

FreeSandal

樹莓派, 樹莓派之學習, 樹莓派之教育

【鼎革 · 革鼎】：RASPBIAN STRETCH 《六之 F.5 》

2017-11-13 | 懸鉤子 | 發表迴響

既然談了 VNC 伺服器，焉能不提及 Karl J. Runge 先生之出類拔萃的

x11vnc: a VNC server for real X displays

x11vnc allows one to view remotely and interact with real X displays (i.e. a display corresponding to a physical monitor, keyboard, and mouse) with any VNC viewer. In this way it plays the role for Unix/X11 that WinVNC plays for Windows.

It has built-in SSL/TLS encryption and 2048 bit RSA authentication, including VeNCrypt support; UNIX account and password login support; server-side scaling; single port HTTPS/HTTP+VNC; Zeroconf service advertising; and TightVNC and UltraVNC file-transfer. It has also been extended to work with non-X devices: natively on Mac OS X Aqua/Quartz, webcams and TV tuner capture devices, and embedded Linux systems such as Qtopia Core. Full IPv6 support is provided. More features are described here.

It also provides an encrypted Terminal Services mode (- create, - svc, or - xdmvc options) based on Unix usernames and Unix passwords where the user does not need to memorize his VNC display/port number. Normally a virtual X session (**Xvfb**) is created for each user, but it also works with X sessions on physical hardware. See the `tsvnc` terminal services mode of the SSVNC viewer for one way to take advantage of this mode.

I wrote **x11vnc** back in 2002 because **xOrfbserver** was basically impossible to build on Solaris and had poor performance. The primary **xOrfbserver** build problems centered around esoteric C++ toolkits. **x11vnc** is written in plain C and needs only standard libraries and so should work

on nearly all Unixes, even very old ones. I also created enhancements to improve the interactive response, added many features, and etc.

This page including the FAQ contains much information [*]; solutions to many problems; and interesting applications, but nevertheless please feel free to contact me if you have problems or questions (and if I save you time or expense by giving you some of my time, please consider a PayPal Donation.) Do check the FAQ and this page first; I realize the pages are massive, but you can often use your browser's find-in-page search action using a keyword to find the answer to your problem or question.

SSVNC: An **x11vnc** side-project provides an Enhanced TightVNC Viewer package (SSVNC) for Unix, Windows, and Mac OS X with automatic SSL and/or SSH tunnelling support, SSL Certificate creation, Saved connection profiles, Zeroconf, VeNCrypt, and built-in Proxy support. Added features for the TightVNC Unix viewer: NewFBSize, ZRLE encoding, Viewer-side Scaling, cursor alphablending, low color modes, and enhanced popup menu; UltraVNC extensions support for: File Transfer, Text Chat, Single Window, Server Input, and 1/n Scaling extensions, and UltraVNC DSM encryption. The SSVNC bundle could be placed on, say, a USB memory stick for SSL/SSH VNC viewing from nearly any networked computer.

程式也！舉例來說，在樹莓派上如何作逆向 VNC 連接的呢？

Starting Listening VNC Viewer and establishing a reverse connection

You can establish a reverse connection to a Listening VNC Viewer, allowing the Viewer user to bypass the authentication scheme and control the Server User's computer. When a connection is made in this way, VNC Viewer does not connect to VNC Server, but instead waits for VNC Server to connect to it.

This feature is useful if your computer is protected by a firewall that cannot be configured to allow – or by a router that cannot be configured to forward – network communications, thus preventing incoming connections. In a reverse connection, network communications from the Server user's computer are *outgoing*.

Note: Reverse connections are not secure and should only be used in a locked-down environment.

Configuring Listening VNC Viewer

Starting Listening VNC Viewer

- Under Windows or Mac OS X, search for or navigate to the Listening VNC Viewer program.
- Under UNIX, you must start Listening VNC Viewer at the command line. If VNC Viewer is *installed* on your system, run:

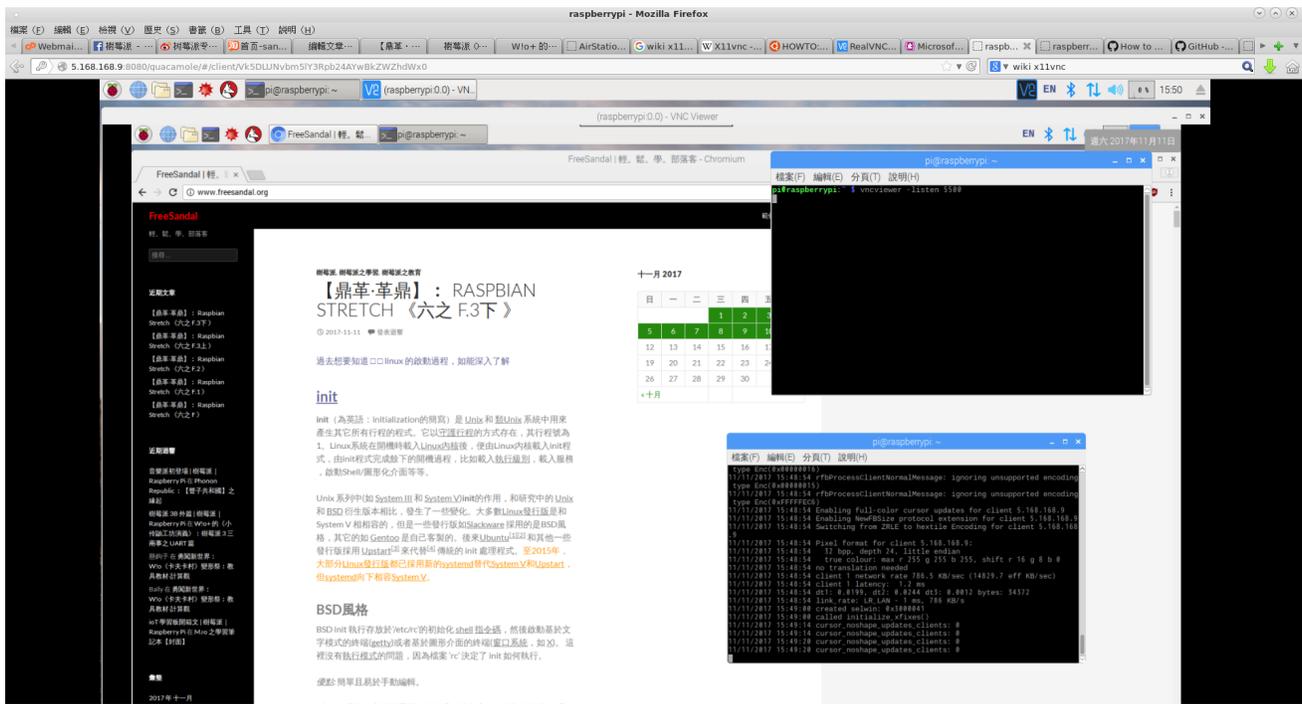
```
vncviewer -listen [<port>]
```

If you use the standalone version of VNC Viewer, run:

```
./<download-file> -listen [<port>]
```

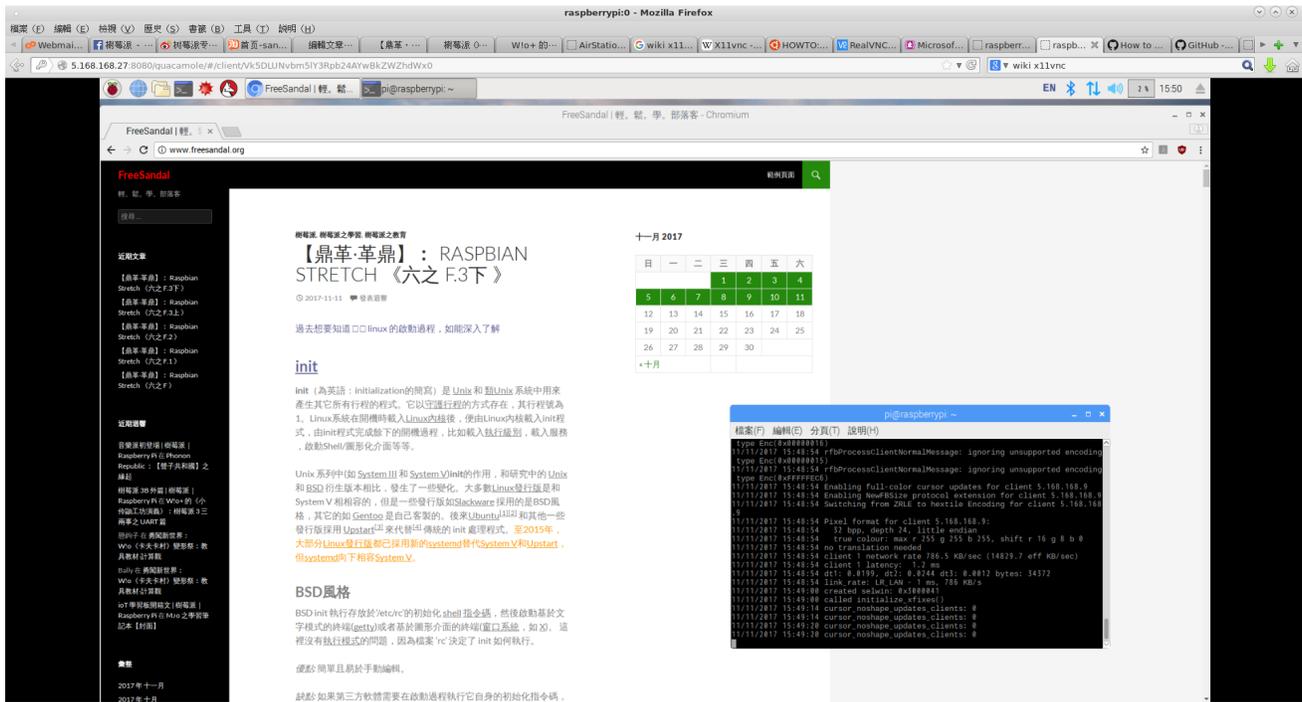
【觀看客戶端】

```
vncviewer -listen 5500
```



【伺服器端】

x11vnc-connect 觀看客戶端 IP :5500



就得了！？還可藉此工具了解 VNC 規範連線效能以及選項運用？！將能自製酪梨 client 乎！！？？

PyGuacamole

A Python client library for communication with Guacamole server (guacd)

Installation

Using pip

```
1 $ pip install pyguacamole
```

From source

```
1 $ python setup.py install
```

Usage

GuacamoleClient handles communication with a running *guacd* server via Guacamole Protocol.

GuacamoleClient must be used by a broker server which handles communication with a Javascript application running in the browser. GuacamoleClient implements the methods that enables communication with *guacd* server (send & receive).

First step should be establishing *handshake* with *guacd* server, then Broker server should handle instruction sending and receiving:

- **send:** send instruction *to* *guacd* server
- **receive:** receive instruction *from* *guacd* server

.....

唯因目前其喜派生二 Python :

```
1 pi@raspberrypi:~ $ sudo pip install pyguacamole
2 Collecting pyguacamole
3   Downloading pyguacamole-0.6.tar.gz
4 Collecting future>=0.15.2 (from pyguacamole)
5   Downloading future-0.16.0.tar.gz (824kB)
6     100% |████████████████████████████████████████| 829kB 298kB/s
7 Requirement already satisfied: six>=1.10.0 in /usr/lib/python2.7/dist-packages (from py
8 Building wheels for collected packages: pyguacamole, future
9   Running setup.py bdist_wheel for pyguacamole ... done
10  Stored in directory: /root/.cache/pip/wheels/f9/6d/74/876c143097972169cef3fa99508be9b
11  Running setup.py bdist_wheel for future ... done
12  Stored in directory: /root/.cache/pip/wheels/c2/50/7c/0d83b4baac4f63ff7a765bd16390d2c
13 Successfully built pyguacamole future
14 Installing collected packages: future, pyguacamole
15 Successfully installed future-0.16.0 pyguacamole-0.6
16
17 pi@raspberrypi:~ $ python
18 Python 2.7.13 (default, Jan 19 2017, 14:48:08)
19 [GCC 6.3.0 20170124] on linux2
20 Type "help", "copyright", "credits" or "license" for more information.
21 >>> from guacamole.client import GuacamoleClient
22 >>> client = GuacamoleClient('127.0.0.1', 4822)
23 >>> client.handshake(protocol='rdp', hostname='localhost', port=3389)
24 >>> instruction = client.receive()
25 >>> instruction
26 '4.size,1.0,4.1024,3.768;'
27 >>> instruction = '5.mouse,3.400,3.500;'
28 >>> client.send(instruction)
29 >>>
```

故待挺進派生三的叻??!!

```
1 pi@raspberrypi:~ $ python3
2 Python 3.5.3 (default, Jan 19 2017, 14:11:04)
3 [GCC 6.3.0 20170124] on linux
4 Type "help", "copyright", "credits" or "license" for more information.
5 >>> from guacamole.client import GuacamoleClient
6 >>> client = GuacamoleClient('127.0.0.1', 4822)
7 >>> client.handshake(protocol='vnc', hostname='localhost', port=5900)
8 Traceback (most recent call last):
9   File "<stdin>", line 1, in <module>
10  File "/usr/local/lib/python3.5/dist-packages/guacamole/client.py", line 158, in hands
11    self.send_instruction(Instruction('select', protocol))
12  File "/usr/local/lib/python3.5/dist-packages/guacamole/client.py", line 136, in send_
13    return self.send(instruction.encode())
14  File "/usr/local/lib/python3.5/dist-packages/guacamole/client.py", line 122, in send
15    self.client.sendall(data)
16 TypeError: a bytes-like object is required, not 'str'
17 >>>
```

