Git & Github

Introdução Engenharia Informática

Mário Antunes

October 27, 2025

Exercícios

Laboratório Prático: Explorar e Usar a Rede

Objetivo: Este laboratório irá guiá-lo através dos fundamentos práticos de redes. Irá inspecionar a sua rede local, explorar a Internet, procurar por serviços e usar ferramentas profissionais como SSH e `rsync`.

Parte 0: Configuração & Preparação

Antes de começar, tem de configurar o seu ambiente com as ferramentas necessárias e um alvo (target) com o qual trabalhar.

1. Instalar Ferramentas de Rede (Linux) Abra o seu terminal e instale os seguintes pacotes, que contêm as ferramentas para os nossos exercícios:

```
$ sudo apt update
$ sudo apt install -y nmap traceroute dnsutils curl python3-pip
```

2. Criar um Par de Chaves SSH Vamos usar autenticação baseada em chaves, o método mais seguro para fazer login em servidores remotos.

```
$ ssh-keygen -t ed25519 -N ""
```

Isto cria uma **chave privada** (`~/.ssh/id_ed25519`) e uma **chave pública** (`~/.ssh/id_ed25519.pub`). **NUNCA** partilhe a sua chave privada.

3. Lançar um Alvo Seguro (Servidor SSH) Precisamos de um servidor "remoto" ao qual nos possamos ligar. Vamos usar o Docker para lançar um contentor de servidor SSH simples e pré-configurado.

1. Crie uma pasta chamada `ssh-server` e entre nela com `cd`.
2. Crie um `custom-openssh-server.Dockerfile`:

   ```
   FROM lscr.io/linuxserver/openssh-server:latest
   RUN apk update && apk add rsync && rm -rf /var/cache/apk/*
   ```

3. Crie um ficheiro chamado `compose.yml`:

   ```
   services:
     ssh:
       build:
         context: .
         dockerfile: custom-openssh-server.Dockerfile
       container_name: ssh
       environment:
         - PUID=1000
         - PGID=1000
         - TZ=Europe/Lisbon
         - USER_NAME=student
   ```

```
      - PUBLIC_KEY_FILE=/config/authorized_keys/student.pub
    volumes:
      - ./authorized_keys:/config/authorized_keys
    ports:
      - "2222:2222" # Mapeia a porta 2222 do Host para a porta 2222 do Contentor
    restart: unless-stopped
```

4. Crie uma pasta para a sua chave pública: `mkdir authorized_keys`
5. Copie a sua chave pública (do passo 2) para esta pasta, para que o servidor confie em si:

   ```
   $ cp ~/.ssh/id_rsa.pub ./authorized_keys/student.pub
   ```

6. Inicie o servidor:

   ```
   $ docker compose up -d
   ```

   Tem agora um servidor SSH a correr em `localhost` na porta 2222.

Parte 1: Exploração da Rede Local (LAN)

Exercício 1: Encontre o Seu Endereço IP Use o comando `ip` para encontrar o endereço lógico do seu computador.

```
$ ip addr show
```

- Identifique a sua interface de rede principal (ex: `eth0` ou `wlan0`).
- Encontre o seu **endereço IPv4** (ex: `192.168.1.50/24`). O `/24` é a sua **máscara de sub-rede**.
- Encontre o seu **endereço MAC** (ex: `link/ether 0a:1b:2c:3d:4e:5f`).

Exercício 2: Verifique a Interface de Loopback Teste se a pilha de rede interna do seu computador está a funcionar. A interface "loopback" (`127.0.0.1` ou `localhost`) é uma interface virtual que aponta para a sua própria máquina.

```
$ ping 127.0.0.1 -c 4
```

Deverá obter uma resposta instantânea. Isto confirma que o serviço de rede do seu sistema está ativo.

Exercício 3: Encontre o Seu Default Gateway O "Default Gateway" (Gateway Padrão) é o endereço IP do seu router—a "porta" da sua LAN para a Internet.

```
$ ip route show default
```

- O output será algo como `default via 192.168.1.1 dev eth0`.
- Agora, faça `ping` a esse endereço para confirmar que consegue alcançar o seu router.

  ```
  $ ping 192.168.1.1 -c 4
  ```

Exercício 4: Veja a Tabela ARP Agora que fez ping ao seu gateway, o seu computador conhece o endereço MAC físico dele. Use a tabela do **Address Resolution Protocol (ARP)** para ver este mapeamento.

```
$ arp -n
```

Verá uma lista de IPs locais e os seus correspondentes endereços MAC. É assim que o seu switch sabe para onde enviar pacotes locais.

Parte 2: Explorar a Rede Alargada (WAN)

Exercício 5: Testar Conectividade Externa (`ping`) Verifique se consegue alcançar um servidor na Internet e meça a sua **latência** (o tempo de ida e volta).

```
$ ping google.com -c 4
```

- Note o valor `time= ...` (ex: `time=15.2 ms`). Esta é a sua latência para os servidores do Google.

Exercício 6: Traçar o Caminho (`traceroute`) Descubra o caminho exato que os seus dados levam para chegar a um servidor. O `traceroute` mostra cada "salto" (hop) (router) ao longo do caminho.

```
$ traceroute 8.8.8.8
```

Verá uma lista de endereços IP, começando com o seu próprio router, depois os routers do seu ISP e, finalmente, os do Google.

Exercício 7: Consultar a "Lista Telefónica" (`dig`) Use o **DNS** para traduzir um nome de domínio num endereço IP.

```
$ dig google.com
```

- Procure na "ANSWER SECTION" para encontrar o registo A (o endereço IPv4).
- **Bónus:** Encontre os servidores de email de `google.com` consultando o registo `MX`.

  ```
  $ dig google.com MX
  ```

Parte 3: Descoberta de Serviços (`nmap`)

Exercício 8: Faça Scan a Si Mesmo Use o **Nmap** (Network Mapper) para procurar portas abertas no seu próprio computador.

```
$ nmap localhost
```

Provavelmente verá a porta 2222 aberta por causa do servidor SSH que iniciou na Parte 0.

Exercício 9: Faça Scan a um Servidor Público Vamos ver que portas estão abertas num website público. `scanme.nmap.org` é um servidor *especificamente* para praticar Nmap.

```
$ nmap scanme.nmap.org
```

- Que serviços estão a correr? Este output mostra-lhe porque é que as portas `80` (HTTP) e `22` (SSH) são importantes.

Parte 4: Operações Remotas Seguras (SSH & `rsync`)

Exercício 10: Ligue-se ao Seu Servidor (`ssh`) É altura de fazer login no servidor SSH que lançou na Parte 0. A sua chave pública já está autorizada.

```
# Ligar ao utilizador 'student' em localhost na porta 2222
$ ssh student@localhost -p 2222
```

- Agora está *dentro* do contentor Docker.
- Execute comandos como `whoami`, `ls -l`, ou `pwd` para provar que está num ambiente diferente.
- Escreva `exit` para sair.

Exercício 11: Sincronize Ficheiros de Forma Segura (`rsync`) O `rsync` é a melhor forma de copiar ficheiros através de SSH. É inteligente e copia apenas as diferenças.

1. Na sua **máquina anfitriã (host)**, crie uma nova pasta e um ficheiro.

   ```
   $ mkdir o-meu-projeto
   $ echo "Este é um ficheiro de teste" > ./o-meu-projeto/README.md
   ```

2. Use o `rsync` para "empurrar" (push) esta pasta para o diretório home do servidor.

   ```
   # Note a flag -e para especificar a porta SSH
   $ rsync -avzP -e "ssh -p 2222" ./o-meu-projeto student@localhost:~/
   ```

3. Faça login novamente no servidor SSH (`ssh student@localhost -p 2222`) e execute `ls`. Verá que `o-meu-projeto` foi copiado.

Parte 5: Projeto - Geo-Traceroute

Exercício 12: Construa um Traceroute Visual Neste projeto, irá combinar `traceroute`, uma API pública e uma biblioteca de mapeamento para visualizar o caminho físico que os seus dados percorrem pelo mundo. Descarregue o código aqui.

O código faz o seguinte:

1. Executa `traceroute` e extrai os endereços IP de cada salto.
2. Procura a localização geográfica de cada IP usando a API `ip-api.com`.
3. Armazena os resultados em cache num ficheiro `cache.json` para evitar repetir consultas e atingir os limites da API.
4. Plota todos os pontos e o caminho num mapa interativo.

Experimente, siga as instruções do `README.md` para configurar o projeto.

Exercício Bónus: Túnel SSH (SSH Tunneling)

Vamos usar o encaminhamento de portas (port forwarding) avançado do SSH para aceder de forma segura a um servidor web "escondido".

1. Faça login no seu contentor SSH:

   ```
   $ ssh student@localhost -p 2222
   ```

2. **Dentro do contentor,** execute um servidor web simples na porta 8000. Esta porta *não* está exposta no seu `docker-compose.yml`, por isso está inacessível a partir do seu host.

   ```
   # Dentro do contentor SSH
   $ cd ~
   $ echo "Olá de dentro do túnel!" > index.html
   $ python3 -m http.server 8000
   ```

3. Abra um **novo terminal do host** (deixe o servidor a correr).

4. Crie um túnel SSH. Este comando diz "encaminha a *minha* porta local 8080 para `localhost:8000` *no servidor remoto*."

   ```
   # Num NOVO terminal do host
   $ ssh -N -L 8080:localhost:8000 student@localhost -p 2222
   ```

   (A flag `-N` apenas abre o túnel sem iniciar uma shell).

5. Abra o browser na sua máquina **host** e vá a `http://localhost:8080`.

6. Deverá ver a mensagem "Olá de dentro do túnel!". Acedeu com sucesso a uma porta escondida através de um túnel SSH seguro.