

## **Workshop 6 – part 1: Configuring SSH -Tunneling X**

### **Intro:**

The SSH server can then forward the X data it receives to the SSH client, which is presumably running on the same computer as the user's X server. This configuration is much simpler to manage than is tunneling most other protocols.

### **Recipe:**

Add two Machines as a server and client to your working environment and follow the below steps:

**1- install openssh-server on the Server machine.**

```
[ root@server ~] # yum install openssh-server
```

**2- In terms of configuration, you must ensure that the server's /etc/ssh/sshd\_config file includes the following line:**

**X11Forwarding yes**

Note: If this line reads no rather than yes, change it to read yes.

**3- Restart the server for changes take effect.**

```
[ root@server ~] # /etc/init.d/sshd restart
```

**4- install openssh-clients on the client machine.**

```
[ root@client ~] # yum install openssh-clients
```

**5- On the client side, the /etc/ssh/ssh\_config file should have the following option set:**

**ForwardX11 yes**

**6- On your client, connect to your server. Be certain to tell ssh to allow X11 forwarding.**

```
[ root@client ~] # ssh -x root@server-ip
```

**7- now, on your client test the below commands.**

```
[ root@server ~] # nautilus
```

```
[ root@server ~] # firefox
```

## **Workshop 6 – part 2: Configuring SSH - Logins without Password**

**Intro:** This exercise guides you through the process of enabling SSH logins without passwords. You will require access to two computers, an SSH client and an SSH server. The SSH server must already be configured to accept logins.

**1-** Log into the SSH client system as the user who will be performing remote access.

**2-** Type the following commands to generate a version 2 SSH key:

```
$ cd ~/.ssh  
$ ssh-keygen -t rsa id_rsa
```

**3-** Step 2 generates two files: id\_rsa and id\_rsa.pub . Transfer the second of these files to the SSH server computer in any way that's convenient — via a USB flash drive, by using scp , or by any other means. Copy the file under a temporary name, such as temp.rsa, to ensure you don't accidentally overwrite a like - named file on the server.

**4-** Log into the SSH server system. If you use SSH, you'll need to type your password.

**5-** Add the contents of the file you've just transferred to the end of the ~/.ssh/authorized\_keys file.

**6-** Now login to the server to check the connectivity.