Tema 4

Redes de Comunicações

Objetivos:

- Conceito de endereço IP
- Máscaras
- Rotas
- Configuração de rede em Linux
- Serviços de Rede
- Acesso Remoto

4.1 Introdução

Os sistemas com capacidades de comunicação em rede possuem uma variedade de identificadores que possibilitam a troca de informação. Estes identificadores funcionam como a morada numa casa, quando se pretende trocar correspondência, ou o número de telefone quando se pretende falar com um amigo. Em ambos os casos existem identificadores que permitem que a informação chegue ao seu destino, e se identifique a origem.

Neste trabalho iremos explorar como estes identificadores estão relacionados e qual a sua utilidade para a comunicação na Internet. Utilizaremos máquinas virtuais para simular uma rede local e explorar comandos de monitorização e configuração de redes.

Importante: Neste guião, recorremos a diversos comandos e ficheiros UNIX. Alguns são mesmo específicos de certas distribuições de Linux, nomeadamente das derivadas da distribuição Debian/Ubuntu. Também iremos manipular certas configurações e fazer diversas operações que requerem permissões de administrador. Por isso, recomenda-se que faça os exercícios numa máquina virtual criada propositadamente com um sistema operativo Debian, Ubuntu ou um derivado desses. Para facilitar, fornecemos uma imagem

comprimida de um disco virtual já preparado com um sistema Lubuntu.

Exercício 4.1

Se não tem já a máquina virtual da disciplina instalada, descarregue e descomprima o ficheiro do disco virtual fornecido.

Crie uma máquina virtual, indicando as opções:

Nome: `labi-lubuntu` (ou outro qualquer).

Tipo de sistema operativo: Linux.

Versão de sistema operativo: Ubuntu (64bit).

Memória: 512 MB.

Disco rígido: "Use an existing virtual hard drive file" e escolha a imagem que descarregou. Pode igualmente ativar a memória *cache* do controlador SATA de forma a acelerar o acesso ao disco.

De seguida, configure a máquina para ter duas interfaces de rede. A primeira do tipo *NAT*[a], que servirá para comunicação com a Internet; e a outra do tipo *Internal Network*, que servirá para comunicação com outras máquinas virtuais a correr no mesmo hospedeiro.

[a] veja `http://en.wikipedia.org/wiki/Network_address_translation`

Os exercícios abaixo deverão ser feitos na máquina que acabou de criar, a não ser que indiquem explicitamente outro procedimento.

4.2 Configuração de rede de um computador

A configuração de rede de um computador envolve diversos componentes, podendo ser feita de forma mais ou menos automatizada. Neste guião será utilizada a forma manual, que é a tipicamente presente em servidores e outros dispositivos. Em várias distribuições é possível utilizar o serviço `NetworkManager` para configurar os mesmos parâmetros.

Os mais importantes são: endereços, interfaces, encaminhamento e serviço de resolução de nomes. De seguida iremos abordar a configuração de alguns parâmetros.

Exercício 4.2

Abra um terminal, execute os comandos **`ifconfig`** ou **`ip address list`** e verifique:

1. Quantas interfaces de rede existem;
2. O endereço Internet Protocol v4 (IPv4)[1] de cada interface;
3. O(s) endereço(s) Internet Protocol v6 (IPv6)[2] de cada interface;
4. O endereço físico (Media Access Control (MAC)[3]) de cada interface;
5. A máscara de rede de cada interface;
6. Relacione o número de interfaces reportados no *Linux*, com o número reportado pelo *VirtualBox*.

(Pode experimentar o mesmo comando no sistema hospedeiro, se for Linux ou outro UNIX. Em Windows pode usar o comando **`ipconfig /all`**)

Além de mostrar a configuração das interfaces, o comando **`ifconfig`** também pode ser usado (pelo super administrador) para definir ou alterar essa configuração. No entanto, é mais fácil e mais usual recorrer a programas gestores de interfaces de rede, que lêem as configurações guardadas em ficheiros do sistema e aplicam-nas da mesma forma que o **`ifconfig`**.

O gestor de interfaces de rede nativo em sistemas Ubuntu é o conjunto de programas a que se chama coletivamente **`ifupdown2`** (poderá ter de ser instalado). Este gestor mantém as configurações no ficheiro **`/etc/network/interfaces`**.[1]

Este ficheiro permite especificar se as interfaces devem ser configuradas usando algum método dinâmico (ex, Dynamic Host Configuration Protocol (DHCP)[4]), ou através de uma configuração estática. Por exemplo, a configuração seguinte define que a interface **`eth2`** será configurada dinamicamente, enquanto a **`eth3`** será configurada estaticamente:

```
auto eth2
iface eth2 inet dhcp

auto eth3
```

[1] Pode consultar **`man 5 interfaces`** para detalhes da sua sintaxe.

```
iface eth3 inet static
    address 192.168.0.1
    netmask 255.255.255.0
    gateway 192.168.0.254
```

Repare que no caso da interface **eth3**, é necessário fornecer todos os parâmetros essenciais à sua operação.

Exercício 4.3

Edite o ficheiro **/etc/network/interfaces** (pode usar a aplicação **gedit** quando super-administrador) e especifique uma configuração dinâmica da primeira interface de rede (*NAT*) e uma configuração estática da segunda interface de rede (*Internal Network*). Nesta última, defina uma rede 192.168.56.0/24 e sem gateway.

Pode aplicar a configuração através do comando **ifup nome-da-interface**. O comando **ifup -a** aplica as configurações a todas as interfaces indicadas com **auto** no ficheiro **interfaces**. (Este comando é usado num dos scripts de arranque do sistema operativo.) Use o comando **ifconfig nome-da-interface** ou **ifconfig** para verificar o estado atual de uma ou de todas as interfaces. O comando **ifdown nome-da-interface** permite desactivar a configuração.

Pode necessitar desativar os interfaces através de **ifdown** antes que o comando **ifup** surja efeito.

(Necessita de permissões de administrador (usando **sudo**)para os comandos neste exercício.)

Também é comum haver outro gestor de interfaces de rede que corre no ambiente gráfico e que se configura através da interface gráfica. No caso do *Lubuntu*, temos o gestor **NetworkManager** e o interface de configuração **nm-connection-editor**, que se pode encontrar no menu principal, acedendo a *Preferences->Network Connections*.

Exercício 4.4

Utilize a interface gráfica para aplicar uma configuração que define todas as interfaces com endereços obtidos dinamicamente (DHCP). Pode aceder à configuração através do menu *Iniciar->Preferences->Network Connections*

4.3 Tabela de Endereços Físicos

Os dispositivos com capacidade de comunicação possuem endereços únicos que os identificam. O sistema operativo mantém uma tabela onde regista informação sobre as estações vizinhas conhecidas. Em *Linux* é possível a um administrador do sistema listar as entradas desta tabela executando o comando **`arp -an`**[2]. A Figura 4.1 demonstra a tabela que pode encontrar.

```
root@linux:~# arp -an
? (10.0.2.2) em 52:54:00:12:35:02 [ether] em eth0
? (192.168.56.100) em 08:00:27:fe:45:4e [ether] em eth1
root@linux:~#
```

Figura 4.1: Resultado do comando **`arp -an`**.

Exercício 4.5

Execute o comando **`arp -an`**. Verifique que endereços estão presentes.
Se estiver numa rede privada (em sua casa) veja se o endereço IPv4 de um computador da mesma rede está presente na tabela.
Repita o comando num computador físico da sala, ou no hospedeiro e compare resultado.

4.4 Tradução de nomes em endereços IP

Os nomes que utilizamos para aceder a conteúdos HyperText Transfer Protocol (HTTP)[5] não são os utilizados para as comunicações. Na realidade o endereço IPv4 (versão 4 ou versão 6) é que é utilizado a quando do estabelecimento de uma ligação. O Domain Name System (DNS)[6] é um serviço que permite traduzir nomes (ex. **`www.ua.pt`**) em endereços IPv4 e vice versa.

Exemplos de alguns nomes:

```
www.ua.pt
www.up.pt
www.sapo.pt
www.antena3.pt
www.fcporto.pt
```

[2]Necessita do pacote **`net-tools`**

```
www.scp.pt
www.sporting.pt
www.slbenfica.pt
www.google.com
www.google.pt
www.facebook.com
```

Exercício 4.6

A configuração de DNS de um sistema *Linux* encontra-se em **`/etc/resolv.conf`**. Visualize o conteúdo deste ficheiro e registe qual o servidor de DNS que está a utilizar.

Compare este valor com o presente no anfitrião da máquina virtual.

Exercício 4.7

Utilize o comando **`host`** (ex., **`host www.ua.pt`**) para obter os endereços associados a cada um dos nomes anteriormente listados (resolução direta). Seja curioso. Procure e registe endereços repetidos, múltiplos endereços ou outras situações que considere anómalas.

Exercício 4.8

Da mesma forma que é possível traduzir nomes em endereços, também é possível realizar a operação inversa. Utilizando o mesmo comando (ex., **`host 193.136.92.123`**), verifique qual a correspondência inversa (de endereço para nome). Procure identificar se a resolução direta e inversa produzem resultados compatíveis.

4.5 Conectividade e rotas

Até agora sabemos que é possível comunicar usando o endereço IPv4 do servidor. Também já abordámos o serviço que permite converter nomes em endereços IPv4. Resta saber como a informação atravessa a Internet. O segredo está no conceito de *rota de encaminhamento*.

O comando **route** permite listar (e modificar) as rotas de encaminhamento.

Exercício 4.9

De forma a determinar as rotas existentes, execute **route -n**. Verifique qual a rota por omissão (*default*) do sistema. Que outras rotas tem?
(Para perceber a tabela, consulte **man 8 route**, secção **OUTPUT**.)

Existem dois comandos particularmente relevantes no domínio do diagnóstico do estado das redes e das sua rotas: **ping** e **traceroute**. O primeiro (**ping**) permite enviar um pacote especialmente construído que instrui o destinatário a responder. Pode ser utilizado para determinar a existência de conectividade **bidireccional** e mesmo o atraso nas comunicações (*Round Trip Time*). O segundo comando (**traceroute**), mais complexo, permite identificar a rota utilizada para comunicar com o destino.

Exercício 4.10

Execute o comando **ping** para cada um dos destinos da tabela abaixo e registe o tempo médio de comunicação. Pode também verificar que algumas ligações apresentam ocasionalmente perdas de pacotes. Detecta uma correlação entre tempo, perdas e distância física?

Enviando pacotes especialmente construídos e processando as notificações enviadas de volta por cada *Router*, é possível identificar os dispositivos numa rota. O programa **traceroute** implementa este mecanismo de sinalização. A Figura 4.2 mostra a rota que existe entre servidores na Universidade de Aveiro e os servidores de **www.google.pt**. Para cada uma das entradas é mostrado o nome, endereço IPv4 e o tempo médio de resposta.

Devido às regras de segurança aplicadas na Universidade de Aveiro, não é possível utilizar o programa **traceroute** dentro da rede de clientes da universidade (ex, **eduroam**).

Durante a aula recomenda-se utilizar um serviço Web que permite executar o comando **traceroute** desde um servidor remoto em Portugal. Naturalmente, a origem dos pacotes não será Aveiro e o resultado será ligeiramente diferente ao que se obteria dentro da Universidade de Aveiro. Para aceder ao serviço, utilize o navegador que tem instalado e insira o endereço: `http://toolbox.3gnt.net/network/`.

Nome	Localização
www.ua.pt	Aveiro, Portugal
www.ipp.pt	Porto, Portugal
www.utl.pt	Lisboa, Portugal
www.utad.pt	Vila Real, Portugal
www.uevora.pt	Évora, Portugal
www.uam.es	Madrid, Espanha
www.univ-paris8.fr	Paris, França
www.cmu.edu	Pittsburgh, EUA
www.bjut.edu.cn	Pequim, China
www.u-tokyo.ac.jp	Tóquio, Japão
www.adelaide.edu.au	Adelaide, Austrália
www.poea.gov.ph	Filipinas

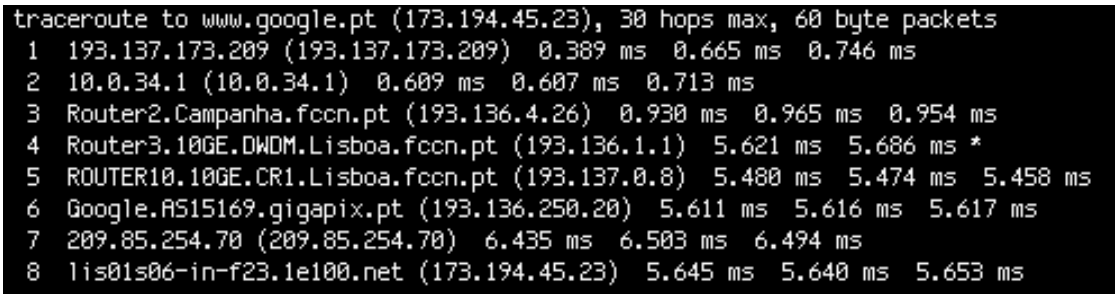

Figura 4.2: Resultado do comando **`traceroute www.google.pt`** executado a partir de um servidor na Universidade de Aveiro.

Preencha o endereço de destino pretendido, selecione a ferramenta (**`traceroute`** ou **`lft`**) e inicie o teste. Ao fim de alguns segundos, aparece o resultado. Pode experimentar vários endereços através desta interface.

Se estiver a realizar este guia fora da Universidade de Aveiro, poderá utilizar o comando

`traceroute endereço` directamente a partir da linha de comandos da máquina virtual.

Exercício 4.11

Para cada um dos endereços anteriormente analisados, obtenha a rota desde o ponto de origem até ao destino. De seguida, analise a rota obtida, identifique e registe:

1. O número de routers na rota.
2. Alguns países por onde o tráfego foi encaminhado.
3. O *Router* com maior atraso.

4.6 Identificação da entidade responsável por uma máquina

Todos os equipamentos possuem uma entidade responsável e esta entidade tem de estar devidamente identificada perante os restantes utilizadores da Internet. Bases de dados disponíveis *online* como, por exemplo, `http://www.whois.sc` permitem consultar esta informação. Se estiver fora da Universidade de Aveiro, pode também consultar esta informação através do comando **`whois`**.

Exercício 4.12

Para cada um dos nomes, registe o nome do titular do registo.

Exercício 4.13

Considerando as rotas obtidas anteriormente, e utilizando o serviço de *Whois* registe qual a entidade responsável (Organization), pelo acesso à Internet de cada um dos destinos.

4.7 Transmissão de informação em redes: `ping`

Até agora tem-se referido que as redes atuais são orientadas à comunicação por pacotes, não tendo sido no entanto observados estes elementos de comunicação. Neste ponto iremos observar o que realmente acontece quando se executa o comando **`ping www.google.pt`**.

Para isso é necessário utilizar uma aplicação que permite escutar (*sniffing*) todo o tráfego

enviado para a rede: *Wireshark*.

De forma a capturar tráfego, execute o *Wireshark* como super-administrador, aceda às opções e defina que quer escutar pacotes na interface de rede *NAT*, que configurou no *VirtualBox*.

Responda às questões:

- Quais os endereços IPv4 e MAC envolvidos nas comunicações?
- Que protocolos são utilizados em cada comunicação?
- Consegue identificar o endereço do servidor de DNS?
- Com o comando **`ping`** consegue saber a que pedido corresponde uma resposta?

Exercício 4.14

Repita o comando **`ping`** para vários endereços. Consegue explicar o funcionamento do comando?

4.8 Transmissão de informação em redes: conteúdo HTTP

O protocolo HTTP é um protocolo de nível aplicacional, muito utilizado para a transferência de informação na Internet. Sempre que acede ao *Google* ou ao *Facebook* está a utilizar este protocolo. Visto ser um protocolo aplicacional, funciona em cima de um outro protocolo chamado Transmission Control Protocol (TCP)[7]. O protocolo TCP permite que vários serviços o utilizem, criando a noção de portas. Cada comunicação usa uma porta diferente e assim é possível comunicar. Isto será abordado nos parágrafos seguintes.

Para transferir a informação, o protocolo HTTP baseia-se no princípio do pedido (*request*) e resposta (*response*). Quando insere um Uniform Resource Locator (URL)[8] no *browser*, é enviado um pedido, ao qual o servior responde com o conteúdo pretendido.

O exemplo que se segue é um destes pedidos. Neste caso, requisita-se a página **`/`** ao servidor **`www.google.pt`**. Como pode verificar, o cliente identifica-se (**`User-Agent`**) e define que tipo de conteúdo aceita (**`Accept`**).

```
GET / HTTP/1.1
Host: www.google.pt
User-Agent: Mozilla/5.0
Accept: text/html
```

Exercício 4.15

O comando **`telnet endereço-do-servidor porta`** permite efetuar uma ligação TCP, sobre a qual se pode transmitir informação. Para verificar a simplicidade do protocolo HTTP, efectue uma ligação ao servidor indicado na porta 80 (ex, **`telnet www.google.pt 80`**) e envie o pedido mostrado acima (copiar/colar). Deverá aparecer muito texto. Consegue entender o seu conteúdo?

Utilize o *Wireshark* e verifique o que realmente acontece.

Exercício 4.16

Utilizando o *browser*, repita o processo para qualquer outro *site*, e analise o resultado com o *Wireshark*.

Consegue identificar os endereços IP e os protocolos utilizados?

Relativamente ao protocolo HTTP, consegue identificar a versão do protocolo, o cliente utilizado, o servidor e o caminho que compõem o URL pedido?

4.9 Acesso a servidores remotos

4.9.1 SSH

Como vimos, é possível realizar ligações a sistemas e trabalhar nestes, como se de um sistema local se tratasse. No passado, o método mais utilizado era o protocolo **`telnet`**, que ainda se pode encontrar em alguns equipamentos mais simples. Atualmente, um dos métodos mais utilizados é o **`ssh`**, acrónimo de Secure Shell.

O **`ssh`** possibilita aceder a uma *shell* num sistema remoto, sobre a qual é possível executar comandos. Ao contrário do **`telnet`**, todas as trocas de informação através do **`ssh`** são seguras. As comunicações são cifradas de forma que não poderão ser percebidas por um atacante que as intercepte. Existem algumas outras funcionalidades que levaram a que o **`ssh`** se tornasse mais popular:

- Suporta níveis de segurança configuráveis.
- Suporta a transferência de ficheiros.
- Suporta a criação de túneis de tráfego (como uma Virtual Private Network (VPN)[9]).
- Suporta a execução de aplicações gráficas remotas.

Para iniciar uma ligação **ssh** basta executar **ssh username@servidor** ou alternativamente, **ssh -l username servidor**.

No caso desta disciplina, o servidor mais utilizado será **xcoa.av.it.pt** enquanto o username terá o formato `labi-tXgY`.

Portanto, ao executar na consola: **ssh labi-tXgY@xcoa.av.it.pt**, o grupo Y da turma X estará a iniciar uma sessão remota no servidor **xcoa.av.it.pt**. Use o seu número de turma e grupo que pode ser fornecido pelo docente.

Recorde estes valores pois serão necessários para toda a disciplina.

Como mecanismo de segurança, o **ssh** cria uma impressão digital dos servidores. Isto permite que os utilizadores tenham a certeza que se estão a ligar ao servidor certo.[3] No caso do servidor atual, e na primeira ligação, será apresentada a seguinte mensagem:

```
The authenticity of host 'xcoa.av.it.pt (193.136.92.147)' can't be established.
ECDSA key fingerprint is SHA256:Se2g3o+sVClY+zPOSNLBP/L5vCfIjo9W+08spExPXbg.
Are you sure you want to continue connecting (yes/no)?
```

Repare que a impressão digital (*fingerprint*) do servidor tem o valor
SHA256:Se2g3o+sVClY+zPOSNLBP/L5vCfIjo9W+08spExPXbg.
Caso este valor seja apresentado, o utilizador sabe que está a ligar-se ao servidor correto. Caso seja diferente, deverá interromper imediatamente a tentativa de ligação. Num cenário real, um utilizador pode verificar o valor junto do administrador do sistema.

As impressões digitais dos servidores previamente acedidos são armazenadas no ficheiro **~/.ssh/known_hosts**. Numa ligação posterior, o **ssh** confirma a impressão digital e não requer intervenção do utilizador. Caso a impressão digital guardada seja diferente da do servidor, é mostrada a mensagem que se segue:

[3] Note que é possível desviar as comunicações que passam na Internet, tal como um carteiro poderia desviar cartas se assim o entendesse.

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@    WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!     @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!
Someone could be eavesdropping on you right now (man-in-the-middle attack)!
It is also possible that a host key has just been changed.
The fingerprint for the ECDSA key sent by the remote host is
SHA256:Se2g3o+sVClY+zPOSNLBP/L5vCfIjo9W+08spExPXbg.
Please contact your system administrator.
Add correct host key in /home/linux/.ssh/known_hosts to get rid of this message.
Offending ECDSA key in /home/linux/.ssh/known_hosts:1
ECDSA host key for xcoa.av.it.pt has changed and you have requested strict checking.
Host key verification failed.
```

Depois da sessão se encontrar estabelecida, todos os comandos introduzidos são executados no servidor remoto, sendo o seu resultado enviado de volta para o cliente local. Pode verificar que utilizadores se encontram ligados executando o comando **who**.

Exercício 4.17

Remova o ficheiro **~/.ssh/known_hosts** do seu sistema (local). De seguida, efetue uma nova ligação ao servidor *xcoa.av.it.pt* e verifique que a impressão digital é a apresentada neste guião. Termine a ligação e volte a estabelecê-la. É apresentada alguma mensagem adicional?

Verifique agora o conteúdo do ficheiro **~/.ssh/known_hosts** e localize a entrada relacionada com o servidor *xcoa.av.it.pt*.

4.9.2 Transferência de ficheiros

O **ssh** não permite apenas executar comandos remotos, permite igualmente transferir ficheiros entre sistemas. É possível transferir ficheiros e diretórios de e para servidores remotos, ou entre servidores remotos.

Para transferir um ficheiro utilizando **ssh**, invoca-se o comando **scp** (*secure copy*). A sintaxe do comando **scp** é a seguinte:

```
scp origem destino
```

A **origem** e o **destino** podem ser simplesmente nomes de ficheiros e/ou diretórios locais e, nesse caso, o comportamento é idêntico ao do comando **cp**. Porém, se a **origem** e/ou o **destino** tiverem o formato **utilizador@servidor:ficheiro**, então indicam um determinado ficheiro (ou diretório) de um certo servidor, acedido pelo utilizador indicado. O exemplo seguinte copia

um ficheiro *teste.txt*, que se encontra na área pessoal do utilizador atual, para a área pessoal do utilizador chamado *user* no servidor *xcoa.av.it.pt*. Num cenário real, terão de se utilizar utilizadores e caminho adequados.

```
scp ~/teste.txt user@xcoa.av.it.pt:/home/user
```

Para copiar o ficheiro de volta, desta vez para o diretório atual, poderia executar-se:

```
scp user@xcoa.av.it.pt:/home/user/teste.txt .
```

Também é possível copiar o ficheiro entre sistemas remotos:

```
scp user1@xcoa.av.it.pt:/home/user/teste.txt user2@xcoa.av.it.pt:
```

Exercício 4.18

Usando o *browser*, descarregue uma imagem para o seu computador, e utilizando o **scp**, copie-a para o servidor *xcoa.av.it.pt*.

Autenticação por chaves

Até agora a autenticação do protocolo **ssh** junto de servidores remotos tem sido efetuada através de um nome de utilizador e de uma senha. Quando se estabelecem ligações a servidores frequentemente, o facto de se introduzirem constantemente estes dados torna-se problemático. Além disso, a utilização deste par de elementos (utilizador e senha), não é a forma mais segura de autenticação. Entre outros problemas de segurança, o uso de senhas curtas ou baseadas em palavras de dicionário pode comprometer o sistema.

O **ssh** permite a utilização de um par de chaves que irá substituir a senha. Estas chaves são constituídas por duas partes (2 ficheiros). Uma é pública e deve ser colocada nos servidores a que pretendemos ligar, a outra nunca deve ser fornecida a terceiros e fica no computador local.[4]

O primeiro passo a fazer é a criação das chaves para autenticação. A chave local deve estar cifrada com uma senha. Esta senha nunca é enviada para o servidor, serve apenas para decifrar o ficheiro da chave privada **id_rsa**. Isto consegue-se executando o comando **ssh-keygen** e seguindo as instruções fornecidas:

```
$ ssh-keygen
```

[4] O conceito de chaves públicas e privadas está fora do âmbito desta disciplina. Se tiver curiosidade pode consultar a página http://pt.wikipedia.org/wiki/Criptografia_de_chave_pública

```
Generating public/private rsa key pair.
Enter file in which to save the key (/home/debian/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/debian/.ssh/id_rsa.
Your public key has been saved in /home/debian/.ssh/id_rsa.pub.
The key fingerprint is:
cd:78:2d:63:12:aa:02:89:22:de:89:4d:cd:3d:8e:73 labi@labi
```

A partir deste momento estarão criados dois ficheiros com as duas chaves:

`/home/debian/.ssh/id_rsa.pub` Chave Pública (a colocar no servidor).

`/home/debian/.ssh/id_rsa` Chave Privada (a manter localmente).

A chave pública deve ser adicionada ao ficheiro **`~/.ssh/authorized_keys`** do servidor remoto para que o **`ssh`** passe a utilizar as chaves criadas em vez do método tradicional para autenticação. Pode acrescentar a chave usando comandos ou um editor no servidor, ou usando o comando **`ssh-copy-id`** a partir do sistema cliente.

Exercício 4.19

Crie um par de chaves no seu computador sem especificar uma palavra passe. Instale a chave pública na sua conta do servidor *xcoa.av.it.pt* e verifique o que acontece quanto volta a estabelecer uma sessão com o servidor.

Volte a criar e instalar um par de chaves mas especificando uma senha. Volte a estabelecer uma sessão com o servidor e verifique o que acontece.

Pode obter mais informação sobre o processo de autenticação se executar **`ssh -v`** em vez de **`ssh`**.

Reencaminhamento do protocolo X11

Também é possível executar aplicações gráficas através de **`ssh`**. Em *Linux*, a interface gráfica é um serviço que recebe pedidos das aplicações clientes (ex., desenhar um botão, apresentar uma imagem) e envia-lhes eventos (ex., tecla pressionada, nova posição do rato, etc). Como a comunicação dos pedidos e dos eventos é feita através de mensagens, usando o protocolo X11,[5] é perfeitamente possível que a aplicação e a interface gráfica se encontrem em sistemas distintos.

Utilizando **`ssh`**, podemos disponibilizar o servidor de X11 local como terminal gráfico para as aplicações remotas.[6] Para isto é necessário executar o **`ssh`** da seguinte forma:

[5] Para mais informação, consultar `http://pt.wikipedia.org/wiki/X_Window_System`.

[6] Note que, nesta situação, o **`ssh`** é um *cliente* local do *servidor* remoto de aplicações, mas essas aplicações remotas são clientes do servidor de X11 que corre no sistema local.

```
ssh -X user@servidor
```

Depois, poderá executar qualquer aplicação gráfica e não apenas comandos de texto nas sessões remotas. Se se pretender iniciar uma sessão sem suporte de reencaminhamento de X11, o **ssh** poderá ser executado com a opção **-x** (minúsculo).

Exercício 4.20

Efetue uma ligação ao servidor *xcoa.av.it.pt* com a opção **-x** e interprete o resultado do comando **midori**.

Volte a repetir a sessão mas desta vez utilizando a opção **-X**. Compare o resultado que obtém ao executar o comando **midori**.

Utilizando o **midori** (no servidor remoto) e o **firefox** (no computador local), aceda ao URL `http://labi2.aws.atnog.av.it.pt/ip/` e compare o resultado. Pode igualmente aceder a outras páginas como `http://my.ua.pt` ou `http://www.sapo.pt` e verificar que em ambos os casos existe conetividade à Internet.

4.10 Para aprofundar o tema

Exercício 4.21

Utilizando o wireshark, capture tráfego e identifique todos os pedidos efectuados. Pode utilizar o filtro **http.request** depois de capturar os pacotes, se quiser visualizar apenas os pedidos de HTTP.

Exercício 4.22

Execute o comando **traceroute** e, através da aplicação *Wireshark*, verifique que pacotes são enviados. Tente encontrar a função do campo **TTL** e como este é utilizado.

Exercício 4.23

Execute o comando **pftp glua.ua.pt** e utilize o utilizador **ftp** sem palavra passe. Capture o tráfego e verifique que dados consegue visualizar na captura.

Caso necessite de instalar este comando, pode fazê-lo através de: **apt install ftp**

Exercício 4.24

É comum nomear os sistemas com personagens e locais de livros, séries, filmes, sagas ou outras obras. Sabendo que os sistemas centrais da Universidade de Aveiro estão na rede `193.136.173.0/24` e recorrendo ao comando **`host`** ou **`dig -x`**, resolva vários endereços IPv4 e determine qual(ais) as obras que são utilizadas para nomear alguns sistemas da universidade.

Poderá ter de instalar o pacote **`dnsutils`**.

Glossário

DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name System
HTTP	HyperText Transfer Protocol
IPv4	Internet Protocol v4
IPv6	Internet Protocol v6
MAC	Media Access Control
SATA	Serial ATA
TCP	Transmission Control Protocol
URL	Uniform Resource Locator
VPN	Virtual Private Network

Referências

[1] J. Postel, *Internet Protocol*, RFC 791 (Standard), Updated by RFC 1349, Internet Engineering Task Force, set. de 1981.

[2] S. Deering e R. Hinden, *Internet Protocol, Version 6 (IPv6) Specification*, RFC 2460 (Draft Standard), Updated by RFCs 5095, 5722, 5871, Internet Engineering Task Force, dez. de 1998.

[3] C. Hornig, *A Standard for the Transmission of IP Datagrams over Ethernet Networks*, RFC 894 (Standard), Internet Engineering Task Force, abr. de 1984.

[4] R. Droms, *Dynamic Host Configuration Protocol*, RFC 2131 (Draft Standard), Updated by RFCs 3396, 4361, 5494, Internet Engineering Task Force, mar. de 1997.

[5] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach e T. Berners-Lee, *Hypertext Transfer Protocol – HTTP/1.1*, RFC 2616 (Draft Standard), Updated by RFCs 2817, 5785, 6266, Internet Engineering Task Force, jun. de 1999.

[6] P. Mockapetris, *Domain names - implementation and specification*, RFC 1035 (Standard), Updated by RFCs 1101, 1183, 1348, 1876, 1982, 1995, 1996, 2065, 2136, 2181, 2137, 2308, 2535, 2845, 3425, 3658, 4033, 4034, 4035, 4343, 5936, 5966, Internet Engineering Task Force, nov. de 1987.

[7] J. Postel, *Transmission Control Protocol*, RFC 793 (Standard), Updated by RFCs 1122, 3168, 6093, Internet Engineering Task Force, set. de 1981.

[8] M. Mealling e R. Denenberg, *Report from the Joint W3C/IETF URI Planning Interest Group: Uniform Resource Identifiers (URIs), URLs, and Uniform Resource Names (URNs): Clarifications and Recommendations*, RFC 3305 (Informational), Internet Engineering Task Force, ago. de 2002.

[9] L. Andersson e T. Madsen, *Provider Provisioned Virtual Private Network (VPN) Terminology*, RFC 4026 (Informational), Internet Engineering Task Force, mar. de 2005.