

Tutorial básico sobre SSH

Índice:

1 - Introdução	1
1.1 - SSH (Secure Shell)	1
2 - Instalação do SSH (secure Shell)	2
2.1 - Sistema operacional e software usado	2
2.2 - Requisitos para instalar o software	2
2.3 - Instalação do software SSH	3
2.4 - Instalação do OpenSSH	12
3 - Soluções de acesso remoto encriptado	19
3.1 - Server	19
3.2 - Clientes	21

1 - Introdução

Nesta documentação utilizaremos o SSH (Secure Shell) para acessos remotos substituindo serviços que apresentam problemas graves de segurança como o FTP, Telnet, POP3, X11 e outros. Estes serviços não possuem nenhum mecanismo de encriptação, facilitando a captura de dados por sniffers.

O SSH (Secure Shell) é muito parecido com o telnet mas possui suporte a criptografia de dados tornando a técnica de sniffer ineficiente, evitando que logins, senhas, cartões de créditos e outros dados sejam grampeados.

O SSH (Secure Shell) possui atualmente várias implementações cliente e servidora, entre elas podemos destacar: ssh , openssh, ossh, sftp (Secure FTP) e outras. Os clientes são distribuídos para quase todas as plataformas existentes e possuem interface gráfica ou trabalham em modo texto. A grande vantagem do SSH (Secure Shell) é o grande suporte que possui: em listas de discussão, profissionais, documentações, News Group e diversas implementações.

Uma outra solução seria o SSLTelnet, pouco usado em função do pequeno suporte que existe para este software.

1.1 - SSH (Secure Shell):

Basicamente o software SSH é Free implementado para suportar os protocolos SSH1 e SSH2, sendo que algumas implementações suportam os dois protocolos simultaneamente como é o caso do openssh na versão 2.3.0 que será apresentada.

As implementações de SSH usam como padrão a porta 22/tcp do TCP/IP, possuem suporte para kerberos, autenticação TIS e socks. Outro ponto positivo é que alguns equipamentos como os roteadores cisco também possuem suporte para ssh ou kerberos.

Entre os pontos responsáveis pela popularidade do SSH podemos destacar : a sua instalação e configuração são muito simples, grande suporte técnico em listas de discussão, estabilidade, escalabilidade, segurança quando bem configurado, suportado por diversos sistemas operacionais (Linux, *BSD, Solaris, IRIX, Digital UNIX, AIX, IBM OS/2, SCO UNIX, HPUX, MAC/OS, Palm, Windows CE, VAX/OPENVMS, BeOS, MS-DOS e outros).

O SSH (www.ssh.fi) é escrito em C ANSI, também possui implementações em java, é um software open source livremente distribuído. No site do SSH (www.ssh.fi) podemos encontrar diversas informações como: relações e correções de Bugs, listas de discussão, novas implementações, acessos aos diversos mirrors pela internet, como comprar clientes SSH com recursos extras e etc.. . Inicialmente abordaremos o SSH1 (www.ssh.fi) e depois o openssh (www.openssh.com) que possui suporte ao ssh1 e ssh2.

O SSH2 possui maior flexibilidade, melhor escalabilidade e uma maior segurança. O SSH2 atualmente é menos usado que o SSH1, como ambos são incompatíveis as instituições/empresas que precisam se comunicar com outras tem que utilizar o SSH1.

Principais recursos para obtermos informações sobre SSH:

Principais sites	http://www.ssh.com http://www.ssh.org http://www.ssh.fi
Lista de discussão	ietf-ssh@clinet.fi
FAQ's	http://www.employees.org/~satch/ssh/faq/ssh-faq.html http://www.tigerlair.com/ssh/faq/ssh-faq.html

2 - Instalação do SSH (secure Shell)

2.1 - Sistema operacional e software usado

Sistema operacional UNIX FreeBSD 2.2.8-RELEASE 32 bits.
Usaremos o software `ssh-1.2.27.tar.gz` versão 1.2.27

2.2 - Requisitos para instalar o software

- Copilador gcc (ANSI), versão 2.7.2.1 ou superior e o GNU make;

Softwares	Site
GCC	ftp://ftp.gnu.org/gnu/make/
Make	ftp://ftp.gnu.org/gnu/make/

- Compactador tar, gunzip e gzip;
- Comandos básicos como: `chmod`, `chown`, `chgrp` e `vi`;
- Software de compactação gzip e gunzip
- Conhecimento básico de UNIX ou LINUX;

- Zlib (necessário no pacote openssh);

Software	Site
ZLib	http://www.freesoftware.com/pub/infozip/zlib/

- OpenSSL 0.9.5a ou maior (necessário no pacote openssh);

Softwares	Site
Ssleay 0.9.0	ftp://ftp.apache-ssl.org/SSLeay/ http://www.apache-ssl.org/ http://www2.psy.uq.edu.au/~ftp/Crypto/ssleay
Openssl 0.9.4	http://www.openssl.org

- PAM, o OpenSSH pode utilizar o PAM (Pluggable Authentication Modules). Isto é opcional.

Software	Site
PAM	http://www.kernel.org/pub/linux/libs/pam/

2.3 - Instalação do software SSH

2.3.1 - Descompactação do software

- Inicialmente estar no diretório onde encontra –se o software compactado:

```
$ cd /usr/local/
```

- Usar o compactador “tar” existente no UNIX

```
$ tar -xvzf ssh-1.2.27.tar.gz
$ chown -R 0 ssh-1.2.27
$ chgrp -R 0 ssh-1.2.27
```

- Entrar no diretório onde o software foi descompactado

```
$ cd ssh-1.2.27
```

```
$PATH=$PATH:/bin:/sbin:/usr/bin:/usr/sbin:/usr/local/bin:/usr/local/sbin:/usr/local/lib:/usr/lib:/usr/local/libexec:/usr/libexec:/usr/X11R6:/usr/X11R6/bin:/usr/include:/usr/local/include
```

2.3.2 - Processo de Instalação

- Executar os “script” que preparará o software de acordo com o ambiente do sistema operacional:

```
$. /configure \  
--prefix=/usr/local/ssh1 \  
--exec-prefix=/usr/local/ssh1 \  
--with-etcdir=/usr/local/ssh1/etc \  
--disable-server-port-forwardings \  
--disable-client-port-forwardings \  
--disable-server-x11-forwarding \  
--disable-client-x11-forwarding \  
--disable-suid-ssh \  
--enable-warnings \  

```

Caso deseje retirar o label da versão do SSH edite:

Para o SSH1 edite version.h

```
#define SSH_VERSION "Versao clean"
```

Obs: Não podemos alterar a versão do protocolo.

Para o SSH2 edite apps/ssh/ssh2version.h

```
#define SSH2_VERSION "Versao clean"
```

Obs: Não podemos alterar a versão do protocolo.

- Iniciar o processo de compilação:

```
$ make
```

- Iniciar o processo de instalação:

```
$ make install
```

2.3.3 - Listagem dos arquivos que compõe a instalação do SSH

Esta listagem mostrará a localização e permissões dos arquivos que compõem o software ssh-1.2.27.

➤ arquivos em /usr/local/ssh1/bin/

```
-rwxr-xr-x 1 root wheel 21234 Sep 13 01:44 make-ssh-known-hosts1
-rwxr-xr-x 1 root wheel 21234 Sep 13 01:44 make-ssh-known-hosts1.old
-rwxr-xr-x 1 root wheel 70291 Sep 13 01:44 scp1
-rwxr-xr-x 1 root wheel 70291 Sep 13 01:44 scp1.old
-rwxr-xr-x 1 root wheel 604365 Sep 13 01:44 ssh-add1
-rwxr-xr-x 1 root wheel 604365 Sep 13 01:44 ssh-add1.old
-rwxr-xr-x 1 root wheel 611038 Sep 13 01:44 ssh-agent1
-rwxr-xr-x 1 root wheel 611038 Sep 13 01:44 ssh-agent1.old
-rwxr-xr-x 1 root wheel 67267 Sep 13 01:44 ssh-askpass1
-rwxr-xr-x 1 root wheel 67267 Sep 13 01:44 ssh-askpass1.old
-rwxr-xr-x 1 root wheel 580160 Sep 13 01:44 ssh-keygen1
-rwxr-xr-x 1 root wheel 580160 Sep 13 01:44 ssh-keygen1.old
-rwx--x--x 1 root wheel 1059554 Sep 13 01:44 ssh1
-rwxr-xr-x 1 root wheel 1059554 Sep 13 01:43 ssh1.old
```

• arquivos em /usr/local/ssh1/sbin/

```
-rwxr-xr-x 1 root wheel 1150579 Sep 13 01:44 sshd1
-rwxr-xr-x 1 root wheel 1150579 Sep 13 01:44 sshd1.old
```

• arquivos em /usr/local/ssh1/etc/

```
-rw-r--r-- 1 root wheel 880 Sep 13 01:43 ssh_config
-rw----- 1 root wheel 525 Sep 13 01:43 ssh_host_key
-rw-r--r-- 1 root wheel 329 Sep 13 01:43 ssh_host_key.pub
-rw-r--r-- 1 root wheel 713 Sep 13 01:43 sshd_config
```

- arquivos em /usr/local/ssh1/etc/

```
-rw-r--r-- 1 root wheel 880 Sep 13 01:43 ssh_config  
(Configuração do ssh cliente )  
-rw----- 1 root wheel 525 Sep 13 01:43 ssh_host_key  
(Chave privada )  
-rw-r--r-- 1 root wheel 329 Sep 13 01:43 ssh_host_key.pub  
(Chave publica )  
-rw-r--r-- 1 root wheel 713 Sep 13 01:43 sshd_config  
(Configuração do ssh server )
```

- O ssh_host_key (chave privada) deve estar com a permissão 700. E os outros com group e owner 0 (zero).

- manuais em /usr/local/ssh1/man/

```
scp1 (1)  
ssh-add1 (1)  
ssh-agent1(1)  
ssh-keygen1 (1)  
ssh1 (1)  
sshd1 (8)
```

- Gerar a chave para de encriptação para o root:

```
$ ssh-keygen  
Initializing random number generator...  
Generating p: .....++ (distance 300)  
Generating q: .....++ (distance 44)  
Computing the keys...  
Testing the keys...  
Key generation complete.  
Enter file in which to save the key (/root/.ssh/identity):  
Enter passphrase:  
Enter the same passphrase again:  
Your identification has been saved in /root/.ssh/identity.  
Your public key is:  
1024 33  
12204154010574373846624727016445974125464900313347803791711436  
39061692552237363182008527147802225325308990680328270193194784
```

```
57540300458993919173353454503296310397555496054636454911974064
08132650849833310148148514001838209857672739639559706840255417
1381032119263698148383855017118827293308030778944617748524827
root@cbpf.br
Your public key has been saved in /root/.ssh/identity.pub
```

- Sugestão para o /usr/local/ssh1/etc/sshd_config.

O arquivo sshd_config possui definições de segurança importantes que serão mostradas abaixo.

O servidor usado como exemplo possui o ip 10.10.10.2/24

```
Port 22
( Porta do ssh )
ListenAddress 10.10.10.2
( endereço ip onde o socket estará funcionando )
HostKey /usr/local/ssh1/etc/ssh_host_key
RandomSeed /usr/local/ssh1/etc/ssh_random_seed
ServerKeyBits 768
( define o número de bits do server )
LoginGraceTime 600
( tempo de espera do sshd até o sucesso do login )
KeyRegenerationInterval 3600
( define o tempo em segundos de regeneração da chave de encriptação,
diminuindo esse tempo podemos evitar que a secção seja capturada e
desencriptada )
PermitRootLogin no
( Não permite o login do root )
IgnoreRhosts yes
( ignora ou não a autenticação do rhosts e shosts )
StrictModes yes
QuietMode no
X11Forwarding no
( habilita o uso de X11 forwarding )
X11DisplayOffset 10
FascistLogging no
( habilita o logging verbose )
PrintMotd yes
( define se o sshd mostra o /etc/motd )
KeepAlive yes
SyslogFacility DAEMON
( habilita o código usado no syslog, podemos usar o código AUTH )
```

```

RhostsAuthentication no
( Habilita a autenticação pelo rhosts ou /etc/hosts.equiv )
RhostsRSAAuthentication no
( Habilita a autenticação pelo rhosts ou /etc/hosts.equiv usando o método de
autenticação RSA )
RSAAuthentication yes
( Especifica se o método de autenticação RSA é aceito )
PasswordAuthentication yes
PermitEmptyPasswords no
UseLogin no
# CheckMail no
# PidFile /u/zappa/.ssh/pid
#AllowUsers aaa
( Define que o user "aaa" tem permissão de efetuar login )
DenyUsers anderson
( Nega o login do user anderson )
#AllowGroups user
( Permite o acesso do grupo user definido em /etc/group )
#DenyGroups user
( nega o acesso do grupo user definido em /etc/group )

AllowHosts 10.10.10.*
( permite o acesso da rede 10.10.10.* /24 )
#DenyHosts *.*.* lowsecurity.theirs.com *.evil.org evil.org
( Nega acesso dos hosts e redes especificados )
# Umask 022
# SilentDeny yes

```

2.3.4 - Como startar o sshd:

```

# ----- inicio do script -----
#!/bin/sh
# start SSHD
if [ ! -d /usr/local/ssh1/sbin/sshd ]
then
    /usr/local/ssh1/sbin/sshd &
fi
# ----- fim do script -----

```

- No freebsd podemos adicionar este script em /etc/rc.local

Outra forma de startar é através do /etc/inetd.conf adicionando esta linha:

```
ssh stream tcp nowait root /usr/local/ssh1/sbin/sshd sshd -i
```

(Neste exemplo podemos usar o tcp-wrapper)

No /etc/services devemos colocar a linha:

```
ssh      22/tcp  #Secure Shell Login
ssh      22/udp  #Secure Shell Login
```

2.3.5 - Utilização do SSH client

Após ter configurado corretamente o servidor sshd podemos testar o login através de um client ssh no unix. Apresentaremos as formas básicas de utilização do ssh client:

Exemplo 1:

```
Ssh -l <user> <host>
# /usr/local/ssh1/bin/ssh -l aaa 10.10.10.2
Host key not found from the list of known hosts.
Are you sure you want to continue connecting (yes/no)? yes
Host '10.10.10.2' added to the list of known hosts.
aaa@10.10.10.2's password:
Last login: Mon Sep 13 05:21:06 1999 from 10.10.10.1
Copyright (c) 1980, 1983, 1986, 1988, 1990, 1991, 1993, 1994
    The Regents of the University of California.  All rights reserved.
No mail.
$
```

Exemplo 2:

```
Ssh <user>@<host>
$ /usr/local/ssh1/bin/ssh aaa@10.10.10.2
aaa@10.10.10.2's password:
```

```
Last login: Mon Sep 13 05:26:15 1999 from server1.aaa.com.  
Copyright (c) 1980, 1983, 1986, 1988, 1990, 1991, 1993, 1994  
The Regents of the University of California. All rights reserved.  
No mail.  
$
```

2.3.6 - Utilização do SCP client

- Uso do utilitário scp usado para transferência de arquivos usando a autenticação encriptada.

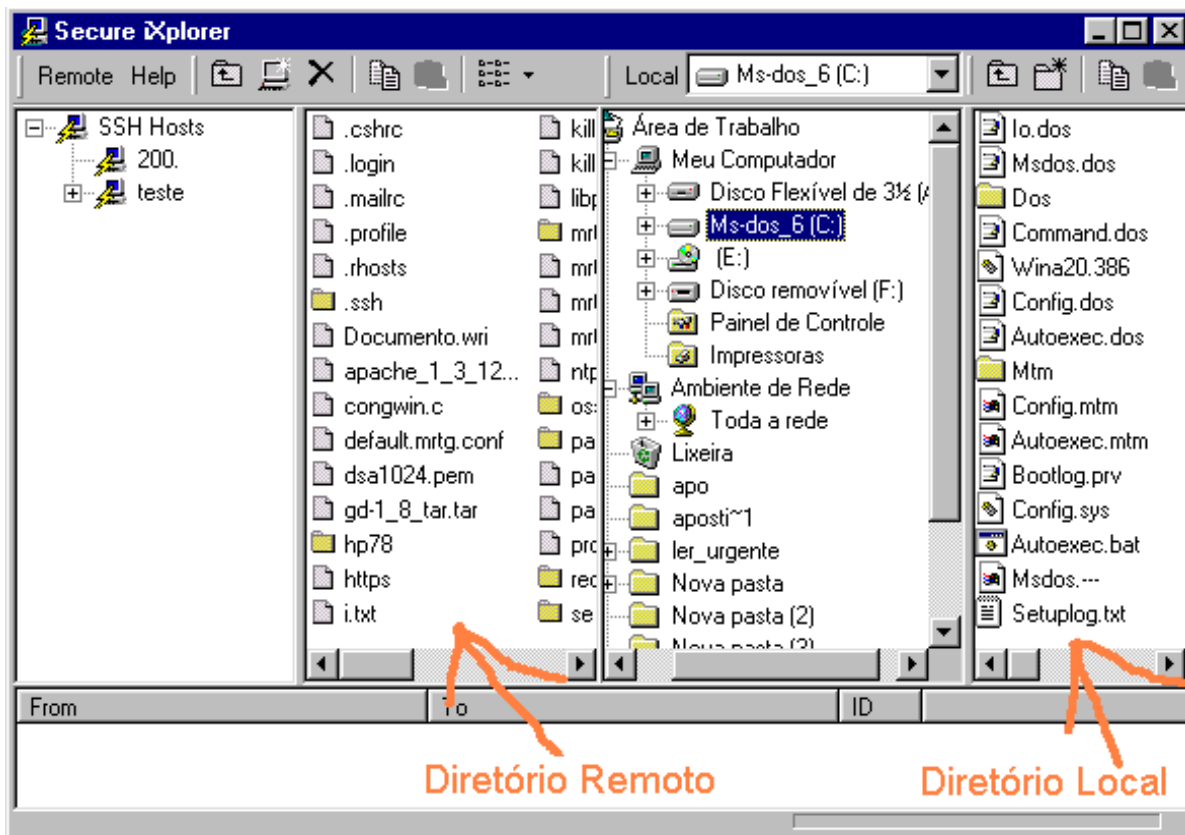
Exemplo do “put”:

```
Scp <nome_completo_do_arquivo_de_transferência>  
<login>@<maquina_remota_que>: / <diretorio_onde_o_arquivo_será_gravado>  
  
Scp /home/aaa/relatorio.txt aaa@10.10.10.2:/home/aaa/arquivos/  
(Neste exemplo o arquivo /home/aaa/relatorio.txt da máquina local será  
transferido para o diretório /home/aaa/arquivos/ da maquina remota 10.10.10.2 )
```

Exemplo do “get”:

```
Scp <login>@<maquina_remota_que>: /  
<nome_completo_do_arquivo_de_transferência> <diretorio_onde_o_arquivo  
será_gravado>  
  
Scp aaa@10.10.10.2:/home/aaa/relatorio.txt /home/aaa/arquivos/  
(Neste exemplo o arquivo /home/aaa/relatorio.txt que está na máquina remota  
10.10.10.2 será transferido para o diretório /home/aaa/arquivos/ da maquina  
local )
```

- Uso do software ixplorer.zip (<http://www.i-tree.org/>) scp gráfico para windows:



2.3.7 - Forward de portas

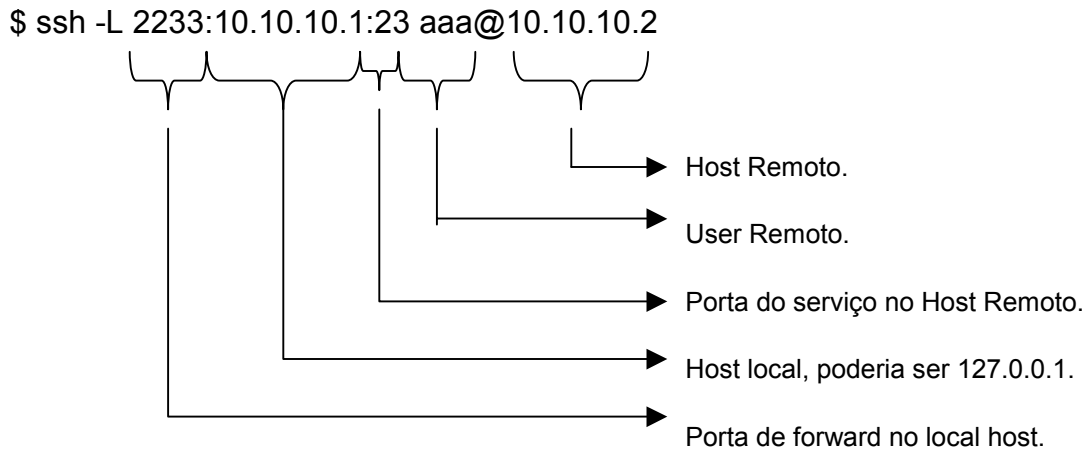
Uma das opções do SSH é criar um tunel encriptado entre o local host e o host remoto, e estabelecer a conexão de outro serviço entre os dois hosts em cima deste tunel encriptado.

Isto pode ser usado para acessos usando serviços como: telnet, ftp, POP3, Xwindows, Imap e outros.

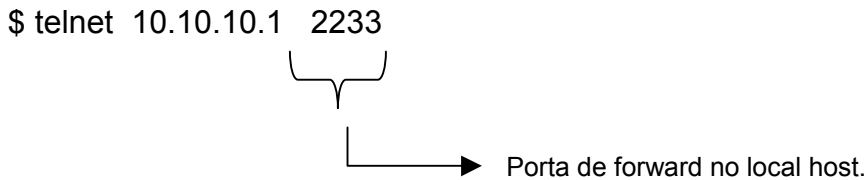
O exemplo abaixo será com o telnet, neste exemplo usaremos os dados:

Local host	10.10.10.1 ou 127.0.0.1
Host Remoto	10.10.10.2
Serviço Remoto	23 (telnet)
Porta local host	2233 (deve ser maior que 1024)

O primeiro passo é criar o telnet encriptado, inicialmente escolheremos a porta local 2233 para o forward da conexão de telnet. A senha do host remoto será solicitada como se estivessemos em uma conexão ssh comum.



O segundo passo é conectar na porta local escolhida para o forward (2233).



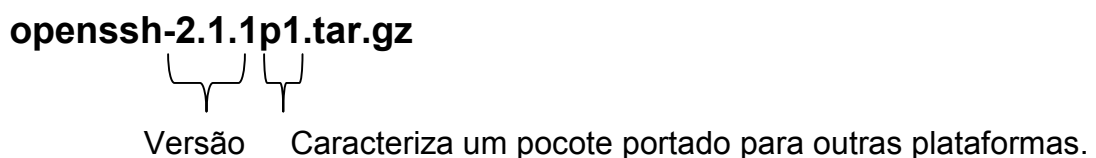
Feito isso a transmissão de dados será encriptada, isto pode ser observado com sniffers. Este método pode ser usado para outros serviços ditos inseguros, bastando apenas apontar a conexão para este serviço.

2.4 - Instalação do OpenSSH

Como foi dito acima o openssh é uma alternativa derivada do ssh 1.2.12, basicamente o openssh possui uma pequena diferença na instalação que será mostrada abaixo. A configuração e utilização não possui grandes diferenças.

O desenvolvimento do openssh é baseado em rigorosos processos de segurança coordenado pelo grupo do OpenBSD. Inicialmente o openssh foi desenvolvido pelo "the OpenBSD Project". Ele traz outros recursos de encriptação além do RSA e DES, como: Blowfish, 3DES, arcfour e cast128-cbc.

E a partir da versão 2.1.0 o openssh foi portado para outras plataformas, este pacote tar.gz é caracterizado pela letra p (portado) após a versão. Exemplo:



Lista de discussão:

Listas	Subscribe mail
Geral	openssh-unix-announce@mindrot.org
Desenvolvedores	openssh-unix-dev@mindrot.org

News Group: comp.security.ssh

Archive List: <http://marc.theaimsgroup.com/?l=secure-shell&r=1&w=2>

Teremos inicialmente que instalar os pacotes: Ssleay, openssl e zlib. Esta instalação foi realizado em um linux Slackware kernel 2.0.35.

2.4.1 - Instalação do Ssleay

```
$ tar -xvf SSLeay-0.9.0b.tar
$ cd SSLeay-0.9.0b
$ ./Configure linux-elf
$ make
$ make install
$ find . -name ssleay -print
./apps/ssleay
$ cp apps/ssleay /usr/local/bin/
# Copiar o binário ssleay para
# /usr/local/bin/ caso este
# exista na versão ssleay usada
```

2.4.2 - Instalação do Zlib

```
$ tar -xvzf zlib.tar.gz
$ cd zlib-1.1.3
$ ./configure
$ make
$ make install
$ make install
```

2.4.3 - Instalação do OpenSSL

```
$ tar -xvzf openssl-0.9.6.tar.gz
$ cd openssl-0.9.6
$ ./config
$ make
$ make test
$ make install
```

2.4.4 - Instalação do OpenSSH

```
$ tar -xvzf openssh-2.3.0p1.tar.gz
$ cd openssh-2.3.0p1
$ ./configure --enable-suid-ssh
$ make
$ make install
```

Caso deseje retirar o label da versão edite a linha abaixo em version.h:

```
#define SSH_VERSION "Versao clean"
```

Obs: Não podemos alterar a versão do protocolo.

2.4.5 – Arquivos de configuração e binários

➤ Em /usr/local/etc:

```
$ ls -l /usr/local/etc/
total 7
-rw-r--r-- 1 root  root    895 Nov 12 17:49 ssh_config
-rw----- 1 root  root    668 Nov 12 17:50 ssh_host_dsa_key
-rw-r--r-- 1 root  root    600 Nov 12 17:50 ssh_host_dsa_key.pub
-rw----- 1 root  root    525 Nov 12 17:49 ssh_host_key
-rw-r--r-- 1 root  root    329 Nov 12 17:49 ssh_host_key.pub
-rw-r--r-- 1 root  root   1292 Nov 12 17:49 sshd_config
```

➤ Binários gerados:

Daemon ssh, Pode ser iniciado standalone ou através do inetd. O item 2.3.4 descreve como inicializa -lo.

```
$ ls -l /usr/local/sbin/sshd*  
-rwxr-xr-x 1 root  root  630172 Nov 12 17:49 /usr/local/sbin/sshd  
$
```

Cliente ssh, o item 2.3.5 descreve como usa -lo. O 2.3.6 descreve como usar o scp para transferência de arquivos.

```
$ ls -l /usr/local/bin/ssh* /usr/local/bin/scp /usr/local/bin/sftp*  
  
-rwx--x--x 1 root  root  672556 Nov 12 17:48 /usr/local/bin/ssh  
-rwxr-xr-x 1 root  root  529800 Nov 12 17:48 /usr/local/bin/ssh-add  
-rwxr-xr-x 1 root  root  197448 Nov 12 17:48 /usr/local/bin/ssh-agent  
-rwxr-xr-x 1 root  root  533664 Nov 12 17:48 /usr/local/bin/ssh-keygen  
-rwxr-xr-x 1 root  root  4 Nov 12 17:48 /usr/servers/bin/scp -> scp2  
-rwxr-xr-x 1 root  root  1153356 Nov 12 17:48 /usr/servers/bin/scp2  
lrwxr-xr-x 1 root  wheel  5 Nov 12 17:48 /usr/local/bin/sftp -> sftp2  
lrwxr-xr-x 1 root  wheel  12 Nov 12 17:48 /usr/local/bin/sftp-server -> sftp-  
server2  
-rwxr-xr-x 1 root  wheel  550538 Nov 12 17:48 /usr/local/bin/sftp-server2  
-rwxr-xr-x 1 root  wheel  696335 Nov 12 17:48 /usr/local/bin/sftp-server2.static  
-rwxr-xr-x 1 root  wheel  1316683 Nov 12 17:48 /usr/local/bin/sftp2  
$
```

➤ Arquivo de configuração do sshd:

Arquivo de configuração sshd_config localizado em /usr/local/etc/ .

```
$ more /usr/local/etc/sshd_config  
Port 22  
# Define a porta usada pelo sshd  
#Protocol 2,1  
# Define o protocolo ssh1 ou ssh2  
ListenAddress 192.10.10.10  
# Define o ip onde será "montado" o servidor sshd  
HostKey /usr/local/etc/ssh_host_key  
# Localização da chave privada
```

```
ServerKeyBits 768
# Tamanho da chave privada
LoginGraceTime 600
# Tempo de espiração do login sem autenticação
KeyRegenerationInterval 3600
# Tempo de espiração da chave
PermitRootLogin no
# Não permite que o root log remotamente
IgnoreRhosts yes
# ignora as definições nos arquivos .rhosts
RhostsRSAAuthentication no
# Não permite autenticação RSA para as definições nos .rhosts
StrictModes yes
X11Forwarding no
# Não permite forward de sessões xwindows
X11DisplayOffset 10
PrintMotd yes
KeepAlive yes
SyslogFacility AUTH
# Define que serão logados via "syslogd" os acessos e autenticações
LogLevel INFO
# Define nível de log "syslogd"
RhostsAuthentication no
# Não permite autenticação Rhosts
/usr/local/etc/ssh_known_hosts
# arquivo que guardará as chaves públicas do servidores
RhostsRSAAuthentication no
# Não permite rhosts rsa autenticação
RSAAuthentication yes
# Permite autenticação usando o algoritmo RSA
PasswordAuthentication yes
PermitEmptyPasswords no
# Não permite autenticação com user contendo senhas em branco
#SkeyAuthentication no
# Habilita suporte a S/Key
#KbdInteractiveAuthentication yes
#KerberosAuthentication no
#KerberosOrLocalPasswd yes
#AFSTokenPassing no
#KerberosTicketCleanup no
# Kerberos TGT Passing does only work with the AFS kaserver
#KerberosTgtPassing yes
# Habilita suporte ao kerberos
CheckMail no
#UseLogin no
# Uncomment if you want to enable sftp
Subsystem sftp /usr/local/libexec/sftp-server
```

```
# habilita o acesso via sftp (Secure ftp)
#MaxStartups 10:30:60
$
```

A restrição de ips acessando o seu openssh server pode ser feita através do /etc/hosts/deny e etchosts.allow:

```
$ more /etc/hosts.deny
all:all
$ more /etc/hosts.allow
sshd:10.10.10.1 10.10.10.2
$
```

➤ Arquivo de configuração do ssh:

Arquivo de configuração ssh_config localizado em /usr/local/etc/ .

```
$ more /usr/local/etc/ssh_config
# Host *
# ForwardAgent yes
  ForwardX11 no
# Permite foward de Xwindows
  RhostsAuthentication no
# Suporte a acesso .rhosts
  RhostsRSAAuthentication no
# Suporte a acesso .rhosts via algoritmo RSA
  RSAAuthentication yes
# Suporte ao algoritmo RSA
  PasswordAuthentication yes
  FallBackToRsh no
  UseRsh no
# Habilita uso de "rsh" via ssh
# BatchMode no
  CheckHostIP yes
# StrictHostKeyChecking no
  IdentityFile ~/.ssh/identity
# Guarda em ~/.ssh/identity as chaves públicas dos servidores
  Port 22
# Define a porta de acesso do ssh client
  Protocol 2,1
# Deifne o suporte aos protocolos ssh1 e ssh2
# Cipher blowfish
# Suporte ao algoritmo blowfish
  EscapeChar ~
```

```
# Define a Tecla de escape, "exit"  
$
```

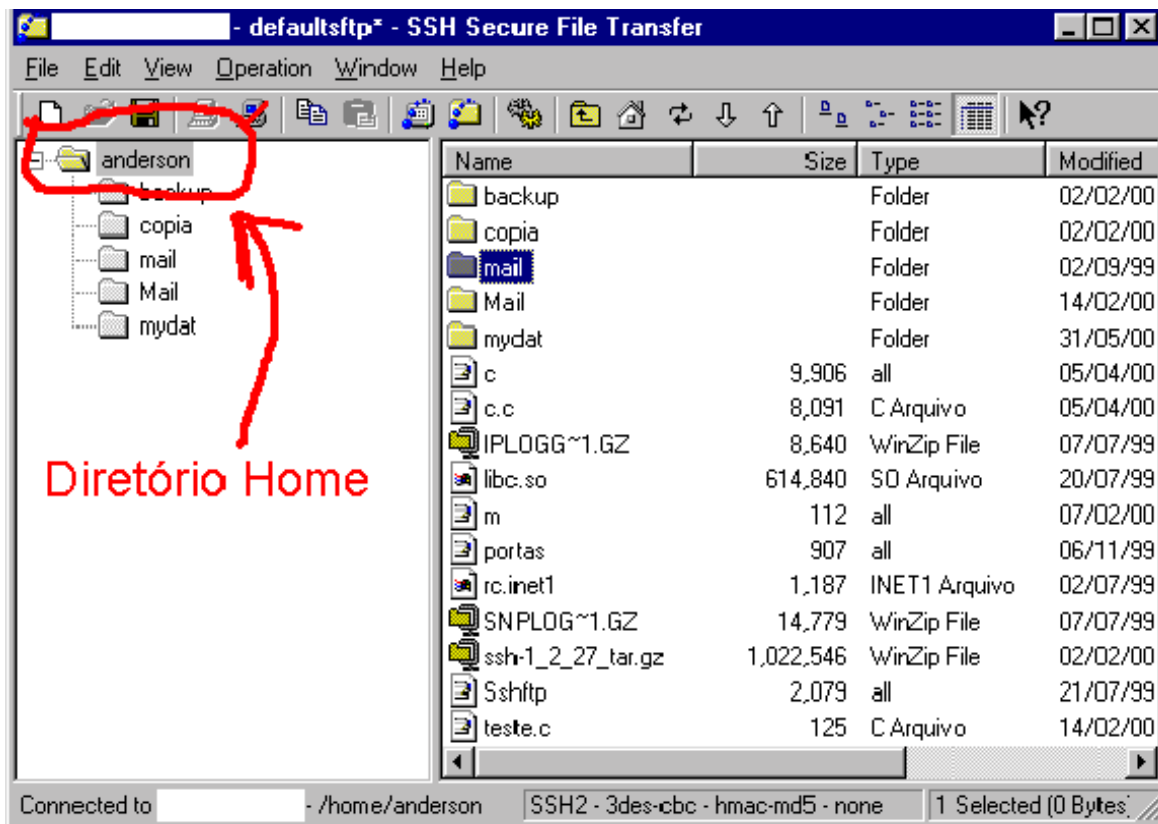
2.4.6 – Recursos do scp gráfico no protocolo SSH2

Este software é o *SSHWin-versao.exe* encontrado no site www.ssh.com, ele possui licença para uso acadêmico.

Basta descomentar a linha abaixo em `sshd_config`:

```
Subsystem sftp /usr/local/libexec/sftp-server
```

Após conectar o servidor notaremos que o usuário não pode descer de nível no seu diretório home, ficando restrito ao home diretório como acontece no ftp. Outra diferença é a flexibilidade com o windows que o software *SSHWin-versao.exe* possui, podemos transferir arquivos como se estivessemos usando as facilidades de um software de ftp tradicional para windows.



3 - Soluções de acesso remoto encriptado

3.1 - Server

- LSH/psst (implementação do SSH2)

<http://www.net.lut.ac.uk/psst/>

- **OpenSSH**

Um projeto iniciado para o openbsd, compatível com a versão ssh1.

Este é um dos melhores SSH server, pois, possui suporte ao protocolo SSH1 e SSH2, também é desenvolvido pelo mesmo grupo que coordena o desenvolvimento do OpenBSD. Quando a transferências de arquivos é vital este

software é o mais aconselhado ao lado do SSH2, também é aconselhado a utilização dos clientes F-Secure para SSH2 ou do site www.ssh.com para SSH2.

Site: <http://www.openssh.com/>

➤ OSSH

<ftp://ftp.pdc.kth.se/pub/krypto/ossh/>

➤ SSH1

Sites:

<http://www.ssh.fi/>

<http://www.ssh.org/>

<http://www.datafellows.com>

➤ **SSH2**

Sites:

<http://www.ssh.fi/>

<http://www.ssh.org/>

Quando a transferências de arquivos é vital este é o mais aconselhável ao lado do OpenSSH (com suporte a SSH2), também é aconselhado a utilização dos clientes F-Secure para SSH2 ou do site www.ssh.com para SSH2.

➤ Telnet – SSL com SSLtelnet and MZtelnet

<ftp://ftp.uni-mainz.de/pub/internet/security/ssl/>

[ftp://ftp.zedz.net/pub/replay/linux/redhat/.](ftp://ftp.zedz.net/pub/replay/linux/redhat/)

➤ Windows NT Server

<http://marvin.criadvantage.com/caspian/Software/SSHD-NT/default.php>

3.2 - Clientes

- BeOS: The BeOS R4 port of SSH1 for Intel and PowerPC is

www.be.com/beware/Network/ssh.html

- “BetterTelnet 2.0bX with SSH patch”

<http://www.cstone.net>

- Cédric Gourio also developed a Java based SSH for his diploma

www.cl.cam.ac.uk/~fapp2/software/java-ssh

- DataFellows’s “F-Secure SSH Client for Macintosh and Windows”:

Um dos melhores clientes SSH1 e SSH2 para windows, porém este software é pago.

<http://www.datafellows.com>

- DOS Client

<http://www.vein.hu/~nagyd/#ssh>

- Fresh Free FiSSH

<http://www.massconfusion.com/ssh/>

- Fsh “Fast remote command execution”

<http://www.lysator.liu.se/fsh/>

- Gnome SSH Client

Cliente SSH com interface gráfica para Gnome.

<http://zephyr.webhop.net/gnome-ssh.html>

➤ Explorer – scp gráfico para windows
<http://www.i-tree.org/>

➤ MacSSH (implementação do SSH2)

<http://www.macssh.com/>

➤ Mindterm (escrito em JAVA)

<http://www.mindbright.se/mindterm/>

➤ NSH

<http://www.networkshell.com/>

➤ “NiftyTelnet 1.1 SSH” (implemetação do SSH1)

<http://www.lysator.liu.se>

<http://andrew2.andrew.cmu.edu>

➤ OS/2

<ftp://hobbes.nmsu.edu/pub/os2/apps/internet/telnet/client/ssh-1.2.27-b1.zip>

➤ Palm Pilot

www.isaac.cs.berkeley.edu/pilot/

<ftp.zedz.net/pub/crypto>

➤ Putty

Este talvez seja o mais flexível cliente para SSH1 permite conectar nos mais diversos ssh1 servers, so superado pelo F-Secure que é pago. Porém não possui uma interface gráfica para o scp (para arquiteturas intel i386), ao contrário dos clientes ssh2 da F-Secure ou do site www.ssh.com. O software que implementa o scp (para arquiteturas intel i386) ainda é executado via linha de comandos e a sua versão gráfica atualmente não é muito agradável.

Este software também possui facilidade e rapidez na instalação, economizando bastante trabalho do suporte técnico.

<http://www.chiark.greenend.org.uk/~sgtatham/putty/>
<ftp://ftp.chiark.greenend.org.uk/users/sgtatham/putty-latest/>

scp para windows

<http://www.chiark.greenend.org.uk/~sgtatham/putty/>
<ftp://ftp.chiark.greenend.org.uk/users/sgtatham/putty-latest/>

➤ Secure CRT

<http://www.vandyke.com/>

➤ SRP

<http://www.kermit-project.org/k95.html>
<http://srp.stanford.edu/srp/>

➤ SSH Plugin (escrito em JAVA)

<http://www.mud.de/se/jta/doc/plugins/SSH.html>

➤ SSH Win32 ports

<http://guardian.htu.tuwien.ac.at/therapy/ssh/>

➤ Telneat (implementação do SSH1 para windows)

<http://telneat.lipetsk.ru/>

➤ Tera Term Pro

www.zip.com.au/~roca/download.html
<http://hp.vector.co.jp/authors/VA002416/teraterm.html>

➤ TTSSH (implementação do SSH1 para windows)

<http://www.zip.com.au/~roca/ttssh.html>

➤ UNIX Client

<http://www2.wiwi.uni-marburg.de/~leich/soft/secpanel/>

- VAX/OpenVMS

www.er6.eng.ohio-state.edu/~jonesd/ssh
SSH1 client www.free.lp.se/fish

- Windows CE

www.movsoftware.com/sshce.htm

- Windows Client

<http://www.pragmasys.com/SecureShell/>

4 - Bibliografia:

Sites:

- BSD - “<http://www.bsd.org>”;
- Conectiva - “<http://www.conectiva.com.br>”;
- FreeBSD - “<http://www.freebsd.org>”;
- ISS - “<http://www.iss.net>”;
- LDP - “<http://ldp.conectiva.com.br>”;
- Linux.com – “<http://www.linux.com>”;
- Linux.org - “<http://www.linux.org>”;
- Linux Unicamp - “<http://www.linux.unicamp.br>”;
- Linux Usp - “<http://www.linux.usp.br>”;
- NetBSD - “<http://www.netbsd.org>”;

- Net-Security - "<http://www.net-security.org>";
- Olinux - "<http://www.olinux.com.br>";
- OpenBSD – "<http://www.openbsd.org>";
- OpenSSH - "<http://www.openssh.org>" e "<http://www.ssh.com>";
- OpenSSL - "<http://www.openssl.org>";
- RNP – "<http://www.rnp.br>";
- SSH - "<http://www.ssh.org>", "<http://www.ssh.fi>" e "<http://www.ssh.com>";
- SSL - "<http://www.ssl.org>";
- Unicamp - "<http://www.security.unicamp.br>";

Livros:

- Computer Networks; Andrew S. Tanenbaum, Prentice Hall PTR, 1996;

Revistas:

- Security Magazine - "<http://www.securitymagazine.com.br>";

Espero que este artigo ajude a outras pessoas, qualquer dúvida ou sugestão mailme

Autor: Anderson Alves de Albuquerque
e-mail: a.alves@montreal.com.br
Agradecimentos: Marita Maestrelli