

Remotely logging on to a UNIX system using Putty

Written by [Dave Pawlowski](#), September 4, 2012

Introduction

This course is sort of odd. One of the main skills that I want you to learn is how to work with a Linux based operating system. The reason this is odd is that we will be using Windows machines. Hopefully that says something to you about how useful Linux can be, since we're going through the trouble. Anyway, the way that you will learn UNIX is by remotely logging into a Linux machine from these windows machines. This is what we will learn today.

Now this all may seem funny to you, but the truth is one of the best things about Linux is the ease in which you can perform remote operations. Honestly, that's a lot of what Linux is all about. In fact, using the software that you will learn about in this document will make it seem like you're sitting in front of those systems anyway. And the best part is, the software is free! Which is good, because chances are you are going to have to do some homework in this class, or complete a project or two. In order to do these things, you have a couple options:

1. You could break in to this lab and risk being arrested and expelled
2. You could remotely log in to one of the Linux systems on campus and work from home, a coffee shop, the library or where ever.

I suggest option 2. Now, if you happen to be so lucky that you have a Linux computer at your disposal, which includes a Mac, then this is simple. Linux was designed with remote operations in mind. We will cover what to do in that case later. If you are a windows user, then you need a few things. The first of which is a program called PuTTY. PuTTY is what's called a ssh client, and it's free. The ssh stands for secure shell, which is the current standard system for performing remote operations. It is also the Linux command that you would use if you want to log into another machine.

PuTTY basically turns a Windows machine into a Linux one, and allows you to open up a "terminal" on a remote Linux machine. Now, that's all fine and everything, but PuTTY doesn't do everything. See, if you were sitting in front of Linux machine, you would first open up a terminal and then start doing stuff with it, which would probably cause all sorts of interesting graphical windows to open up. When you use PuTTY alone, this would not happen, and you would be limited to working on everything in a single

terminal window. This is fine, but doesn't make use of all the tools at your disposal very effectively. The reason for this is that the graphical interface software on a Linux machine is different from that on a Windows machine.

However, all is not lost. A couple pieces of software allow you to configure your Windows machine so that it can run these graphical programs. This is what is covered in this document.

If your computer does run Linux, you will only need to do this when you use one of the computers in this room. Otherwise the following command will be useful:

```
% ssh -Y username@computername.emich.edu
```

where 'username' is the username that gives you access to the computer computername (for example chuck, emunix, or yipe), and "%" is a generic terminal prompt.

Background Information

The Linux graphical windowing system is called [X11](#), which is also known as X Windows, or most commonly X. On a Mac, you will find the X11 Application in either the Applications folder or one layer deeper in Utilities. Note that X Windows has nothing at all to do with Microsoft Windows.

One feature of X is called "X forwarding", which basically means that a graphical program runs on one computer, but the user interacts with it on another computer. Think Microsoft remote desktop, only on a program-by-program or window-by-window basis. If both computers are running Linux, this whole process is managed in the background and all you have to do is add that -Y flag as shown above. However, if you are working at a Windows machine, all you need to be able to interact with the Linux graphical programs are an X server that runs on Windows, and an SSH client, both of which are freely available.

An X what? And an SSH who?

Client and server are terms that are thrown around all the time in the world of web design and web applications. For us, in the context of X, the client is the computer running the program and the server is the remote computer that you're sitting in front of. This is actually backwards from normal web usage, typically the server is the remote computer that's serving you.

An aside on web stuff

For example, the emich web page is hosted by a server on campus. When you access the page at home, you are not sitting in front of the server. Web designers have to develop their software while keeping in mind the advantages of running programs server side

or client side. Server side programs have the advantage that whoever developed them knows that the programs will generally run on any system, because the software that runs them is actually on the server. If you're familiar with web programming, PHP code is typically executed server side.

Programs that are run client side require the user to have relevant software installed on their local machines. JavaScript is probably the most obvious client-side language. You have all probably experienced something on the web not working because you didn't have JavaScript turned on, or it wasn't up to date, etc.

Some software you need

SSH Client

We will be using the free and widely used Windows program PuTTY as our ssh client. You can download it [here](#).

PuTTY doesn't need to be installed; you just download it and put it somewhere, like the desktop or in "Program Files". Once downloaded, just double click the icon, shortcut, or whatever to open it up.

X Server and Fonts

The standard X.org X server has been ported for Windows; it's called [Xming](#). There are two files that you need, and they do actually need to be installed (this has already been done on the Sherzer computers). Here are the direct links:

- [The X server](#)
- [Fonts for the X server](#)

Yes, you do need both of these.

Both of these packages have Windows installers that you should be able to get through on your own, and I suggest you install them both into the same directory, which is the default option anyway.

Starting Everything Up

Start Xming

The first thing to start up is Xming and you want to do this by choosing XLaunch in the Xming folder; you should have a start menu item for it. If you're using a local firewall, you may be asked if you want to unblock it. You don't have to. When XLaunch starts, you will get a Display Settings window where there are a few options to choose from.

Stick with Multiple Windows then click next and you'll see a Session type. Select **Start no client** and then click next again. After a couple more "nexts" you will get to a Finish configuration screen. Here, you can save your configuration to a file that you will then be able to double click in the future to get Xming started, without having to go through all these steps. /bf Do This!

Also note that when Xming is started, there will be an "X" icon in your system tray. If you hover the mouse over it, it should say something like "Xming server - 0:0". If the last bit isn't "0:0", make a note as you'll need that in a bit.

Start PuTTY

Double click on the PuTTY icon to start up PuTTY. You'll notice there are all sorts of configuration options, but don't worry about those, we'll only use a few. You can do all sorts of cool things like change the color of the terminals that you open and use a specific font size and type. As you get the hang of using PuTTY feel free to change these at will. For example, I was always a bit preferential to the blue background and yellow lettering. However, as I gotten a bit older, I've found that my eyes can't handle the Michigan colors anymore.

The first screen is the one that tells PuTTY what computer system you would like to log into. This information goes in the **hostname** dialog box. A hostname is basically the name of a computer. You are familiar with hostnames in the context of web URLs. The first part of a URL, after the http://, but before the next "/" is the hostname. Typically, it is comprised of the host's local name (the name of the Mac in my office is chuck) and the name of the domain (such as emich.edu). If you put the two together, chuck.emich.edu, you are providing a specification that is uniquely defined on the internet. In other words, by putting that domain in the address box in Firefox, you will access my computer.

Sort of. Actually, it's the computers IP address that uniquely defines a computer on the internet. The nice name, like www.google.com, is so people can actually remember where to go. Its much easier that remembering a series of 11 or more numbers. The name of a computer is mapped to an IP address using something called the Domain Name System (DNS).

Ports

Now when you try to access my computer, the program you are trying to access it with tries to do so using something called a port. For example, when Firefox tries to access my computer (when you type in chuck.emich.edu and hit enter), it tells my computer it wants access to port 80. The reason I mention this is because back in PuTTY, you'll notice a dialog box for **port**. This allows you to specify which port you want to try to access. Think of ports in the context of shipping. Every ship that comes into Boston Harbor, for example, must go to a specific port (or they will be yelled at by customs). Ports allow remote access to a computer to be done securely. When trying to remotely log in to a computer using SSH, as you are doing with PuTTY, it is usually the case (99.9% of the

time) that you will want to access port 22 on the remote machine. This is the default in PuTTY, so don't worry about changing it.

I mentioned security. If you try to log in to pretty much any computer on campus from off campus, it won't work. The reason is that IT security at Eastern really doesn't like SSH, and so they have setup a firewall that blocks port 22 on every computer. Actually IT security have blocked access to every port on chuck (and most every other computer), except port 80 (HTML). If you are on campus, then you are within the Eastern firewall, and thus you just need to have access to whatever system you want (have a username and password) i.e. chuck.emich.edu, and you will be able to do so. Chuck is an exception to this rule, as I was granted special permission to allow remote access. So if you are off campus, you can log directly in to chuck.

So now that you have something in the **hostname** box, probably chuck.emich.edu, and are specifying the correct port, 22, you could go ahead and click **Open** to log on. If you do that, you'll get access to chuck, but you won't be able to use any graphical programs. First, you need to tell PuTTY that you want to allow X11 forwarding. In other words, you want to see those graphical programs!

On the left sidebar in PuTTY, click on the Connections/X11 sub-panel. Then click **Enable X11 Forwarding** and in the **X display location** dialog box, put "localhost:0.0" (unless you didn't have the "0:0" in the section above, then use the numbers that you noted there instead). Leave the **MIT-Magic-Cookie-1** radio button checked.

Now click back to the main **Session** panel on the left. Put a name that describes the system you are accessing in the text box in the middle right below "Saved Sessions" (such as: chuck). Click "Save". From now on, you can double click that saved session and you will skip having to mess with the configuration options.

Press **Open** to start PuTTY. Put in your username and password and hit enter, and the next thing you know you should be logged in. To test that Xming is doing its job, type the command "xcalc &". A little calculator should pop up that you can interact with.

Tips

- It's important to start Xming **before** you try to start the graphical program you want to run. Otherwise, there won't be anything on your end to draw the windows!
- Putting an ampersand ('&') after any command puts it in the background: i.e. "emacs &" will start emacs and let you continue to use the PuTTY terminal. This is not just a PuTTY thing, its a Linux thing, but it bares mentioning here.
- Any programs that you start will close (without saving!) if you close the PuTTY session. So, you know, be careful. This isn't Microsoft Word that saves everything every 2 seconds.

- If you need another terminal, instead of running another PuTTY session, try running “xterm &” in your current session. xterm is the Linux command that starts another terminal.
- Remember, you are working remotely, which means that you are dependent on your internet connection for everything you do. Opening up graphics-heavy programs (i.e. a web browser (don’t do that)) or having lots of terminals open will cause things to be slow. In most situations, you will use a terminal or two, a text editor (emacs), and maybe a plot or two running, and as long as your internet connection is decent, things should happen relatively quickly.