber quais serviços de chat – mais especificamente, serviços Extensible Messaging and Presence Protocol (XMPP) – se houver, estão disponíveis:

```
$ dig +short _xmpp-client._tcp.gmail.com. SRV
20 0 5222 alt3.xmpp.l.google.com.
5 0 5222 xmpp.l.google.com.
20 0 5222 alt4.xmpp.l.google.com.
20 0 5222 alt2.xmpp.l.google.com.
20 0 5222 alt1.xmpp.l.google.com.
```

A opção +short é para exibir apenas a seção de resposta relevante.
_xmpp-client._tcp é o formato prescrito pela RFC 2782.
SRV: especifica em qual tipo de registro estamos interessados.

SRV (RFC 2782): SerVice locator Records. São um mecanismo de descoberta genérico.

Se você tiver um XMPP como o Jabber, poderá usar esse endereço para entrada de configuração.

Aplicação

- Você pode facilmente executar um servidor HTTP simples que serve apenas o conteúdo de um diretório de duas maneiras: usando Python ou usando netcat (nc).
- Com Python, para o conteúdo de um diretório, você faria o seguinte:

```
$ python3 -m http.server
Serving HTTP on :: port 8000 (http://[::]:8000/) ...
::ffff:127.0.0.1 - - [21/Sep/2021 08:53:53] "GET / HTTP/1.1" 200 -
```

Use o módulo integrado do Python http.server para o conteúdo do diretório atual (ou seja, o diretório a partir do qual você iniciou este comando).

Ele confirma que está pronto para servir pela porta 8000. Isso significa que você pode inserir http://localhost:8000 em seu navegador e verá o conteúdo do seu diretório lá.

Isso mostra que uma solicitação HTTP na raiz (/) foi emitida e atendida com êxito (o código de resposta HTTP 200).

Aplicação

- Se você quiser fazer coisas mais avançadas, além de servir um diretório estático, considere usar um servidor Web adequado, como o NGINX.
- Você poderia, por exemplo, executar o NGINX usando Docker com o seguinte comando:

```
$ docker run --name mywebserver \
        --rm -d \
        –v  "$PWD":/usr/share/nginx/html:ro \
        -p 8042:80 \
        nginx:1.21
```

Chame o contêiner em execução mywebserver; você verá isso ao emitir um comando docker ps para listar os contêineres em execução.

O --rm remove o contêiner na saída e o -d o transforma em um daemon (desconecte do terminal, execute em segundo plano).

Monta o diretório atual ($PWD) no contêiner como o diretório de conteúdo de origem NGINX. Observe que $PWD é uma maneira bash de endereçar o diretório atual.

A imagem do contêiner a ser usada (nginx:1.21) usando implicitamente o Docker Hub

Disponibiliza a porta 80 interna do contêiner no host via 8042. Isso significa que você poderá acessar o servidor web via http://localhost:8042 em sua máquina.

Aplicação

- Agora vamos ver como podemos usar **curl** para interagir com qualquer tipo de URL, para obter o conteúdo do servidor Web que lançamos no exemplo anterior (certifique-se de que ele ainda esteja em execução ou reinicie-o em um servidor separado).

```
$ curl localhost:8000
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
                      "http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>Directory listing for /</title>
</head>
<body>
<h1>Directory listing for /</h1>
<hr>
<ul>
<li><a href="app.yaml">app.yaml</a></li>
<li><a href="Dockerfile">Dockerfile</a></li>
<li><a href="example.json">example.json</a></li>
<li><a href="gh-user-info.sh">gh-user-info.sh</a></li>
<li><a href="main.go">main.go</a></li>
<li><a href="script.sh">script.sh</a></li>
<li><a href="test">test</a></li>
</ul>
<hr>
</body>
```

Secure Shell

- Secure Shell (SSH) é um protocolo de rede que usa criptografia para oferecer serviços de rede seguros.
- Por exemplo, acessar uma máquina virtual na nuvem com um endereço IP 63.32.106.149.
- O nome de usuário fornecido por padrão é ec2-user.

```
$ ssh \
    -i ~/.ssh/lml.pem \
    ec2-user@63.32.106.149
...
https://aws.amazon.com/amazon-linux-2/
11 package(s) needed for security, out of 35
available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-26-8-138 ~]$
```

Use o arquivo de identidade ~/.ssh/lml.pem em vez de uma senha. Fornecer explicitamente esse arquivo é uma boa prática.

A máquina de destino SSH no formato nomedeusuario@host.

Assim que o processo de login for concluído, entramos na máquina de destino e podemos usá-la como se fosse local.

Secure Shell

- Se você executa um servidor SSH, ou seja, permite que outras pessoas façam ssh em sua máquina, então você deve desabilitar a autenticação por senha. Isso força os usuários a criarem um par de chaves e compartilhar a chave pública com você, que você adiciona a ~/.ssh/authorized_keys e permite o login por meio deste mecanismo.
- Use ssh -tt para forçar a alocação de pseudo-tty.
- Faça export TERM=xterm ao fazer ssh em uma máquina, caso esteja tendo problemas de exibição.
- Configure tempos limite para sessões ssh em seu cliente. Pelo usuário, isso geralmente é feito por meio de ~/.ssh/config, onde você pode definir as opções ServerAliveInterval e ServerAliveCountMax para manter suas conexões ativas.
- Se você estiver tendo problemas, tente iniciar o ssh com a opção -v, fornecendo detalhes sobre o que está acontecendo nos bastidores (também, tente várias instâncias de v, como –vvv, para informações de depuração mais refinadas).

Transferência de Arquivos

- Para copiar de e para sistemas remotos, você pode usar a ferramenta básica **scp** (abreviação de "secure copy") que funciona sobre SSH.
- Uma vez que o padrão do **scp** é **ssh**, precisamos ter certeza de que temos a senha (ou melhor ainda, autenticação baseada em chave) em vigor para que funcione.
- Vamos supor que temos uma máquina remota com o endereço IPv4 63.32.106.149 e queremos copiar um arquivo da nossa máquina local para lá:

```
$ scp copyme \
    ec2-user@63.32.106.149:/home/ec2-user/
copyme
```

A fonte é o arquivo copyname no diretório atual.

O destino é o diretório /home/ec2-user/ na máquina 63.32.106.149.

Transferência de Arquivos

- Sincronizar arquivos com **rsync** é muito mais conveniente e rápido que **scp**. O **rsync** também usa SSH por padrão.
- Vamos transferir arquivos de ~/data/ da máquina local para o host em 63.32.106.149:

```
$ rsync -avz \
      ~/data/ \
      mh9@:63.32.106.149:
building file list ... done
./
example.txt

sent 155 bytes  received 48 bytes  135.33 bytes/sec
total size is 10  speedup is 0.05

$ ssh ec2-user@63.32.106.149 -- ls
example.txt
```

-a para arquivo, -v para detalhado e -z para usar compressão

Diretório de origem.

Destino no formato usuário@host.

Verifique se os dados chegaram executando um ls na máquina remota.

Transferência de Arquivos

- Se você não tiver certeza do que o rsync fará, use a opção --dry-run. Essencialmente, ele lhe dirá o que fará sem realmente realizar a operação.
- rsync também é uma ótima ferramenta para realizar backups de diretórios porque pode ser configurado para copiar apenas arquivos que foram adicionados ou alterados.
- Não se esqueça do: depois do host! Sem ele, o rsync seguirá em frente e interpretará a origem ou destino como um diretório local.
- Ou seja, o comando funcionará bem, mas em vez de copiar os arquivos para a máquina remota, ele irá parar na sua máquina local.
  - Por exemplo, user@example.com como destino seria um subdiretório do diretório atual chamado user@example.com/.

Dynamic Host Configuration Protocol (DHCP)

Usando dhcpdump para detectar pacotes DHCP na interface wlp1s0

```
$ sudo dhcpdump -i wlp1s0
  TIME: 2021-09-19 17:26:24.115
    IP: 0.0.0.0 (88:cb:87:c9:19:92) > 255.255.255.255 (ff:ff:ff:ff:ff:ff)
    OP: 1 (BOOTPREQUEST)
 HTYPE: 1 (Ethernet)
  HLEN: 6
  HOPS: 0
   XID: 7533fb70
   ...
OPTION:  57 (  2) Maximum DHCP message size 1500
OPTION:  61 (  7) Client-identifier         01:88:cb:87:c9:19:92
OPTION:  50 (  4) Request IP address        192.168.178.42
OPTION:  51 (  4) IP address leasetime      7776000 (12w6d)
OPTION:  12 ( 15) Host name                 MichaelminiiPad
...
```

Network Time Protocol (NTP)

- Serve para sincronizar relógios de computadores em uma rede.
- Por exemplo, usando o comando **ntpq**, um programa de consulta NTP padrão, você poderia fazer uma consulta explícita ao servidor de horário da seguinte forma:

-p: mostrar uma lista de peers conhecidos pela máquina, incluindo seu estado.

```
$ ntpq -p
     remote           refid      st t when poll reach   delay   offset  jitter
==============================================================================
 0.ubuntu.pool.n .POOL.          16 p    -   64    0    0.000    0.000   0.000
 1.ubuntu.pool.n .POOL.          16 p    -   64    0    0.000    0.000   0.000
 2.ubuntu.pool.n .POOL.          16 p    -   64    0    0.000    0.000   0.000
 3.ubuntu.pool.n .POOL.          16 p    -   64    0    0.000    0.000   0.000
 ntp.ubuntu.com  .POOL.          16 p    -   64    0    0.000    0.000   0.000
 ...
 ntp17.kashra-se 90.187.148.77    2 u    7   64    1   27.482   -3.451   2.285
 golem.canonical 17.253.34.123    2 u   13   64    1   20.338    0.057   0.000
 chilipepper.can 17.253.34.123    2 u   12   64    1   19.117   -0.439   0.000
 alphyn.canonica 140.203.204.77   2 u   14   64    1   91.462   -0.356   0.000
 pugot.canonical 145.238.203.14   2 u   13   64    1   20.788    0.226   0.000
```

Dúvidas?