

Internet Connection Share Module

Google Summer of Code 2008 Project

Cihangir Beşiktaş

`cihangirbesiktas@gmail.com`

Internet Connection Share Module Google Summer of Code 2008 Project
by Cihangir Beşiktaş

Dedication

I thank my mentor Bahadır Kandemir for his support during the project period.

Table of Contents

1. INTRODUCTION	1
1.1. The Aim of the Project.....	1
1.2. Requirements	1
1.2.1. Hardware Requirements	1
1.2.2. Software Requirements	1
2. Internet Connection Sharing	2
2.1. Settings.....	2
2.1.1. Network Settings	2
2.1.2. Iptables	2
2.1.3. DHCP Server	2
2.2. The Code	3
2.2.1. GUI Implementation.....	3
2.2.2. GUI Signals	3
3. VPN Connection.....	5
3.1. Settings.....	5
3.2. The Code	5

Chapter 1. INTRODUCTION

1.1. The Aim of the Project

The aim of this project is to add new features to Pardus operating system's Network Manager. These features are internet connection sharing and VPN supports which are served with a easy-use graphic interface for users.

Internet connection sharing module gives the users a chance to share their internet with their clients easily. Because they only select interfaces and hit the apply button. They do not need to do lots of configuration settings. This module does every settings for them.

VPN support gives users an opportunity to connect VPN servers with a user-friendly interface. The only thing users have to do is just filling the related fields of the interface.

1.2. Requirements

There are some requirements to use these features. These requirements can be groued into two sections. One is hardware requirements. The other is software requirements.

1.2.1. Hardware Requirements

For hardware, you need to have two network interface card. It is recommended (most laptops have) to have one wireless NIC and one ethernet NIC. With these, it is advised to use wireless NIC for your internet connection and ethernet NIC for the sharing of it.

1.2.2. Software Requirements

For software, all you need is having iptables, dhcp server installed in your Pardus system. Iptables is for network adress translation, and dhcp server is for supplying ip, gateway, dns...,etc to internal hosts that you will share your connection with.

Chapter 2. Internet Connection Sharing

2.1. Settings

2.1.1. Network Settings

To use this module, some network settings have to be done. First, a profile for going to internet is created by using NM's interface. After the profile settings is done, the profile state is changed to "up" by clicking the checkbox of the profile. Then another profile for internet connection sharing is created in the same way. For this profile, if the user wants to use default ip, netmask, gateway setting, he/she chooses the automatic setting. But if not, the user has to assign ip, netmask and gateway. Here gateway has to be ip address of the other profile.

However, the `net.ipv4.ip_forward` sysctl variable should be 1 to forward ip packets. With this option, packets coming from internal clients is forwarded to go to internet.

2.1.2. Iptables

Iptables is one of the most used firewall applications in linux. It also provides network address translation. NAT is used in this project to do NAT on clients packets. The following command enables nat:

```
$iptables -t nat -A POSTROUTING -o eth1 -j MASQUERADE
```

Here eth1 is the interface of the NIC that provides internet connection.

2.1.3. DHCP Server

With the rules above, the communication between the internal hosts and internet is done. But we have to provide ip, router, dns...,etc information for the internal hosts. So to do this, we use a dhcp server and configure it correctly. The following file contains correct datas to do this:

```
$vi /etc/dhcp/dhcpd.conf

ddns-update-style interim;
ignore client-updates;

max-lease-time 500;
default-lease-time 500;
option domain-name-servers 193.140.100.210;
option routers 10.10.10.1;
```

```
option subnet-mask 255.255.255.0;
subnet 10.10.10.0 netmask 255.255.255.0 {
    range 10.10.10.2 10.10.10.254;
}
```

Here 10.10.10.1 is the router and also the ip address of the NIC that is for sharing. The internal hosts gateway will be this value. And subnet 10.10.10.0 with netmask 255.255.255.0 tells us that the internal network address is 24 bit and 10.10.10.0. The range is the ip address interval of internal hosts.

2.2. The Code

The code is written with python programming language. GUI is done with qt3 API and Qt3 Designer.

2.2.1. GUI Implementation

The GUI of internet connection sharing module consists of some labels which give information to users, 2 comboboxes for selecting profiles, 1 checkbox for enabling comboboxes and disabling sharing process, 1 Apply button for enabling sharing process, 1 Cancel button for discarding without save.

2.2.2. GUI Signals

QT language has the signal and slot feature to process some code when some events is done on GUI elements. In this project this feature is used to process necessary code segments.

When the sharing button that is at the right up corner of NM's GUI is hit, the dialog opens. While opening, in the code segment, the ethernet and wireless based profiles are loaded into the comboboxes. Regarding to the previous state of sharing process, the checkbox can be enabled or disabled.

When the checkbox is hit, some codes are processed and the below GUI elements are enabled. Then user selects the profiles in the comboboxes correctly.

After that, when the Apply button is hit, some codes are processed again and the following works is done:

1. Do network settings - This is done according to profile settings that users specify. If they select auto configuration, a set of ip, netmask, gateway addresses is used. If not, user's values are used.
2. Do DHCP server settings - This is done by comar call which is used in Pardus for interprocess communication. It also uses dbus. Here, our application is share, our model is Net.Share, method is CheckShare. The variables that are application, model, method are required for dbus call. After we

call this dbus call, some code segments that is registered to the system are processed. In this process, our dhcp server's configuration file is written to provide necessary infos.

3. Up and down the states of the interfaces (this is for routing table.) - Also this is done by a dbus call.
4. Start DHCP server. This is done by dbus call,too. Here the application is "dhcp", model is "System.Service", method is "start".
5. Start Iptables
6. Load Nat rules
7. Show information box that informs everything is OK. This is done by the function `KMessageBox.information()`.
8. When Ok on the information box is hit, close the dialog.

If any problem occurs, it is informed with an error box.

If you want to disable sharing of your connection, you just deselect the checkbox. When checkbox is deselected, state information of DHCP and Iptables is gotten. Then if DHCP is up, it is stopped. If Iptables is up, its nat rules are flushed to avoid rule overflow, and then it is stopped.

Chapter 3. VPN Connection

3.1. Settings

VPN(Virtual Private Network) is useful to make a connection more securely. In this project, openvpn application is used to provide VPN connection. It is an open source project and mostly used.

The only thing to make a vpn connection using openvpn is to configure openvpn's configuration file correctly. In this scenario, we are client and we are trying to connect an openvpn server.

The connection is done using network sockets. So we need to specify the ip or domain of the client, the protocol, the device type(tap or tun) for connection info. And also we need to specify CA certificate, CERT certificate, key certificate and perhaps cryptographic chipher to communicate securely.

All the information mentioned above is asked from users with a well-done dialog which is in NM's GUI.

3.2. The Code

The code is done in the same way of internet sharing. GUI consists of labels, linedits that ask for connection name, domain, port, comboboxes that ask for protocol, cryptographic chipher, qt file dialogs that ask for CA certificate, CERT certificate, key certificate files.

Users that will make a VPN connection firstly create a profile of VPN device. To do this, in NM's GUI, they click "New Connection" button, then in coming dialog, they select tun or tap in VPN Device section. After that they fill up the related fields. And push "Apply" button. Their configuration is ok now. To connect, they just click the profile's checkbox.