VNC/Servers

Parent page: Internet and Networking >> VNC

Conteúdo

Introduction

A VNC server is a program that shares a desktop with other computers over the Internet. You will need a VNC server if you want other people to see your desktop. Every VNC server has different strengths and weaknesses and is appropriate for different uses. This page will discuss each of the VNC servers available in Ubuntu, and ways to configure them for most common uses of VNC.

Common security options

The most important thing when setting up a VNC server is to only let the right people access your desktop. The safest way to do that is usually to have someone sitting at the desktop deciding who gets to use it, but that's not always practical - for example, if you want to log in to your own computer from somewhere else.

If you want to confirm each connection manually, you should look for these options:

- **Request access each time** - pop a window up asking whether to allow each connection as it comes in.

- **view-only access** - allow VNC clients to view the destkop, but not to change anything. As well adding a little security, this avoids problems with both of you fighting over control of the mouse.

Using these two options will give you the most security. Requesting access each time will ensure that nobody can connect without you noticing, and view-only access will mean that they can't change anything without asking you to do it for them.

If you want to access your desktop when nobody is sitting at it, these options will be more useful:

- **Only allow local connections** - only let people connect if they already have access to your computer.

- **Start your VNC server in "once" mode** - tell your VNC server to allow one connection, then block anything after that.

- **Set a password** - require people to send a password before they can connect.

These three options should give you a secure set-up, so long as they're used with port-forwarding. Only allowing local connections means that only people with user accounts on your computer can access your desktop. Starting the server in "once" mode means that people with user accounts on your computer would have to log in to your desktop between the time you start your VNC server and the time you connect from your VNC client. Setting a password means that, if anyone did try to connect in that brief interval, they probably wouldn't be able to get in before you noticed and stopped the server.

Note: you must set a password if you want to use the in-built VNC client in Mac OS X.

vino

Vino is the default VNC server in Ubuntu to share your existing desktop with other users.

To configure vino from within GNOME, go to System > Preferences > Remote Desktop

- To set vino to request access each time, tick **Allow other users to view your desktop** in the Remote Desktop configuration window.

- There's no way to set vino to only listen for the next connection.

- To set a password, tick **Require the user to enter this password:**, and enter a hard-to-guess password.

- To put vino in view-only mode, untick **Allow other users to control your desktop**.

- To only allow local connections, open a terminal and run the command:

  ```
  gsettings set org.gnome.Vino network-interface lo
  ```

- To allow connections from anywhere, open a terminal and run the command:

  ```
  gsettings reset org.gnome.Vino network-interface
  ```

x11vnc

x11vnc is a VNC server that is not dependent on any one particular graphical environment. Also, it facilitates using in a minimal environment, as it has a tcl/tk based GUI. It can be started while your computer is still showing a login screen.

⚠ It is helpful to ensure you have uninstalled any other VNC programs first so that they don't interfere with x11vnc.

As a quick proof of concept to test your connectivity, as per the man page, one may create a password file via:

```
x11vnc -storepasswd
```

It will respond with:

```
Enter VNC password:
Verify password:
Write password to /home/USERNAME/.vnc/passwd?  [y]/n y
Password written to: /home/USERNAME/.vnc/passwd
```

One may execute the following in a terminal:

```
x11vnc -auth guess -forever -loop -noxdamage -repeat -rfbauth /home/USERNAME/.vnc/passwd -rfbport 5900 -shared
```

Here a few settings that would be common to adjust depending on your environment:

- To set x11vnc to request access each time when set without a password, include the **-nopw -accept popup:0** options.

- To set x11vnc to only listen for the next connection, include the **-once** option.

- To set x11vnc to continually listen for connections, include the **-forever** option.

- To put x11vnc in view-only mode, include the **-viewonly** option.

- To set x11vnc to only allow local connections, include the **-localhost** option.

Have x11vnc start automatically via upstart in any environment (<=Utopic)

```
sudo nano /etc/init/x11vnc.conf
```

```
# description "Start x11vnc at boot"

description "x11vnc"

start on runlevel [2345]
stop on runlevel [^2345]

console log

respawn
respawn limit 20 5

exec /usr/bin/x11vnc -auth guess -forever -loop -noxdamage -repeat -rfbauth /home/USERNAME/.vnc/passwd -rfbport 5900 -shared
```

Have x11vnc start automatically via systemd in any environment (Vivid+)

```
sudo nano /lib/systemd/system/x11vnc.service
```

```
[Unit]
Description=Start x11vnc at startup.
After=multi-user.target
```

```
[Service]
Type=simple
ExecStart=/usr/bin/x11vnc -auth guess -forever -loop -noxdamage -repeat -rfbauth /home/USERNAME/.vnc/passwd -rfbport 5900 -shared

[Install]
WantedBy=multi-user.target

sudo systemctl daemon-reload
sudo systemctl enable x11vnc.service
```

Have x11vnc automatically start in Kubuntu

One may create a startup script via:

```
nano ~/.kde/Autostart/x11vncstart.sh
```

```
x11vnc -auth guess -forever -loop -noxdamage -repeat -rfbauth /home/USERNAME/.vnc/passwd -rfbport 5900 -shared
```

```
chmod +x ~/.kde/Autostart/x11vncstart.sh
```

Have x11vnc automatically start in Ubuntu

In Ubuntu (but not Kubuntu or Xubuntu) x11vnc needs superuser access, and needs the `-auth /var/lib/gdm/:0.Xauth -display :0` options to be specified on the command-line. The argument value for the -auth option may be found previously with `x11vnc -findauth`.

You can run x11vnc before you've logged in by typing something like this:

```
sudo x11vnc -safer -localhost -once -nopw -auth /var/lib/gdm/:0.Xauth -display :0
```

If you find a blank screen, check the x11vnc FAQ entry on headless servers.

Alternatively, you can add the following lines to the bottom of your /etc/gdm/Init/Default to have x11vnc start after your gnome login does (note that /etc/gdm/Init/Default does not exist on some Ubuntu devices):

```
# Start the x11vnc Server
/usr/bin/x11vnc <options>
```

krfb

Krfb is the default VNC server in Kubuntu. Because it's highly integrated with KDE, running it in other environments is difficult.

To configure krfb, go to System Settings > Sharing > Desktop Sharing > Configure....

- To set krfb to request access each time, tick **Confirm uninvited connections before accepting**
- To set a password, type a hard-to-guess password into the **Password** input box.
- To put krfb in view-only mode, untick **Allow uninvited connections to control the desktop**.
- There's no built-in way to only allow local connections, although see below for a solution.

Once mode

Krfb doesn't have a built-in way to accept the next connection then stop listening for connection attempts. However, the following Python script will listen for a single connection then exit krfb:

Esconder número das linhas

```
#!/usr/bin/python

# Load extra functionality from the 'socket' and 'os' modules
from socket import socket, AF_INET, SOCK_STREAM
from os import execl

# Listen for a connection
server = socket(AF_INET, SOCK_STREAM) # This is an Internet (TCP) connection
server.bind(('127.0.0.1', 5900))      # Listen for a local connection on port 5,900
server.listen(1)                      # Listen for exactly 1 connection
sock = server.accept()[0]             # Accept the connection

# Attach krfb to this connection
execl('/usr/bin/krfb', 'krfb', '--kinetd', str(sock.fileno()))
```

To use this script, open your favorite text editor and paste the contents in. Make sure that the initial '#' character is the very first character in the file, save the file as **krfb.py**, and set the file's permissions to make it executable. Although this simple program won't open a window of any kind, it will quietly wait for the next VNC client to connect to your computer, then pass the connection through to krfb.

This script will only listen for local connections. To allow connections from anywhere, change `127.0.0.1` to `0.0.0.0` in the script.

Invitations

Krfb lets you create "invitations", or individual passwords that are deactivated after an hour or after one use. These are a handy way of giving people one-time access to a computer, but only provide limited security. For example, if you send someone an invitation by e-mail or instant messaging, an attacker could read your invitation message as it went over the Internet and use it to log in.

Invitations can be useful when you want to let other people view your desktop, but you still need to follow the normal precautions when letting other people view your desktop.

tightvncserver

Whereas most VNC servers share your desktop, tightvnc creates a completely new desktop, not attached to any actual screen. This makes it much less useful for some things (like remote help), but much more useful for others (like creating a public area for collaboration). If tightvncserver won't start, you might need to uncomment the `$fontpath` lines in `/etc/vnc.conf`.

Like x11vnc, tightvnc is designed to be run from the command-line. To start it, type:

```
tightvncserver -nolisten :1
```

This will tell tightvnc to listen for VNC connections on port 5901 from anywhere on the Internet. Without the **-nolisten tcp** option, tightvnc will also listen for a different type of connection (X11 instead of VNC), which isn't usually very useful. Tightvnc's unusual design means that it can't create a remote desktop on the standard VNC port (5900) if you have an ordinary desktop running on your computer.

- There's no way to set tightvncserver to request access each time.
- There's no way to set tightvncserver only to accept the next connection, although see below for a similar solution.
- Tightvncserver always requires a password, and will ask you to specify one the first time it's run.

- To set tightvncserver to only allow local connections, include the **-localhost** option.

Once mode

Tightvncserver can't be set to accept the next connection then stop listening for connection attempts. But it can be set to automatically disconnect each client when the next client connects, and can be stopped after your connection is disconnected. To only allow local connections and automatically disconnect clients, start tightvnc by typing:

```
tightvncserver -nolisten tcp -localhost -nevershared :1
```

Then when your client is disconnected by the next client connecting, type:

```
tightvncserver -kill :1
```

Customising your session

By default, tightvncserver provides a session with a simple window manager and a terminal. The first time tightvncserver runs, it creates a `~/.vnc/xstartup` file that you can use to customise your session. Here is an example file that would give you a GNOME desktop:

Esconder número das linhas

```
#!/bin/sh

# Change "GNOME" to "KDE" for a KDE desktop, or "" for a generic desktop
MODE="GNOME"

#Uncommment this line if using Gnome and your keyboard mappings are incorrect.
#export XKL_XMODMAP_DISABLE=1

# Load X resources (if any)
if [ -e "$HOME/.Xresources" ]
then
        xrdb "$HOME/.Xresources"
fi

# Try a GNOME session, or fall back to KDE
if [ "GNOME" = "$MODE" ]
then
        if which gnome-session >/dev/null
        then
                gnome-session --session=ubuntu-2d &
        else
                MODE="KDE"
        fi
fi

# Try a KDE session, or fall back to generic
if [ "KDE" = "$MODE" ]
then
        if which startkde >/dev/null
        then
                startkde &
        else
                MODE=""
        fi
fi

# Run a generic session
if [ -z "$MODE" ]
then
        xsetroot -solid "#DAB082"
        x-terminal-emulator -geometry "80x24+10+10" -ls -title "$VNCDESKTOP Desktop" &
        x-window-manager &
fi
```

Your changes will take effect the next time you start tightvncserver.

TigerVNC

TigerVNC was originally based on the (never-released) VNC 4 branch of TightVNC. It is stable and actively maintained, being around since 2009 and included in most popular distributions. In particular, it supports compositing window managers without requiring a fallback mode, such as with Gnome Shell. When using with the TigerVNC viewer it also uses TLS encryption by default.

TigerVNC is available in Ubuntu 17.04 and newer:

```
sudo apt install tigervnc-standalone-server tigervnc-viewer
```

On older Ubuntus, go to https://github.com/TigerVNC/tigervnc/releases to find the latest release, since it is not yet in an apt repository. Download and install:

```
wget https://bintray.com/artifact/download/tigervnc/stable/ubuntu-14.04LTS/amd64/tigervncserver_1.6.0-3ubuntu1_amd64.deb
sudo dpkg -i tigervncserver_1.6.0-3ubuntu1_amd64.deb
sudo apt-get -f install
```

Its syntax is very similar to tightvncserver, start it as your user with:

```
vncserver :1
```

And stop it with:

```
vncserver -kill :1
```

See `man vncserver` for options. Avaiable options are similar but not identical to tightvnc.

TigerVNC can also replace x11vnc to attach to the local display using the provided `x0vncserver` binary:

```
x0vncserver -display :0
```

More detailed usage information is available here.

Start TigerVNC vncserver at boot

The ubuntu install package also registers a system service, making it easy to define listening vnc servers on startup. Edit the file `/etc/default/vncserver` and add the display number and user to start as:

```
VNCSERVERS="1:myusername"
```

Then enable the service at boot with:

```
sudo update-rc.d vncserver defaults
```

Similar applications

- GNU Screen and tmux allow you to open, share, disconnect, and later return to text-based terminals.

- **directvnc** is a VNC server that shares a Linux framebuffer instead of a desktop.
- **linuxvnc** is a VNC server that shares a text-based console instead of a desktop.
- **xrdp** is a server for Microsoft's Remote Desktop protocol, a client for which comes with all modern versions of Windows.
- **xserver-xephyr** allows you to create a desktop within a desktop on a single computer.
- Apple Remote Desktop is a desktop sharing application for Mac OS that includes a VNC server.
- Apple Screen Sharing is a default application in Mac OS X that allows incoming VNC connections.

Troubleshooting

Having Compiz enabled may interrupt screen updates with some servers and clients. Using -noxdamage with x11vnc can prevent this.

External links

* http://www.karlrunge.com/x11vnc/

VNC/Servers (editada pela última vez em 2020-09-15 23:53:52 por rg141)