Getting Started - Windows

Physics 91SI Handout 01

Alex Ji and Zahan Malkani

1 Introduction

The first day of class, we're going to jump right into how to navigate a UNIX system and Python programming. In order to do this, we need you to do some initial legwork of installing some programs. **Please go through this handout and have an SSH client with X11 forwarding installed for the first day of class (March 29th).**

A reminder: bring your laptop to the first day of class. Our lectures will involve you doing some follow-along coding, which needs a laptop.

There are three steps to this. First, you have to pick and install an SSH client, configuring a connection that forwards X11 to you. Second, you have to install and open an X11 server. Finally, you should try logging into `corn.stanford.edu` and opening emacs with a GUI.

2 Installing an SSH Client

2.1 What's an SSH Client?

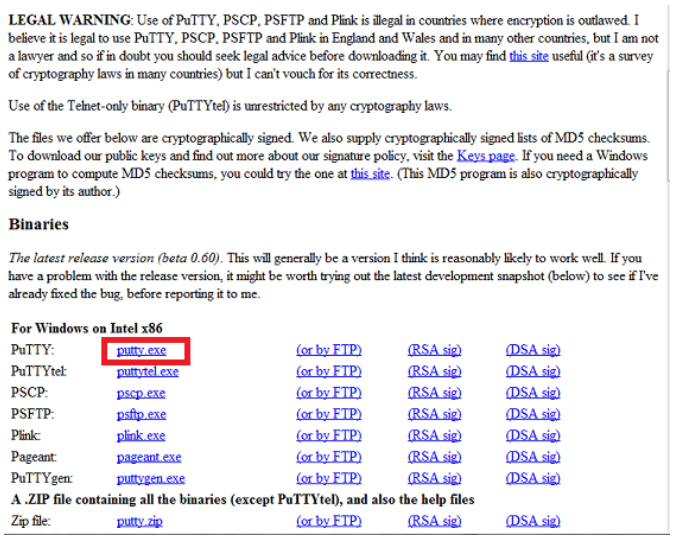
Since Windows isn't built on the same architecture as UNIX, you have to install a separate program to be able to access UNIX computers. This program is called an SSH client.

There are two clients that I'll tell you about: PuTTY and SecureCRT. You only have to choose one. Personally I use SecureCRT, but only because it's what I started using first. They are basically the same, but the options are located in different menus.

2.2 PuTTY

PuTTY is a popular, lightweight, and free SSH client. To start, go to `http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html` and download the file "putty.exe". Save it somewhere you can get to easily, like your desktop. (If you are wondering, encryption is legal in the US and most other countries. If you don't know what type of processor you have, it is probably x86.) See Figure 1

Once you have downloaded PuTTY, go ahead and just open the executable that you've downloaded. This should open a window that looks like Figure 2. Go ahead and type `corn.stanford.edu` into the "Host Name" field, then push "Open". A console window will pop up. Login with your SUNet ID and password, and you're good to go!



LEGAL WARNING: Use of PuTTY, PSCP, PSFTP and Plink is illegal in countries where encryption is outlawed. I believe it is legal to use PuTTY, PSCP, PSFTP and Plink in England and Wales and in many other countries, but I am not a lawyer and so if in doubt you should seek legal advice before downloading it. You may find this site useful (it's a survey of cryptography laws in many countries) but I can't vouch for its correctness.

Use of the Telnet-only binary (PuTTYtel) is unrestricted by any cryptography laws.

The files we offer below are cryptographically signed. We also supply cryptographically signed lists of MD5 checksums. To download our public keys and find out more about our signature policy, visit the Keys page. If you need a Windows program to compute MD5 checksums, you could try the one at this site. (This MD5 program is also cryptographically signed by its author.)

Binaries

The latest release version (beta 0.60). This will generally be a version I think is reasonably likely to work well. If you have a problem with the release version, it might be worth trying out the latest development snapshot (below) to see if I've already fixed the bug, before reporting it to me.

For Windows on Intel x86

PuTTY:	putty.exe	(or by FTP)	(RSA sig)	(DSA sig)
PuTTYtel:	puttytel.exe	(or by FTP)	(RSA sig)	(DSA sig)
PSCP:	pscp.exe	(or by FTP)	(RSA sig)	(DSA sig)
PSFTP:	psftp.exe	(or by FTP)	(RSA sig)	(DSA sig)
Plink:	plink.exe	(or by FTP)	(RSA sig)	(DSA sig)
Pageant:	pageant.exe	(or by FTP)	(RSA sig)	(DSA sig)
PuTTYgen:	puttygen.exe	(or by FTP)	(RSA sig)	(DSA sig)

A .ZIP file containing all the binaries (except PuTTYtel), and also the help files

Zip file:	putty.zip	(or by FTP)	(RSA sig)	(DSA sig)

Figure 1: The link to download PuTTY

If you login to the same place a lot with the same settings, you can save the settings easily by changing things in this "PuTTY Configuration" window, typing a name into the "Saved Sessions" box, and pressing "Save". Loading this session uses all the same options you used before.

2.3 SecureCRT

SecureCRT is an SSH client that you can get for free while you are at Stanford. The only extra feature I know about is the ability to have tabbed terminal consoles. To download, go to `http://itservices.stanford.edu/service/ess/pc/securecrt`. Download the software appropriate for your computer. (If you don't know whether you have a 32 or 64 bit computer, download one and if it doesn't install then download the other one.) This actually installs both SecureCRT and SecureFX, which is just fine because both programs are useful.

After the installation is complete, open the SecureCRT program and you'll get a window that looks something like Figure 3. Click the New Session button at the top, and it brings up a "New Session Wizard". Use SSH2, type `corn.stanford.edu` into Hostname, and your SUNet ID into Username (Figure 4). Use SFTP as your FTP protocol, then name the connection whatever you like and give it a description if you want. After you confirm, your new connection should appear in the list shown in Figure 3, and you can double-click that to open that connection. It brings up a dialog for you to enter your SUNet password. I suggest not saving the password. After you enter the password, you are in!

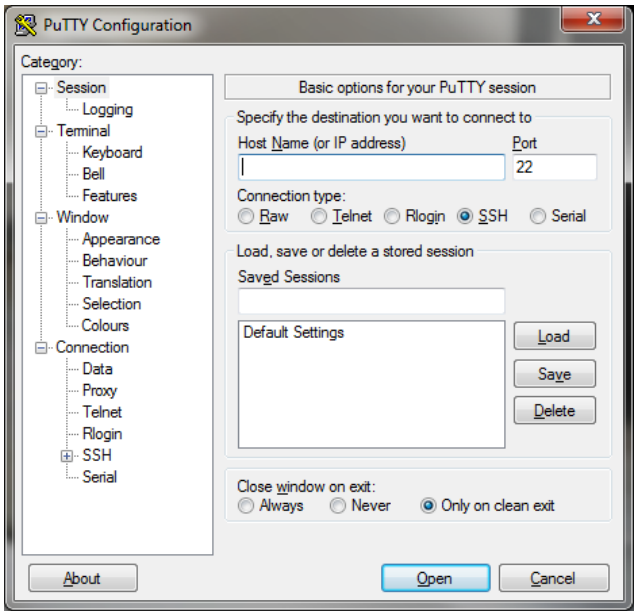


Figure 2: What you see when you open PuTTY

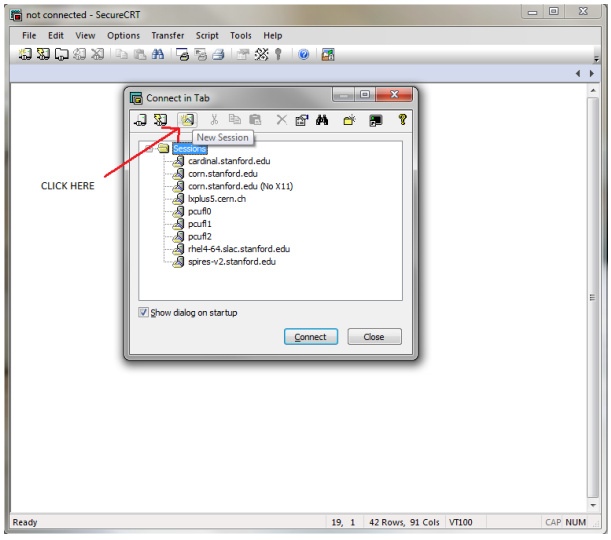


Figure 3: What you see when you start SecureCRT

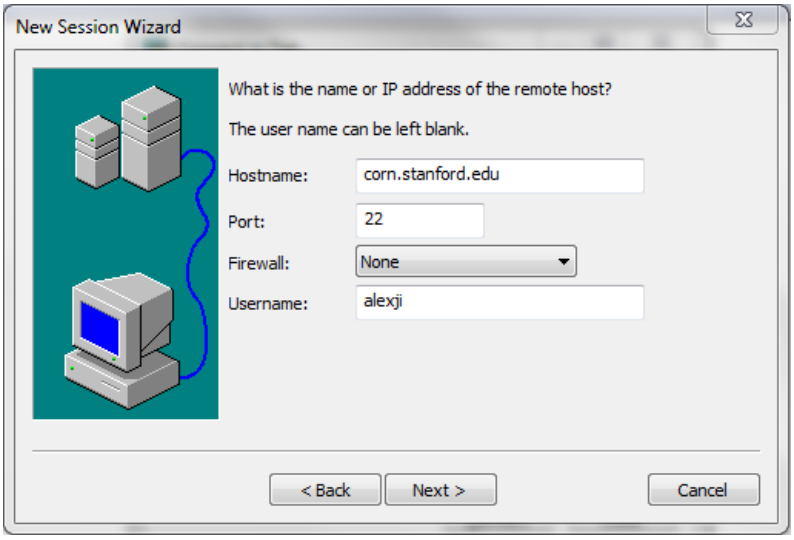


Figure 4: Creating a new session in SecureCRT. Use SSH2 as your SSH protocol and SFTP as your FTP protocol.

3 Installing an X11 Server

3.1 What's an X11 Server?

An SSH client allows you to connect to remote computers, but by default the only way you get information back from the computer you've connected to is through the terminal. This is not that great for beginners, since using the terminal for things like emacs is a bit counterintuitive (for one, your mouse doesn't do anything). It's also not good when you use plotting programs, since plots are usually shown in separate windows that aren't terminals.

The primary way to create windows in UNIX that aren't the basic terminal is through something called X11. To display these windows on your computer, you'll have to enable something called X11 forwarding, which allows the remote computer to send you X11 packets that display things on your computer. You'll need to install an X11 server on your computer to interpret these X11 packets. The most common of these is from a package called Cygwin.

X11 forwarding will be good for using the emacs GUI as well as showing plots. However, it doesn't work well when you are not very close to the computer you are connecting too. In other words, if you're not on Stanford campus, using X11 can be very slow.

3.2 Installing Cygwin XWin Server

This section is based on this CS107 website by Matt Spitz: `http://www.stanford.edu/class/cs107/other/x11tutorial.htm`. However, it is updated because Cygwin updated to make life easier.

First, go to `http://x.cygwin.com/` and download `setup.exe`. Run this program and push next a lot. You want to "Install from Internet" and choose a root directory (the default is fine). Push next a lot more, and eventually you'll get to the Select Packages page (Figure 5).

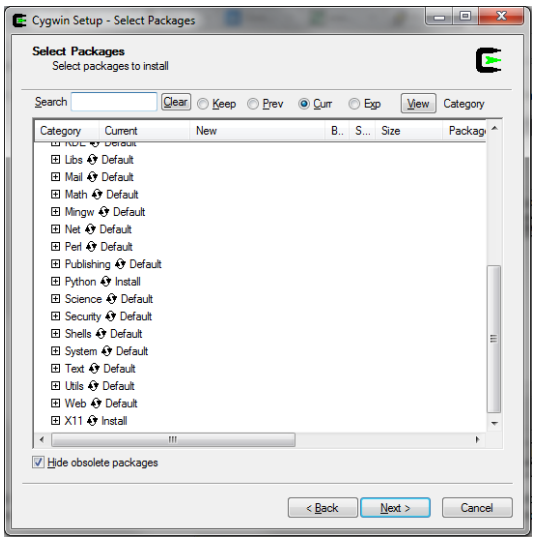


Figure 5: What you see when you open the Cygwin installer

Here, scroll down to the X11 category and click "Default" so it changes to "Install", as in Figure 5. Then click the "+" button next to "Net" and scroll down to the package called "openssh". Click the "Skip" button, and your screen should look like Figure 6.

Hit next again, and wait for Cygwin to install. It can take quite a while. [twiddles thumbs]

When you're done, you can leave the shortcut boxes checked, although I don't think you need to.

3.3 Using Cygwin's XWin Server

The full Cygwin package actually allows you to emulate the UNIX environment on Windows. It doesn't actually duplicate the underlying UNIX structure on your computer, it's just a collection of Windows packages that mimic UNIX syntax and functionality. If you didn't understand that, don't worry, it's not very important.

The only thing we will officially use Cygwin for is to create an X11 server that lets us display X11 windows on our laptop. To do this, go to the Start Menu, find the Cygwin-X folder, and find the XWin Server icon (see Figure 7). Click that and you will have started an X11 server. It's that easy! You can make a shortcut to this on your desktop for convenience.

3.4 Enabling X11 Forwarding for PuTTY

When you open PuTTY and get the PuTTY Configuration screen (Figure 2, go to Connection -¿ SSH -¿ X11 and click "Enable X11 forwarding". You're done! (It's probably a good idea to save this as a session so you don't have to check it every time.)

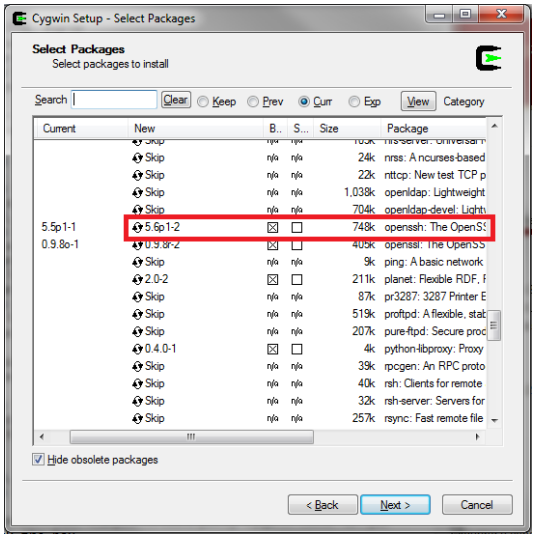


Figure 6: About what your screen looks like when you have properly selected openssh

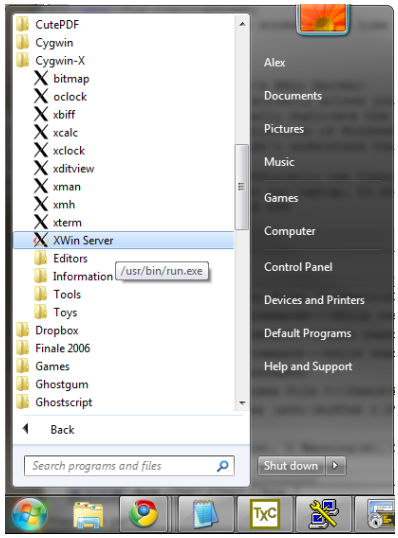


Figure 7: Where to find XWin Server. You can also see the background of the .tex I am using to write this document because of the wonderful translucent Windows 7 Start Menu.

3.5 Enabling X11 Forwarding for SecureCRT

In the Connect menu you first get when you open SecureCRT (where you have a list of connections), right-click the connection you want and click Properties. Go to Connectine -¿ Port Forwarding -¿ Remote/X11 and check "Forward X11 packets". Click OK, and you're done! Note that you have to do this for each different connection you create that you want X11 forwarding on. I actually have two connections to `corn`, one with X11 enabled and one without.

4 Open Emacs

Now try logging into `corn` from your SSH client with X11 enabled and type `emacs &` into the command prompt. After some lag (a LOT of lag if you're not on Stanford campus), you should get a popup emacs window. You can start typing some stuff, and you can click around the menu bar at the top. We'll talk a little bit about this at the first class.

5 Other Useful Stuff

5.1 FTP Clients

An FTP Client is a program that lets you transfer files between your computer and another computer. In particular, at some point you'll probably want to transfer files between your computer and the `corn` computers. SecureCRT comes equipped with SecureFX, which is a good FTP client. You can also download FileZilla as a free FTP client. The PuTTY webpage has one too, but I've never used that before.

5.2 Color in SecureCRT

If you've tried both of the clients, the first thing you'll notice is that PuTTY sometimes gives pretty colors while SecureCRT does not. You can change the default settings of SecureCRT to allow colors. Go to Properties for the connection you want to fix, then go to Terminal -¿ Emulation and change your Terminal to Linux. Check the "ANSI Color" box, but DON'T check the "Use color scheme" box unless you know what color scheme you're using.

This doesn't look quite as pleasing as the PuTTY interface, but it still works.