

# Linksys NSLU2 Network Attached Storage (NAS)

## 1 Introduction

The Linksys NSLU2 is a low cost Network Attached Storage (NAS) device distributed with an embedded Linux OS from SnapGear. Linksys has publicly released source code for the NSLU2 on their GPL code center, making this device popular to hackers.

The NSLU2 contains two USB ports for a USB hard drive or USB flash drive; however, other devices such as an 802.11 USB adaptor could be attached to these ports. In addition to the two USB ports, the NSLU2 includes a standard Ethernet port so that the device may be attached to one's local LAN.

The information below is presented in two sections. One section describes how to customize the native NSLU2 software, while the other section presents way to replace the native software with a customized version.

**Table 1 NSLU2 Summary**

<b>Chipset</b>	Intel IXP420 Network Processor (ARM). Includes 8MB of flash and 32MB of RAM.
<b>OS</b>	<a href="#">SnapGear Embedded Linux</a>
<b>Firmware</b>	Version V2.3R24
<b>Ports</b>	2 USB 2.0, 1 10/100 RJ-45
<b>Product Link</b>	<a href="http://www.linksys.com/products/product.asp?grid=35&amp;scid=43&amp;pid=640">http://www.linksys.com/products/product.asp?grid=35&amp;scid=43&amp;pid=640</a>
<b>Hacking Info</b>	<a href="http://www.batbox.org/nslu2-linux.html">http://www.batbox.org/nslu2-linux.html</a> <a href="http://www.nslu2-linux.org/">http://www.nslu2-linux.org/</a> <a href="http://groups.yahoo.com/group/nslu2-linux/">http://groups.yahoo.com/group/nslu2-linux/</a> <a href="http://sourceforge.net/projects/nslu/">http://sourceforge.net/projects/nslu/</a> source for the Unslung image

## 2 Resources

- [Linksys NSLU2 Home page](#)
- Batbox (Jim Buzbee) - [Linux on the NSLU2](#). The primary site for information for customizing the NSLU2.
- [NSLU2 Linux development group](#)
- [NSLU2 Linux discussion group](#) (Yahoo)

## 3 Native Configuration

The NSLU2 firmware can be downloaded from the Linksys GPL Code center. This image includes the SnapGear OS, a boot loader and other components outlined below. By default, access is relatively restricted on this device, and is primarily restricted to web

interface. Any hard drives used by the NSLU2 are formatted with as a Linux ext3 file system, so it is possible to mount these drives to a Linux box (or any host that understands an ext3 file system).

and install new applications with out replacing the Linksys firmware. Any changes made with out modifying the flash image are not persistent after rebooting the NSLU2. It is possible to modify and replaced the Linksys firmware image in order to make those changes persistent.

Common changes made to the standard NSLU2 including enabling telnet and installing additional programs like NSF or as demonstrated on Tom's Hardware an Apple iTune's server. Those changes are described below, in the "Customization" section.

Default IP Address	
Root Password	Not known
Admin Password	admin
Home Directory Telnet	Disabled

**Table 2 Default NSLU2 Configuration Parameters**

## 4 Image

The NSLU2 is loaded with a SnapGear embedded Linux distribution. This image can also be downloaded from the Linksys website. This image contains four image components: RedBoot, system configuration, kernel, and RAM Disk.

These components can be extracted from the Linksys image using the "splitnsul" utility from Brian Lantz. Or the Linux "dd" utility can be used to manually extracted the appropriate sections.

### 4.1 Image Components

#### 4.1.1 RedBoot

RedBoot is an open source debugger and bootloader developed by RedHat. It is based on the eCos operating system and is commonly used with embedded Linux systems running on ARM, MIPS, MN10300, PowerPC, SH (Hitachi), NEC v850, and x86 (Intel) platforms.

RebBoot can provide debugging support over serial and Ethernet interfaces using gdb. It also provides utilities to manage flash images including removal, download, update, and booting.

Linksys does not offer source code for RedBoot, but source code is available from RedHat.

See the [RedBoot website](http://sources.redhat.com/redboot/) (<http://sources.redhat.com/redboot/>) for more information on this component.

### **4.1.2 System Configuration**

Persistent configuration data is stored in the System configuration component.

### **4.1.3 Kernel**

SnapGear Linux is a freely available, open source, embedded Linux distribution. It supports platforms such as the NSLU2 that do not have a Memory Management Unit (MMU). According to the SnapGear website, SnapGear Linux is in use in more than 20 million devices.

SnapGear uses the uClibc library and includes tool chains for various platforms. More information is available from the [SnapGear website](http://www.snapgear.org/snapgear/about.html) (<http://www.snapgear.org/snapgear/about.html>).

### **4.1.4 RAM Disk (RAMFS)**

The RAM Disk contains the file system, libraries, configuration files, scripts, executables, etc. The RAM Disk is a compressed RAM file system (RAMFS) that starts at 0x160010 in the Linksys image file.

Once can extract this image, decompress it, and mount it via a loop back interface.

## **5 Customization**

There are a variety of individuals working to customize the NSLU2. Modifications can be separated into those that are persistent across device reboots and those that are not. Persistent changes require a modification to the flash image, while non-persistent changes involve a change to the in memory (RAM) system.

Ranging from those making non-persistent changes to services or application running from RAM, while others have rewritten the flashed image.

This image includes the SnapGear OS, a boot loader and other components outlined below. By default, access is relatively restricted on this device, but its possible to enable telnet and install new applications with out replacing the Linksys firmware. Any changes made with out modifying the flash image are not persistent after rebooting the NSLU2. It is possible to modify and replaced the Likysys firmware image in order to make those changes persistent.

Common changes made to the standard NSLU2 including enabling telnet and installing additional programs like NSF or as demonstrated on Tom's Hardware an Apple iTunes's server. Those changes are described below, in the "Customization" section.

## **5.1 Non-Persistent Modifications**

Non-persistent modifications to the NSLU2 involve changes not saved to disk. These changes last until the device is rebooted. The most common non-persistent change involves enabling telnet access to the NSLU2.

### **5.1.1 Mounting an External Hard Drive**

Plug the USB drive into a Linux box. If Linux recognizes the USB drive then there will be a message in /var/logs/messages.

After attaching the device run “dmesg” to gather information about the partitions on the drive. Assuming the drive is recognized as /dev/sdb there are typically three partitions: sdb1, sdb2, sdb3. sdb1 is the data partition, the sdb2 partition contains configuration data, and sdb3 is swap space.

Mount the data or configuration partition with the standard mount command:  
Mount /dev/sdb2 /mnt/usbdrive

(You will need to create the usbdrive directory)

### **5.1.2 Enabling Telnet Access**

Enabling telnet access is somewhat of a hack. Telnet can be enabled/disabled by accessing a hidden web page in the NSLU2’s web interface, but to finalize access one must alter a password file stored on the primary hard drive. In many respects enabling telnet is a persistent change – once you enable it via the web interface it remains active even after a reboot. Even though these changes are persistent, no direct user modification of the firmware is ever performed, so this change is classified as a non-persistent change.

To enable telnet, go to the root NSLU2 web configuration site and go to /Management/telnet.cgi. This should load a new page asking if you want to enable telnet access. Click on the button to toggle telnet as appropriate.

This enables telnet access for the root or the “ourtelnetrescueuser” user, but we don’t have access to their passwords, so one more hack must be performed. Use the NSLU2 web configuration utility to create a new user account and password. This information is saved to the primary hard drive and can be edited if you mount the drive (the drive is using an ext3 file system, so any Linux box should be able to mount it). Edit the “passwd” file on the hard drive and either change the root or telnet user’s password to your new account password, or edit your shell to be “/bin/sh” instead of “/dev/null”.

Once this is done, you should be able to log into the device.

### **5.1.3 Creating and Installing a New Application**

A new application can be built using the SnapGear tool chain and sources (use the SnapGear tool chain from the Linksys WRV54G firmware). Once built, the executable can be installed on the primary hard drive and manually executed. Without changing the

firmware image, it is not possible to automate the startup of the new application. The application must be manually restarted each time the NSLU2 is restarted.

## **5.2 *Persistent Modifications***

The most straight forward method for modifying the NSLU2 firmware is to download the NSLU2 firmware from Linksys and separate (unpack) it into its four components (RedBoot, system config, kernel, and RAM Disk) using the “splitnsul” utility.

Uncompress and mount the ramdisk and add any desired scripts and executables. When finished making changes to the ramdisk, compress it and use the splitnsul to re-pack the image components. The new, packed, image can now be uploaded to the NSLU2.

### **5.2.1 Unslung**

NSLU2 hackers have developed a very customized firmware image known as “Unslung” based on the SnapGear firmware. It provides the same functionality as the original Linksys firmware, but mounts the root file system to an external disk instead of a RAM disk. Mounting to the external drive frees up 10 MB of RAM. It also implements a package manager system that provides easy upgrade options for the NSLU2.

The addition of the package manager, ipkg, allows existing applications to be easily updated or new application to be installed. Applications can be installed or updated over the network just by issuing a command to the package manager. One could write new applications that could be installed, updated, or removed using the package manager.

Also, since the root file system is no longer mounted in RAM, much larger and more demanding applications can be ran on the NSLU2.

The Unslung firmware is under active development, with new features pending. There is an additional Unslung branch called OpenSlug, whose goal is to completely replace the Linksys firmware including the kernel and bootloader. The OpenSlug product is still very experimental.