2. Acesso Remoto com OpenSSH

2.1 Introdução

Este laboratório tem por objetivo realizar a instalação, configuração e apresentação dos possibilidades de utilização do OpenSSH que é uma implementação open source do SSH, alternativo ao código proprietário da suíte de softwares Secure Shell, oferecido pela SSH Communications Security.

Para execução deste laboratório, criaremos o usuário **aluno1** nos *hosts* Debian-Server e Debian-Client:

```
root@Firewall:/# adduser aluno1
root@client:/# adduser aluno1
```

2.2 Instalando e configurando o OpenSSH-Sever

2.2.1 Instalação do OpenSSH-Server

Para instalar o OpenSSH-Server utilizaremos o gerenciador de pacotes APT, para isso execute os seguintes comandos:

```
root@Firewall:/# apt-get update
root@Firewall:/# apt-get install openssh-server
```

2.2.2 Configuração do OpenSSH-Server

Com o OpenSSH-Server devidamente instalado, a primeira tarefa que realizaremos é entrar no seu diretório de configuração:

```
root@Firewall:/# cd /etc/ssh
```

Dentro deste diretório, existe dois arquivos principais para configuração do serviço:

- ***sshd_config***: Arquivo de configuração do Servidor OpenSSH.
- ***ssh_config***: Arquivo de configuração do cliente OpenSSH.

Vamos então, editar o arquivo de configuração do Servidor:

```
root@Firewall:/# vim /etc/ssh/sshd_config
```

Dentro deste arquivo, você encontrará várias linhas que dizem respeito aos arquivos de criptografia de serviço, o que não é muito aconselhável de trocar, então, abordaremos as principais linhas, você pode alterar as linhas para testar suas funções:

Port 22 - Esse valor é padrão, caso queira alterar a porta onde o SSH vai trabalhar. Lembre-se que, se caso alterada a porta, será necessário toda vez que for iniciar uma conexão com o Servidor, indicar em qual porta se conectar.

Exemplo:

```
$ ssh usuario@ip –p porta
$ ssh dorneles@172.16.1.1 –p 2210
```

Protocol 2 - Protocolos aceitos pelo Servidor. O valor já vem padrão nas versões recentes.

LoginGraceTime 60 - Essa linha indica o tempo limite, em segundos, que ele terá para fazer o Login após indicar que deseja iniciar uma conexão SSH com o Servidor. Nesse caso: sessenta segundos.

PermitRootLogin

- **without-password / prohibit-password**: Se esta opção estiver definida, o login por autenticação via senha não será permitida para o root, porém, essa opção permite o login do root via autenticação por chave pública-privada (chave de confiança).
- **no**: Se definido “no”, esta opção não permitirá o login diretamente com root. A ideia é fazer com que a conexão seja feita com usuário normal, e em seguida já na máquina remota, seja invocado o usuário root. Isso dificulta a vida de quem quer acessar de forma indevida.
- **yes**: Se definido “yes”, esta opção permitirá o login diretamente com root.

AllowUsers dorneles ademir - Caso você não queira liberar o SSH para todo mundo, você pode utilizar esta opção e indicar quais usuários podem se conectar via SSH. Caso esta linha não esteja em seu arquivo, fique à vontade para adicioná-la. Neste exemplo, estou permitindo acesso apenas aos usuários dorneles e ademir.

DenyUsers dorneles - Essa linha faz o oposto da anterior e bloqueia o acesso SSH para um usuário, no exemplo o usuário dorneles não conseguira se autenticar no OpenSSH-Server.

PermitEmptyPasswords no - Esta linha permite, ou não, que o SSH aceite senhas vazias. Utilize o padrão ’no’, para garantir uma segurança maior.

ListenAddress 0.0.0.0 - Especifica o endereço IP das interfaces de rede que o servidor sshd servirá requisições. Múltiplos endereços podem ser especificados separados por espaços. A opção **Port** deve vir antes desta opção.

Banner /etc/issue.net - Se você quer exibir uma mensagem antes do Prompt de Login, a mensagem é especificada através desta linha. Você pode utilizar esta funcionalidade para listar algumas regras de uso do SSH, por exemplo.

X11Fowarding no - Esta linha indica se o Servidor permitirá que os aplicativos executem aplicativos gráficos remotamente. Dependendo da sua velocidade de Upload, isso pode ser um problema. Deixe como 'no'.

Fique à vontade para alterar e realizar os testes.

Após adicionar ou alterar linhas no arquivo sshd_config, salve-o e reinicie o serviço SSH:

```
root@Firewall:/# systemctl restart ssh.service
```

Para verificar se o serviço está rodando corretamente:

```
root@Firewall:/# systemctl status ssh.service
```

Faça algumas verificações para testar se o serviço está rodando, e se está escutando na porta correta:

```
root@Firewall:/# ps aux | grep ssh
root@Firewall:/# nmap localhost -p 22
```

Observação:

Para utilizar a ferramenta *Nmap* é necessário instalar o pacote referente a esse serviço:
root@Firewall:/# apt-get install nmap

2.2.3 Testando o acesso ao servidor Firewall via SSH

Via prompt de comando Linux (VM Client)

Teste o funcionamento do OpenSSH-Server acessando a máquina "Client" com o usuário aluno1 e a partir dela executando o seguinte comando:

```
root@client:/# ssh aluno1@172.16.1.1
```

Observação:

Caso você tenha alterado o parâmetro referente à porta.
root@client:/# ssh aluno1@172.16.1.1 –p <porta>

Via software Putty (estação hospedeira)

Nesta atividade realizaremos o acesso ao servidor a partir do sistema hospedeiro (Windows 8) das estações de trabalho do laboratório.

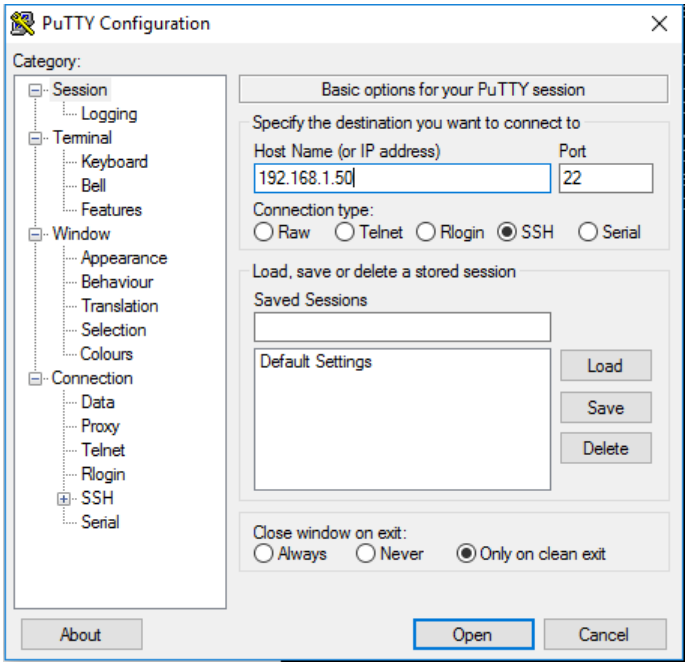
Para isso será necessário verificar o endereço IP entregue pelo DHCP da rede do Senac para a

sua VM Debian Firewall. O endereço IP que será utilizado é o da interface enp0s3 definida como bridge. Utilize um dos comandos listados abaixo:

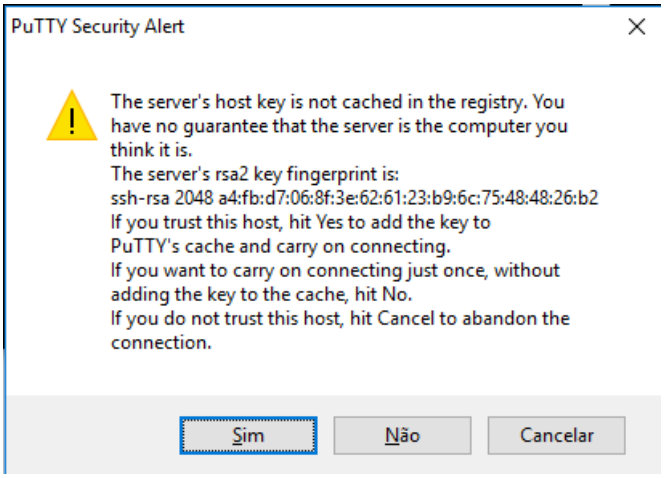
> ***root@Firewall:/# ip addr show dev enp0s3***

Copie do diretório "publico" da rede o utilitário Putty disponibilizado pelo professor, ou faça o download a partir do seguinte endereço https://www.chiark.greenend.org.uk/ sgtatham/putty/latest.html.

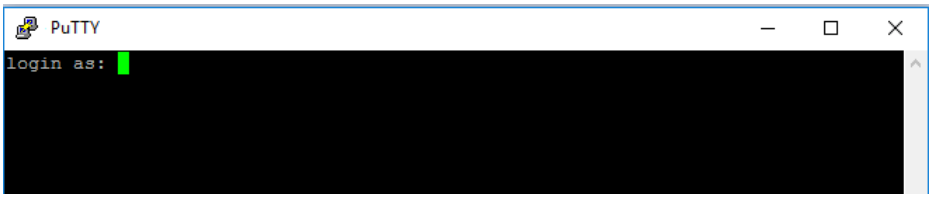
Execute o utilitário e preencha as informações de acordo com o seguinte exemplo:



Assim que aparecer a chave enviada pelo servidor, aceite-a clicando em "sim".

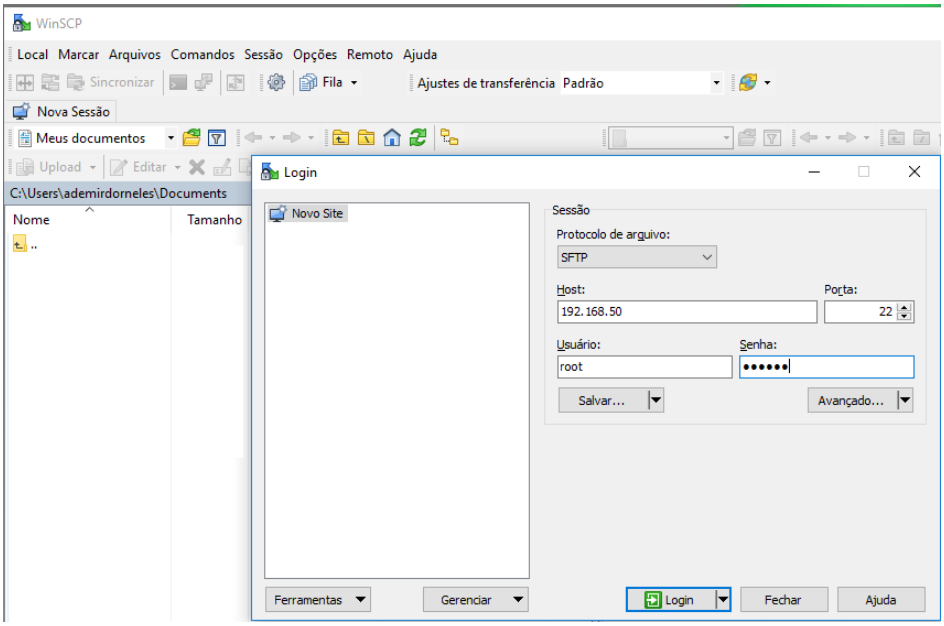


No prompt aberto, digite o usuário e senha para acesso ao servidor:

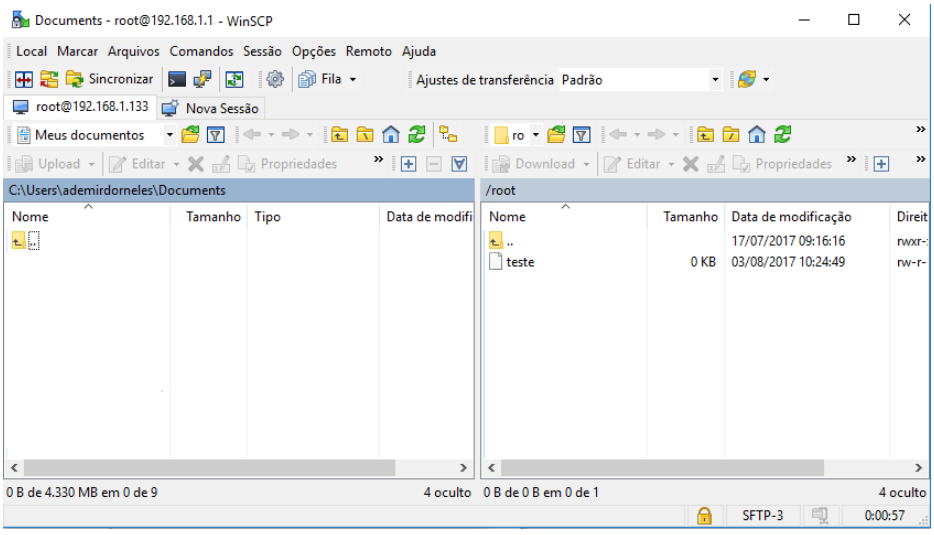


Via softawre WinSCP (estação hospedeira)

Abra o WinSCP a partir do ícone disponível na área de trabalho e preencha as informações de acordo com o seguinte exemplo:



Após aceitar a chave enviada pelo servidor, abrirá a janela (conforme figura abaixo) onde no lado direito poderão ser acessados os arquivos do servidor e o lado esquerdo os arquivos da máquina local.



2.3 Copiando arquivos com o comando SCP

Para copiar arquivos entre as máquinas local e remota (debian-Firewall e debian-client), utilizamos a mesma lógica do comando "*cp*", porém no SSH utilizamos o "*scp*".

Sintaxe do comando:

```
# scp <origem> <destino>
```

Vamos realizar a tarefa na prática então, na máquina Debian-Firewall, crie o arquivo **teste.txt** no diretório */home* do usuário aluno1, e altere as permissões de acesso para r(leitura) e w(escrita) para qualquer usuário do sistema:

```
root@Firewall:/# touch /home/aluno1/teste.txt
root@Firewall:/# chmod 666 /home/aluno1/teste.txt
```

Efetue logon na máquina Debian-Client com o usuário aluno1. Lembre-se que agora, a origem ou o destino, podem ser remotos. Por exemplo:

- Copiando um arquivo da máquina remota (Debian-Firewall) para a máquina local (Debian-Client):

  ```
  root@client:/# scp aluno1@172.16.1.1:/home/aluno1/teste.txt /home/aluno1
  ```

  Verifique se foi realizada com sucesso a cópia do arquivo:

  ```
  root@client:/# ls /home/aluno1
  ```

- Enviando um arquivo da máquina local (Client) para a máquina remota (Firewall):

  ```
  root@client:/# scp /home/aluno1/teste.txt aluno1@172.16.1.1:/tmp
  ```

2.4 Conectando o servidor utilizando SFTP

Para se conectar a um servidor usando o sftp, o comando é:

```
root@client:/# sftp usuario@servidor
```

Caso o servidor OpenSSH estiver configurado para escutar em uma porta diferente da 22, é preciso indicar a porta no comando, incluindo o parâmetro "-o port=", como no exemplo abaixo:

```
root@client:/# sftp -o port=2222 usuario@servidor
```

Após efetuar a autenticação, o usuário poderá executar alguns comando possíveis para o novo prompt que foi aberto:

Comando "***put***" = usado para realizar upload de um arquivo da máquina local para a remota.

Comando "***get***" = usado para realizar download de um arquivo da máquina remota para a máquina local.

Para navegar na estrutura de diretórios da máquina remota (servidor), use os comandos "cd

diretorio/" (para acessar o diretório), "cd .." (para subir um diretório), "ls" (para listar os arquivos) e "pwd" (para visualizar o diretório corrente, atual). Veja um exemplo na tabela a seguir:

```
aluno1@client:~$ sftp -o port=2222 aluno1@172.16.1.1
Connecting to servidor ...
Password: ********
sftp>ls
teste.txt
sftp>get aula1.txt (copia arquivo da máquina remota (Firewall) para a máquina local (Client))
```

Executar comandos na máquina local:

Comando "lcd" (local cd): permite trocar o diretório local

Comando "lls" (local ls): permite listar conteúdo do diretório local

Comando "lmkdir" (local mkdir): permite criar diretório local

Comando "lpwd" (local pwd): permite verificar qual é o diretório local de trabalho

2.5 Acessando o servidor com chave de autenticação

Agora que a máquina virtual cliente já está comunicando com o servidor, é necessário pensar na segurança desta informação que está sendo transmitida entre ambos. Para isso, trabalharemos com criptografia assimétrica com chave pública e privada.

Na máquina Debian-Firewall, abra uma sessão com o usuário aluno1 e crie o diretório *.ssh* dentro do diretório */home/aluno1*, o ponto na frente indica que o diretório é oculto.

```
aluno1@Firewall:/$ mkdir /home/aluno1/.ssh
```

Na máquina debian-client, abra uma seção com o usuário aluno1 e crie a chave:

```
aluno1@client:/$ cd /home/aluno1
aluno1@client:/$ ssh-keygen -t rsa
```

Este comando vai gerar os arquivos "id_rsa" e "id_rsa.pub", dentro do seu diretório ".ssh" da home do usuário.

A chave pública, deve ser enviada então para a máquina Debian-Firewall:

```
aluno1@client:/$ scp ~/.ssh/id_rsa.pub aluno1@172.16.1.1:~/.ssh/authorized_keys
```

Para enviar a chave de forma mais prática pode ser utilizado o comando *ssh-copy-id*:

```
aluno1@client:/$ ssh-copy-id 172.16.1.1
```

Após o envio, faça o teste de acesso ssh ao servidor, se não ocorreu problemas na criação e envio da chave, durante o acesso não solicitará senha.

```
aluno1@client:/$ ssh aluno1@172.16.1.1
```