SSH – Secure Shell

Miguel Frade

Departamento de Engenharia Informática
Instituto Politécnico de Leiria

October 9, 2014

O SSH

O que é o SSH?

- É um protocolo. Quando nos referimos ao protocolo SSH escreve-se em maiúsculas;
- Cifra as ligações entre computadores;
- Serviços de segurança implementados no SSH:
  - autenticação; confidencialidade; e integridade;
- Existem várias aplicações que implementam o SSH:
  - putty, WinSCP, Open-SSH, etc.

O que não é o SSH?

- não é um produto
- não é uma *shell* tipo *bash*
- não permite armazenar a informação cifrada

O SSH usado nas aulas

Nos sistemas GNU/Linux usa-se a implementação `OpenSSH`

- Modelo cliente-servidor

Instalar o OpenSSH

```
sudo apt-get install openssh-server openssh-client
```

Verificar instalação

```
ssh -V                    # cliente SSH
sudo service ssh status # Servidor
```

Como se usa o SSH

Logins remotos

- substitui o `telnet`
- especificar o utilizador com a opção `-l user`
- ou em alternativa com a opção `user@` seguido do nome ou IP do servidor
- entre sistemas Linux a especificação do utilizador pode ser omitida se nome for o mesmo no cliente e no servidor

Exemplos

```
ssh -l jonas exemplo.pt
ssh jonas@exemplo.pt
ssh 192.168.226.3 # utilizador omitido
```

Como se usa o SSH

Cópia segura de ficheiros

- Permite transferir ficheiros entre computadores de forma segura
- Sintaxe: `scp` *`origem destino`* tal como o comando `cp`
- a *`origem`* e o *`destino`* podem especificar um computador remoto

Exemplos

```
# origem remota, destino local
# "." representa a diretoria atual
 scp user@exemplo.pt:ficheiro.txt .
 scp user@exemplo.pt:ficheiro.txt dir
 scp user@exemplo.pt:ficheiro.txt /dir

# origem local, destino remoto
# o "." representa a home da conta destino
  scp ficheiro.txt user@exemplo.pt:.
```

Como se usa o SSH

Montar uma diretoria remota

- com o `sshfs` é possível montar uma diretoria remota
- todo o tráfego é cifrado
- é preciso instalar: `sudo apt-get install sshfs`
- permite o acesso a ficheiros remotos a partir de outras aplicações
  - editores de texto, exploradores de ficheiros, etc

Exemplo

```
# diretoria mnt deve estar vazia antes deste comando
  sshfs user@192.168.226.3:dir mnt

# confirmar o mount realizado no comando anterior
  mount

# desfazer o mount
  fusermount -u mnt
```

Como se usa o SSH

Execução remota de comandos

- a partir de um cliente, executar comandos em servidores
- uma solução possível para tarefas repetitivas
- sintaxe: `ssh user@exemplo.pt comando`

Executar o comando `ls` em vários servidores

Script

```
#!/bin/sh
for pc in servidor1 servidor2 servidor3 servidor4
do
  ssh $pc ls
done
```

Exercícios

1. Login remoto com `SSH`
   - ligue-se via `SSH` ao computador do seu colega
   - verifique o endereço `IP`
   - termine a ligação escrevendo `exit`
2. Cópia segura de ficheiros
   - crie uma directoria com 3 ou mais ficheiros `.txt`
   - copie os ficheiros para o computador do seu colega com `scp`
   - repita o passo anterior usando o `sshfs`
   - no final remova o `mount` que fez com o `sshfs`
3. Execução remota de comandos
   - execute o comando `df -H` no computador do seu colega
   - sem criar uma sessão interativa

Port Forwarding

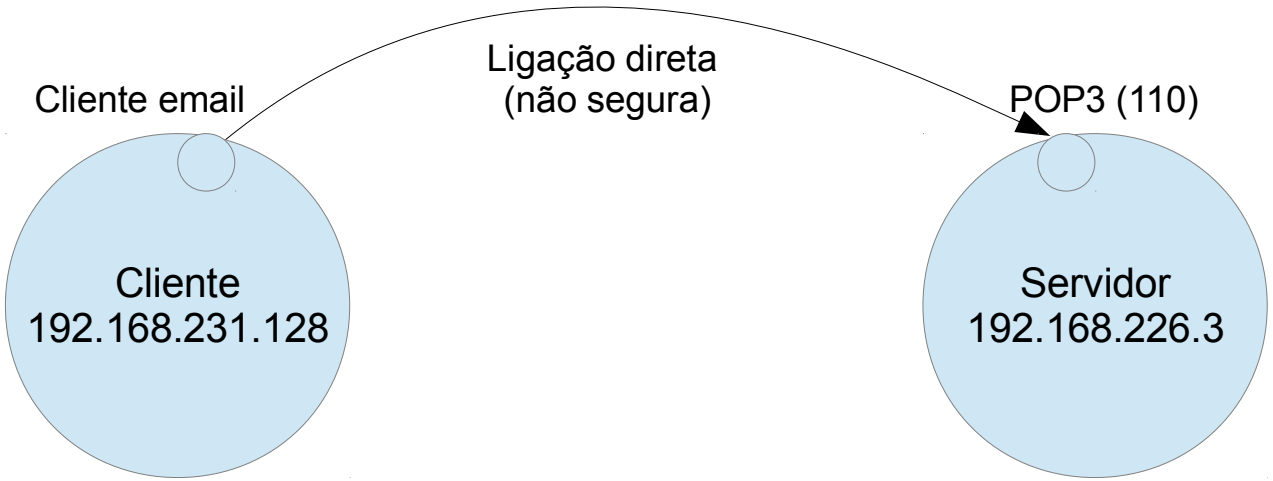
Reencaminhamento

- também é possível aumentar a segurança de aplicações TCP/IP não seguras
- criando um túnel seguro específico para a aplicação a segurar
- é necessário:
  - o serviço a reencaminhar tem de ser TCP e só pode usar um porto para funcionar
  - o servidor ter o serviço `SSH` e ter conta nesse serviço
  - criar o túnel `SSH`
  - alterar a configuração do cliente para usar o túnel

Port Forwarding

Exemplo de acesso ao `POP3` sem túnel

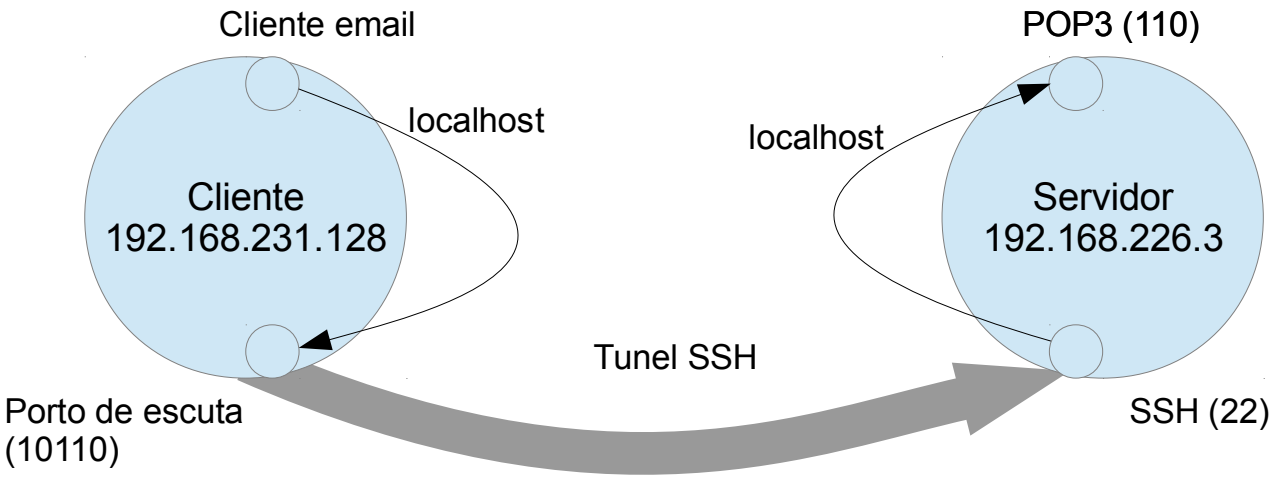
- configuração do cliente de email
  - IP = `192.168.226.3`
  - Porto = `110`



Port Forwarding

Exemplo de acesso ao `POP3` com túnel `SSH`

- criação do túnel (a executar no cliente):
  - `ssh -N -L10110:localhost:110 user@192.168.226.3`
- configuração do cliente de email
  - IP = `localhost`
  - Porto = `10110`



Port Forwarding

O que siginificam os parâmetros

- `ssh -N -L10110:localhost:110 user@192.168.226.3`
  - `-N` sessão não interativa, que não recebe comandos
  - `-L` fica à escuta num porto local
  - `10110` porto escolhido para ficar à escuta no cliente
  - `localhost:110` para onde reencaminhar quando chegar ao servidor
  - `user@192.168.226.3` autenticação `SSH` no servidor

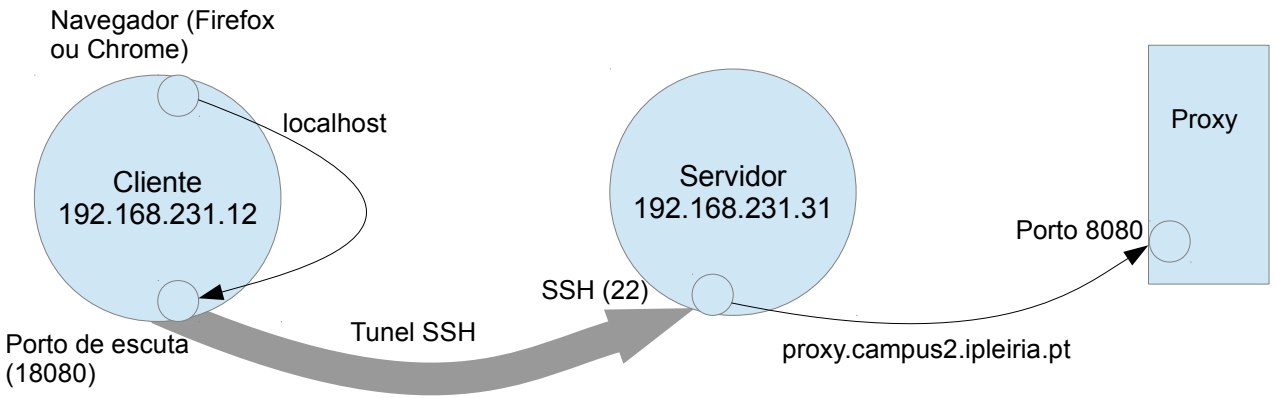
Port Forwarding

O mesmo túnel pode ser criado com um comando executado no servidor. Nesse caso ficaria:

- `ssh -N -R10110:localhost:110 user@192.168.231.128`
  - `-N` sessão não interativa, que não recebe comandos
  - `-R` fica à escuta num porto remoto (no cliente)
  - `10110` porto escolhido para ficar à escuta no cliente
  - `localhost:110` para onde reencaminhar no servidor
  - `user@192.168.231.128` autenticação `SSH` no computador cliente

Exercício

1. configure o seu navegador para usar o proxy `proxy.campus2.ipleiria.pt`, teste a configuração
2. crie um túnel local `SSH` de acordo com a figura (substituir os endereços `IP` pelo seu e do seu colega)
   - altere a configuração do navegador para usar o túnel
   - abra uma página web para testar o túnel
   - termine o túnel e abra outra página web, o que acontece?
3. crie o mesmo túnel `SSH`, mas agora de forma remota



Autenticação de servidores

Os servidores SSH são autenticados, na 1ª vez pergunta:

```
mfrade@kubuntu:~$ ssh 192.168.226.3
  The authenticity of host '192.168.226.3' can't be established.
  ECDSA key fingerprint is
      80:37:7c:fd:40:5f:8f:fa:c8:99:9b:55:db:ac:65:7a.
  Are you sure you want to continue connecting (yes/no)? yes
```

- adiciona a chave do servidor SSH ao ficheiro `~/.ssh/known_hosts`
- serve para evitar ataques *man-in-the-middle*
- listar chaves armazenadas:
  - `ssh-keygen -l -f ~/.ssh/known_hosts`
  - ou `ssh-keygen -H -F` *nome_ou_IP*
- apagar chaves armazenadas:
  - `ssh-keygen -R` *nome_ou_IP*

Autenticação assimétrica

O SSH permite fazer autenticação sem usar senha:

1. usa chaves assimétricas: K_{U_A} e K_{R_A}, que são geradas com o seguinte comando:
   - `ssh-keygen -t rsa`
   - que gera os ficheiros `~/.ssh/id_rsa` e `~/.ssh/id_rsa.pub`
2. a chave pública (K_{U_A}) tem de ser copiada para o servidor com seguinte comando
   - `ssh-copy-id user@192.168.226.3`
3. a chave privada (K_{R_A}) tem de ser colocada em memória com o comando `ssh-add`
   - pede a *passphrase*
4. depois desta configuração o SSH faz a autenticação com chaves assimétricas e deixa de pedir a senha

Exercício

1. Configure autenticação assimétrica para se ligar via `SSH` ao computador do seu colega
2. Teste a configuração:
   - faça *login* remoto
   - use o `scp` ou o `sshfs`
3. Edite o ficheiro `~/.ssh/authorized_keys` e apague a linha correspondente à chave do seu colega
4. Teste novamente

Bibliografia

- `man ssh`
- `man scp`
- `man sshfs`
- `man ssh-keygen`
- Daniel J. Barret e Richard E. Silverman, "*SSH, The Secure Shell – the definitive guide*", Fevereiro de 2001, O'Reilly