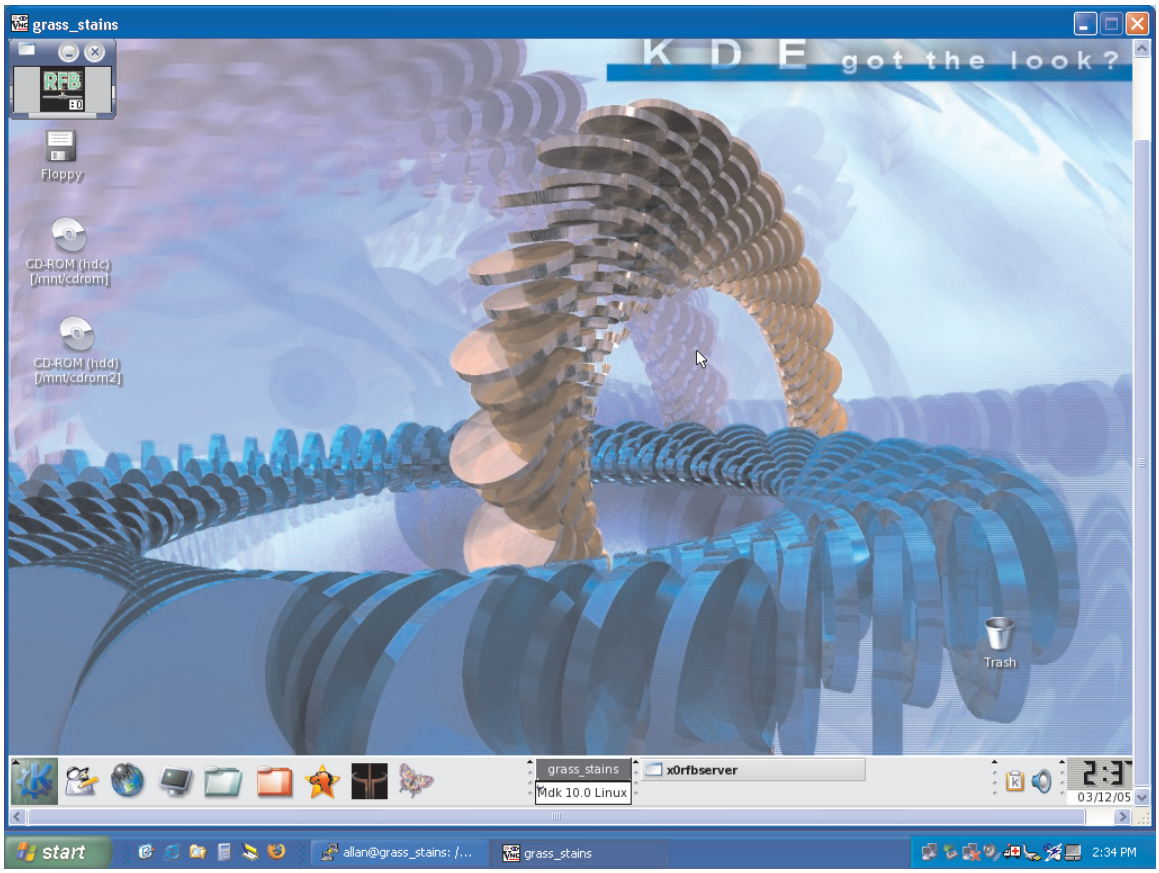
VPN-ssh

by Al Brown allan@allan2.com

Bringing Your Linux Desktop Along

Use Windows or Linux Clients



Take Your Desktop Along With You

The ability to take your desktop along with you may be mostly a parlor trick, but occasionally can be quite handy. If however this convenience opens your system up to snoops and hackers, any advantage will swiftly be overshadowed. Therefore, we will make use of a combination of a common but insecure VPN (Virtual Private Network) solution along with the port tunneling capabilities of ssh, the secure shell. Put simply, we will use ssh to communicate via an encrypted connection which is very hard to spy upon, and the information that we pass over this connection is the VPN providing our home desktop to wherever we happen to be. Since most Linux installations will by default have both ssh and VPN capabilities, this explanation will focus mainly on using a Windows client to connect to your Linux desktop.

Getting Started

The place to start is on your Linux machine. Ssh must be running and your network setup must allow for outside access to port 22, which is where the ssh daemon listens for connections. A quick check at the command line with ps ax <enter> should show a sshd process running. If not, check to see that an openssh package has been installed. Remember to adjust your firewall settings appropriately. While you are on this machine you should note the ip address of this machine that is seen by the outside world. This is not necessarily the address on the local network. An easy way to check for the correct address is to use your browser and go to any internet site that provides whoami info. If you are using an ISP that provides you with a dynamic address, you may have to check this regularly as it can change. Unfortunately, the added cost of a static address is usually not attractive.

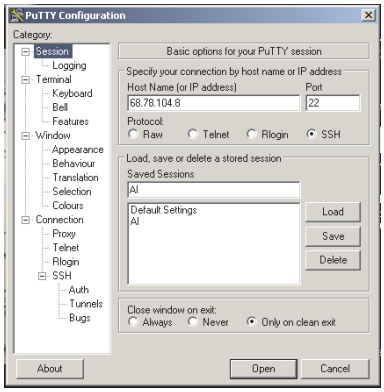
The VPN Server

Next, a check for a VPN server is in order. This is where you need to know your acronyms. The software to serve up or display your desktop can be called a VPN (Virtual Private Network), VNC (Virtual Network Computing), or RFB (Remote Frame Buffer) server. Actually there are even more options than this, but this example is going to stick with the remote frame buffer option. The RFB server uses the active X server display which is typically :0. A VPN or VNC on the other hand will open another display which will be referred to as :1 or some other number. Communication is typically via port 5900. But before we get too distracted, the software choice here is x0rfbserver. You may have a GUI interface to x0rfbserver or you can start it at the command line with something like /usr/bin/x0rfbserver or /usr/bin/x0rfbserver :0. I happen to be using a Mandrake 10.0 system, so starting the RFB server is done via the Virtual Network Connection GUI, which prompts you for a password for the server. On other systems you may have to use vncpasswd at the command line to set the password for the server.

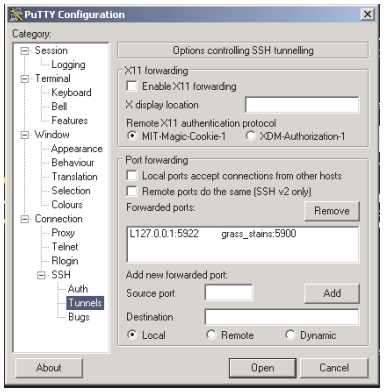
Windows Components

Now, let's grab the components for the Windows side. Putty is an open source ssh client for Windows. It can be found at http://www.chiark.greenend.org.uk/~sgtatham/putty/. The latest release at this time is 0.57. The download of interest is putty.exe. This executable file can be dropped directly on the desktop for easy access. Launching this program brings up a small window where you need to add the ip address of the machine that is the ssh server. This is the address that we noted earlier. Now, in the category section, go to Connection>SSH>Tunnels. Here

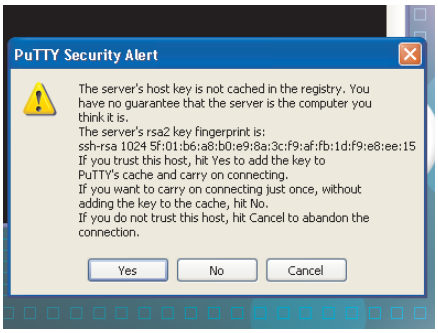
The VPN and Putty icons on the Window's desktop.



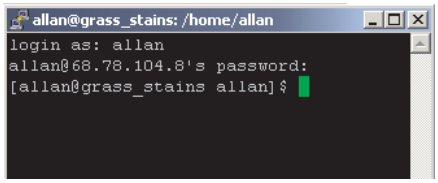
The main Putty window, where you can save numerous sessions.



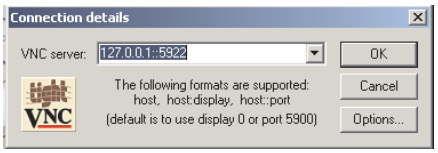
The area of Putty for entering port forwarding information.



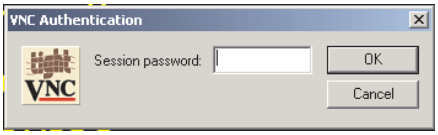
Connecting on a Windows machine for the first time to your Linux server via ssh with the Putty client brings up this security dialog.



The ssh connection is complete. Use the command line to navigate the system, or minimize this window and start the VPN for a graphical interface.



The main TightVNC window. This is where you enter the loopback address and an unused port number.



Session password is easy to forget since it is not your login password, but one specifically for the VNC server.

you enter in the port forwarding area labeled Source Port, the loopback address 127.0.0.1 and an unused port address separated by a colon. I have used 127.0.0.1:5922. In the destination section you must enter the name or address of the machine that is running your desktop. This time it is the local network name/address. To this we add a colon and the port number that the VPN uses. Here I have used grass_stains:5900. (grass stains was my name in Quake 2.) Once you have entered this information, remember to click add. Now you can go back to the Session area and give this configuration a name and click save. The next time you launch Putty, choose the name of the session and click Load. To open a connection between your home server and your remote machine, click open.

For the Windows VPN client, TightVNC is an open source solution. It is available at http://www.tightvnc.com/download.html. Start with just the viewer executable which is only 155 kbytes for tightvnc-1.2.9_x86_viewer.zip. From this zipped file, extract vncviewer.exe to the desktop. Launch this program and in the box named Connection details, enter the loopback address and the port number that you used in the port forwarding area of Putty. Be sure to separate the address and port number by two colons. I have entered 127.0.0.1::5922. Close this app for now. Your entries will remain.

Time to Test

So at this point we should be ready for a test. With sshd and x0rfbserver running on the home machine, we launch Putty on the remote Windows machine which of course has a connection to the internet. Load the previously saved session and click open. If everything goes well, there should be a message about the RSA key of the server not being stored in the Registry. We can choose to store the key or just connect once. At this point, we should be prompted for a login name and then for our password. With the correct entries we should have the command prompt on our home machine. Now we minimize the Putty window and launch the VNCviewer. Click okay on connection details and then enter the password for the VNCserver (x0rfbserver), and click okay again. This is the moment of truth. After a short delay we should see our home desktop. The response time depends on our connection speed and how the network is configured.

At this point we can use any program on our desktop system just as though it was local, except that there will be a fair amount of lag in screen changes. So, this is not meant for graphically intensive applications such as action games. However, if you want to find a piece of information that

you have at home, or write or read your home email, or run an application that you only have on your home machine, you are in business.

Moving Files

Perhaps now you need to copy a file from your home machine to your remote machine. Ssh is not particularly meant to be used for transferring files, and copy and paste is not going to work between our home machine and our remote machine, so we will get some help from the Putty Secure Copy client, PSCP. PSCP is used from the command prompt and run from the directory where it resides. So to copy the file /home/text from the server grass_stains as user allan to the file c:\temp\text.txt, you would type: pscp allan@grass_stains:/home/text c:\temp\text.txt <enter>. This action is separate from your VPN and any running Putty ssh session, so you will be prompted for your password. File transfer will follow. Of course, this really isn't going to work. The reason is that grass_stains is not a fully qualified domain name. So in order to get the file, grass_stains should be replaced by the internet address of our server that we used before. Now it works.

End Session Cleanly

When you have had enought of this fun, make sure that you type exit <enter> at the ssh prompt so as to exit cleanly from your Putty ssh session.

The Linux Connection

I have not said anything so far about connecting from a Linux machine. Here you already should have the ssh client and you can use a VPN client like TightVNC. Or since you have an X client available, you may want to use the X forwarding capability of ssh. If X forwarding is enabled, starting an application on the server will result in it running in a window on your remote machine. Start by connecting with ssh at the command line and experiment from there. In this example I would connect by entering ssh -L 5922:grass_stains:5900 68.78.104.8 <enter> at the command line. This is of the form ssh -L port:host:hostport hostname, or as I would say ssh -L fakeport:sshservername:realvpnport serveraddress.

The Last Word

If you are looking for this type of technology for a business setting where security is extremely important, I would suggest looking into some of the SSL solutions which are considered to be even more secure that ssh. However, for personal use, the techniques that I have outlined here of tunneling information through a secure shell connection can afford you a reasonable secure, cost effective, and easily implemented solution. Enjoy!

Resources:

http://www.uk.research.att.com/archive/vnc/sshvnc.html - Making VNC more secure using SSH
http://www.uk.research.att.com/archive/vnc/sshwin.html - SSH-protected VNC: the case of the Windows client and the Unix server
http://www.chiark.greenend.org.uk/~sgtatham/putty/ - Putty
http://www.tightvnc.com/download.html - TightVNC
http://the.earth.li/~sgtatham/putty/0.57/htmldoc/ - Putty User Manual
http://www-128.ibm.com/developerworks/library/l-keyc.html - ssh key management