COMP220 - VNC Sessions and SSH Tunnels

James *Logan* Mayfield

1 Unencrypted VNC Step-by-step

The follow steps will set up an insecure, remote desktop to the server.

1. (**Do only once**). Configure and launch your VNC desktop server
   (a) Log in to the server using SSH.
   (b) Launch your VNC server session:

   ```
   (no frills. default options)
   > vncserver

   (set desktop size to 1024x768. Can use other sizes. )
   > vncserver -geometry 1024x768
   ```

   Read the *vncserver* man page for more options. There are a few things you need to look for when launching your desktop server:

   - The very first time you launch *vncserver* you'll have to set a password. This password is *only for logging in to your vncserver session* and for security reasons ***should not match the password you use to log in with ssh.***. You can use the command *vncpasswd* to change this password (read the man page).
   - When you launch your desktop the resultant output will tell you a new desktop is launched and call it something like *cs.monm.edu:1*. The number, in this case 1, is your *desktop number*. The name is just the name of the server itself. ***Remember your desktop number***
   - If you ever want to end your desktop session, then you can use the *-kill* command. If your desktop number is 3, then

     ```
     > vncserver -kill :3
     ```

     will shutdown the vncserver. In general, you can leave the session running indefinitely.

   (c) Once you've launched your vnc server you can exit your SSH session. The VNC server will stay running in the background.

2. Logging in to the desktop using RealVNC.
   (a) Locate and launch RealVNC viewer. (Can be downloaded for free from `http://www.realvnc.com/download/viewer/`).
   (b) If your desktop number is n then the server you're connecting to is ***cs.monm.edu:****n*. When prompted give your *vnc password, not your SSH password.*

At this point you should see a new window showing a desktop. The desktop environment is called LXDE. It's meant to be lightweight and therefore easy to push over a network. The bird looking icon in the bottom left is like the Windows start menu, it'll let you find and launch programs graphically. When in doubt, you can still pull up a terminal and use the CLI. It's *XTerm* under "System Tools".

2 Securing VNC with SSH

There are two reasons to run your VNC server through an SSH tunnel.

1. VNC traffic is unencrypted and data will be sent across the web in the clear. This is why you should not use the same password for VNC and SSH. If you're a security conscious user, and you should be, then you should want to secure your VNC session.

2. Open, insecure VNC sessions have a tendency to fall prey to brute-force login attempts. This means someone will invariably try common username and password combos on your session in order to gain access to your session and thereby the server. It's unlikely they'll guess your username and password. So, the result of these attacks is effectively a Denial of Server. The VNC server will lock itself down for a very long period. This causes you to have to manually kill the server and restart. If you want to avoid this, then secure your session with the instructions below; they effectively make your VNC server invisible to the web.

The following instructions allow you to setup a secure vnc session using Windows and PuTTy to establish SSH connections.

1. On the server: Configure and launch your vnc sever for localhost only connections. (Again, do only once)

   ```
   > vncserver -localhost -geometry 1024x768
   ```

2. On the client: Setup an SSH tunnel with PuTTy.

   (a) Go to Connection > SSH > Tunnels. If your desktop number is n then your desktop port number is $59n$ with n expressed in two digits. For example, desktop 3 has port number 5903.

   - Source Port: $59n$ (i.e. your desktop port. This can be other numbers, but this is as "safe" choice)
   - Destination: *localhost:*$59n$. The number after the colon must be your desktop port.

   You'll probably want to save the settings for this PuTTy Connection so that you don't have to repeat this step every time you connect.

   (b) Using PuTTY with the tunnel configuration, connect to server and *leave the SSH connection open* in the background.

3. Using your VNC client, connect to the server but rather than *cs.monm.edu:n* use *localhost:59n* or whatever the destination for your tunnel happened to be.

Tunnels in *nix and OS-X

If you're using *nix or OS-X, then an SSH tunnel can be setup via the command-line. Let's say your server username is *joeshmo* and your VNC desktop number is 10. Then, on your personal machine, you'd use the following terminal command to establish an SSH tunnel to your vnc server:

```
ssh -Nf joeschmo@cs.monm.edu -L 5910:localhost:5910
```

Setting up the VNC server itself is the same. Just use the *-localhost* option. For more details on SSH servers read the *ssh* man page. You might also check out the program *autossh* which can be used to setup and maintain tunnels and SSH connections.