

Remote Desktop With P2P VNC

VNC is a well-known tool for remote desktop view and control. The two computers establish a TCP connection so that one of them can access the display of the other. However, almost always these two computers are behind a firewall/router and do not have a real IP to be accessed from the Internet. In such a case one of them can do **port forwarding** on the router/firewall and this would allow the connection to be established.

The problem is that more often than not, none of the parts that want to establish such a remote connection have access to the firewall that separates/protects it from the real Internet. Sometimes it can even be several levels deep behind the firewall (several layers of firewalls). However, if you have access to an external server (for example a server in the cloud) there is still a workaround and things can be fixed to work. For more details read the docs, which explain the tricks and hacks that can be used in such a case in order to establish a secure connection to a remote desktop.

This project has some scripts that simplify the installation of a P2P server, and the connection between the two clients.

1 Basic Installation And Usage

1.1 Installation Of A P2P Server

For more security and flexibility, the installation is done inside a chroot environment (built with debootstrap), or inside a container (built with docker).

Installation in chroot

```
cd /var/chroot/  
git clone https://github.com/dashohoxha/p2p.git  
cp p2p/install/settings.sh p2p/cfg.sh  
vim p2p/cfg.sh  
nohup nice p2p/install/chroot.sh p2p/cfg.sh &  
tail -f nohup.out
```

After the installation is done, you can start and stop the service with:

```
/etc/init.d/chroot-P2P stop  
/etc/init.d/chroot-P2P start
```

Installation in container

```
git clone https://github.com/dashohoxha/p2p.git  
cp p2p/install/settings.sh p2p/cfg.sh  
vim p2p/cfg.sh
```

```
nohup nice p2p/install/docker.sh p2p/cfg.sh &
tail -f nohup.out
```

After the installation is done, you can start and stop the service with:

```
docker start P2P
docker stop P2P
```

1.2 Sharing The Desktop (On The Client)

To share the desktop we need a VNC server. Make sure that **x11vnc** is installed:

```
sudo apt-get install x11vnc
```

Then get the scripts from GitHub:

```
git clone https://github.com/dashohoxha/p2p.git p2p-vnc
```

Set the P2P server that you want to use, for example:

```
p2p-vnc/client/use.sh p2p.btranslator.org
```

Now share the VNC port and start an **x11vnc** server.

```
p2p-vnc/client/start_x11vnc.sh
```

KEY: b8e1f1e779

Give it to the remote part in order to access your desktop.
To stop the connection run: p2p-vnc/client/stop.sh b8e1f1e779

The script will return a randomly generated key which you should give to the other part which needs to access your desktop.

There are more options to try. Change them at the config file **vnc.rc** or from the command line:

```
p2p-vnc/client/start_x11vnc.sh --help
```

Usage: p2p-vnc/client/start_x11vnc.sh [OPTIONS]

Share the VNC port and start an x11vnc server.
The options from command line override the settings
on the config file 'vnc.rc'.

--help	display this help screen
--vnc_port=<port>	set the VNC port (5900)
--window=yes	share a single window, not the whole desktop
--viewonly=yes	the desktop cannot be controlled remotely
--shared=yes	more than one computer can connect at the same
time --forever=yes	keep listening for more connections (don't exit when the first client(s) disconnect)

1.3 Accessing The Remote Desktop (From The Other Client)

To access a remote desktop we need a VNC client. Make sure that **vncviewer** is installed:

```
sudo apt-get install vncviewer
```

Then get the scripts from GitHub:

```
git clone https://github.com/dashohoxha/p2p.git p2p-vnc
```

Set the P2P server that you want to use, for example:

```
p2p-vnc/client/use.sh p2p.btranslator.org
```

Now connect to the remote VNC port and start **vncviewer**.

```
p2p-vnc/client/start_vncviewer.sh b8e1f1e779
```

You will need to know the number of the key that was created by the remote desktop.

See also the usage:

```
p2p-vnc/client/start_vncviewer.sh --help
```

```
Usage: p2p-vnc/client/start_vncviewer.sh [OPTIONS] [<key>]
```

```
Connect to the remote VNC port and start vncviewer.  
The options from command line override the settings  
on the config file 'vnc.rc'.
```

```
--help          display this help screen  
--vnc_port=<port> set the VNC port (5900)
```

1.4 Closing A Connection

From any (or both) of the clients, the connection can be closed with:

```
p2p-vnc/client/stop.sh b8e1f1e779
```

This will cleanup the keys on the server, close the ssh tunnels, and close the programs that were started locally (x11vnc, vncviewer, etc.)

2 Other Use Cases

2.1 Sharing A Single Window

Use the option **--window=yes** from command line (or set it on **vnc.rc**) to share only a single window (not the whole desktop). When this option is 'yes', the mouse will become like a + (cross-hair) and you will be able to select a window.

2.2 Make A Demonstration To One Or More People

Start **x11vnc** with options like these:

```
p2p-vnc/client/start_x11vnc.sh --viewonly=yes --shared=yes
```

The remote viewers will not be able to control your desktop (or window) and more than one viewers will be able to connect.

2.3 Share The Desktop Permanently

Suppose that time after time you need to access the computer at home remotely. Usually, after **vncviewer** is closed, **x11vnc** is closed automatically. But if the option **--forever=yes** is used, **x11vnc** will keep listening for more connections (after the first client(s) disconnect).

```
p2p-vnc/client/start_x11vnc.sh --forever=yes
```

Note: Keep in mind that a cron job on the P2P server will automatically delete keys older than one day. Without the key, you will not be able to access the desktop even if **x11vnc** is still listening. If this is not what you want, try to fix this on the P2P server (either stop the cron or make the period longer).

2.4 Share The Linux Console (Tty)

This is very similar to sharing the desktop, but it can share the Linux consoles (from **tty2** to **tty6**):

```
p2p-vnc/client/start_linuxvnc.sh --tty=3
```

2.5 Share Your Local Webserver

Usually web developers use a local webserver for building an application. What would you do to show the current prototype to the customer or to another developer? You can share your desktop, or you can share only the window of the browser. But you can also share your local webserver (port 80 and 443).

It can be done like this:

- On you side you run:

```
sudo p2p-vnc/client/port_share.sh 80  
3e41a200bd  
sudo p2p-vnc/client/port_share.sh 443  
c9277c986d
```

You need **sudo** in this case because **80** and **443** are privileged ports and can be forwarded only by root.

- On the other side, your partner should run:

```
sudo p2p-vnc/client/port_connect.sh 80 3e41a200bd  
sudo p2p-vnc/client/port_connect.sh 443 c9277c986d
```

The connection keys are provided to him by you. He should make sure that he does not have any local webserver running on ports 80 and 443. Now he can open **localhost** or **127.0.0.1** on his browser and he will directly access your local webserver.

- To stop sharing, both of you can run:

```
sudo p2p-vnc/client/port_stop.sh 3e41a200bd  
sudo p2p-vnc/client/port_stop.sh c9277c986d
```

The same way can be used for sharing other ports/services as well.

Author: Dashamir Hoxha <dashohoxha@gmail.com>

Date: 02 February 2014