

Aula Exercício de Introdução ao SSH

Henrique Gemignani Passos Lima

25 de outubro de 2011

Sumário

1	O que é SSH, e principais características	2
2	Como usar	2
2.1	PuTTY	2
2.2	OpenSSH Client	2
3	Utilidades para o SSH	2
3.1	Shell Remoto	2
3.2	Transferência de Arquivos	3
3.3	Túneis	4
3.4	Git	4
4	Métodos de Autenticação	5
4.1	Senha	5
4.2	Chaves	5
4.2.1	Agente SSH	5
5	Segurança	6
6	Túneis	7
6.1	Local -> Remoto	7
6.2	Remoto -> Local	8
6.3	Dinâmico	9
6.4	X	9

1 O que é SSH, e principais características

SSH, Secure Shell, é um protocolo de rede que permite uma conexão segura entre dois computadores através de um ambiente de rede inseguro.

Principais características:

1. Autentica o servidor, impedindo ataques do estilo “man-in-the-middle”.
2. Conexão criptografada: senhas não passam em branco pela rede.

2 Como usar

Existem diversos clientes, mas nessa aula veremos apenas dois:

2.1 PuTTY

Embora essa aula foque apenas na versão Windows do PuTTY, é importante notar que esse é um programa multi-plataforma e possui versões para Linux.

2.2 OpenSSH Client

É um programa de terminal. Aqui vai uma lista de exemplos de comandos:

1. `ssh destino` ; Conecta-se em destino, usando o mesmo usuário que o seu atual.
2. `ssh jose@destino` ; Conecta-se em destino, com o usuário “jose”.
3. `ssh jose@destino -p 22000` ; Conecta-se em destino na port 22000.
4. `ssh -i chaveprivada destino` ; Conecta-se em destino, usando `chaveprivada` como chave privada de autenticação.

3 Utilidades para o SSH

3.1 Shell Remoto

O principal e mais simples uso do SSH é fazer um login com um shell numa máquina remota. Com isso podemos trabalhar em uma máquina remota através de uma rede.

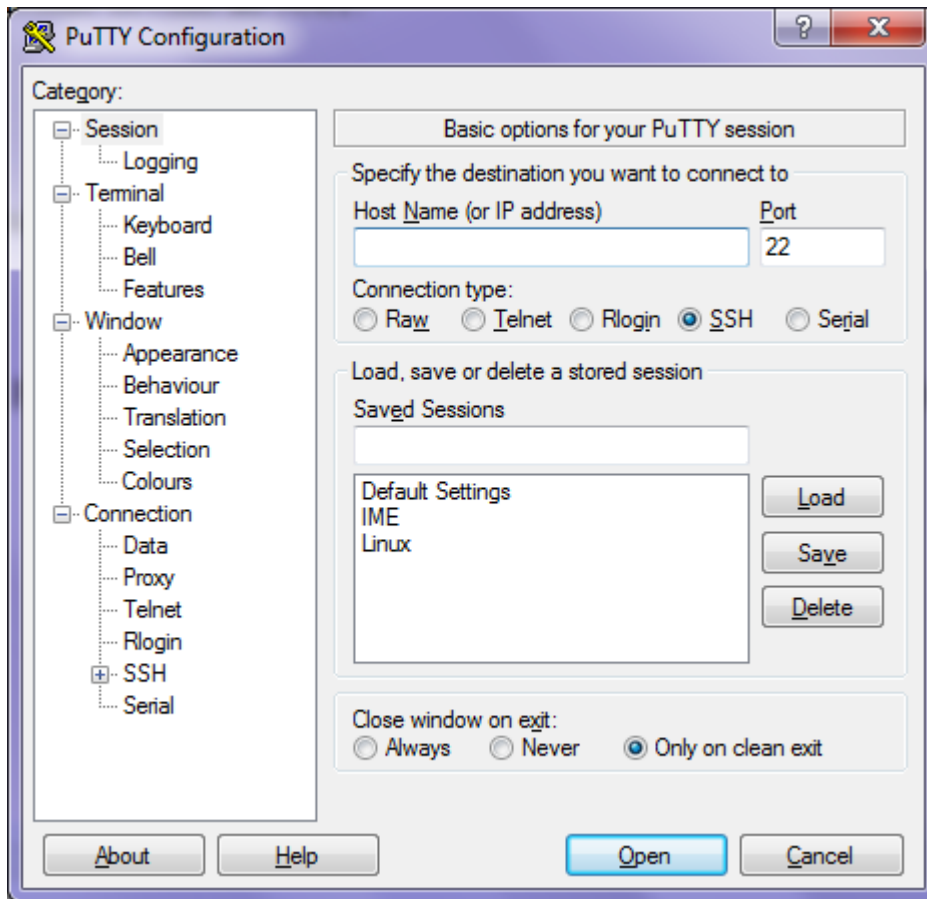


Figura 1: Tela inicial do PuTTY

Exemplos:

1. Conecte-se com a sua máquina do trabalho e continua a editar documentos usando o vim.
2. Conectar-se em um servidor HTTP e configurar o acesso de um novo usuário.

3.2 Transferência de Arquivos

É possível transmitir arquivos de uma máquina para outra por meio do SSH através do `scp` ou `sftp`.

O `scp` é simplesmente um `cp` encapsulado dentro de uma conexão SSH. Permite copiar arquivos locais para máquinas remotas, e de máquinas remotas para local.

O `sftp` é um protocolo desenvolvido com o objetivo de providenciar uma alternativa segura ao FTP.

3.3 Túneis

Túneis permitem encapsular protocolos inseguros na conexão segura do SSH, como por exemplo acessar e-mails.

Também é possível utilizar túneis para burlar problemas devido ao NAT.

3.4 Git

Git é um sistema de controle de versão distribuído, que utiliza túneis SSH para fazer a transferência de dados de maneira segura, além de utilizar chaves para autenticação.

Para comparação, considere esse comando git:

```
git clone git@github.com:uspgamedev/horus_eye.git
```

Esse comando abre uma conexão SSH para o servidor `github.com` usando o usuário `git` e executa o clone do git no caminho `uspgamedev/horus_eye.git`.

4 Métodos de Autenticação

4.1 Senha

Aparece um prompt onde você digita a senha, que é então transmitida de maneira segura para o servidor, e que então verifica se a senha é válida.

4.2 Chaves

Chaves são utilizadas para autenticação automática, e são compostas de duas partes: a pública e a privada.

A chave pública, você deve disponibilizar para os servidores no qual você deseja autenticar-se, enquanto que a chave privada deve permanecer apenas no seu cliente.

Uma dos parâmetros para abrir uma conexão SSH é o caminho para o arquivo de chave privada que deseja usar. O cliente SSH tentará autenticar-se com o servidor usando essa chave, e obterá sucesso apenas se o servidor conhecer (e aceitar) a chave pública associada.

Como a chave em si é utilizada para a autenticação, deve-se tomar cuidado para impedir acesso indevido à chave. Uma opção de segurança é criptografar a chave com uma “passphrase” .

4.2.1 Agente SSH

Um agente é um programa que possui uma lista de chaves privadas, onde o seu cliente, no mesmo computador, tem acesso. Como o agente guarda as chaves de forma descriptografada, não é necessário digitar senhas ao usar o agente.

É possível encaminhar o agente através da conexão, permitindo utilizar as mesmas chaves privadas no destino.



Nome	Tipo
 Exemplo.ppk	PuTTY Private Key File
 Exemplo.pub	Arquivo PUB

Figura 2: Uma chave, composta pela parte privada e pública.

Cuidado: use agentes SSH, em particular se encaminhar-los pela conexão, apenas se você confia nos administradores da máquina destino, pois estes podem acessar os arquivos temporários que o agente usa!

5 Segurança

O cliente possui uma tabela de servidores conhecidos, onde cada item é uma tupla (hostname, endereço IP, chave pública).

Ao conectar com um servidor, este envia a sua chave pública, e seu cliente verifica a integridade desta chave. Se a resposta for negativa, a conexão é terminada. Senão, de agora em diante, todo dado transportado pela conexão é criptografado usando a chave pública do servidor.

Isto garante, desde de que o cliente conheça propriamente a chave pública de fato do servidor, que ataques do estilo “man-in-the-middle” são impossibilitados, pois apenas o servidor têm conhecimento de sua chave privada.

6 Túneis

Túneis são formas de encapsular e encaminhar conexões arbitrárias com a conexão SSH.

6.1 Local -> Remoto

O seu cliente SSH local escuta conexões numa port TCP de sua máquina, encaminha através da conexão segura até um servidor SSH, e este encaminha seus pacotes para o destino, em uma port especificada.

Note que a port local, o destino e a port no destino são especificadas por você ao abrir a conexão SSH.

O exemplo abaixo pode ser replicado no OpenSSH Client com o comando `ssh user@servidor -L8080:destino:80`. Note que o destino pode ser fisicamente o próprio servidor SSH.

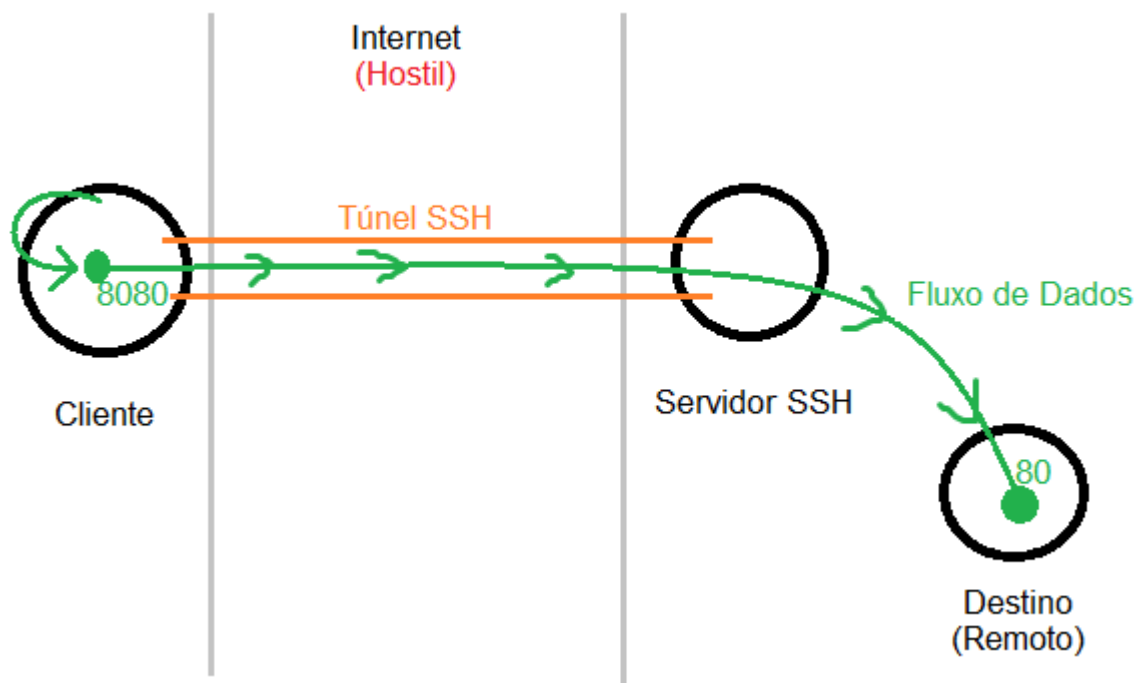


Figura 3: Exemplo de Túnel Local -> Remoto

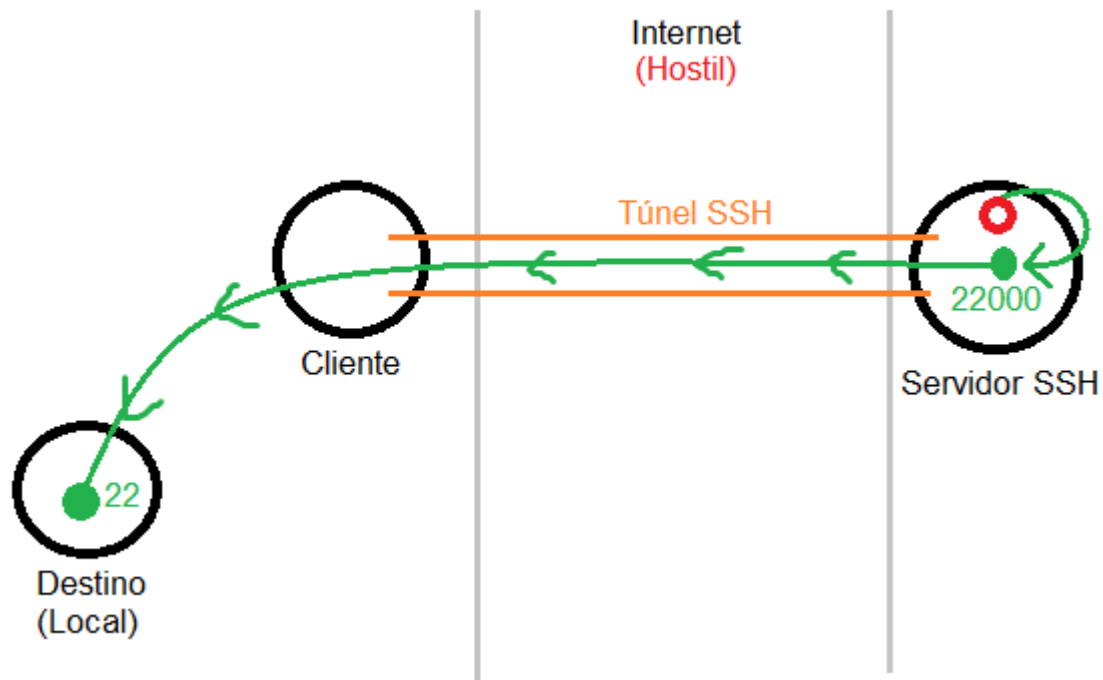


Figura 4: Exemplo de Túnel Remoto -> Local

6.2 Remoto -> Local

Você configura o servidor SSH para que ele escute uma port específica, e abre um túnel de conexão segura com ele. O servidor então envia todo pacote na port escutada para o destino local em uma port específica, através do seu cliente.

Neste caso também, as ports e o destino são especificados por você, na criação do túnel.

Em geral, o destino é a própria máquina onde o cliente está sendo executado.

O exemplo abaixo pode ser replicado no OpenSSH Client com o comando `ssh user@servidor -R8080:destino:80`. Note que o destino pode ser fisicamente o próprio servidor SSH.

6.3 Dinâmico

O seu cliente SSH escuta em uma port especificada em modo de entrada, podendo também restringí-la a um endereço específico. Sempre que uma conexão é aberta com essa port, o protocolo da conexão é usado para decidir o destino final, e a conexão é encaminhada para o servidor pela conexão segura, que então encaminha por sua vez para o destino.

Isto é útil para criar proxies de maneira segura.

6.4 X

O X é um sistema de gerenciador de janelas, que faz distinção entre servidor (que encapsula o gerenciador de janelas) e clientes (programas que utilizam e exibem janelas). Isso permite que façamos túneis de gerenciadores de janelas.

Um túnel X do SSH, é um tipo de “display virtual” na sua sessão remota, que corresponde a um servidor X na sua máquina local.

Ao abrir um programa remotamente que usa janelas (um cliente X), ele tentará usar este “display virtual”, que é túnel de conexão segura para seu gerenciador de janelas local.

Assim, você pode usar programas remotamente, ainda aproveitando da interface gráfica de janelas :P