

## Setup

Hardware used:

Dell Optiplex 780
Intel Core 2 Duo 2.93 Ghz x 2
Ubuntu 12.04 64-bit
3.7 Gbs of ram
120 Gbs of disk space

Essential Software:

GNU Linux Kernel v3.2.\* [kernel.org](http://kernel.org)

git [git-scm.com](http://git-scm.com)

Suggested Software:

cscope [cscope.sourceforge.net](http://cscope.sourceforge.net)

ctags [ctags.sourceforge.net](http://ctags.sourceforge.net)

## Installation

From our git repository:

Open up a terminal and type the following commands.

1	git clone https://github.com/COP5641-2012-Group-Project/linux.git
2	cd linux
3	git checkout v3.2-usb_no_boot

Once this is done you should set up the .config file for your system. Our project does not require any special kernel configuration so you can customize it or use the default .config that came with your distro. If that won't work for you, you can try the following command, however this is not a perfect solution so some google-ing of how to setup .config files is suggested.

4	make config localmodconfig
---	----------------------------

After you have done this you need to replace the hard coded serial number with the one belonging to your device. This is found in [drivers/usb/core/hub.c](#) on line 1905.

5	vim drivers/usb/core/hub.c
---	----------------------------

**Note: some USB devices, like HID devices, won't have a serial number. In these cases serial will be NULL**

Now that your USB serial is the source and your .config file is setup you are ready to build the kernel.

6	sudo make -j5
7	sudo make -j5 modules_install

To install your newly created kernel.

8	sudo make install
9	sudo update-grub

From the source and our patch:

If you already have the source for the 3.2.\* GNU Linux kernel you can just apply our patch with the following commands

1	cd <LINUX SRC DIR>
2	wget http://ww2.cs.fsu.edu/~easton/KDD/v3.2-usb_no_boot.patch

You should then verify the md5. It should be “883898ad59a1a58f481ec324d52c80a5”

3	md5sum v3.2-usb_no_boot.patch
---	-------------------------------

You then need to apply the patch.

4	git apply v3.2-usb_no_boot.patch
---	----------------------------------

After you have done this you need to replace the hard coded serial number with the one belonging to your device. This is found in [drivers/usb/core/hub.c](#) on line 1905.

5	vim drivers/usb/core/hub.c
---	----------------------------

**Note: some USB devices, like HID devices, won't have a serial number. In these cases serial will be NULL**

Now that your USB serial is the source and your .config file is setup you are ready to build the kernel.

**You did setup your .config file... right?**

6	sudo make -j5
7	sudo make -j5 modules_install

To install your newly created kernel.

8	sudo make install
9	sudo update-grub

## Functionality

Allows a user to add an additional layer of authentication. Specifically, it requires that users authenticate themselves using something they have (the USB device) rather than just something they know (a password).

This does not provide the option for more than one device to be used as a key; however it could easily be extended to accommodate that.

## Use Case

If the user had physically locked down their computer, password protecting the BIOS changing computers boot sequence to disallow booting from removable media, then this addition to the GNU Linux kernel would indeed add some measure of additional security.

If the user had not implemented the aforementioned security measures, this system could easily be overcome with a Linux live CD.