VNC

NOTE: for the following examples, 192.168.1.34 will be the REMOTE machine (VNC server).

Setup VNC Server (on Slackware)

USING TigerVNC: On your remote machine, install fltk, then install TigerVNC via slackpkg

```
   slackpkg install fltk
   slackpkg install tigervnc
```

Start the VNC server:

```
   vncserver
```

You should get a message stating that the VNC server has started on DISPLAY 1.

NOTE: VNC listens to port 5900 or 5901 by default so make sure you can accept connections from that port.

NOTE: If you are using TigerVNC, the server can also listen on port 5800 for web-browser VNC connections.

Setup VNC Viewer (on Slackware)

USING TigerVNC: On your local machine, install fltk and TigerVNC via slackpkg

```
   slackpkg install fltk
   slackpkg install tigervnc
```

Start the VNC viewer:

```
   vncviewer
```

Connect to your VNC server:

```
   In the "VNC server:" box put the IP address/URL of your VNC server and
the display number:
   192.168.1.34:1
```

Tunnel VNC Through SSH (from Slackware)

Open an SSH connection:

```
   ssh -L 5901:localhost:5901 -N -f -l bob 192.168.1.34
```

```
Where,
    -L 5901:localhost:5901 = Connections to local port 5901 is forwarded to
remote port 5901 on the remote machine.
    -N : Just forward ports, do not execute a remote command.
    -f : Make ssh to go to background before command execution.
    -l bob : 'bob' is the username.
    192.168.1.34 : is the remote machine (target VNC Server).
```

Connect with VNC:

```
   vncviewer
   Enter "localhost:1" (or "127.0.0.1:1") into the 'VNC server:' box
```

Tunnel VNC Through SSH (from Windows with Putty [V0.62])

Install Putty on the LOCAL machine.

Open putty and set the following values:

```
   Session >
   HostName: 192.168.1.34
   Connection Type: SSH
   Connection > SSH > Tunnels >
   Source port: 5901
   Destination: 192.168.1.34:5901
```

Start the connection by hitting the OPEN button.

Connect with VNC:

```
   vncviewer
   Enter "localhost:1" (or "127.0.0.1:1") into the 'VNC server:' box
```

Reverse VNC Connection

(Connecting to a listening VNC viewer (Useful for family IT support))

Start The VNC Viewer (on Slackware) In Listen Mode-

```
   vncviewer -listen 5500
   WHERE: 5500 is the port to listen on (5500 is default).
```

IF YOUR INCOMING CONNECTION IS COMING THROUGH A FIREWALL (most home routers) do not forget to forward the port in the router (5500 is the default port for reverse VNC connections).

Start the VNC SERVER on Windows and attach a listening viewer-

```
   Start the VNC Server
   RIGHT CLICK on the VNC server icon in your task bar and select "ATTACH
LISTENING VIEWER"
   Put in the IP address of the viewer machine: 192.168.1.34
   **NOTE** - If the incoming connection is coming into your home network
from the internet
              (through a forwarded port), make sure to connect to the
ROUTER'S address (the
              internet address) and NOT your local address. This can be
found by going to a
              "whatsmyip" website with a broswer.
```

Starting ''Xvnc'' on-demand

A nice refinement is to configure the remote host to automatically create for each user a permanent virtual terminal.

The virtual terminal gets created when the user first connects. The user can detach from their terminal and reattach from elsewhere, providing true persistent terminal service. The terminal persists until either

(i) the user destroys it by logging out, or

(ii) the system operator destroys it by rebooting.

Remote host listener

The remote host needs an `inetd` running. Assign each user a display in `inetd.conf`. The lines in `inetd.conf` might look something like this:

```
5901 stream tcp wait arfon  /usr/bin/Xvnc Xvnc -Log *:syslog:30
```

```
passwordFile=/home/arfon/.vnc/passwd  -inetd -query localhost -once
5902 stream tcp wait metaed /usr/bin/Xvnc Xvnc -Log *:syslog:30
passwordFile=/home/metaed/.vnc/passwd -inetd -query localhost -once
5903 stream tcp wait doe    /usr/bin/Xvnc Xvnc -Log *:syslog:30
passwordFile=/home/doe/.vnc/passwd    -inetd -query localhost -once
5904 stream tcp wait roe    /usr/bin/Xvnc Xvnc -Log *:syslog:30
passwordFile=/home/roe/.vnc/passwd    -inetd -query localhost -once
```

This arranges that when a VNC viewer connects to port 5901, an `Xvnc` is started for user `arfon`, creating virtual terminal `:1`. User `metaed` uses port 5902 and gets terminal `:2`, and so on down the list.

If this is the first time you’ve used `inetd`, you might need to get it running. Example:

```
# chmod 755 /etc/rc.d/rc.inetd
# /etc/rc.d/rc.inetd start
# pgrep -l inetd
27849 inetd
```

Connection procedure

To use, there is a two step procedure. First, the end user connects using `ssh` to the remote host and tunnels their assigned port number to their physical terminal. In PuTTY, user `arfon` would go to Connection, SSH, Tunnels, and set Source to `5901` and Destination to `localhost:5901`. An equivalent `ssh` command line would be:

```
$ ssh -L 5901:localhost:5901 arfon@terminalserver.example.com
```

Second, the end user connects using `vncviewer` to the local end of the tunnel (`localhost:5901`).

I've seen at least one VNC viewer (bVNC) that has builtin support for `ssh` and combines these into one step, but haven't tried that.

Display manager configuration

This depends entirely on which display manager you have available, and is really outside the scope of this how-to. Here is just a simple example. Supposing you want to use bog-standard `xdm`. You would:

- Open `/etc/X11/xdm/xdm-config`.
- Edit the line `DisplayManager.requestPort: 0`.
- Disable the line by inserting a `!` at column 1. This lets `xdm` listen on its UDP port for session requests. As noted below under Security considerations, you do NOT punch a hole in your firewall for this port.
- Open `/etc/X11/xdm/Xaccess`.
- Edit the line `#* #any host can get a login window`.
- Change the `*` at column 2 to `localhost`.
- Enable the line by deleting the `#` at column 1. This lets only virtual displays local to your remote

host get a login window.

You might not want your `xdm` trying to manage the console. If that is so:

- Open `/etc/X11/xdm/Xservers`.
- Edit the line `:0 local /usr/bin/X :0`.
- Disable the line by inserting a # in column 1. This tells `xdm` not to try to manage the console.

You will also need to change the remote host's default runlevel from 3 to 4. This tells `init` to run your display manager by spawning `/etc/rc.d/rc.4`, and respawn it if it dies.

- Open `/etc/inittab`.
- Edit the line `id:3:initdefault:`.
- Change 3 to 4 in column 4.
- To make the change immediately without rebooting, run `telinit 4`.

For extra credit, you can customize `rc.4` by creating a file `/etc/rc.d/rc.4.local`. See `rc.4` for details.

After authentication, `xdm` runs the user's `$HOME/.xsession` script. You, or they, will have to provide one. A dirt simple one could be:

```
xterm &
twm
```

Security considerations

There are several layers of security available to protect the remote virtual terminals from unauthorized use.

Firewall

The first layer of security is the remote host's firewall. Firewall configuration is outside the scope of this how-to, but here is a brief outline of what you need. The firewall ought to deny remote connections by default. So if you already block all ports by default, no change is needed. You will NOT be punching holes in your firewall to make this work. Supposing your remote host provides only SSH service, and you use the `nftables` firewall, you probably have a rule chain something like this:

```
type filter hook input priority 0; policy drop;
ct state vmap { 0x1 : drop, 0x2 : accept, 0x4 : accept }
iifname "lo" accept
tcp dport 22 accept
```

With such a firewall in place, you are properly secured.

Encryption

Configured as recommended above, all VNC traffic between the remote host and the local terminal is encrypted, because it travels on the `ssh` tunnel.

Authentication

There are several authentication layers available.

1. An `ssh` key is needed to create the tunnel.
2. The terminal can be secured using `vncpasswd`, to prevent another authenticated user from “walking up to your terminal”.
3. The `-query localhost` option tells `Xvnc` to ask a display manager process for an assist. The display manager prompts the user to authenticate using a password, just the way a physical terminal would, before creating the user session.
4. The user can `xlock` the display before detaching. This will prompt for the user password.

References

- `Xvnc` man page (https://tigervnc.org/doc/Xvnc.html)
- `inetd` man page
- `ssh` man page
- `xdm` man page
- `init` man page
- `nftables` Wiki (https://wiki.nftables.org)

Sources

- Originally written by arfon
- Contributions by metaed

howtos, software, X Windows, X11, VNC, TightVNC, Putty, author arfon

From:
https://docs.slackware.com/ - **SlackDocs**

Permanent link:
https://docs.slackware.com/howtos:window_managers:vnc

Last update: **2023/03/13 18:26 (UTC)**