How to Enable Legacy FileVault on Mac OSX 10.7 Lion

The purpose of this guide is to show how to create new users WITH "Legacy FileVault" enabled on upgraded or fresh installs of OSX 10.7 Lion.

For years, the "solution" given to us by Apple for data security was "FileVault". Starting in OSX 10.3 Panther, users could "encrypt" their home directories. Home directory encryption is better than nothing, but there are still some very real problems. If your system is physically compromised (i.e., taken from you), the opposition will not have access to all the data in your home directory; however they can still make inferences by looking at all the other unencrypted data on your system. Worse yet, if you are foolish enough to use Sleep or Hibernate, it is theoretically possible for a skilled opposition to recover your encryption keys from the unencrypted temporary space on the drive.

The REAL solution has been, and always will be, Whole Disk Encryption (WDE) at boot time based on a strong user provided key. This combined with good security practices will render a system almost impervious to any opposition, whether that opposition is a nosy co-worker, professional hacker, or government agency. In an ideal world, the BEST WDE would also offer "plausible deniability". Plausible deniability would render encrypted hard drives that were mathematically and legally indistinguishable from random data and boot from an easily discarded SD card or thumb-drive. Plausible deniability would also allow the user to "boot" into a completely different environment from the one they normally work in which would be hidden within the encrypted disk and indistinguishable.

Starting with OSX 10.7 Lion Apple has FINALLY woken up to the concept of WDE, albeit not the ideal kind with plausible deniability. Still, it is better than only home directory encryption, and WAY better than being naked on the battlefield. The new system is just an extension of FileVault, called in some cases "FileVault 2" and in other cases just "FileVault" where the pervious system is call "Legacy FileVault".

By design, OSX 10.7 Lion is supposed to be installed as an "Upgrade" to an existing OSX 10.6 Snow Leopard system, however there are several methods for installing OSX 10.7 Lion on a clean hard-drive. In the upgrade situations, if the user had a FileVault encrypted home directory, the encryption system is carried forward into the New OSX 10.7 Lion install and named "Legacy FileVault". It behaves exactly like OSX 10.6 Snow Leopard in every respect. The one area that differs is that new users on the system are unable to activate "Legacy FileVault" for their home directories. Any attempt to use the FileVault format from the Security and Privacy System Preference Plane will result in the user being required to "upgrade" to the new WDE FileVault.

Having said that WDE is better than home directory encryption, why does this guide exist? Apple's new WDE FileVault does provide better security than its previous iteration, however that added security comes with a price. WDE FileVault is not 100% compatible with various configurations and installs of Apple Bootcamp software, which allows Mac owners to run multiple operating systems on the same computer. WDE FileVault is also not compatible with users running OSX 10.7 Lion on non-Apple-approved hardware such as being done in the active Hackintosh OSX86 communities. Finally, because there is only 1 encryption key to unlock the disk in WDE, there are situations were multiple users of the same computer need to be blocked from each other with strong encryption, but not share the same encryption key.

What you need to have to complete this guide:
1. A fulling working OSX 10.7 Lion system
2. (optional) Access to a OSX 10.6 Snow Leopard system running FileVault.
3. Some method to move files between the two aforementioned computers (thumb drive, network, email, etc)
4. Be semi-comfortable with the Terminal and some UNIX commands
5. An understanding that modern computers have different users and user permissions.

6. About 30 minutes of time.

Disclaimer
Your data is WAY more important than your hardware. READ THROUGH THIS GUIDE COMPLETELY before trying it. If you are not comfortable or don't understand what is going on, don't do it. If you do this wrong, it could break your computer, burn down your home/office, cause your spouse to leave you, and/or piss off Apple. I am in no way responsible for anything you do. Having said THAT, unless you do something REALLY foolish like "sudo rm -rf /" you should be fine. This guide is very benign and messing up the steps would only affect the user you are trying to create and shouldn't spill over into other users or the system.

Of course, the best of all words would have you performing this guide on a fresh install of OSX 10.7 Lion on a clean system with no user data, but I guess that's unrealistic for most people.

A Note about users
This guide can be done on a single user system. However, you might find that it will be easier on some steps to preform the guide logged in as a secondary user NOT as the primary user. Conversely, I strongly recommend that if you are working with a "production" system (i.e. the computer you do "work" from, or your primary home computer) you take the time to setup a second user and try preforming this guide on THAT SECOND USER BEFORE TRYING IT ON THE PRIMARY. One can perform all the steps to the second account while logged into the primary account. I would advise you to log into the second account before following the guide just to make sure the account works. This way if you totally mess up, you are not endangering your primary user account. After you have performed this a couple of times, and feel confident in the procedure, you can then log into the second account and preform the steps on the primary.

How to read this guide:

Commands that you need to execute will be **BOLD**
Messages back from the system will be in *Italics*

Because this guide deals with the creation of a user, these commands are very specific to the USERNAME and PASSWORD of the user you are trying to create. Apple, like all modern *nix systems, has a "Full Name", "Short Name", and "Password" for each user. My Full Name is McKinley H. Tabor, the Short Name I typically use is mckinleytabor, and my Password is none of your business 🙂. In this guide I will use "McKinley H. Tabor", "mckinleytabor", and "mckinleyspassword" in places where that data is needed. You will of-course need to replace these with your own corresponding information, unless of course your name is McKinley Tabor, in which case please contact me!

This guide is written for and tested by people who have a working knowledge of computers. There are some high level concepts here, but I will try and break them down into byte-size chunks. In some cases where there is a lack of uniformity and specific instructions are not feasible, I will give high level direction such as "copy files from here to there". In cases where there is uniformity I will give the entire commands in sequence, thus making "copy and paste" easier. While you are NEVER supposed to Copy and Paste commands into the terminal from some strange guy's website, we all do it and I guess I'm as trustworthy as anyone.

Step 1: The Chicken and the Egg.

NOTE: If you do not have a working OSX 10.6 Snow Leopard computer, it is possible to still create a working Legacy FileVault on OSX 10.7 Lion, however you will LOSE the ability to recover from a forgotten password by using the

"Master Password" recovery option. In practice this may not be a big deal for you. If you wish to proceed without the Master Password recovery option, skip to step 3.

FileVault on OSX 10.7 Lion requires the system to generate a Certificate and Keychain for it to work. The only means by which you can generate a Certificate and Keychain for FileVault on OSX 10.7 Lion is to enable FileVault. Of course, doing so will encrypt your entire drive, and possibly break BootCamp and most definitely will break your Hackintosh. The trick is to get a Certificate and Keychain into OSX Lion WITHOUT enabling FileVault. As it turns out, FileVault's Certificate and Keychain did not change much between OSX 10.6 Snow Leopard and OSX 10.7 Lion. Therefor it is possible to copy the FileVault Certificate and Keychain from the older system and use it on the newer one.

On both systems the FileVault Certificate and Keychain are located in:

/Library/Keychains/

You will need to copy two files from this directory on the OSX 10.6 Snow Leopard to the same directory on OSX 10.7 Lion. Of course, if you have not been using FileVault on OSX 10.6 Snow Leopard then these two files WILL NOT BE THERE.

The two files are:

FileVaultMaster.cer
FileVaultMaster.keychain

One of these files (FileVaultMaster.cer) is only readable by the "root" user, so you will need to copy them via elevated permission in the terminal.

On the OSX 10.6 Snow Leopard system Open Terminal (Hard Drive -> Applications -> Utilities -> Terminal)
From the Prompt:

```
sudo cp /Library/Keychains/FileVaultMaster.* ~/Desktop
```

You will be prompted for your password, enter it to continue.

You will see your two files now on your desktop.

Depending on your configuration, you may may need to run this next command in the Terminal in order to work with the files you just copied.

```
sudo chmod a+rw ~/Desktop/FileVaultMaster.*
```

If prompted for your password, please enter it.

At this point you will need to copy these two files from this OSX 10.6 Snow Leopard computer to your new OSX 10.7 Lion computer. You can do this via any method. Place these two files on the desktop of the OSX 10.7 Lion computer.

Step 2: Installing the FileVault Certificate and Keychain into OSX 10.7 Lion.

On the OSX 10.7 Lion machine open up terminal, we are going to copy the FileVaultMaster files into place and set their permissions properly.

```
sudo cp ~/Desktop/FileVaultMaster.* /Library/Keychains
sudo chown root:wheel /Library/Keychains/FileVaultMaster.*
sudo chmod 600 /Library/Keychains/FileVaultMaster.cer
sudo chmod 644 /Library/Keychains/FileVaultMaster.keychain
```

You have now "installed" the FileVaultMaster files on the OSX 10.7 Lion system.

Step 3: Confirm User Information and set Variables

Our netx steps will confirm that we are working with the right user and setup some BASH global variables to make life easier for us. These will be SBUSERNAME, SBUID, and SBGID. From here on out these steps

```
umask 077
export SBUSERNAME="mckinleytabor"
export SBUID=$(id -u $SBUSERNAME)
export SBGID=$(id -g $SBUSERNAME)
echo Username $SBUSERNAME - UserID $SBUID GroupID $SBGID
```

If all goes well you should get back something like:

```
Username mckinleytabor - UserID 501 GroupID 20
```

Note on UserID: Each user on the system as their own UserID number. On OSX 10.7 Lion (an other versions os OSX) the first user of the system, you one you created at install, has a UserID of 501. If you are running this procedure on a second user as a test, that user will have a UserID of 502, 503, 504, etc depending on how many users you have created over time. This guide will work to create a LegacyFileVault for ANY user regardless if it's the first user or the three-hundredth. Because these commands are being done as the "Super User", it will also work to create a LegacyFileVault for the user you have currently logged in.

Step 4: Go into the Users Directory

In most cases you may be already in the correct directory, but it never hurts to make sure

```
cd /Users/"$SBUSERNAME"
pwd
```

The "pwd" command will show the directory you are currently in, if all goes well you should see:

```
/User/mckinleytabor
```

Step 5: Create the sparsebundle

These are the commands to generate the encrypted sparsebundle that will be the FileVault. The "sparsebundle" is a type of disk image used by Apple for various things. If you're interested in more information on them, it can be found here: http://en.wikipedia....ki/Sparse_image

There are three considerations here: size, password, and the Master Password Recovery.

Size. The size of your sparsebudle determines the maximum amount of data you can store in it. The actual size of the sparsebundle on the disk will change based on the data contained within. There have been some discussions about the use of the "autostretch" switch when creating the sparsebundle so as to avoid a maximum data top end. We have not tested the use of autostretch and for the time being recommend you stick with a hard cap, albeit a large one. Pick a size based on the overall capacity of your disk. In a single user computer, it would not hurt to have the sparsebundle be 90% of the total disk capacity. Undersizing the sparsebundle is far more detrimental than oversizing it. The sizes will be noted as Gigabytes so 300g is 300 gigabytes, 1000g is a terabyte. In the examples below I have used "300g" to denote the creation of a three hundred gigabyte sparsebundle. Please change this number to the size that best suits you.

One more note about size. If this is new system install, or setting up Legacy FileVault on a new user, there should be no problem with disk size. However, if you are setting up Legacy FileVault for an established user with LOTS of data, the overall disk size might be an issue. During the data "Population" in step 9, you will be doubling the data for the user on the disk for a short time. Therefore, if you have a 300 Gigabyte hard drive, and the user has 200 Gigabytes of data, it will be impossible to do the data population in step 9, because in doing so you will run out of space on the disk. There are two ways around this. First is to create the Legacy FileVault for a new user and move the data over from the old user once you have established that everything is working. Second is to backup the data off the machine to an external source, delete the data off the machine, and then copy it back once Legacy FileVault is working.

Password. You MUST use the same password as the one you have for the User you are setting up FileVault for. If these passwords are different, you will get an error when logging in.

Master Password Recovery. The Master Password Recovery is based on the transfer of the FileVaultMaster files from steps 1 and 2. Your Master Recovery Password will be different from the password you use for the sparsebundle and will have been set on the OLD OSX 10.6 Snow Leopard Machine. Ergo, you will never be asked to set a master password on this OSX 10.7 Lion. The Master Password is used in two scenarios. First, if you forget your normal password that unlocks the sparsebundle and logs you into the system. Second, if there have been too many bad password attempts.

THIS IS IMPORTANT. If you had to skip steps 1 and 2 because you did not have a OSX 10.6 Snow Leopard machine running FileVault so you could copy FileVaultMaster files, you will need to create a sparsebundle without referencing the FileVaultMaster files. This will impair your system only slightly, and it will still work in normal day-to-day operations.

Because the sparsebundle creation command is different depending on whether or not you have copied the FileVaultMaster files, I have listed both. In both cases I am still creating a 300g sparsebundle, so edit the command as necessary for your own size. Also the "\" characters allow for a single command to have multiple lines. These characters can be omitted if you are typing them out on a single line.

Command if you HAVE the FileVaultMaster files.

```
hdiutil create -size 300g \
-encryption -agentpass \
-certificate /Library/Keychains/FileVaultMaster.cer \
-uid $SBUID -gid $SBGID \
-mode 0700 -fs "HFS+J" -type SPARSEBUNDLE -layout SPUD \
-volname "$SBUSERNAME" "$SBUSERNAME".sparsebundle
```

Command if you DO NOT have the FileVaultMaster files.

```
hdiutil create -size 300g \
-encryption -agentpass \
-uid $SBUID -gid $SBGID \
-mode 0700 -fs "HFS+J" -type SPARSEBUNDLE -layout SPUD \
-volname "$SBUSERNAME" "$SBUSERNAME".sparsebundle
```

(The only difference is the removal of the third line)

When asked for a password, USE THE SAME PASSWORD you used when the account was created.

If it all works you should get back:

```
created: /Users/mckinleytabor/mckinleytabor.sparsebundle
```

Step 7: Set FileVault Permissions

This will set the sparsebundle to have the correct permissions for the user you are creating it for.

```
chown -R "$SBUSERNAME":staff "$SBUSERNAME".sparsebundle
```

Step 8: Mount sparsebundle in a temporary place to check it.

Now that the sparsebundle has been created, we will want to mount it up and test it to make sure it works before populating it and attaching it to a user. We will create a temporary directory in the user's home folder and mount the sparsebundle to that directory. This temporary directory will be removed in a later step.

```
mkdir sbdest
hdiutil attach -owners on -mountpoint sbdest \
-stdinpass "$SBUSERNAME".sparsebundle
```

You will be prompted for the sparsebundle password and if all goes well you should see something like:

```
/dev/disk2 Apple_partition_scheme
/dev/disk2s1 Apple_partition_map
/dev/disk2s2 Apple_HFS /Users/mckinleytabor/sbdest
```

Step 9: Populate the sparsebundle

This next command will move all the data you need into the sparsebundle for the user to use it as their home directory. It is important, however, before we do the step that you reflect on just how much data you are going to copy. In doing this we will effectively DOUBLE the size on the user's home folder until we can confirm the sparsebundle login and delete all the unencrypted data. If this is LESS than the total size of the disk, then go head with the next command. If it is more STOP NOW and consider making a second user to create the Legacy FileVault for or backing up all of your data off this machine, deleting the data once it has been backed up, then copying it back after we can confirm Legacy FileVault is working.

```
rsync -avxHE ./ sbdest/ \
--exclude="$SBUSERNAME".sparsebundle/ --exclude="sbdest/"
```

This step will take ether a few moments or several hours depending on the size of your data and the speed of your machine. If you followed the directions properly and took time to reflect on the data size, you'll know if you need to wait at the machine, or come back later. There will be lots of messages flying access the screen, these are just telling you what files are being copied.

Step 10: Unmount sparsebundle and remove temporary directory

After all the data has been moved it's time to unmount our now populated sparsebundle and remove the temporary directory we created in step 8.

```
hdiutil detach sbdest
rmdir sbdest
```

Step 11: Modify User's Profile to use sparsebundle

Heretofore everything in the guide has been nondestructive from a system standpoint. (unless you have been moving and deleting data to get it to fit in the sparsebundle, in which case God help you) This means that up until now everything that has been done should NOT affect the way our system boots or the way you log into it. At this point, you could stop and simply delete the sparsebundle from the directly in which you created it, and it would be like you never tried doing any of this.

These next steps WILL affect how your system logs into a user space, and doing them wrong WILL fubar your user account. We will be be making backups along the way so you can recover. But just remember that from here on out, there be dragons. One last time, please reference my notes above about creating second users and about how I'm not responsible for what you do. 🙂

Apple keeps profile login information separate from the normal places users go. We are going to dive deep into the directories and alter a user profile to use the sparsebundle as the home folder.

First change the directory to the location where user profile data is kept by the system.

```
cd /private/var/db/dslocal/nodes/Default/users/
```

Next BACKUP the user profile we have been working with just in case you mess up.

```
cp "$SBUSERNAME".plist "$SBUSERNAME".plist.backup
```

Apple stores the user profile information in a binary format, we will need to convert this into text so we can edit it. We will later convert the edited file back into the binary format.

```
sudo plutil -convert xml1 "$SBUSERNAME".plist
```

There is an old civil war about UNIX editors. There are a couple of good ones ones on OSX, but for this we are going to use nano.

Edit the file:

```
sudo nano "$SBUSERNAME".plist
```

This file will contain lots of data about the user.

Look for the "home" key, it should look like this:

```
<key>home</key>
<array>
<string>/Users/mckinleytabor</string>
</array>
```

You are going to add a couple of lines to the end of this "home" key. In essence you will make it look like:

```
<key>home</key>
<array>
<string>/Users/mckinleytabor</string>
</array>
<key>home_loc</key>
<array>
<string>&lt;home_dir&gt;&lt;url&gt;file://localhost/Users/mckinleytabor/mckinleytabor.sparsebundl
</string>
</array>
```

Under the The "home_loc" and "array" keys, the "string" key is all on one line. Here, like in the rest of the guide, you will need to swap out "mckinleytabor" for the real short name of user you are setting up the Legacy FileVault for.

After you have edited the file, you can save the file in nano by Ctrl-O (O as in Oscar), then Ctrl-X to exit.

Finally you will need to convert the plist file back to binary format

plutil -convert binary1 "$SBUSERNAME".plist

If anything goes wrong on the login, you need only to copy the "$SBUSERNAME".plist.backup file back to "$SBUSERNAME".plist overrating edited file.

Step 11: (option) Clean up Unused folder in Home Directory

This is another Chicken and Egg issue. After populating the sparsebundle, there will be lots of files left over in the unencrypted home folder. You can at this point delete those files, however that is not recommend because you have not, as of yet, established that your sparsebundle login works, and its much more difficult "go back" if all your files are locked up in the sparsebundle and nowhere else. If you are working on a second user however, you can go head and delete everything but sparsebundle folder.

The problem comes in when you have "successfully" logged into the user account with the Legacy FileVault. At that point OSX mounts the sparsebundle as the users home folder, obscuring any and all data it once contained. It's still on the disk taking up space, but however you are unable to get access to it logged in as the Legacy FileVault user. The best option for a "post login" clean up is to do the cleaning from the terminal while logged into another account. You will also need to be the "Super User" when doing this because OSX locks away users home folders from each other.

Step 12: Login and Enjoy

Logout and Log Back in as the Legacy File Vault User. If all went well you should see your Desktop and Documents just as they were. You might have a warm feeling around your backside knowing that your ass is covered should someone take your system. Enjoy that feeling along with the power and the civil liberties encryption gives us all.

Step 13: After-thoughts

You may get a message about "Updating" to the new FileVault. You will of course NOT want to do this. Just click no, and go about your business.

A good way to test your Legacy FileVault is to open the System Preferences and look under Security & Privacy. You should now see "Legacy FileVault". If you did not have access to a OSX 10.6 Snow Leopard system and had to skip steps 1 and 2, there will be an option to Set a Master Password in this Legacy FileVault tab. However because the sparsebundle you created did not reference this, setting a Master Password will have no effect your system.

Encryption is sort of a religious requirement in my line of work. But encryption alone is not good enough to protect you. You need to combine strong encryption with good security procedures in order to maximize your protection. I would encourage you to take the time to read up on how to protect your data and your identity.

Acknowledgements :

The idea and some of the sequencing for this article came from Fabio Maione. <http://lab.maiux.com...t-in-os-x-lion>

A "Living" version of the document can be found on my website at: http://www.taborcg.com