

Laboratório SSH / FTP

Por Lucca Loyale Pinho Vilela

SSH X TELNET

- Apesar de ambos serem protocolos de comunicação usados para acessar dispositivos remotos e permitirem que usuários se conectem e executem comandos nestes mesmos, a principal diferença entre SSH e TELNET é que enquanto o SSH garante a integridade dos dados utilizando criptografia e autenticação por meio de chaves, o Telnet não possui essa mesma autenticação e utiliza senhas simples. Além disso é importante ressaltar que o SSH foi criado por Tatu Ylönen em 1995, e o protocolo Telnet em 1969 pelas forças armadas americanas.

Acessando o servidor SSH no Windows com o Bitvise

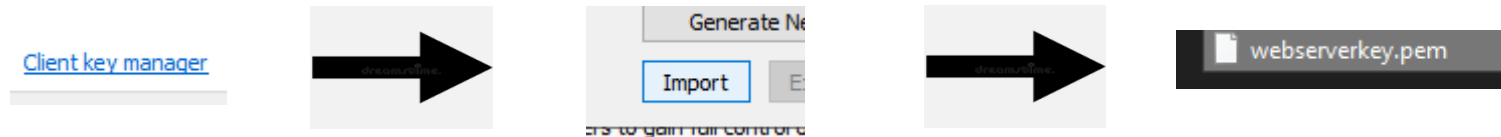
Para acessar a minha máquina virtual da AWS por meio do SSH utilizando o Bitvise pela primeira vez preciso seguir esses passos:

1 – Colocar o ip público da instância da AWS, a porta do SSH (22) e o usuário padrão (ubuntu), e colocar o método de autenticação como “publickey” (chave pública)

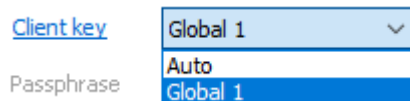
Server		Authentication	
Host	<input type="text" value="52.23.185.219"/>	Username	<input type="text" value="ubuntu"/>
Port	<input type="text" value="22"/>	Initial method	<input type="text" value="publickey"/>
<input type="checkbox"/> Enable obfuscation			

Acessando o servidor SSH no Windows com o Bitvise

2 – Agora, é preciso clicar no client key manager e importar a chave gerada no site da AWS (arquivo .pem / .ppk) .

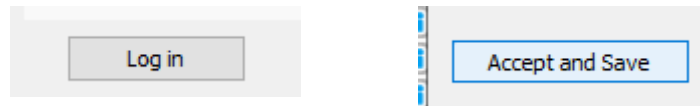


3 – Após isso, mude a client key para essa chave, no meu caso, é a primeira chave, então, global 1 (você também pode modificar o nome das suas chaves, se desejar) .

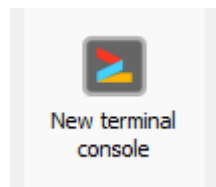


Acessando o servidor SSH no Windows com o Bitvise

5 – Agora, é preciso clicar em “log in” e “accept and save”.



6 – Agora, clique em “new terminal console” do lado, e então já teremos acesso à essa máquina por meio de SSH !



```
The list of available updates is more than a week old.  
To check for new updates run: sudo apt update  
  
The programs included with the Ubuntu system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.  
  
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by  
applicable law.  
  
To run a command as administrator (user "root"), use "sudo <command>".  
See "man sudo_root" for details.  
  
ubuntu@ip-172-31-31-254:~$
```

Mudando para o usuário root e criando um usuário

Para criar um usuário você pode escrever o comando “sudo su” para executar os próximos comandos como usuário root. Após isso digite “adduser nomedousuario”, e digite sua senha. Também serão pedidas mais informações, como nome completo, número de telefone, etc. Mas deixe estas outras informações em branco por agora . Após isso já estará criado o novo usuário !

```
ubuntu@ip-172-31-31-254:~$ sudo su
root@ip-172-31-31-254:/home/ubuntu# adduser lucca
Adding user `lucca' ...
Adding new group `lucca' (1001) ...
Adding new user `lucca' (1001) with group `lucca' ...
Creating home directory `/home/lucca' ...
Copying files from `/etc/skel' ...
New password:
```

Acessando os arquivos de configuração do ssh

Para acessar o diretório desses arquivos, utilize o comando cd:

```
cd /etc/ssh/
```

E o comando ls para lista-los.

```
root@ip-172-31-31-254:/etc/ssh# ls
moduli          ssh_host_dsa_key      ssh_host_ecdsa_key.pub  ssh_host_rsa_key      sshd_config
ssh_config      ssh_host_dsa_key.pub  ssh_host_ed25519_key    ssh_host_rsa_key.pub  sshd_config.d
ssh_config.d    ssh_host_ecdsa_key    ssh_host_ed25519_key.pub ssh_import_id
root@ip-172-31-31-254:/etc/ssh#
```

Utilizando o comando “cat” podemos ler seu conteúdo.

Ex: cat sshd_config ou cat ssh_config

SSHD_CONFIG X SSH_CONFIG

A diferença entre estes dois arquivos é que o `sshd_config` é usado para configurar o servidor SSH, como por exemplo, sua segurança.

Já o `ssh_config` configura o cliente SSH em relação às conexões que ele realiza, como a sua autenticação e criptografia.

Bloqueando o usuário root

Para bloquear o root temos que editar o `sshd_config` com “`vim sshd_config`”, apertar insert e editar esta parte para não permitir seu login:

```
# Authentication:  
  
#LoginGraceTime 2m  
PermitRootLogin no  
#StrictModes yes  
#MaxAuthTries 6  
#MaxSessions 10
```

Após a edição no vim, aperte esc e escreva `:wq` para salvar e sair, depois disso reinicie o servidor SSH com `systemctl restart ssh`.

Permitir acesso ao servidor apenas com o meu usuário

No mesmo arquivo, colocamos o trecho “AllowUsers nomedousuario” para permitir apenas ele.

```
#MaxSessions 10  
AllowUsers lucca  
#PubkeyAuthenticat
```

Após isso, é necessário salvar, sair e reiniciar o servidor ssh com os mesmos comandos do slide anterior.

Utilizando o comando SCP

Irei utilizar o comando SCP para pegar uma chave do meu usuário dentro da minha máquina remota. Primeiramente, preciso descomentar estas linhas no mesmo arquivo de configuração sshd_config. E depois reinicie o servidor ssh.

```
PubkeyAuthentication yes
```

```
# Expect .ssh/authorized_keys2 to be disregarded by default in future
AuthorizedKeysFile      .ssh/authorized_keys .ssh/authorized_keys2
```

```
# To disable tunneled clear t
PasswordAuthentication yes
#PermitEmptyPasswords no
```

Utilizando o comando SCP

Agora, dentro do diretório do meu usuário eu crio uma pasta `.ssh`, e dentro dela um arquivo `authorized_keys`. Lembrando que para criar esta chave eu preciso estar no usuário selecionado para ela, então utilizei o comando `“su lucca”` para mudar para ele.

```
root@ip-172-31-31-254:/home/lucca# su lucca
lucca@ip-172-31-31-254:~$ ls
```

```
mkdir .ssh
```

```
lucca@ip-172-31-31-254:~$ cd .ssh/
lucca@ip-172-31-31-254:~/.ssh$ touch authorized_keys
```

Utilizando o comando SCP

Então utilizo o comando ssh-keygen para gerar a chave e escolher um nome.

```
lucca@ip-172-31-31-254:~/.ssh$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/lucca/.ssh/id_rsa): chavelucca
Enter passphrase (empty for no passphrase):
```

E então copio minha chave e colo nas chaves autorizadas.

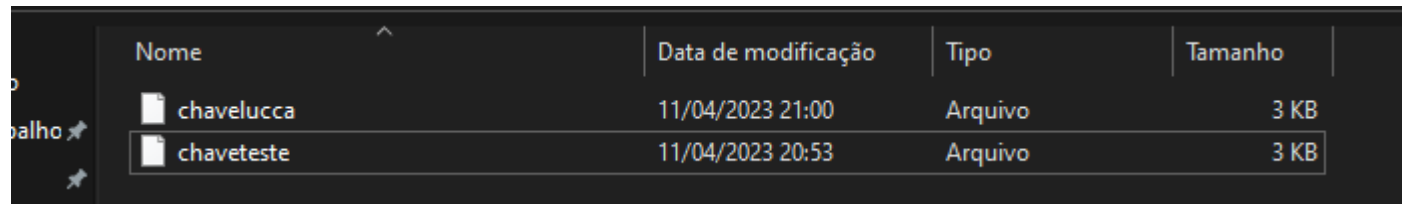
```
authorized_keys chavelucca chavelucca.pub
lucca@ip-172-31-31-254:~/.ssh$ cat chavelucca.pub > authorized_keys
```

Utilizando o comando SCP

E então, utilizo esse comando para copiar minha chave para o Windows, e digito minha senha:

```
PS C:\Users\PlugN'Play\userssh> scp lucca@52.23.185.219:/home/lucca/.ssh/chavelucca .  
lucca@52.23.185.219's password:
```


E assim, a chave está no meu diretório userssh



Nome	Data de modificação	Tipo	Tamanho
chavelucca	11/04/2023 21:00	Arquivo	3 KB
chaveteste	11/04/2023 20:53	Arquivo	3 KB

Utilizando o comando SCP

Agora, para mandar por exemplo uma pasta com o arquivo nomes:

Nome	Data de modificação	Tipo	Tamanho
 nomes.txt	11/04/2023 21:02	Documento de Te...	1 KB

Eu utilizo esse comando:

```
PS C:\Users\PlugN'Play\userssh\teste> scp -r * lucca@52.23.185.219:/home/lucca/
```

Lembrando que apenas utilizei o * pois eu já estava dentro de uma pasta que eu queria mandar para a máquina remota. Assim, os arquivos já apareceram:

```
root@ip-172-31-31-254:/home/lucca# ls
teste
root@ip-172-31-31-254:/home/lucca# cd teste/
root@ip-172-31-31-254:/home/lucca/teste# ls
nomes.txt
root@ip-172-31-31-254:/home/lucca/teste# cat nomes
cat: nomes: No such file or directory
root@ip-172-31-31-254:/home/lucca/teste# cat nomes.txt
lucca
lucca
```

Pergunta LPI

Qual arquivo de configuração você precisará editar para alterar as opções padrões do cliente SSH?

A. /etc/ssh/sshd_config

B. /etc/ssh/ssh_client

C. /etc/ssh/client

D. /etc/ssh/ssh

E. /etc/ssh/ssh_config

- A resposta certa é a letra E, como vimos antes, ali estão as opções padrões do cliente.

Pergunta LPI

Qual parâmetro no arquivo configuração do SSH define os usuários que podem se logar no servidor?

- A. AllowUsers
- B. DenyUsers
- C. AllowUser
- D. UsersAllow
- E. UsersDeny

Resposta A, AllowUsers.

Servidor FTP

Para instalar o proftpd, utilize:

```
5 sudo apt update
6 sudo apt-cache search ftpd
7 sudo apt install proftpd -y
```

E no powershell, faça:

```
PS C:\Users\PlugN'Play\ftp> ftp 54.173.37.181
Conectado a 54.173.37.181.
220 ProFTPD Server (Debian) [::ffff:172.31.81.98]
200 UTF8 set to on
Usuário (54.173.37.181:(none)): teste
331 Password required for teste
Senha:
230 User teste logged in
```

Fazendo download e upload

No powershell, utilizo esse comando(get) para pegar um arquivo texto:

```
ftp> get nomesdois.txt
200 PORT command successful
150 Opening ASCII mode data connection for nomesdois.txt (29 bytes)
226 Transfer complete
ftp: 31 bytes recebidos em 0.00Segundos 31.00Kbytes/s.
ftp>
```

E para mandar um arquivo para a máquina remota uso o put.

```
ftp> put C:\Users\PlugN'Play\ftp\upload.txt
200 PORT command successful
150 Opening ASCII mode data connection for upload.txt
226 Transfer complete
ftp: 8 bytes enviados em 0.15Segundos 0.05Kbytes/s.
ftp>
```

Assim, o arquivo já aparece no local:

```
root@ip-172-31-81-98:/home/teste# ls
nome.txt  nomesdois.txt  upload.txt
```

Limitar acesso por horário

Para isso, acessamos `/etc/proftpd/proftpd.conf` na máquina remota, e editamos esse arquivo de texto, colocando este comando abaixo, limitando o horário de acesso das 18:00 às 23:59

```
time_allow_anon_writable 1800-2359
```

Após isso, reiniciar com `systemctl restart proftpd`.

Utilizando regras PAM

Primeiro instalamos o pacote PAM:

```
root@ip-172-31-81-98:/etc/proftpd# sudo apt-get install libpam-pwdfile  
Reading package lists... Done
```

E depois, no mesmo arquivo proftpd.conf, colocamos a seguinte condição:

```
AuthPAMConfig proftpd
```

Bloquear usuário fatec

Para bloquear o usuário fatec, voltamos em `/etc/` e editamos por texto o arquivo `ftpusers`, e dentro dele colocamos o nome do usuário que não queremos que acesse.

```
# /etc/ftpusers: list of users disallowed FTP access. See ftpusers(5).
root
daemon
bin
sys
sync
games
man
lp
mail
news
uucp
nobody
fatec
~
~
```