Using GadgetPC as a Secure Shell (SSH) Router

Date: 30th July, 2009

Document Revision: 1.01

BiPOM Electronics

16301 Blue Ridge Road, Missouri City, Texas 77489
Telephone: (713) 283-9970 Fax: (281) 416-2806
E-mail: info@bipom.com
Web: www.bipom.com

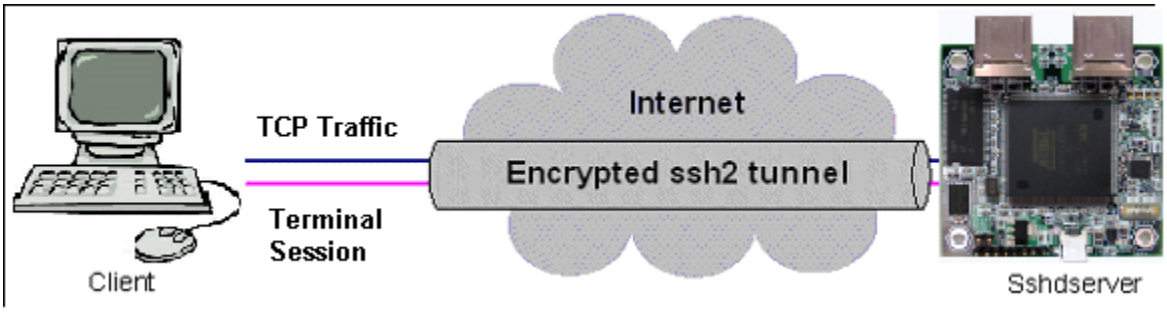
© 2009 by BiPOM Electronics, Inc. All rights reserved.

No part of this work may be reproduced in any manner without written permission of BiPOM Electronics.

All trademarked names in this manual are the property of respective owners.

Overview

Port forwarding, also known as tunneling, is a way to forward otherwise insecure TCP traffic through SSH Secure Shell. For example, you can secure POP3, SMTP, and HTTP connections that would otherwise be insecure.



As you can see from the diagram above, GadgetPC can serve as a low power ssh server.

Implementing SSH Router

Parts Required

1 x GadgetPC
1 x ADP-5V1A-MiniUSB Power Adapter
1 x DUB-E100 Ethernet card
1 x Ethernet Cable
1 x USB Flash Drive running Linux (optional)

Connect the various components as shown in *Figure 1*.

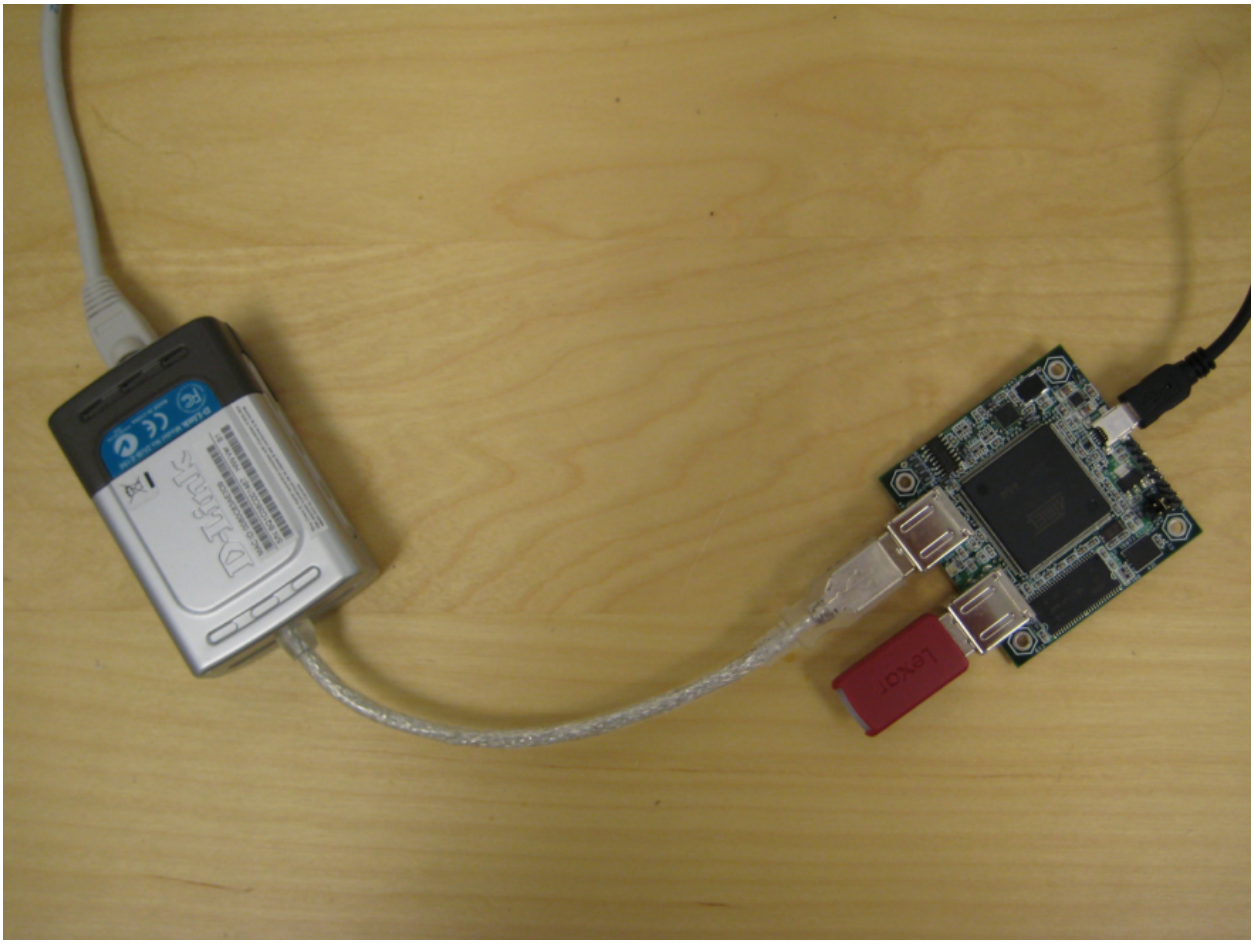
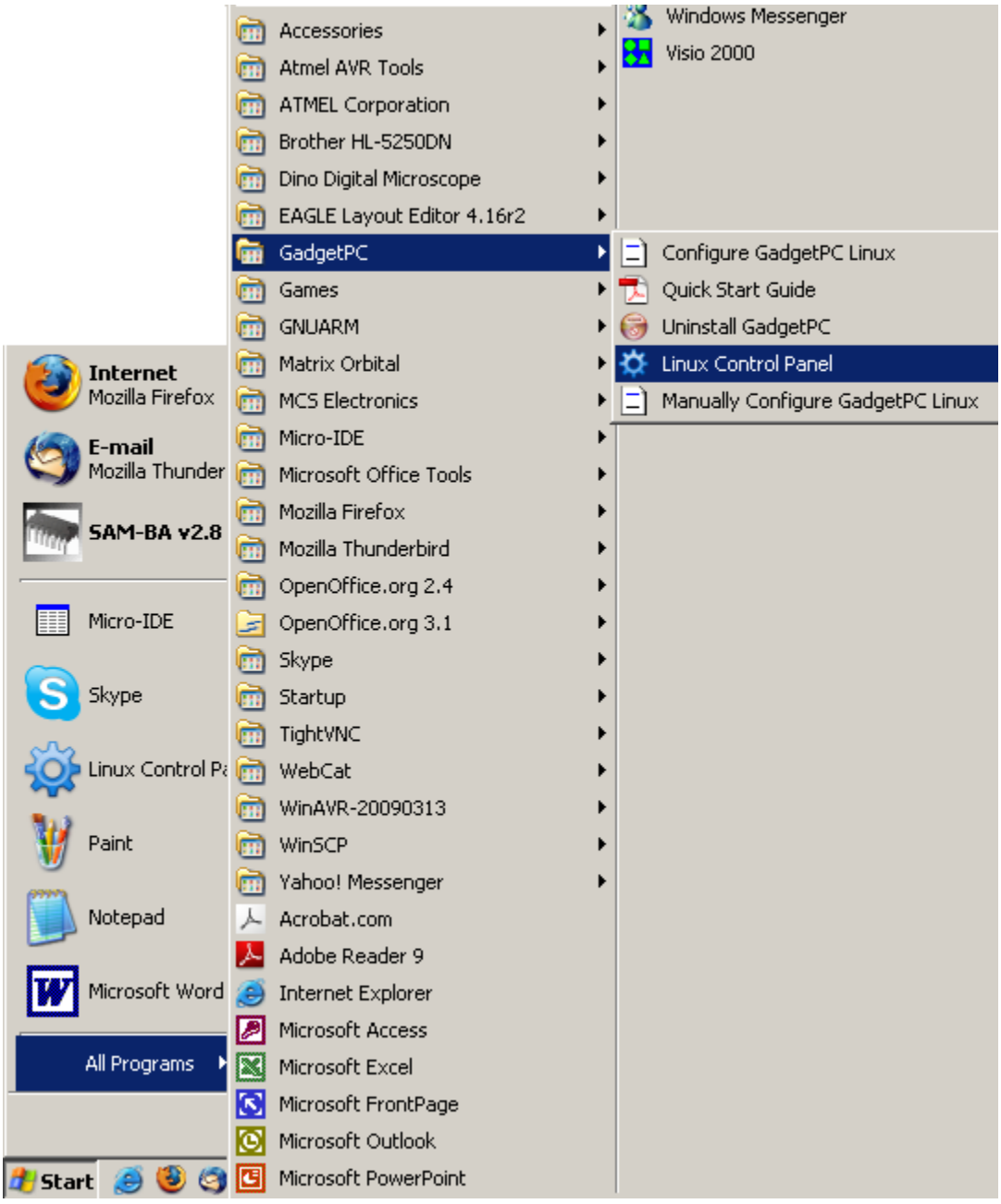


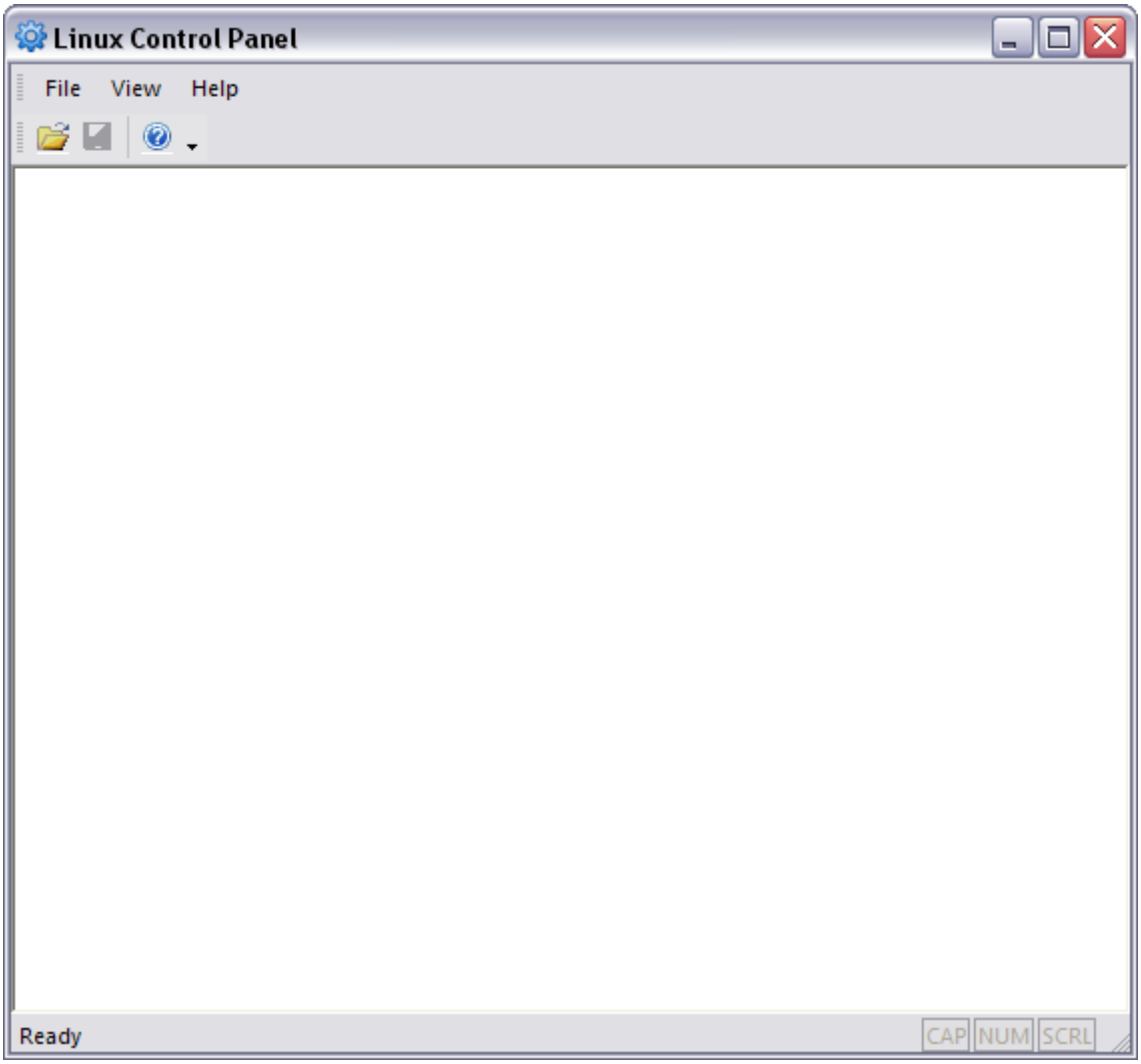
Figure 1.

Software Setup

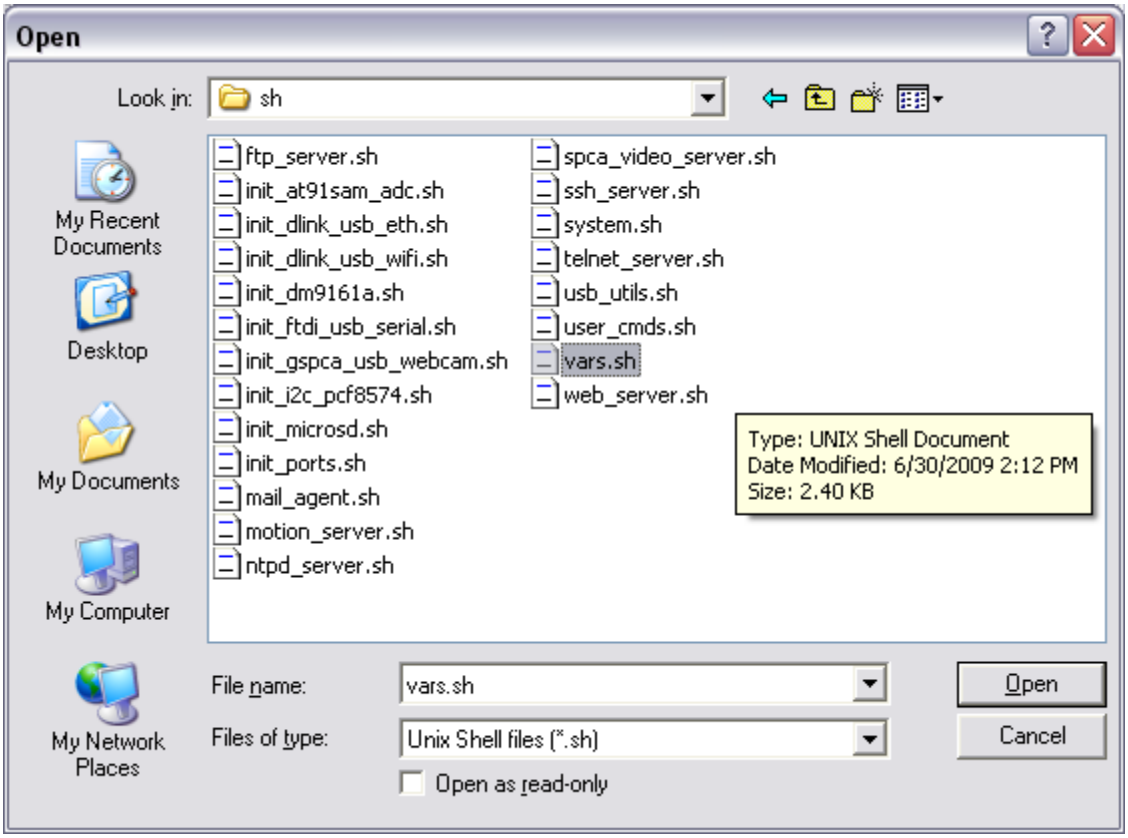
We need to edit the Linux vars.sh file to enable certain configuration options.

1) Open **Linux Control Panel** that comes with GadgetPC setup. (Start Menu -> All Programs -> GadgetPC -> Linux Control Panel)

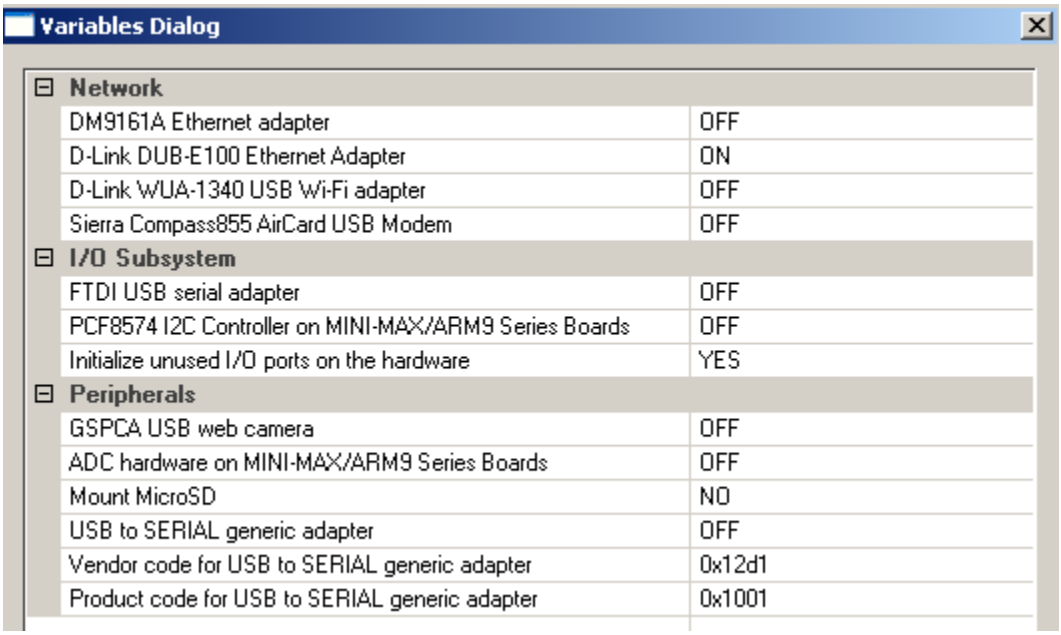

Internet
Mozilla Firefox
E-mail
Mozilla Thunder
SAM-BA v2.8
Micro-IDE
Skype
Linux Control Pa
Paint
Notepad
Microsoft Word
All Programs
Start
Accessories
Atmel AVR Tools
ATMEL Corporation
Brother HL-5250DN
Dino Digital Microscope
EAGLE Layout Editor 4.16r2
GadgetPC
Games
GNUARM
Matrix Orbital
MCS Electronics
Micro-IDE
Microsoft Office Tools
Mozilla Firefox
Mozilla Thunderbird
OpenOffice.org 2.4
OpenOffice.org 3.1
Skype
Startup
TightVNC
WebCat
WinAVR-20090313
WinSCP
Yahoo! Messenger
Acrobat.com
Adobe Reader 9
Internet Explorer
Microsoft Access
Microsoft Excel
Microsoft FrontPage
Microsoft Outlook
Microsoft PowerPoint
Windows Messenger
Visio 2000
Configure GadgetPC Linux
Quick Start Guide
Uninstall GadgetPC
Linux Control Panel
Manually Configure GadgetPC Linux



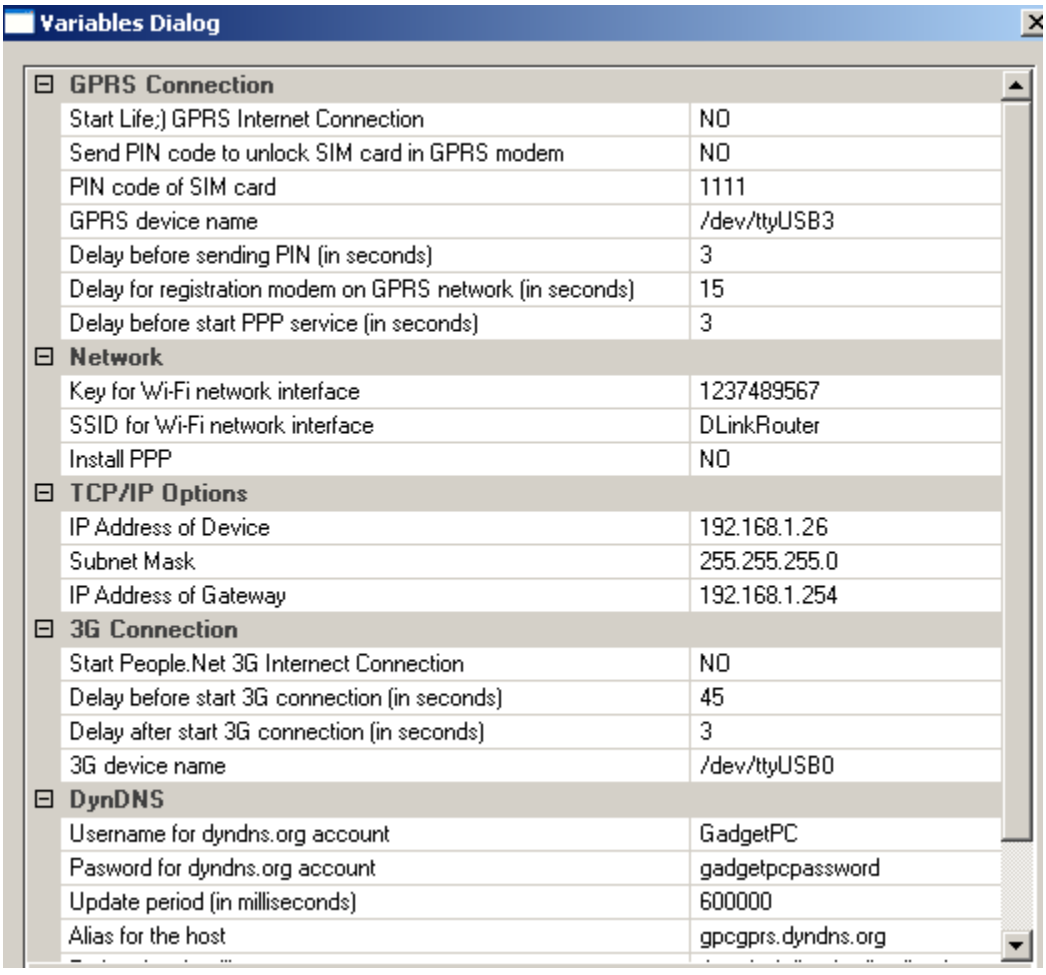
2) Click File -> Open. By default, program will start from folder where GadgetPC was installed. Go to **sh** folder and select **vars.sh** file.



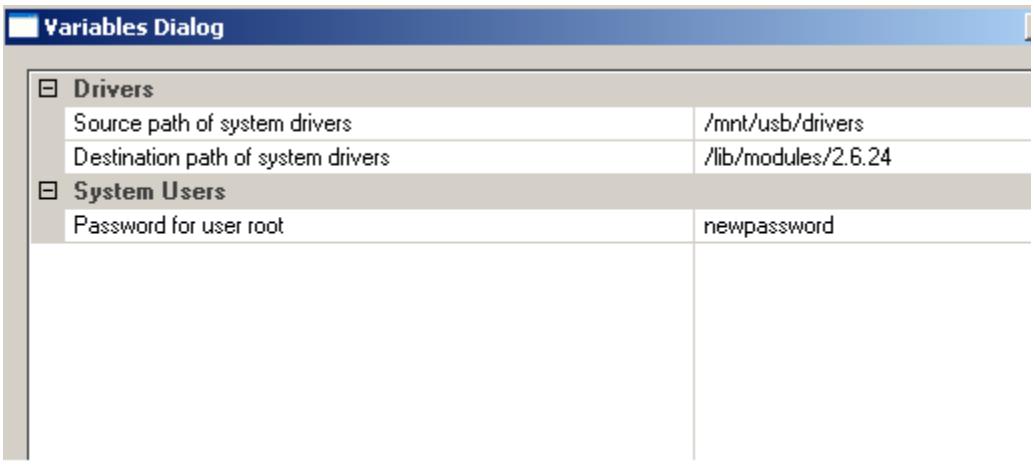
3) Double-click **Hardware and Peripherals** and enable **DUB-E100** (Turned ON).



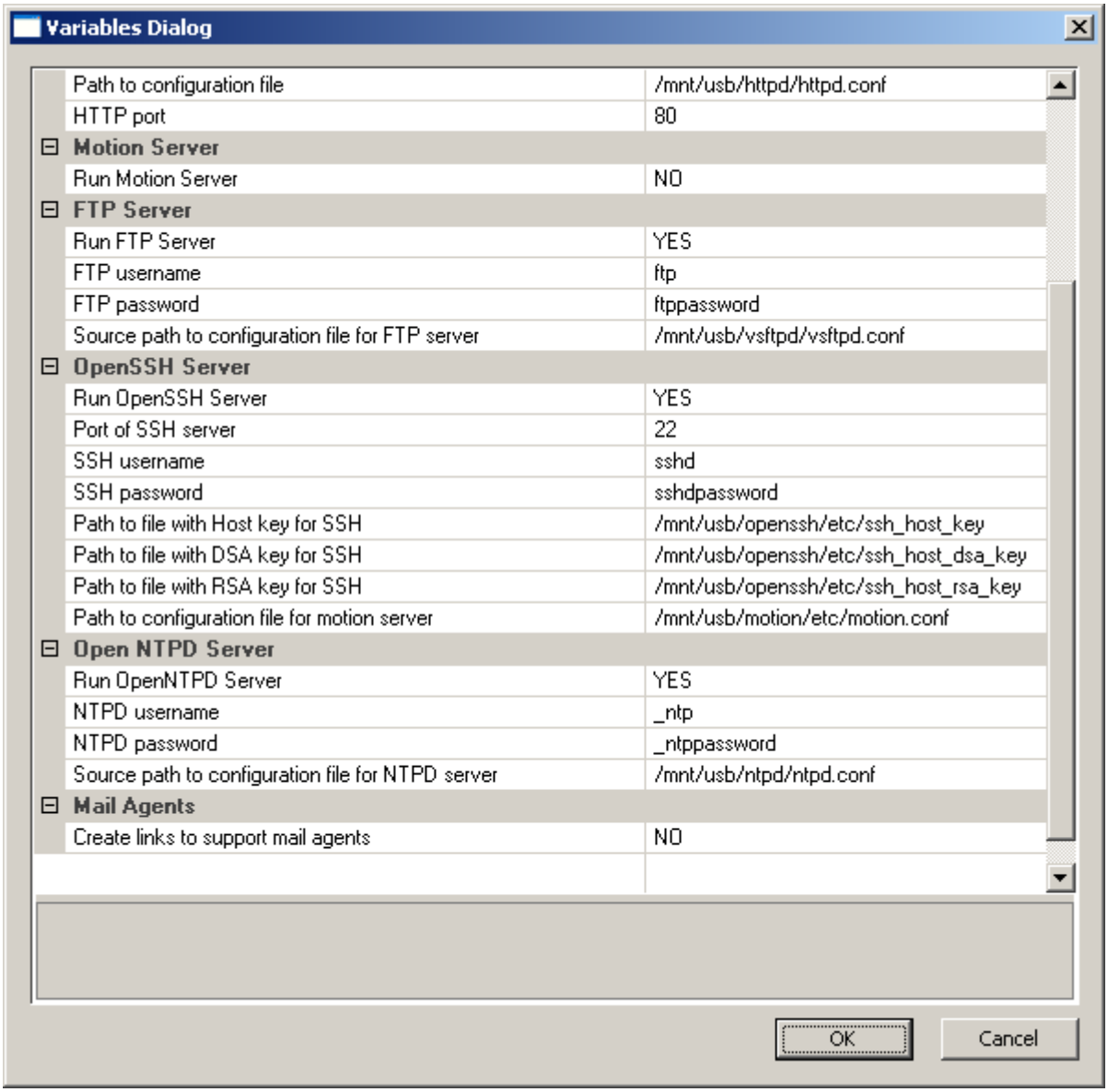
4) Double-click **Network** and assign an IP address, subnet mask, and gateway IP address located under **TCP/IP Options**.



5) Double-click **System** and assign a root password.



6) Double-click **Software** and enable **OpenSSH Server** (Run OpenSSH Server: YES), enable **OpenNTPD** Server (Run OpenNTPD Server: YES), set the **SSH port** (default: 22), and also assign username and password for both servers.



7) Save the **vars.sh** file.

NOTE: Every time you access vars.sh file, you either need to have a temporary copy on your computer so that you can replace the actual copy on the USB Flash Drive or remove the USB Flash Drive from the GadgetPC and insert the USB Flash Drive into the computer for direct editing.

8) Insert the USB Flash drive to GadgetPC.

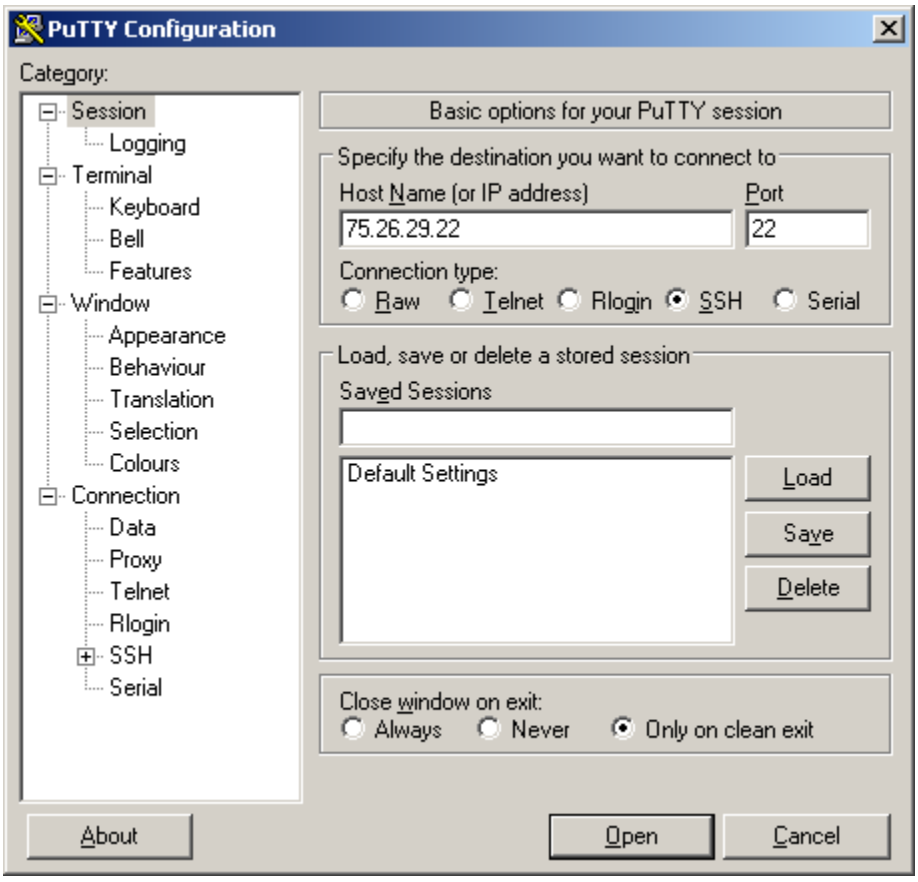
9) Apply or cycle power to GadgetPC (Turn OFF/ON).

10) After about 30-45 seconds, Linux will boot up and GadgetPC will be ready to to port-forward.

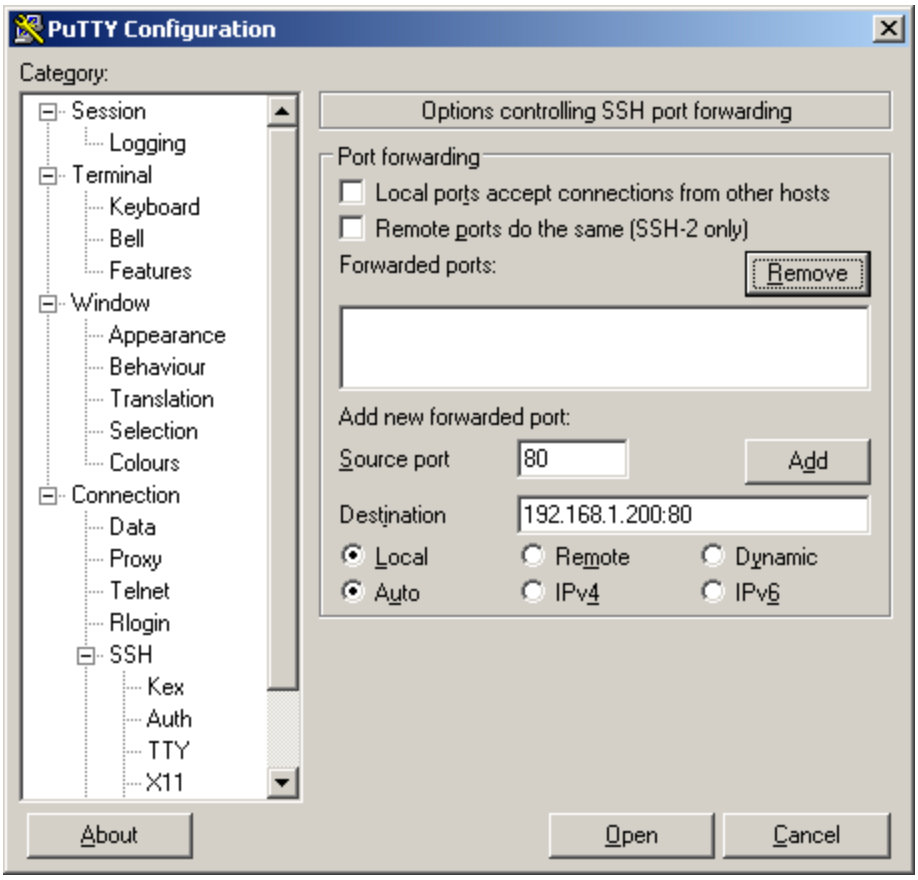
Port Forwarding with **PuTTY**

1) Download the latest free telnet/ssh client **PuTTY**:
http://the.earth.li/~sgtatham/putty/latest/x86/putty.exe

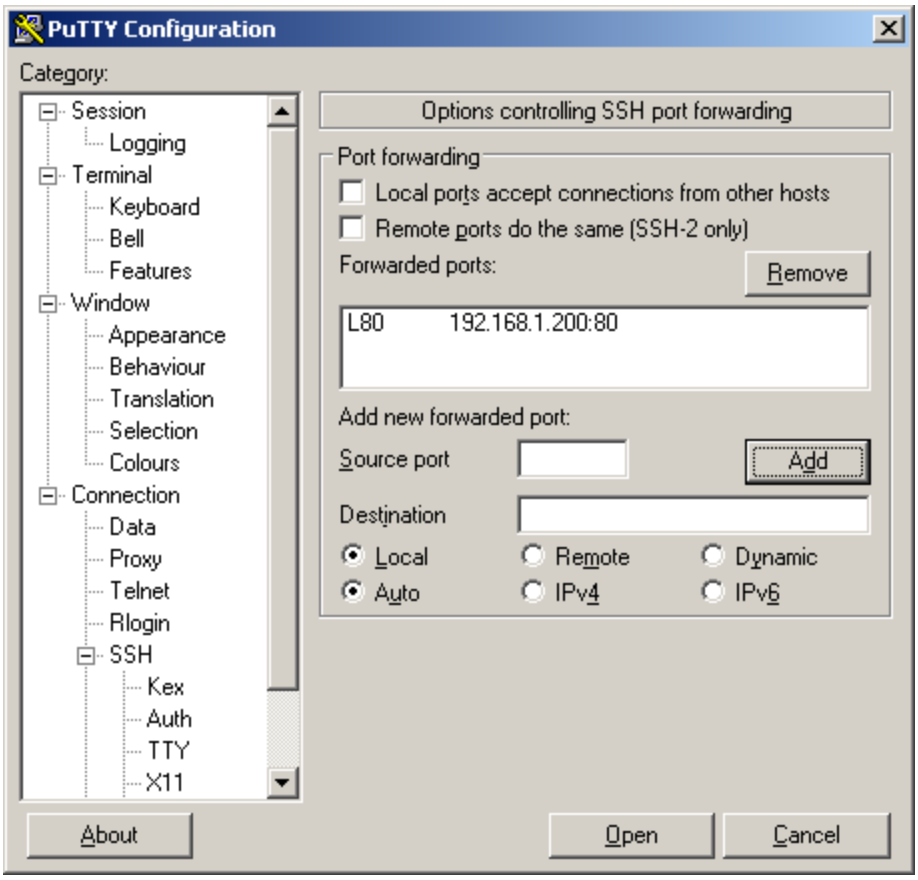
2) Start **PuTTY**, specify the host you want to connect to (e.g. 75.26.29.22), the port (in this case, we use the default port setting: 22), and the SSH protocol.



3) Click the Connections -> SSH -> **Tunnels** tab and specify local port, remote host and port and click the **Add** button.

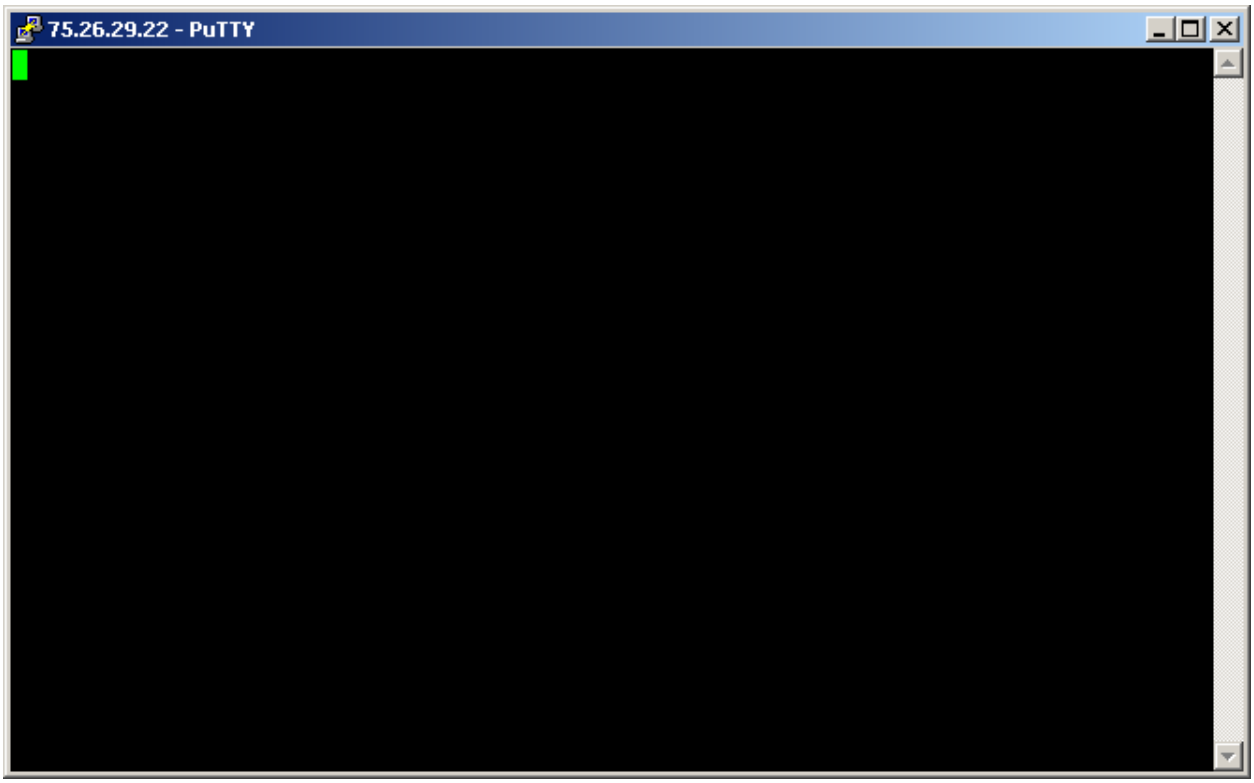


4) You can go back to the **Session** Tab to **Save** your settings; otherwise, click **Open**.



5) A terminal should open. You should now be able to login with your sshd username and password.

Note: If you receive a Security Alert, answer Yes to continue.



The above example is the equivalent to the UNIX 'ssh' command:
ssh -L 80:192.168.1.200:80 *your_username_here*@75.26.29.22

It will forward the local port 80 to host '192.168.1.200', port 80, via the SSH connection. **PuTTY** is an SSH client, so it will still spawn a terminal window. When you're finished using your forwarded connection(s) you can close this window.

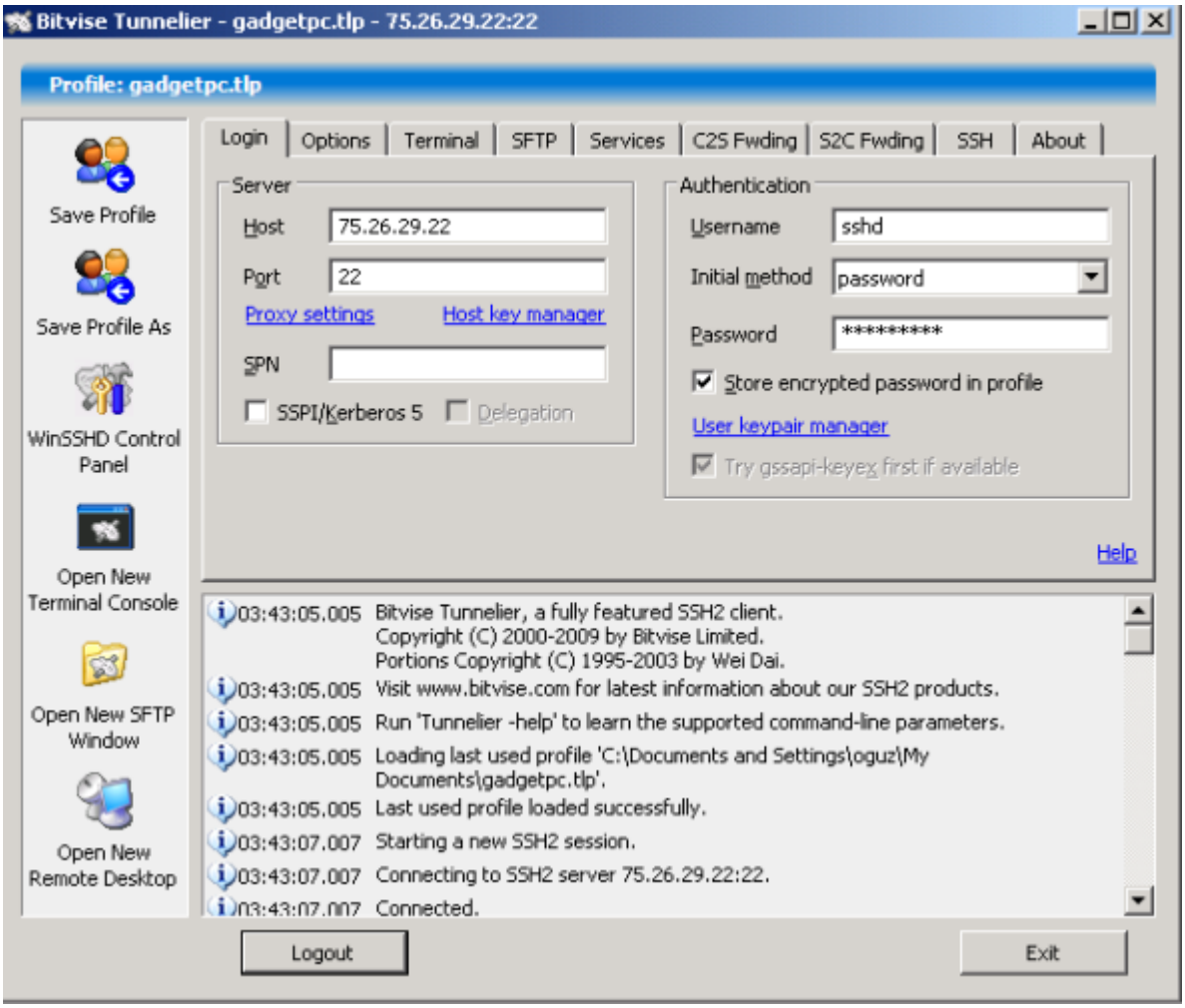
6) In our example, in order to use the forwarded port, we can access the remote web server over forwarded port 80 connection. Open a browser and type http://localhost/. Port 80 for localhost gets diverted to the remote computer on the remote LAN.

Port Forwarding with **Tunnelier**

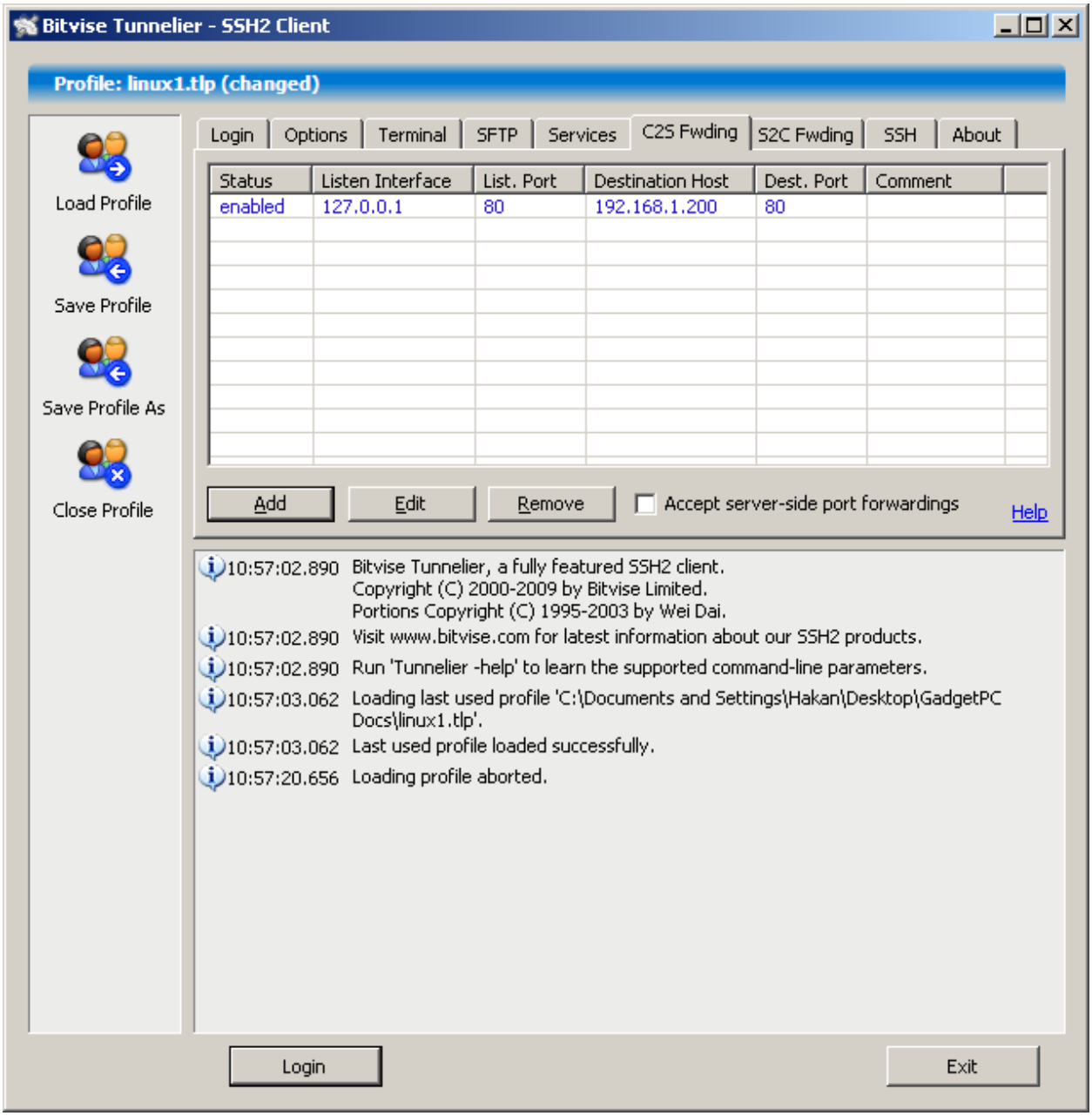
1) Download the latest free telnet/ssh client **Tunnelier**:
http://dl.bitvise.com/Tunnelier-Inst.exe

2) Start **Tunnelier**, specify the host you want to connect to (e.g. 75.26.29.22) and the port (in this case, default port setting: 22) under **Server**.

3) Enter your sshd username and password under **Authentication**.



4) Under **C2S Fwding** Tab, you may forward desired ports.



5) Save the profile (optional).

6) Click **Login**.

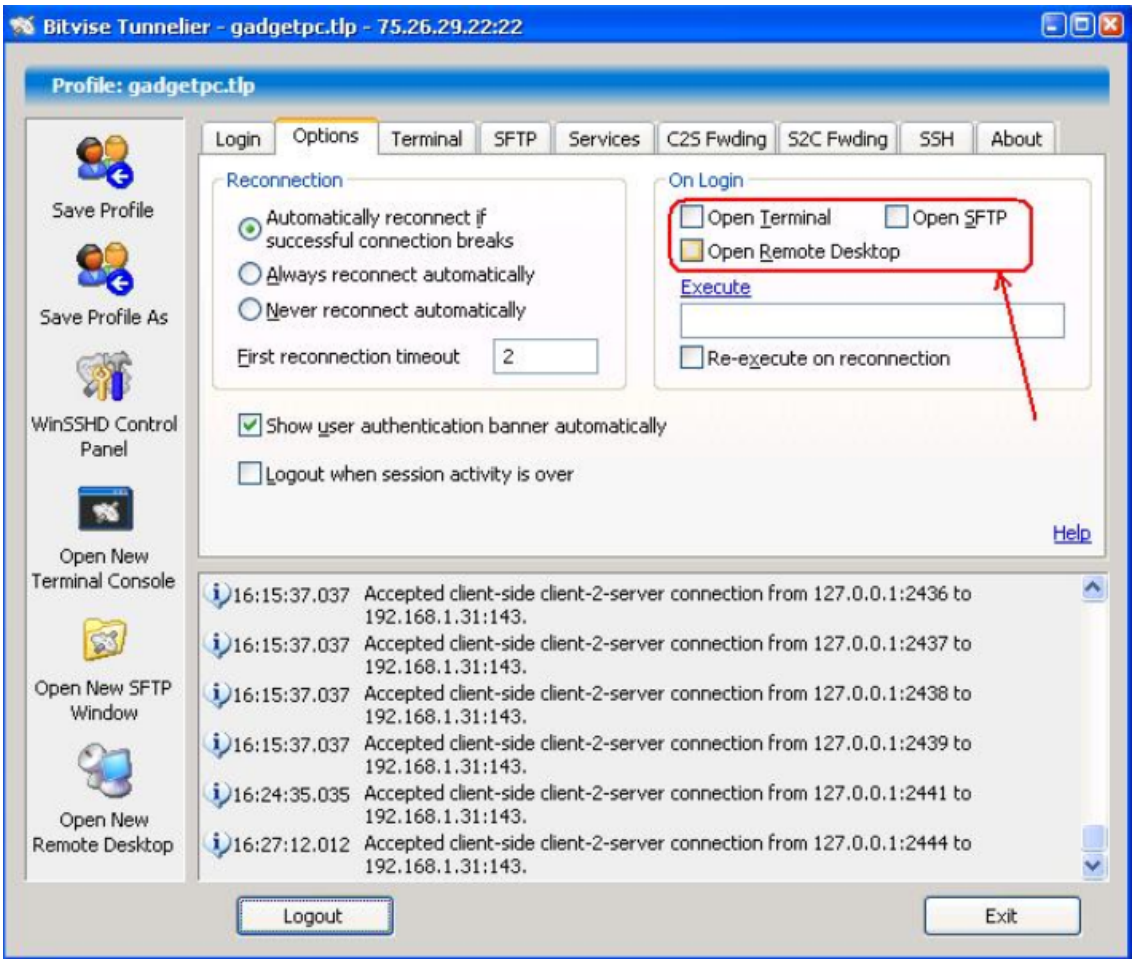
Note: If you receive a Host Key Verification, answer **Accept for this Session** to continue.

7) A terminal should open. You should now be able to give commands.

8) Upon connection through **Tunnelier**, if you do not want **Terminal, Remote Desktop** and **SFTP** to start automatically, please click on **Options** tab in Tunnelier and uncheck

Open Terminal
Open SFTP
Open Remote Desktop

checkboxes as shown:



9) In our example, in order to use the forwarded port, we can access the remote web server over forwarded port 80 connection. Open a browser and type http://localhost/. Port 80 for localhost gets diverted to the remote computer on the remote LAN.