



- stands for Virtual Network Computing
- it allows you to remotely view and control a PC desktop, and will run on a wide variety of operating systems
- VNC and Security
 - access to a VNC desktop means security is very important

VNC and security

- VNC uses encryption when making an initial connection
 - when you login

- however once connected
 - all VNC data is unencrypted
 - a malicious user could snoop your VNC session

VNC and security

- a number of VNC scanning programs exist
 - which will scan a subnet looking for PCs which are listening on one of the ports which VNC uses

- tunneling VNC over a SSH connection allows you to use VNC to access your computer
 - with all traffic strongly encrypted, and optionally compressed

Running VNC

- log into mcgreg and run vnc4server

```
■ $ ssh mcgreg.comp.glam.ac.uk  
Password:  
$ vnc4server  
You will require a password to access your desktops.  
  
Password:  
Verify:  
  
New 'X' desktop is mcgreg:1  
$ exit  
Connection to mcgreg closed.
```

Running VNC

- you will be prompted for a password, note this does not have to be the same as your login password

- note the New 'X' desktop is value, write this down

- now run the vncclient from your start bar menu
 - you will need to give it the server name
`mcgreg.comp.glam.ac.uk:1`
 - note that `mcgreg.comp.glam.ac.uk:1` was used in this example, yours might be `mcgreg.comp.glam.ac.uk:6` you need to check the output from when you ran `vnc4server`

Running VNC

- you will be prompted for your VNC server name
 - in this example it was: `mcgreg.comp.glam.ac.uk:1`

- now you will be prompted for your VNC password

- how many `vnc4servers` are being run on mcgreg?
 - hint use the command `ps -aux` and `grep` and `wc` to find this out

- now shutdown your `vncclient` and log into another GNU/Linux workstation and run `vncclient` again

VNC over SSH

- as VNC is insecure we can tunnel VNC traffic over SSH
 - VNC becomes as safe as SSH

- firstly we need to connect a secure pipe to the vnc4server running on mcgreg

- we initiate this from our GNU/Linux workstation

- ```
$ ssh -g -A -X -N -T -L5901:localhost:5901 mcgreg.comp.glam.ac.uk
```

## VNC over SSH

- now we start our `xvncclient`
  - however we tell it to connect to `localhost:1`
  
- which maps onto `localhost:5901`, which is the other end of the secure pipe
  
- does the connection appear slower?
  - why?



## Tidying up

- please shutdown your `vnc4server` once you have finished this tutorial

```
$ ssh mcgreg.comp.glam.ac.uk
Password:
$ vnc4server -kill :1
```

- remember in your case `:1` might be `:6` as defined when you created the server instance
- finally examine the `ssh` manual page and find out what the parameters `-g -A -X -N -T -L5901:localhost:5901` mean when given to `ssh`