Remote Logins

The purpose of this document is to explain how you can login to the Computer Science Department's UNIX network from a remote PC, such as one you have at home or at work. Since there is no need to explain this if you already have a Linux machine at home or work that you know how to use, I only describe what you need to do if you have a Windows host or a Macintosh host.

The computer that you have at home is the client computer; the one in school is the server. The computer that is in front of you is the local host; the one to which you want to connect is the remote host. Thus, you want to connect your local computer, the client, to the remote computer, which is the server.

About Secure Shell

Before the World Wide Web really took hold, the streets of the Internet were relatively safe for kids of all ages. No one took many precautions about bad guys trying to break into places. Thus, to remotely login to a computer over the Internet, one ran a very simple and not particularly secure program called *telnet* or *rlogin*. As more and more people started using the World Wide Web, more people started snooping around the more obscure streets of the Internet. The swelling crowd included people trying to break into machines.

The problem with *telnet* and *rlogin* was that they sent everything that people typed on their local machines to the remote machines as clear text. People could "tap" the wires and capture this clear text. Thus, the snoops could capture usernames and passwords and therefore gain access to computer accounts that they had no right to access. Along came the Secure Shell Protocol. The Secure Shell Protocol, known as SSH, basically created a technology by which everything that a person typed on the local machine was encrypted on the local machine, transported across the Internet in encrypted form, and decrypted at the remote machine. This made it pretty near impossible to capture passwords. SSH was eventually revised and became SSH2.

System Administrators have pretty much prevented anyone from using the insecure telnet and *rlogin* utilities by closing the ports that they use for communication and turning off the telnet and *rlogin* servers on their machines. They force you to use SSH.

To remotely login to a UNIX host, you need to obtain an SSH client program. The SSH client will make a connection to the SSH server on the remote machine, create a terminal window for you, and let you login to the machine. After you have logged in, you can work just as if you were seated at the computer using a terminal window at the console.

Obtaining SSH Clients

There are several free SSH clients. ***OpenSSH*** is an open source version developed for the ***OpenBSD*** project. It is available at http://www.openssh.com. Alternatives for *Windows* and *Macintosh* are at http://www.openssh.com/windows.html and http://www.openssh.com/macos.html respectively. (If you are using OS X on a *Macintosh*, you do not need to download an SSH client, because ***OpenSSH*** is built into OS X. However, it does not have a GUI; you have to type on the command line.)

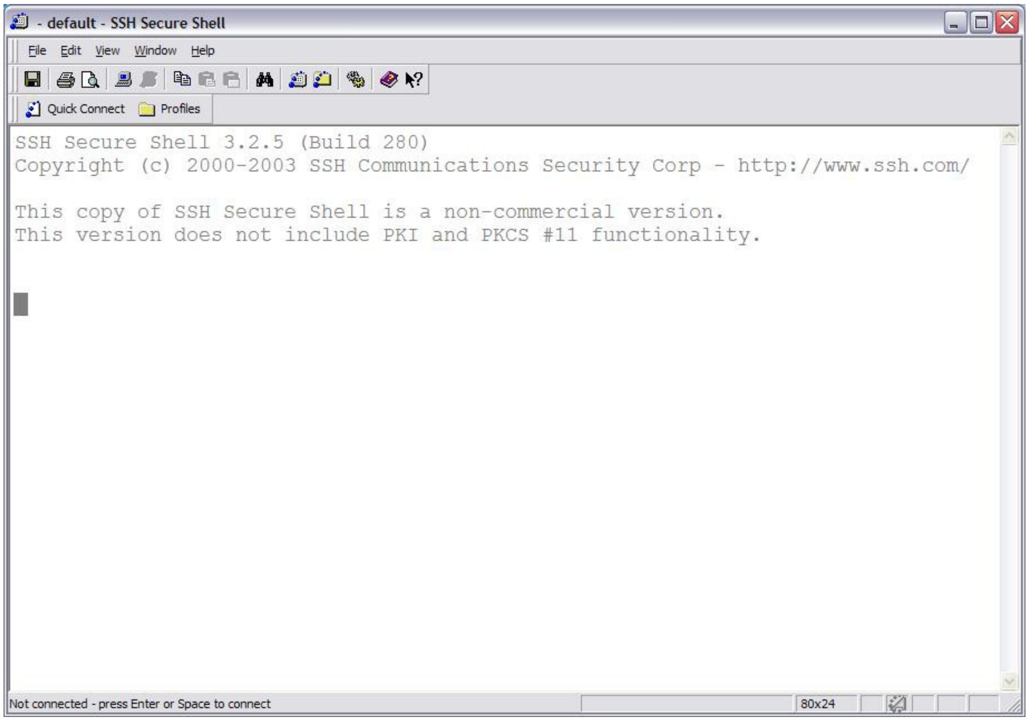
PuTTY ssh is another free version for Windows operating systems, available at http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html. My preference for Windows is the original SSH client, which is no longer supported. It can be downloaded from my website at http://www.compsci.hunter.cuny.edu/~sweiss/resources.php#Applications.

Installing an SSH Client

This is pretty much a trivial task. The clients referenced above are all installers; simply start the executable file and answer the questions that the installer wizard poses to you.

Using Windows SSH Clients

This will vary depending upon the particular SSH client that you chose. If you use *SSH Secure Shell Client* from SSH itself, the window will look like Figure 1initially.

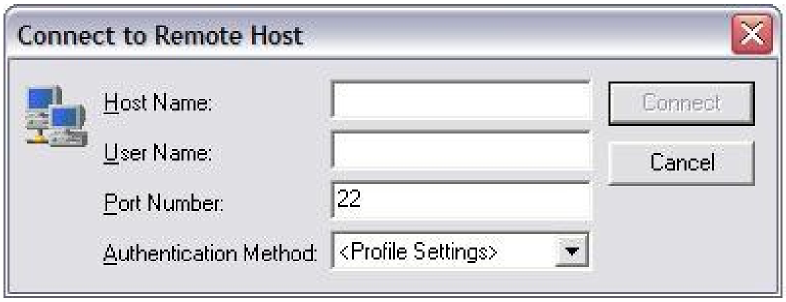


```
SSH Secure Shell 3.2.5 (Build 280)
Copyright (c) 2000-2003 SSH Communications Security Corp - http://www.ssh.com/

This copy of SSH Secure Shell is a non-commercial version.
This version does not include PKI and PKCS #11 functionality.
```

Not connected - press Enter or Space to connect

Figure 1: Initial State of SSH Client

The first time that you use it, you should click on the *Quick Connect* button. It will display a dialog box like the one in Figure 2. Enter the information to connect to the server: for our server, the host name is `eniac.geo.hunter.cuny.edu` and the user name is your user name on our system. After you have filled in the fields, click the Connect button.



Port Number: 22
Authentication Method: <Profile Settings>

Figure 2: Connection Dialog

The *Password Dialog* box will open, as shown in Figure 3.

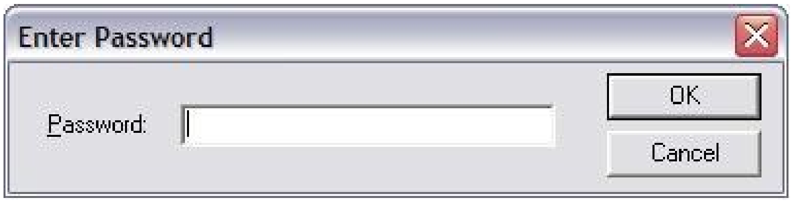
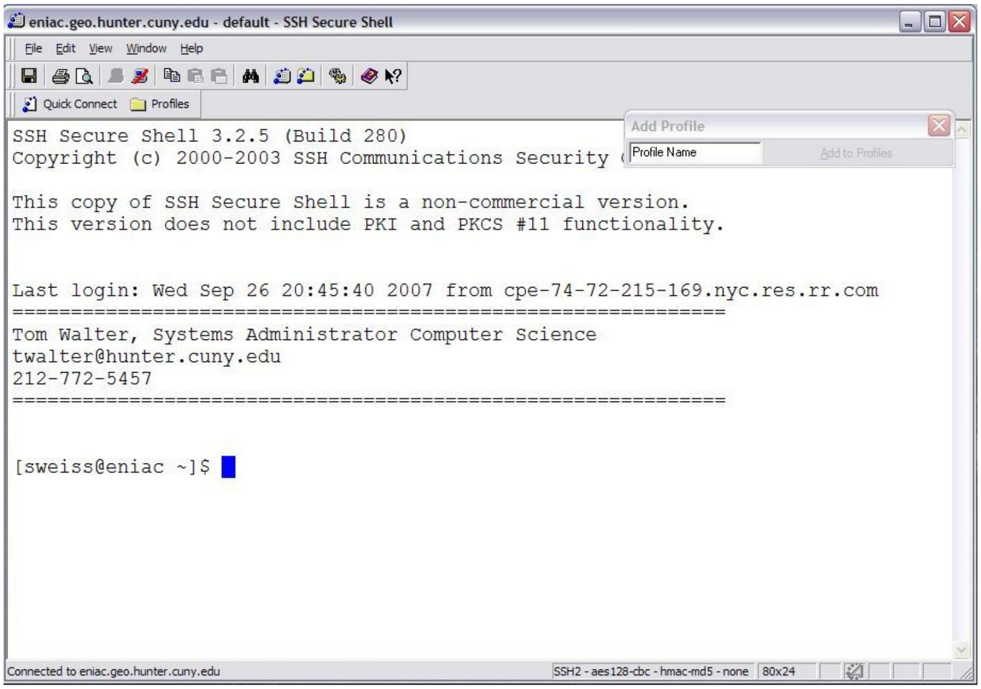


Figure 3: Password Dialog Box

After you enter the correct password, the screen will look like Figure 4. In the *Add Profile* text box, type a name for this connection. I usually use the name of the host. Any name will do.



```
SSH Secure Shell 3.2.5 (Build 280)
Copyright (c) 2000-2003 SSH Communications Security (

This copy of SSH Secure Shell is a non-commercial version.
This version does not include PKI and PKCS #11 functionality.


Last login: Wed Sep 26 20:45:40 2007 from cpe-74-72-215-169.nyc.res.rr.com
==============================================================
Tom Walter, Systems Administrator Computer Science
twalter@hunter.cuny.edu
212-772-5457
==============================================================


[sweiss@eniac ~]$ 
```

Connected to eniac.geo.hunter.cuny.edu
SSH2 - aes128-cbc - hmac-md5 - none
80x24

Figure 4: Add Profile Dialog Box

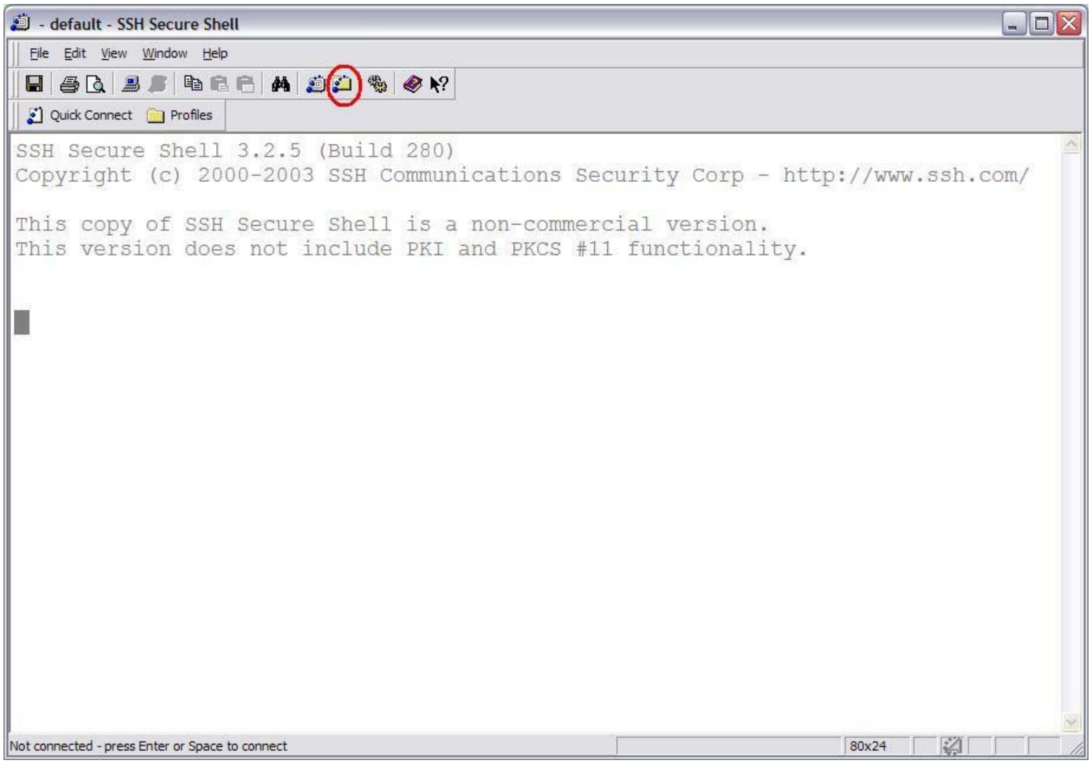
After you have done this, the next time that you want to connect to that host, instead of using the *Quick Connect* button, you can use the *Profiles* button. When you click it, a list of the profiles that you have saved will be displayed, and you can click the one that you want to open. SSH will prompt you for the password alone, and you will be logged in (assuming you entered the correct password and still have access to the system.)

When you logout of the system, SSH will automatically disconnect the client from the server, and you can simply close SSH.

SFTP

You can use the SSH client for transferring files between the client and the server. Each client's interface is somewhat different, but the idea is the same: to upload files, you select the files on the local machine, select the directory into which they are to be copied on the remote machine, and click some button to transfer them.

SSH Secure Shell Client has a drag-and-drop interface that is very convenient. Click the icon circled in red in Figure 5.



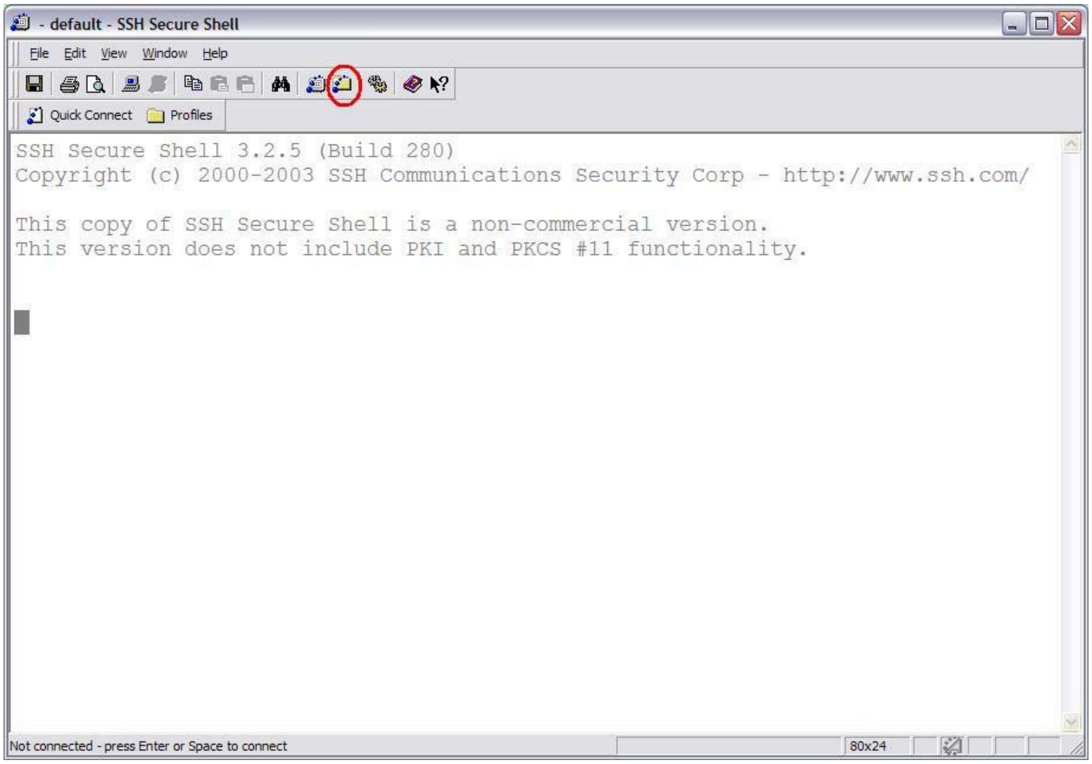
```
SSH Secure Shell 3.2.5 (Build 280)
Copyright (c) 2000-2003 SSH Communications Security Corp - http://www.ssh.com/

This copy of SSH Secure Shell is a non-commercial version.
This version does not include PKI and PKCS #11 functionality.
```

Not connected - press Enter or Space to connect

80x24

Figure 5: File Transfer Icon

A window like that shown in Figure 6 will be displayed. You can drag the files that you want to upload from Explorer windows into the right hand pane of this window. You can also download files by the reverse drag-and-drop.


- default - SSH Secure Shell
File Edit View Window Help
Quick Connect
Profiles
Not connected - press Enter or Space to connect
80x24

```
SSH Secure Shell 3.2.5 (Build 280)
Copyright (c) 2000-2003 SSH Communications Security Corp - http://www.ssh.com/

This copy of SSH Secure Shell is a non-commercial version.
This version does not include PKI and PKCS #11 functionality.
```

Figure 6: File Transfer Icon