Administração de redes

Servidor OpenSSH

O servidor OpenSSH que utiliza os protocolos SSH1 e SSH2(mais seguro) é o substituto dos comandos rlogin, rcp e telnet, e utiliza autenticação criptografada entre dois computadores através do arquivo /etc/passwd.

O servidor OpenSSH possue um daemon servidor chamado sshd, comandos para computadores clientes ssh e scp e o comando ssh-keygen para gerar chaves criptografadas.

Instalação

Pacotes necessários para instalação do servidor OpenSSH no Debian

```
# apt-get install openssh
```

Pacotes necessários para instalação do servidor OpenSSH no Fedora.

```
# yum install openssh
```

Arquivos de configuração

O principal arquivo de configuração do servidor openssh é o arquivo /etc/ssh/sshd_config e do cliente /etc/ssh/ssh_config, onde é gerado na instalação em conjunto com as chaves públicas criptografadas /etc/ssh/ssh_host_dsa_key e /etc/ssh/ssh_host_rsa_key.

Existem os arquivos para autenticação de hosts remotos e de chaves públicas criptografadas:

Hosts remotos

- /home/usuario/.rhosts
- /home/usuario/.shosts
- /etc/hosts.equiv
- /etc/shosts.equiv

Chaves públicas criptografadas

- /home/usuario/.ssh/known_hosts
- /etc/ssh_known_hosts

Existem várias formas de autenticar logins de usuários pelo daemon sshd:

1º Forma	Por login remoto através dos arquivos /home/usuario/.rhosts, /home/usuario/.shosts, /etc/hosts.equiv ou /etc/shosts.equiv, no qual o usuário é autenticado automaticamente sem verificar uma senha, o que torna muito similar ao antigo rlogin o que torna inseguro é não é recomendado.

2º Forma	O sshd usa criptografia de chaves públicas para identificar o host remoto, portanto a chave pública do host remoto é definida no arquivo /etc/ssh_known_hosts (que não pode ser lido por todos) do host local e no arquivo /home/usuario/.ssh/known_hosts(~/.ssh/known_hosts) do usuário. Caso o host remoto possa comprovar a chave pública, então usuário é autenticado sem verificar uma senha, portanto se a segurança do host de local estiver comprometida o host remoto também estará.
3º Forma	É o metodo mais seguro e mais chato de configurar, por que para entrar no sistema é requerida uma senha e ter uma cópia da chave pública que terá que estar sempre com o usuário onde ele estiver gravada em disquete, CD ou no seu Laptop.
4º Forma	Esse método é o mais simples, onde o usuário digita um login e senha usando uma conexão criptografada para entrar no sistema, portanto é o mais recomendado.

O arquivo /etc/ssh/sshd_config contém opções relacionadas a autenticação exibidas na tabela abaixo.

Opção	Forma	Descrição
RhostsRSAAuthentication	2º Forma	Permite login através dos arquivos /home/usuario/.shosts, mas requer chave pública
IgnoreRhosts	1º Forma, 2º Forma	Ignora os arquivos ~/.rhosts e /etc/hosts.equiv, e continua usando os arquivos ~/.shosts e e /etc/shosts.equiv.
RSAAuthentication	3º Forma	Permite logins de usuários com chaves públicas criptografadas por usuário.
PasswordAuthentication	4º Forma	Permite login normal

Abaixo segue trecho de configuração do arquivo /etc/ssh/sshd_config recomendada usando as formas 3º e 4º .

```
# /etc/ssh/sshd_config
RSAAuthentication   yes
IgnoreRhosts    yes
RhostsRSAAuthentication   no
PasswordAuthentication   yes
```

Configuração do arquivo sshd_config

Por padrão o arquivo /etc/ssh/sshd_config já vem com algumas linhas descomentadas, portando edite as seguintes linhas caso seja necessário da seguinte maneira:

```
# /etc/ssh/sshd_config
Port 22      # porta do ssh
Protocol 2      # versão do protocolo do ssh

# chaves criptografadas do protocolo versão 2 do ssh
HostKey /etc/ssh/ssh_host_rsa_key
HostKey /etc/ssh/ssh_host_dsa_key

UsePrivilegeSeparation yes      # previlégios para tornar seguro
```


```
# tempo e tamanho gerado da chaves do servidor na versão 1 do protocolo ssh
KeyRegenerationInterval 3600
ServerKeyBits 768

# opções de login
SyslogFacility AUTH
LogLevel INFO

# opções de autenticação
LoginGraceTime 120
PermitRootLogin yes      # permite login de root
StrictModes yes

# permite login com chaves públicas critptografadas
RSAAuthentication yes
PubkeyAuthentication yes

#AuthorizedKeysFile     %h/.ssh/authorized_keys      # arquivo de chaves dos computadores que podem ter acesso

IgnoreRhosts yes      # ignora os arquivos ~/.rhosts e ~/.shosts

# para trabalhar com esta opção como yes, você precisa da chave criptografada em /etc/ssh_known_hosts
RhostsRSAAuthentication no

HostbasedAuthentication no

PermitEmptyPasswords no      # não permite senha vazias

ChallengeResponseAuthentication no

PasswordAuthentication yes      # permite autenticação por senha

# habilita o servidor X através do servdior OpenSSH
X11Forwarding yes
X11DisplayOffset 10
PrintMotd no
PrintLastLog yes
TCPKeepAlive yes

AcceptEnv LANG LC_*

# habilita o servidor sftp
Subsystem sftp /usr/lib/openssh/sftp-server

UsePAM yes
```

Projeto LinuxBRasil
www.linuxbrasil.org.br

Bloqueando acesso de root

Para bloquear acesso ao usuário root altere a linha: PermitRootLogin yes para PermitRootLogin no

```
PermitRootLogin no
```

Bloqueando acesso de ao servidor X

Para bloquear acesso ao servidor X altere a linha: X11Forwarding yes para X11Forwarding no

```
X11Forwarding no
```

Liberando acesso de ao servidor SFTP

Para liberar acesso ao servidor SFTP retire o comentário da linha: # Subsystem sftp /usr/lib/openssh/sftp-server

```
Subsystem sftp /usr/lib/openssh/sftp-server
```

Ao instalar o servidor openssh ele é automáticamente adicionado na inicialização do sistema, mas caso deseje incluí-lo manualmente execute o comando update-rc.d no Debian ou ntsysv no Red Hat / Fedora.

Debian

```
# update-rc.d ssh defaults
```

Red Hat / Fedora

```
# ntsysv
[x] sshd
```

Use a tecla TAB para navegar entre os serviços e a tecla ESPAÇO para ativar e desativar.

Inicializando o servidor sshd

Debian

```
# /etc/init.d/ssh start
```

Red Hat / Fedora

```
# /sbin/service sshd start
```

Testando o servidor openssh

No computador local digite os comandos abaixo:

```
# netstat -nap | grep ":22"
$ telnet localhost 22
```

No computador cliente remoro digite o comando abaixo, onde o IP do servidor é 192.168.1.1:

```
$ ssh -l usuario1 192.168.1.1
The authenticity of host '192.168.1.1 (192.168.1.1)' can't be established.
RSA key fingerprint is 16:04:0d:15:b7:e0:fd:ae:7c:ea:6b:69:55:95:9f:d3.
Are you sure you want to continue connecting (yes/no)? yes     # digite yes para aceitar a chave pública criptografada
Warning: Permanently added '192.168.1.1' (RSA) to the list of known hosts.
aluno1@192.168.1.1's password: **********     # digite a senha
Linux dns1 2.6.18-4-686 #1 SMP Mon Mar 26 17:17:36 UTC 2007 i686

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
You have new mail.
Last login: Sat Jan 19 15:15:41 2008
aluno1@dns1:~$
```

Gerando um chave pública

Para criar uma chave pública criptografada e usar o 3ª Forma de autenticar usuários, onde você precisa levar uma cópia da chave pública do computador remoto para seu computador local e vice-versa, gere a chave e faça um copia pelo comando scp para seu laptop ou grave em disquete ou CD-ROM executando os comandos abaixo.

Execute os comandos abaixo no computador remoto para gerar a chave pública criptografada, apenas apertando a tecla ENTER, que irá gerar os arquivos com as chaves /home/aluno1/.ssh/id_rsa e /home/aluno1/.ssh/id_rsa.pub com o nome de usuário e do computador aluno1@dns1.

Obs: Para fazer os testes o computador remoto possue o endereço IP 192.168.1.1 e a conta de login aluno1 e no computador local possue o endereço IP 192.168.1.2 e a conta de login aluno2.

```
$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/aluno1/.ssh/id_rsa): APERTE ENTER
Enter passphrase (empty for no passphrase): APERTE ENTER NOVAMENTE
Enter same passphrase again: APERTE ENTER NOVAMENTE
Your identification has been saved in /home/aluno1/.ssh/id_rsa.
Your public key has been saved in /home/aluno1/.ssh/id_rsa.pub.
The key fingerprint is:
2c:09:af:34:ff:8f:a9:46:04:3d:b4:ad:6a:73:e8:19 aluno1@dns1
```

Esta chave deverá ser criada tanto no computador local como no computador remoto, e ambas criadas no arquivo /home/aluno1/.ssh/authorized_keys


```
$ cat /home/aluno1/.ssh/id_rsa.pub
```

Copie o arquivo /home/aluno1/.ssh/id_rsa.pub renomeando-o para /home/aluno1/.ssh/authorized_keys

```
$ cp   /home/aluno1/.ssh/id_rsa.pub   /home/aluno1/.ssh/authorized_keys
```

Faça os mesmo passos no computador local, apenas copie com um nome diferente o arquivo /home/aluno1/.ssh/authorized_keys.

```
$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/aluno2/.ssh/id_rsa): APERTE ENTER
Enter passphrase (empty for no passphrase): APERTE ENTER NOVAMENTE
Enter same passphrase again: APERTE ENTER NOVAMENTE
Your identification has been saved in /home/aluno2/.ssh/id_rsa.
Your public key has been saved in /home/aluno2/.ssh/id_rsa.pub.
The key fingerprint is:
39:b7:8b:73:9e:7b:b4:d3:4a:05:39:62:37:dc:a0:a5 aluno2@dns2
```

```
$ cat /home/aluno2/.ssh/id_rsa.pub
```

```
$ cp   /home/aluno2/.ssh/id_rsa.pub   /home/aluno2/.ssh/arquivo_de_chaves
```

Agora no computador local copie a chave para o computador remoto através do comando scp

```
$ scp    /home/aluno2/.ssh/arquivo_de_chaves         aluno1@192.168.1.1:/home/aluno1/.ssh
```

E no computador remoto entre no diretório /home/usuario/.ssh junte as chaves com o comando cat

```
$ cd   /home/aluno1/.ssh/
$ cat    arquivo_de_chaves   >>    authorized_keys
```

Agora copie este novo arquivo authorized_keys do computador remoto para o computador local com o comando scp

```
$ scp    /home/aluno1/.ssh/authorized_keys         aluno2@192.168.1.2:/home/aluno2/.ssh
```

Agora você pode logar entre os computador local e o computador remoto através da chave pública criptografada sem pedir senha.

```
$ ssh   -l aluno1   192.168.1.1
$ ssh   -l aluno2   192.168.1.2
```
