

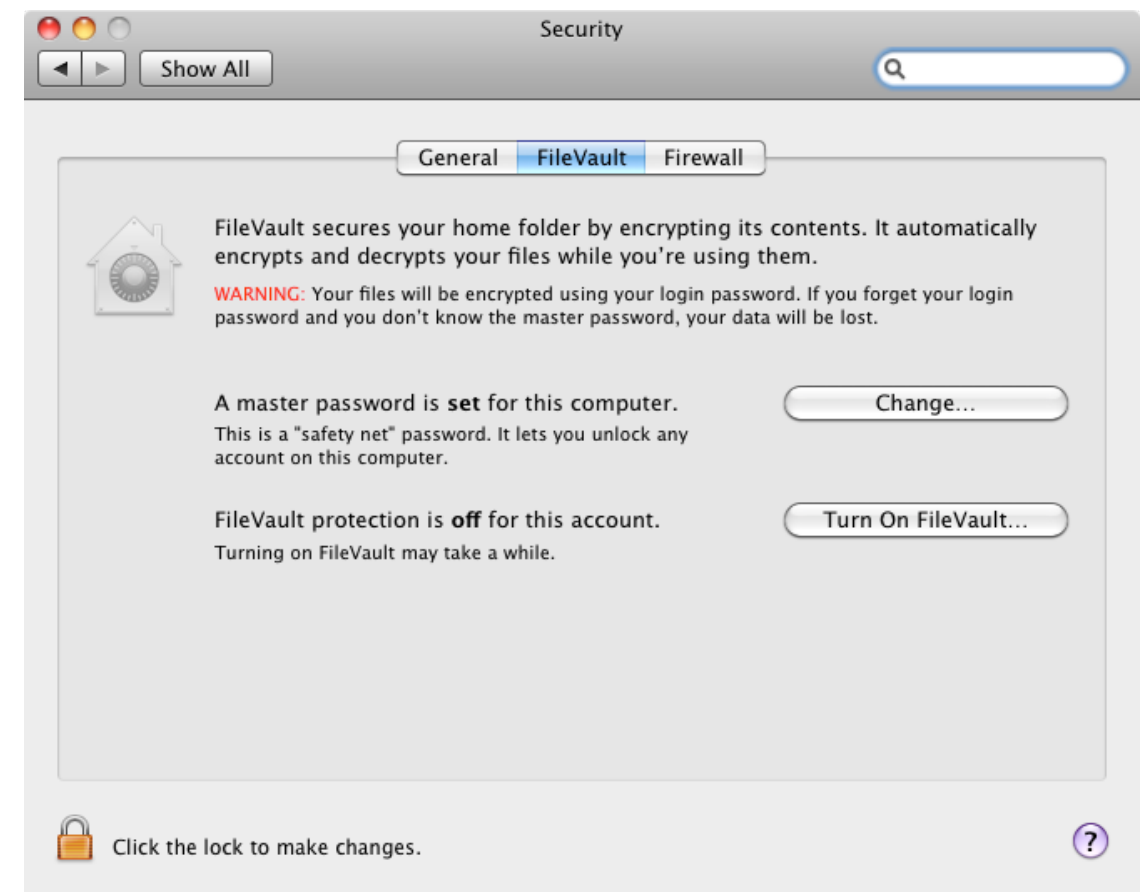
FileVault 2
Decoded

Similar Names, Different Beasts

- Apple completely revamped FileVault in Lion
- Grown from a encryption solution that protected only home folders to one that can protect entire drives
- For simplicity, the older FileVault encryption will be referred to as “FileVault 1”

FileVault | Upsides

- Strong encryption
- Came with the OS, no extra charge to use it
- Designed to work like an decrypted account wherever possible



FileVault | Downsides

- Backups
- Network accounts
- Whole disk encryption not available



Back to the drawing board

- Complete rebuild of FileVault for Lion
 - Uses new virtual volume storage (Core Storage)
- Core Storage encrypted volumes are built on a per-partition basis
- Allows both encrypted and decrypted partitions on the same physical hard drive

How FileVault 2 works

- On startup, the Mac initially boots to a small decrypted partition that only provides access to the tools to unlock the larger encrypted storage.
- When the right authentication is provided, the encryption unlocks and the Mac boots from the Mac's regular OS.
- By unlocking the encryption before the OS boots, the issues with network accounts and backups are solved.

FileVault 2 and Recovery HD

- One of the other new features in Lion was the Recovery HD partition.
- Small hidden partition that provides tools to fix or reinstall the OS.
- To use FileVault 2, you need to have the Recovery HD partition present.
- Why? Because Recovery HD provides the unencrypted space needed to unlock and boot your encrypted Mac.



Name

Password



Sleep

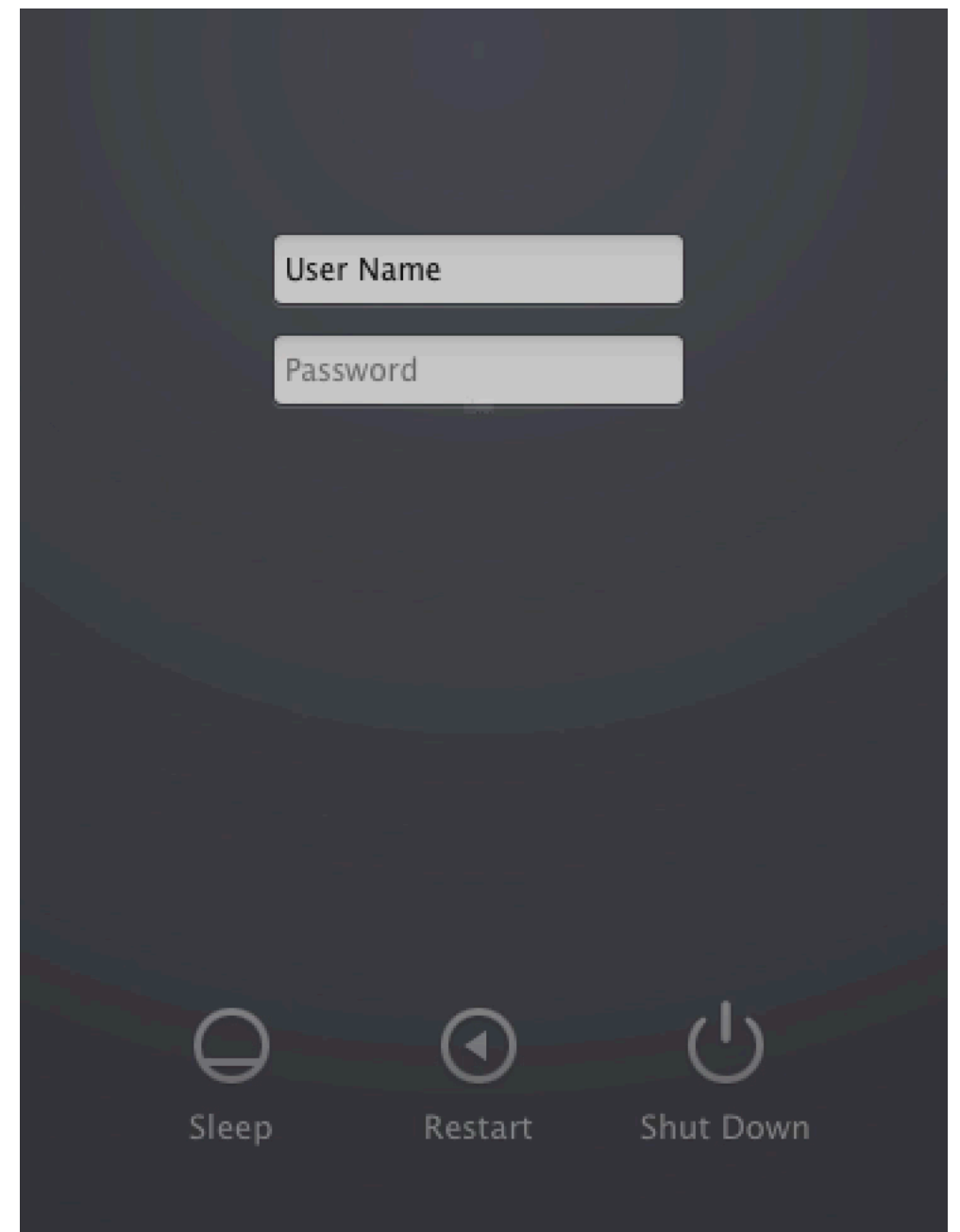
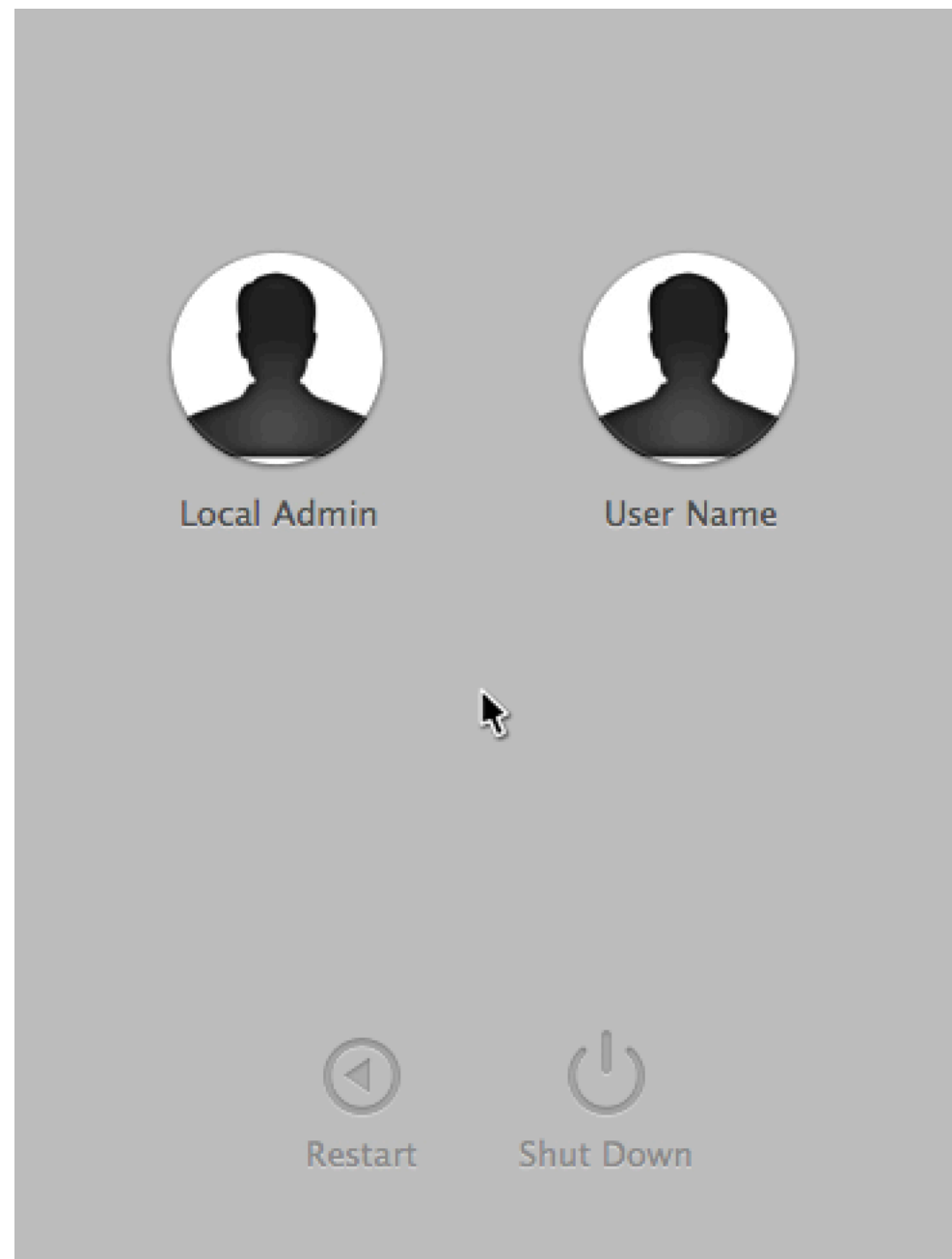


Restart



Shut Down

Using the Personal Recovery Key To Reset Passwords



Institutional Deployment

- With the exception of how the recovery key is generated and handled, setting up FileVault for personal use and setting it up for an institutional deployment can be exactly the same.
- To avoid the administrative headache of tracking multiple individual recovery keys, Apple brought one part of FileVault 1 into FileVault 2.

Sole Survivor - FileVaultMaster.keychain

- What does FileVaultMaster.keychain do?
 - It sets a backdoor to encrypted Macs and is an alternate way to unlock the encryption when the account passwords don't work.
 - Apple calls this the Master Password.
- What role does the password you set as the Master Password actually play?
 - It's the password used to unlock the FileVaultMaster.keychain.



Sole Survivor - FileVaultMaster.keychain

- Wait, what? If all the Master Password does is unlock a keychain, how does it do recovery?
 - FileVaultMaster.keychain's contents are what actually do the recovery.
 - Inside the keychain is a public key (shows up as a SSL certificate) and an accompanying private key.
 - When you have both keys available in the FileVaultMaster keychain, you can use them to unlock or decrypt FileVault 2's encryption.



Sole Survivor - FileVaultMaster.keychain

- You can set a institutional recovery key for a FileVault 2 managed deployment in exactly the same way that you set the recovery key in FileVault 1.
- Crucial difference
 - In FileVault 1, you could have both the private and public key stored in the FileVaultMaster.keychain when you encrypted.
 - In FileVault 2, only the public key can be stored in FileVaultMaster.keychain when you encrypt your Mac.

Creating an Institutional Recovery Key

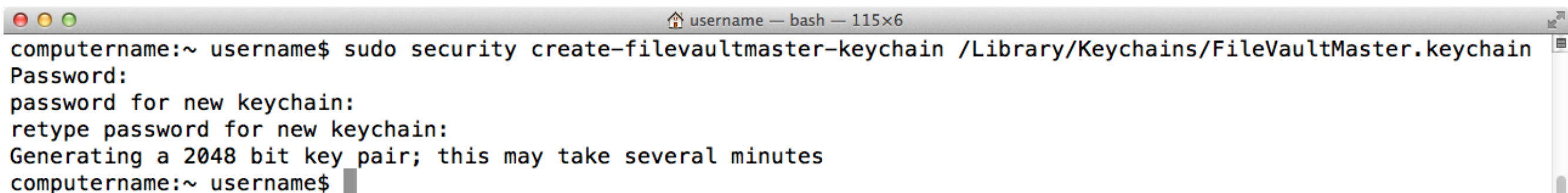
- Make your FileVaultMaster.keychain by setting the Master Password on a specific machine. (You can skip this step if you've already got a set FileVaultMaster.keychain)
 - This FileVaultMaster.keychain will contain both private and public keys.
- Next, make several copies of the FileVaultMaster.keychain file and store the copies in a secure place.
 - A locked safe would be a good place, or in an encrypted disk image on a secured file share.

Creating an Institutional Recovery Key

In 10.7.2 and higher, you can create FileVaultMaster.keychain from the command line.

To create a FileVaultMaster.keychain:

security create-filevaultmaster-keychain /path/to/FileVaultMaster.keychain

A screenshot of a macOS terminal window. The title bar shows 'username — bash — 115x6'. The terminal text shows the command 'sudo security create-filevaultmaster-keychain /Library/Keychains/FileVaultMaster.keychain' being executed. It prompts for a password, then asks for a password for the new keychain, and then asks to retype the password. It also shows a message: 'Generating a 2048 bit key pair; this may take several minutes'. The prompt returns to 'computername:~ username\$'.

```
computername:~ username$ sudo security create-filevaultmaster-keychain /Library/Keychains/FileVaultMaster.keychain
Password:
password for new keychain:
retype password for new keychain:
Generating a 2048 bit key pair; this may take several minutes
computername:~ username$
```

You'll be prompted to set a password. Please enter the Master Password you want to use at this point.

Creating an Institutional Recovery Key

- Once you've got copies, unlock your FileVaultMaster.keychain by running the command below and entering the Master Password when prompted for the password:

security unlock-keychain /path/to/FileVaultMaster.keychain

- After FileVaultMaster.keychain unlocks, go into Keychain Access and access FileVaultMaster.keychain
 - Remove the private key. It will be called **FileVault Master Password Key** and its kind is listed as **private key**.



Name

Password



Sleep



Restart



Shut Down

Deploying an Institutional Recovery Key

- You can deploy the prepared FileVaultMaster.keychain using a variety of methods.
 - Install it via an installer package
 - Include it with your image
 - Copy it to your Macs using your system management tool(s).
- A FileVaultMaster.keychain can be built once and then deployed to as many Macs as needed.



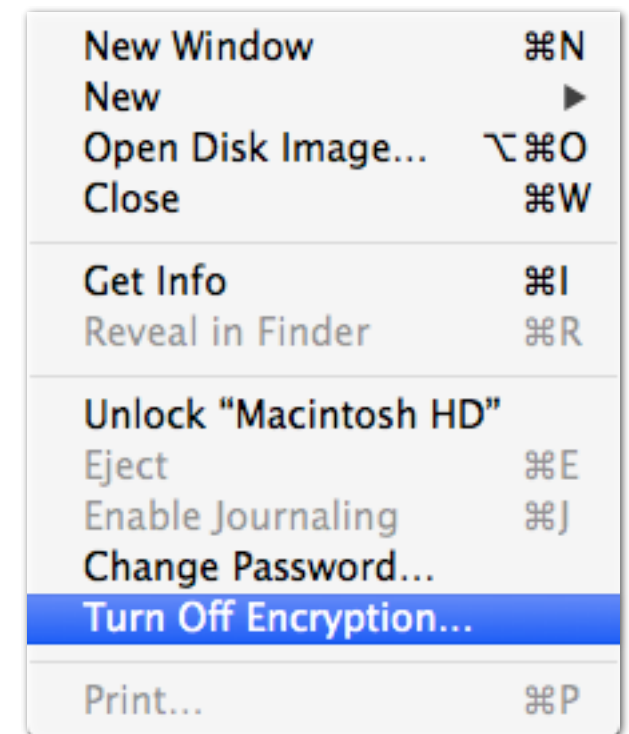
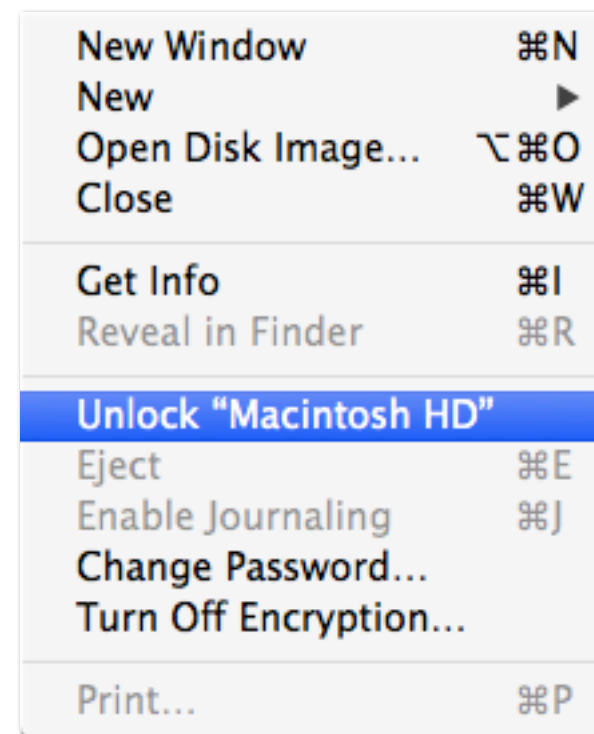
Disaster Recovery

Sometimes bad things happen
to good Macs

- If you have the password to an authorized account available, you can unlock and/or decrypt from Disk Utility or the command line.
- You can also use your recovery key to unlock and/or decrypt from the command line.
- You must unlock the drive before you will be able to decrypt it.

Recovering using your password

- Boot your Mac and hold down ⌘-R (Command – R) to boot from the Mac's Recovery HD partition.
- Use Disk Utility and the password of an authorized user to either unlock or turn off the encryption.

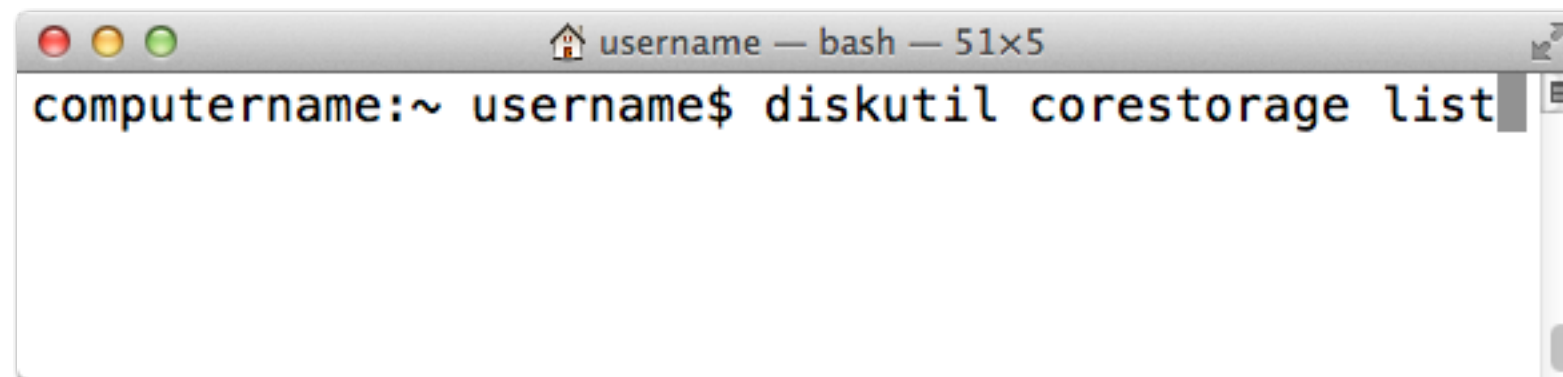


Recovering from Terminal

Boot your Mac and hold down ⌘-R (Command -R) to boot from the Mac's Recovery HD partition.

Open Terminal and use the following to identify the Logical Volume UUID of the encrypted drive.

diskutil corestorage list



Recovering from Terminal

Once you have the UUID, you'll use it to identify the disk to be unlocked or decrypted.

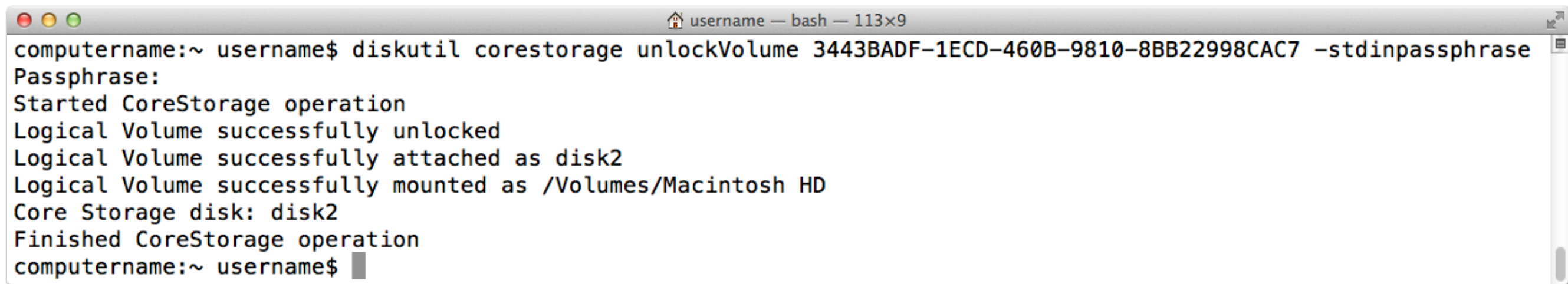
```
username — bash — 71x38
computername:~ username$ diskutil corestorage list
CoreStorage logical volume groups (1 found)
|
+--- Logical Volume Group ABBB1731-262E-4276-B478-90D305A3221F
=====
Name:          Macintosh HD
Status:        Online
Size:          42089914368 B (42.1 GB)
Free Space:    16777216 B (16.8 MB)
|
+--< Physical Volume 794AF504-8DCD-49F9-91A1-0241ABC332C3
=====
Index:         0
Disk:          disk1s2
Status:        Online
Size:          42089914368 B (42.1 GB)
|
+--> Logical Volume Family A58B7C38-B778-47D5-9AD8-B08C09094296
=====
Encryption Status:  Unlocked
Encryption Type:    AES-XTS
Conversion Status:  Complete
Conversion Direction: -none-
Has Encrypted Extents: Yes
Fully Secure:       Yes
Passphrase Required: Yes
|
+--> Logical Volume 3443BADF-1ECD-460B-9810-8BB22998CAC7
=====
Disk:          disk2
Status:        Online
Size (Total):  41754365952 B (41.8 GB)
Conversion Progress: -none-
Revertible:     Yes (unlock and decryption required)
LV Name:        Macintosh HD
Volume Name:    Macintosh HD
Content Hint:    Apple_HFS
computername:~ username$
```

Recovering from Terminal

Unlocking or decrypting using your password

To unlock: *diskutil corestorage unlockVolume UUID -stdinpassphrase*

To decrypt: *diskutil corestorage revert UUID -stdinpassphrase*

A screenshot of a macOS Terminal window. The title bar shows 'username — bash — 113x9'. The terminal text shows the command 'diskutil corestorage unlockVolume 3443BADF-1ECD-460B-9810-8BB22998CAC7 -stdinpassphrase' being executed. It prompts for a passphrase, then displays the following output: 'Started CoreStorage operation', 'Logical Volume successfully unlocked', 'Logical Volume successfully attached as disk2', 'Logical Volume successfully mounted as /Volumes/Macintosh HD', 'Core Storage disk: disk2', and 'Finished CoreStorage operation'. The prompt returns to 'computername:~ username\$'.

```
computername:~ username$ diskutil corestorage unlockVolume 3443BADF-1ECD-460B-9810-8BB22998CAC7 -stdinpassphrase
Passphrase:
Started CoreStorage operation
Logical Volume successfully unlocked
Logical Volume successfully attached as disk2
Logical Volume successfully mounted as /Volumes/Macintosh HD
Core Storage disk: disk2
Finished CoreStorage operation
computername:~ username$
```

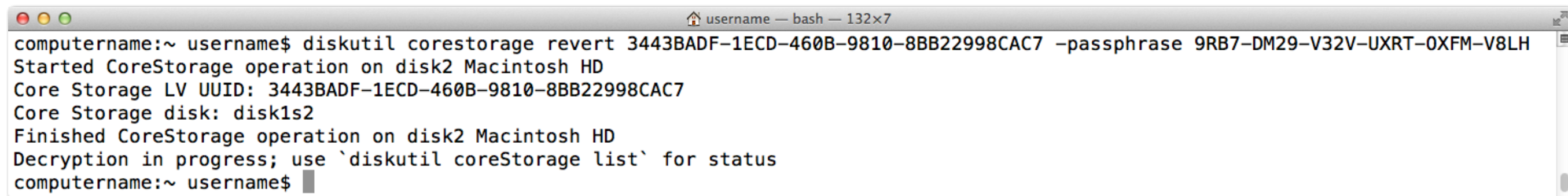
The *-stdinpassphrase* flag will cause the command to prompt you for the password/passphrase of an account that's authorized to unlock the encryption.

Recovering from Terminal

Unlocking or decrypting with an personal recovery key

To unlock: *diskutil corestorage unlockVolume UUID -passphrase recoverykey*

To decrypt: *diskutil corestorage revert UUID -passphrase recoverykey*

A screenshot of a macOS Terminal window. The title bar shows 'username — bash — 132x7'. The command prompt is 'computername:~ username\$'. The command entered is 'diskutil corestorage revert 3443BADF-1ECD-460B-9810-8BB22998CAC7 -passphrase 9RB7-DM29-V32V-UXRT-0XFM-V8LH'. The output shows the operation starting on disk2 Macintosh HD, displaying the Core Storage LV UUID and disk name, and then indicating that decryption is in progress. The prompt returns to 'computername:~ username\$' with a cursor.

```
computername:~ username$ diskutil corestorage revert 3443BADF-1ECD-460B-9810-8BB22998CAC7 -passphrase 9RB7-DM29-V32V-UXRT-0XFM-V8LH
Started CoreStorage operation on disk2 Macintosh HD
Core Storage LV UUID: 3443BADF-1ECD-460B-9810-8BB22998CAC7
Core Storage disk: disk1s2
Finished CoreStorage operation on disk2 Macintosh HD
Decryption in progress; use `diskutil coreStorage list` for status
computername:~ username$
```

This command would be used only if FileVault 2 generated a personal recovery key. This would not apply if you're using FileVaultMaster.keychain as an institutional recovery key.

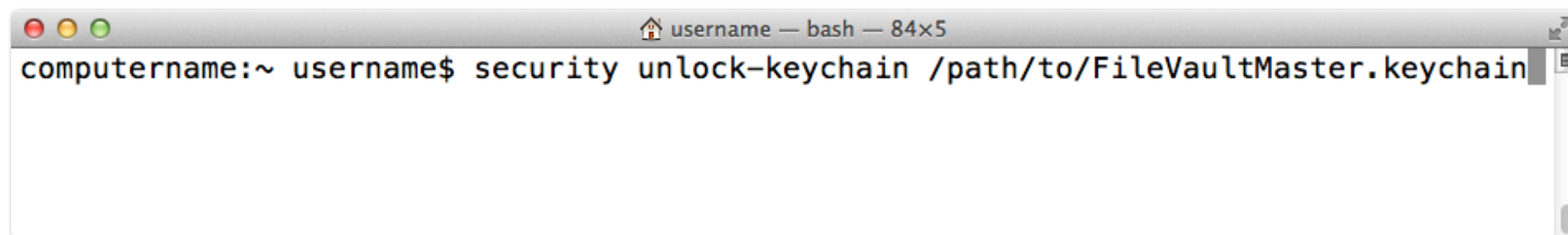
Recovering from Terminal

Unlocking or decrypting with a institutional recovery key

- Boot to Recovery HD or Apple's Internet Recovery
- Copy the FileVaultMaster.keychain with the private key in the keychain from its secured place to a convenient place accessible from Recovery HD.
- To allow the Mac to get access to both the keys inside, unlock the FileVaultMaster.keychain.

To unlock FileVaultMaster.keychain:

security unlock-keychain /path/to/FileVaultMaster.keychain



You'll be prompted a password. Please enter the Master Password at this point.

Recovering from Terminal

Unlocking or decrypting with an institutional recovery key

Once you've unlocked FileVaultMaster.keychain, you can unlock or decrypt the disk.

To unlock:

```
diskutil corestorage unlockVolume UUID -recoveryKeychain /path/to/FileVaultMaster.keychain
```

To decrypt:

```
diskutil corestorage revert UUID -recoveryKeychain /path/to/FileVaultMaster.keychain
```

As long as FileVaultMaster.keychain is unlocked, you should not be prompted for a password.

OS X Utilities



Restore From Time Machine Backup

You have a backup of your system that you want to restore.



Reinstall OS X

Reinstall a new copy of OS X.



Get Help Online

Browse the Apple Support website to find help for your Mac.



Disk Utility

Repair or erase a disk using Disk Utility.

Continue

FileVault 2 Under The Hood

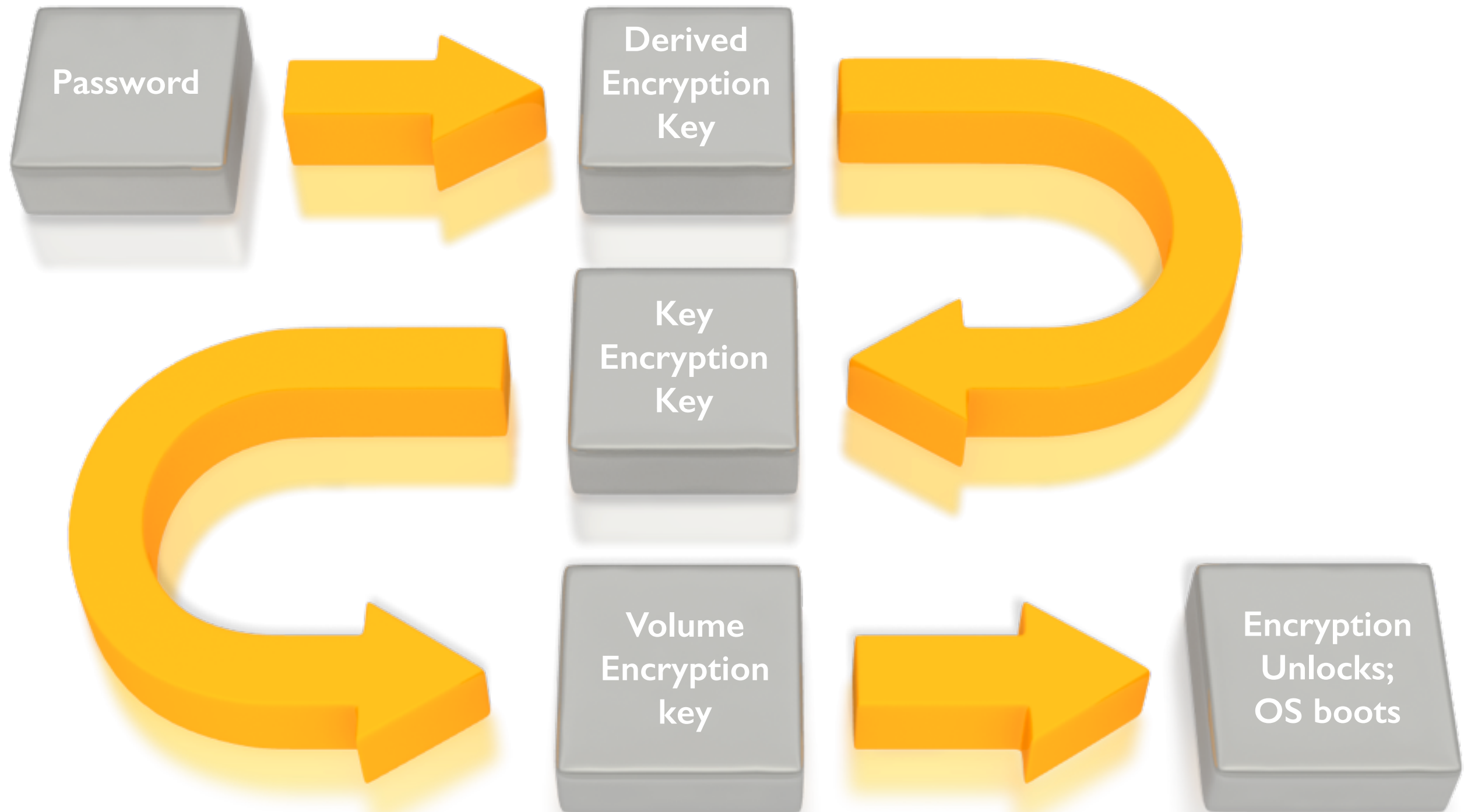


Keys Used By FileVault 2



- › Derived Encryption Key
- › Key Encryption Key
- › Volume Encryption Key

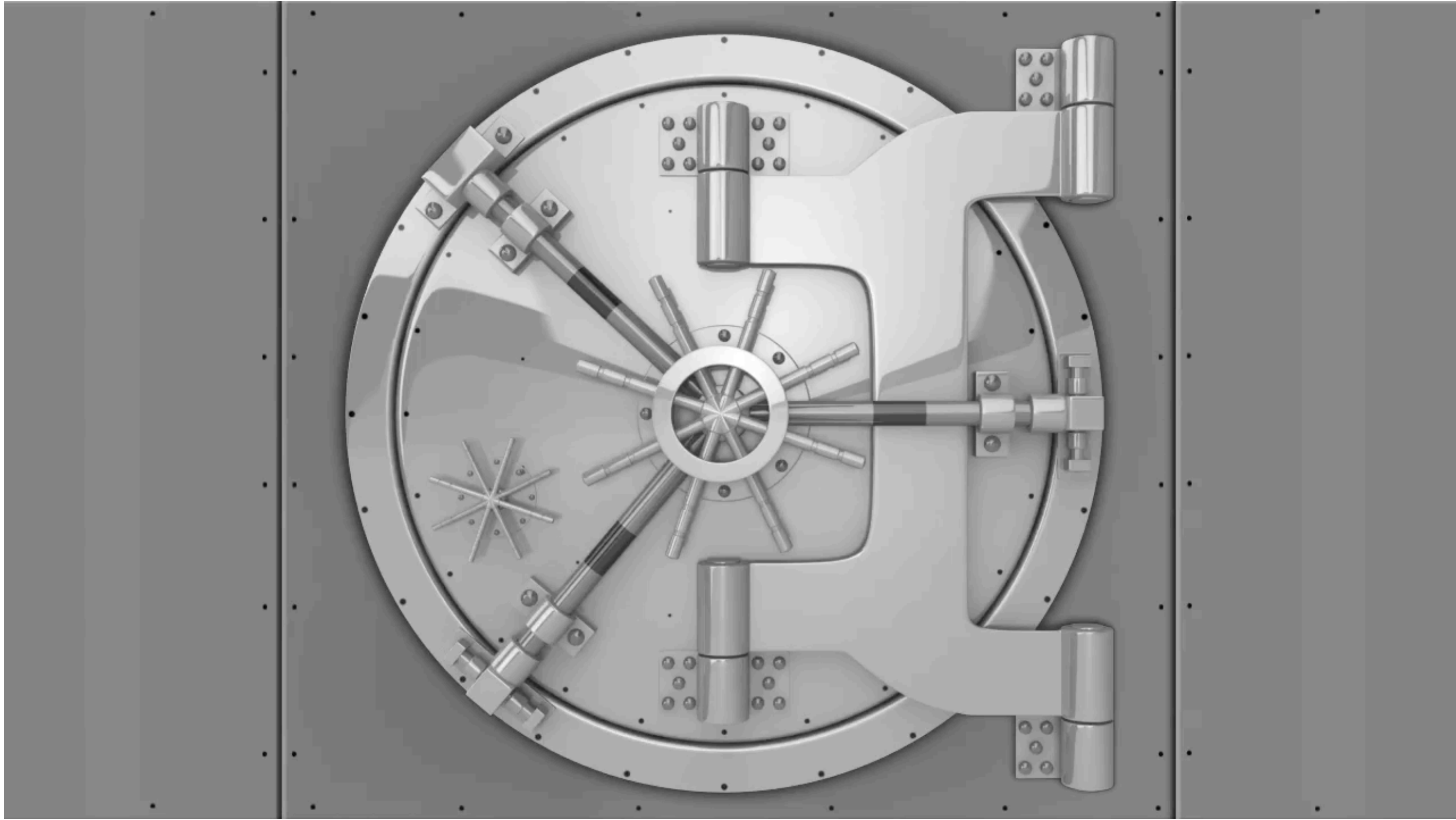
Key-Driven Unlock Process



Generating derived key from the user's password



Key validation and volume unlock



Pre-encryption access



Post-encryption access



Post-encryption access



No key? No access.



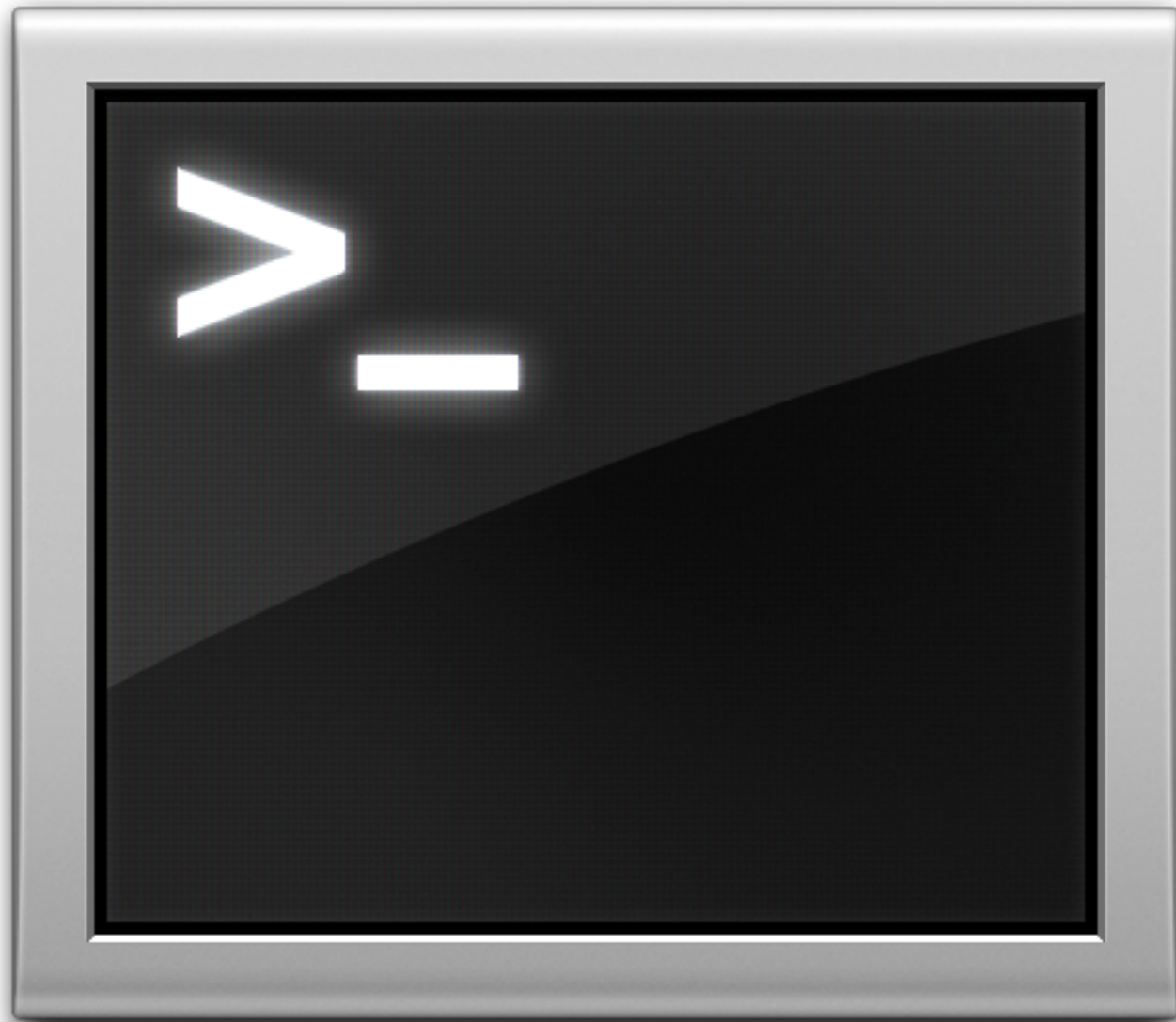
Mavericks' fdsetup



fdsetup overview

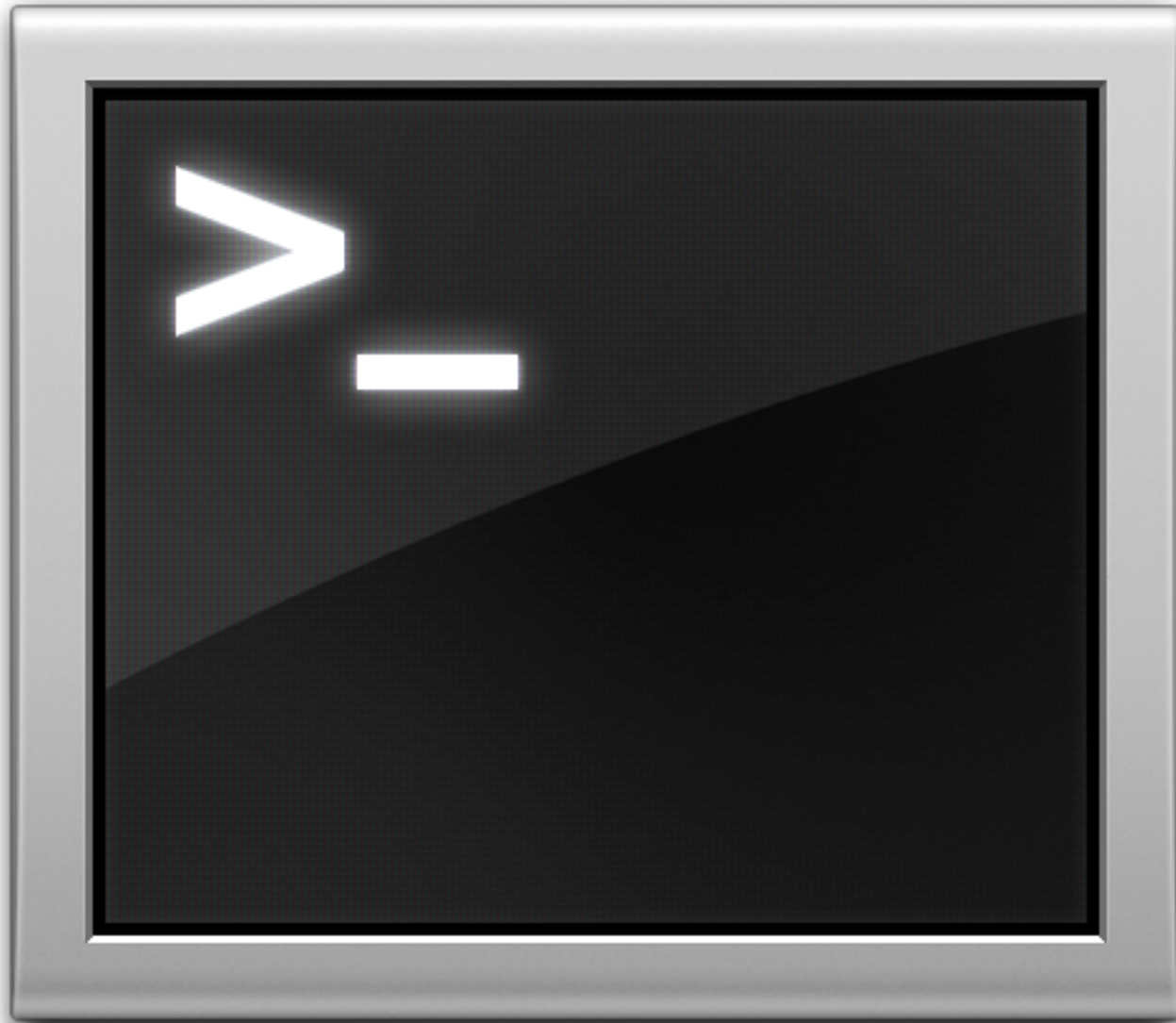


fdsetup commands



- › fdsetup enable
- › fdsetup disable
- › fdsetup add
- › fdsetup list
- › fdsetup remove
- › fdsetup changerecovery
- › fdsetup removererecovery
- › fdsetup sync
- › fdsetup authrestart

fdsetup enable

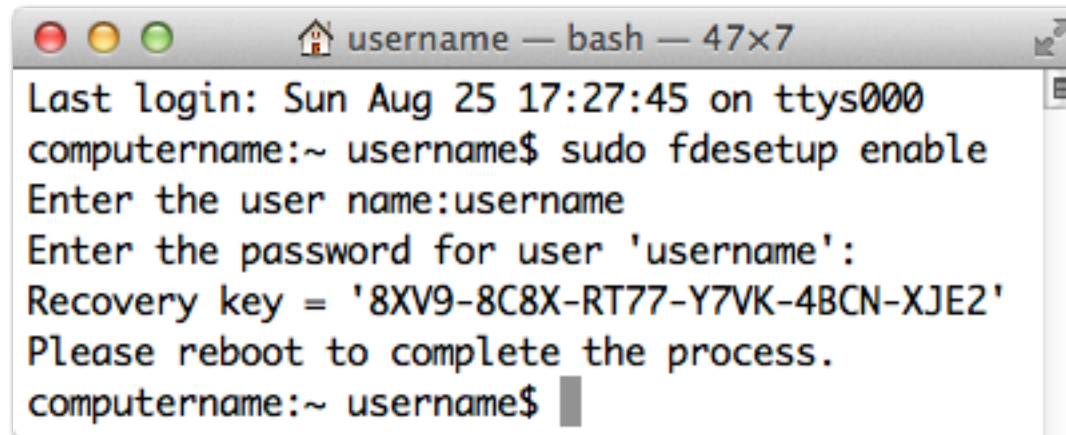


» Activates FileVault 2 Encryption

- Can set FileVault 2 encryption to use:
 - Individual alphanumeric recovery key
 - Institutional recovery key using FileVaultMaster.keychain
 - Both kinds of recovery key simultaneously
- Can enable multiple user accounts at time of encryption activation
- Can import user and certificate information

fdsetup enable

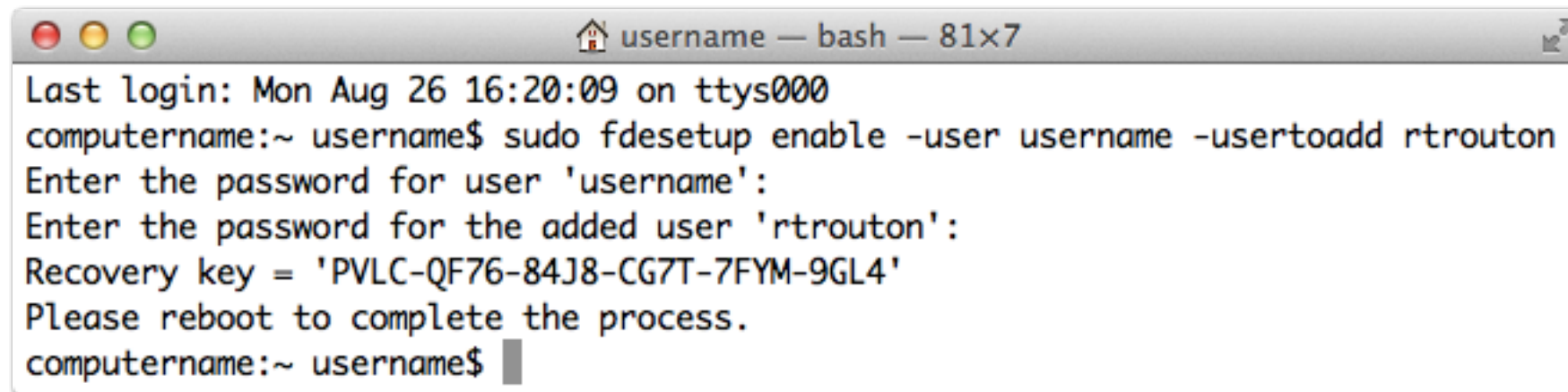
sudo fdsetup enable

A terminal window titled 'username — bash — 47x7' with standard macOS window controls. The terminal output shows the user logging in, running 'sudo fdsetup enable', and receiving instructions to enter a username and password. A recovery key is displayed, and the user is prompted to reboot. The prompt returns to 'computername:~ username\$' with a cursor.

```
computername:~ username$ sudo fdsetup enable
Last login: Sun Aug 25 17:27:45 on ttys000
computername:~ username$ sudo fdsetup enable
Enter the user name:username
Enter the password for user 'username':
Recovery key = '8XV9-8C8X-RT77-Y7VK-4BCN-XJE2'
Please reboot to complete the process.
computername:~ username$
```


fdsetup enable -user

sudo fdsetup enable -user username -usertoadd username



```
username — bash — 81x7
Last login: Mon Aug 26 16:20:09 on ttys000
computername:~ username$ sudo fdsetup enable -user username -usertoadd rtrouton
Enter the password for user 'username':
Enter the password for the added user 'rtrouton':
Recovery key = 'PVLC-QF76-84J8-CG7T-7FYM-9GL4'
Please reboot to complete the process.
computername:~ username$
```

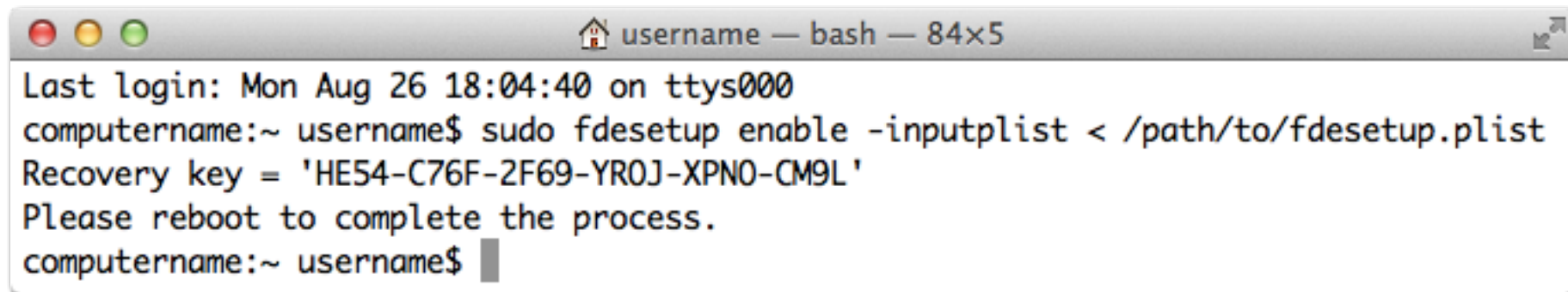
fdesetup enable -inputplist

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
  <dict>
    <key>Username</key>
    <string>localadmin</string>
    <key>Password</key>
    <string>password</string>
    <key>AdditionalUsers</key>
    <array>
      <dict>
        <key>Username</key>
        <string>tom</string>
        <key>Password</key>
        <string>password</string>
      </dict>
      <dict>
        <key>Username</key>
        <string>harry</string>
        <key>Password</key>
        <string>password</string>
      </dict>
    </array>
  </dict>
</plist>
```

Note: All account passwords need to be supplied in cleartext.

fdesetup enable -inputplist

sudo fdesetup enable -inputplist < plistfile.plist

A screenshot of a macOS terminal window. The title bar shows a home icon, the text 'username — bash — 84x5', and window control buttons. The terminal content shows the last login time, the command being executed, the recovery key, and a reboot instruction.

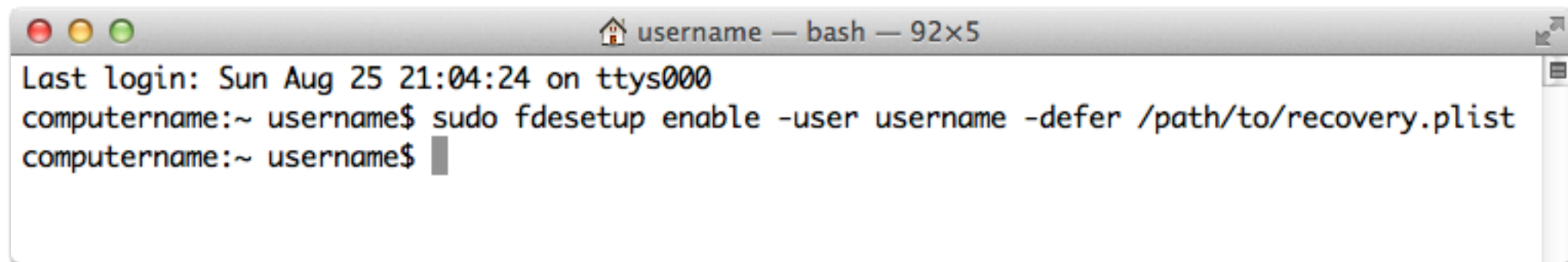
```
username — bash — 84x5
Last login: Mon Aug 26 18:04:40 on ttys000
computername:~ username$ sudo fdesetup enable -inputplist < /path/to/fdesetup.plist
Recovery key = 'HE54-C76F-2F69-YR0J-XPNO-CM9L'
Please reboot to complete the process.
computername:~ username$
```


fdsetup enable -defer

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
  <key>EnabledDate</key>
  <string>2012-07-08 08:05:54 -0400</string>
  <key>HardwareUUID</key>
  <string>00000000-0000-1000-8000-000C293DEFEC</string>
  <key>LVGUID</key>
  <string>B7990442-91D1-4F5E-8F04-DDAC7B3610F2</string>
  <key>LVUUID</key>
  <string>FF837400-B781-496E-8992-D5A11B5A1139</string>
  <key>PVUUID</key>
  <string>E0FD9F00-5D41-4597-A10B-73F0A463CC65</string>
  <key>RecoveryKey</key>
  <string>8MGZ-C7BL-ML2M-J6B4-HN35-ZDOA</string>
  <key>SerialNumber</key>
  <string>VMWVk2fm2om0q0/em3zWsLr2g</string>
</dict>
</plist>
```

fdsetup enable -defer

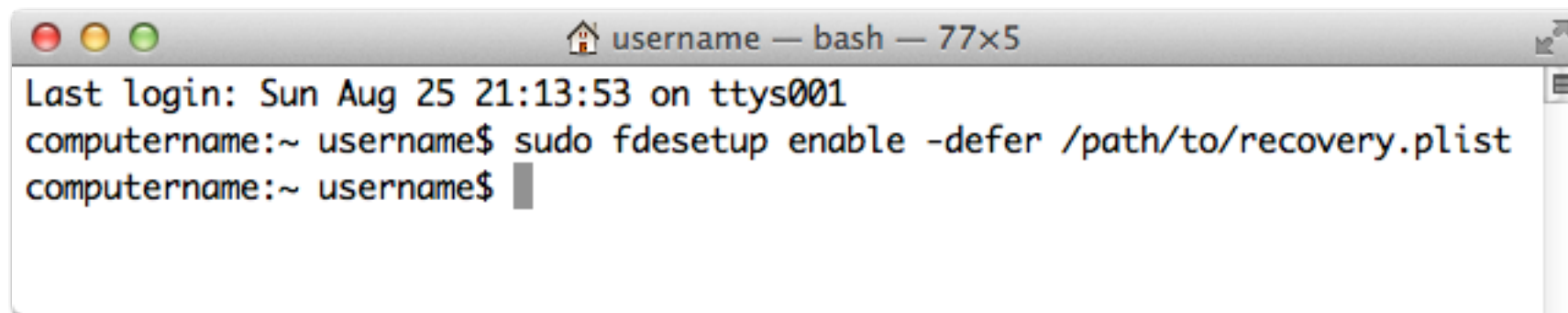
sudo fdsetup enable -user username -defer plistfile.plist

A screenshot of a macOS terminal window. The title bar shows a home icon, the text 'username — bash — 92x5', and window control buttons. The terminal content shows a login message, a command being executed with sudo, and the resulting prompt.

```
username — bash — 92x5
Last login: Sun Aug 25 21:04:24 on ttys000
computername:~ username$ sudo fdsetup enable -user username -defer /path/to/recovery.plist
computername:~ username$
```

fdsetup enable -defer

sudo fdsetup enable -defer plistfile.plist

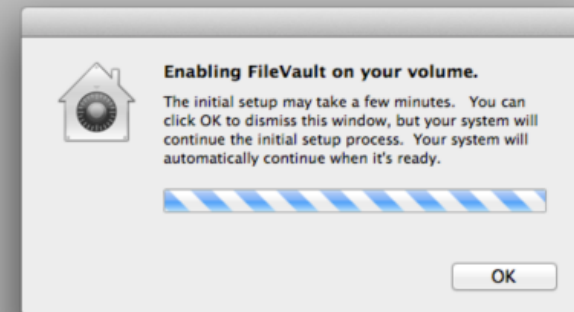
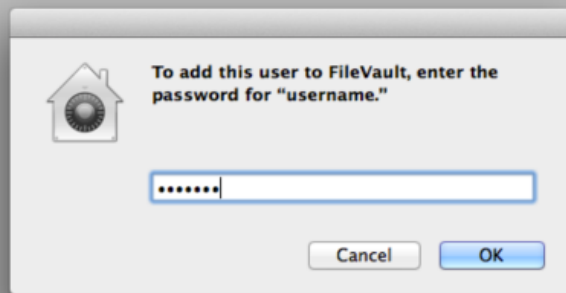
A screenshot of a macOS terminal window. The title bar shows a home icon, the text 'username — bash — 77x5', and window control buttons. The terminal content shows a login message, a command being executed with sudo, and the prompt returning.

```

Last login: Sun Aug 25 21:13:53 on ttys001
computername:~ username$ sudo fdsetup enable -defer /path/to/recovery.plist
computername:~ username$

```

fdesetup enable -defer





Name

Password



Sleep



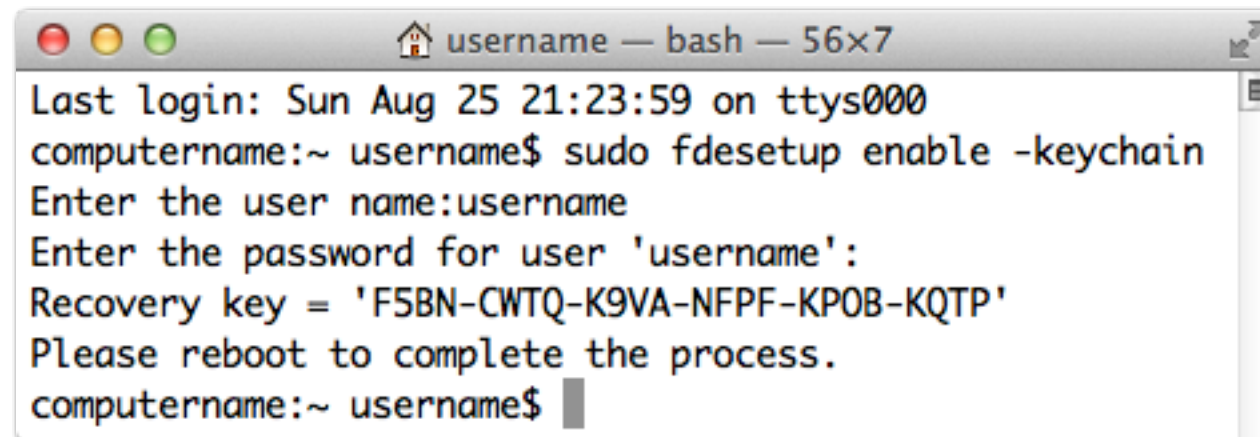
Restart



Shut Down

fdsetup enable -keychain

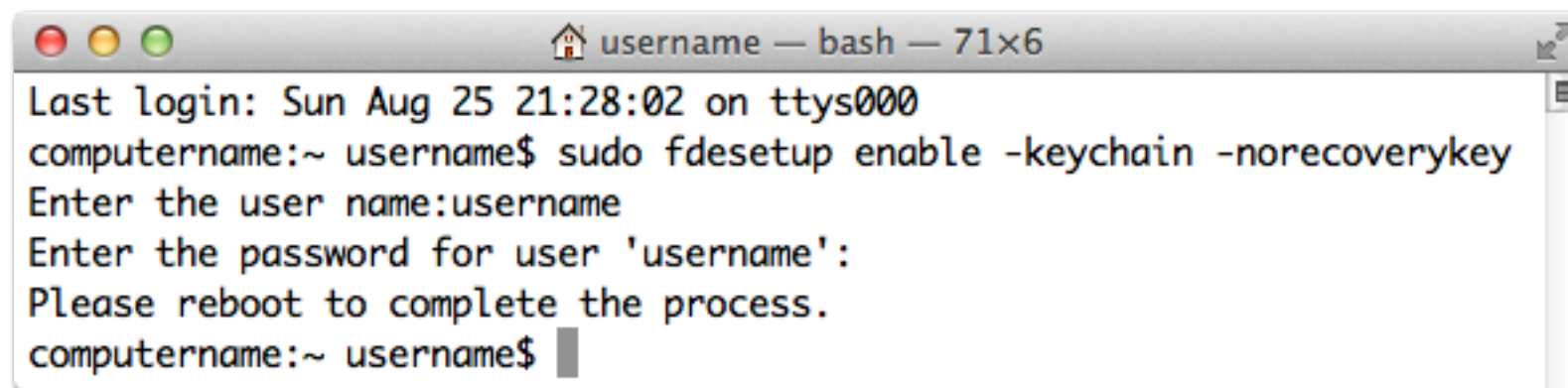
sudo fdsetup enable -keychain

A screenshot of a macOS terminal window. The title bar shows a home icon, the text 'username — bash — 56x7', and standard window control buttons. The terminal content shows the user logging in, typing the command 'sudo fdsetup enable -keychain', and receiving prompts for the username and password. A recovery key is displayed, and a message instructs the user to reboot. The prompt returns to 'computername:~ username\$' with a cursor.

```
username — bash — 56x7
Last login: Sun Aug 25 21:23:59 on ttys000
computername:~ username$ sudo fdsetup enable -keychain
Enter the user name:username
Enter the password for user 'username':
Recovery key = 'F5BN-CWTQ-K9VA-NFPF-KPOB-KQTP'
Please reboot to complete the process.
computername:~ username$
```

fdsetup enable -keychain

sudo fdsetup enable -keychain -norecoverykey

A screenshot of a macOS terminal window. The title bar shows a home icon, the text 'username — bash — 71x6', and window control buttons. The terminal content shows a login message, the command 'sudo fdsetup enable -keychain -norecoverykey' being entered, the user name 'username' being provided, a password prompt, a reboot instruction, and the command prompt returning to 'computername:~ username\$'.

```
username — bash — 71x6
Last login: Sun Aug 25 21:28:02 on ttys000
computername:~ username$ sudo fdsetup enable -keychain -norecoverykey
Enter the user name:username
Enter the password for user 'username':
Please reboot to complete the process.
computername:~ username$
```

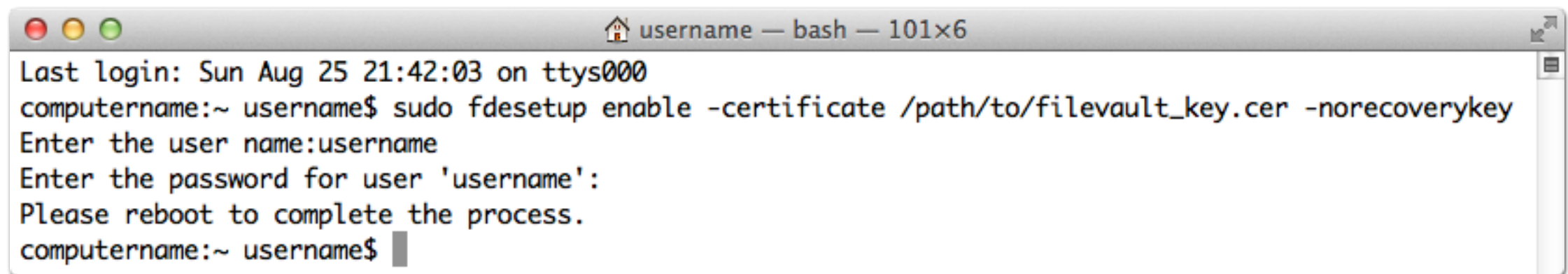

fdsetup enable -certificate

sudo fdsetup enable -certificate cert.cer



fdsetup enable -certificate

sudo fdsetup enable -certificate cert.cer -norecoverykey

A screenshot of a macOS terminal window. The title bar shows a home icon, the text 'username — bash — 101x6', and window control buttons. The terminal content shows a login message, the execution of 'sudo fdsetup enable -certificate /path/to/filevault_key.cer -norecoverykey', prompts for username and password, a reboot instruction, and the command prompt returning to the user.

```
username — bash — 101x6
Last login: Sun Aug 25 21:42:03 on ttys000
computername:~ username$ sudo fdsetup enable -certificate /path/to/filevault_key.cer -norecoverykey
Enter the user name:username
Enter the password for user 'username':
Please reboot to complete the process.
computername:~ username$
```




Mac HD



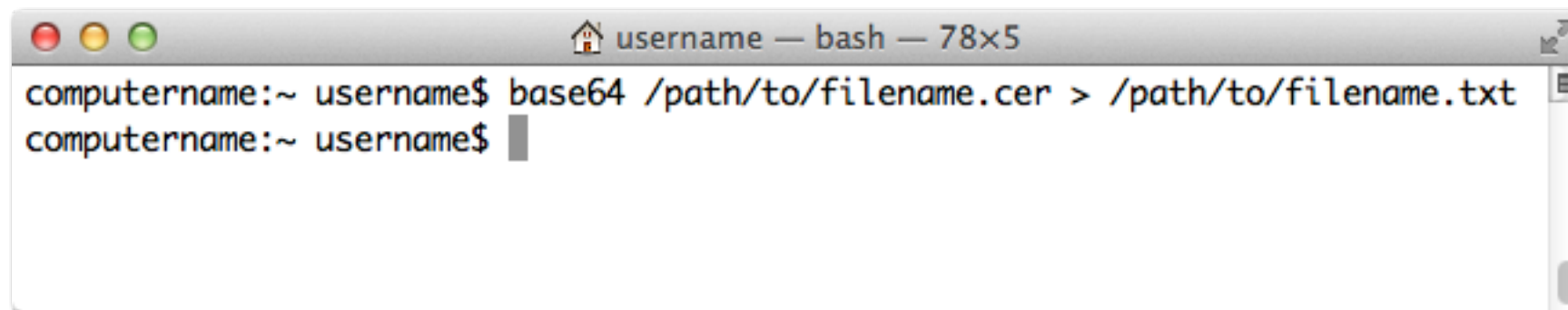
fdsetup enable -inputplist with public key data

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
  <dict>
    <key>Username</key>
    <string>username</string>
    <key>Password</key>
    <string>pass123</string>
    <key>AdditionalUsers</key>
    <array>
      <dict>
        <key>Username</key>
        <string>tom</string>
        <key>Password</key>
        <string>password</string>
      </dict>
      <dict>
        <key>Username</key>
        <string>harry</string>
        <key>Password</key>
        <string>password</string>
      </dict>
    </array>
    <key>Certificate</key>
    <data>
      MIIDLjCCAahagAwIBAgIENsqyWTALBgkqhkiG9w0BAQUwSTEFMB0GA1UEAwWRmlsZVZhdWx0IFJlY292ZXJ5IEtleTEmMCQGA1UE
      DQwddHJvdXRvbnQtdm0zLnRhb21lY2h3b3Jrcy5uZXQwHhcNMjMwODI0MjEzNTI1WhcNMjMwODI0MjEzNTI1WjBJMR8wHQYDVQDD
      DBZGawxlVmF1bHJlY292ZXJ5IEtleTEmMCQGA1UEAwWRmlsZVZhdWx0IFJlY292ZXJ5IEtleTEmMCQGA1UEAwWRmlsZVZhdWx0IFJlY292
      hvCNAQEBAQADggEPADCCAQoCggEBAMJp04TYyH8HC3H4WwV0FPV48I1cdRjhBGi90YJX8cQrgWb72FTDCNbU+fd1ce1++
      KALheMNCmh0IwPshQtUuU9ngpfRYI/L6KIgIaUW4bpUksHXZ0CIwSvfqD6+
      np06wniS48KNPmnyHqxNeYyfU20wAITPznP7w0VjU7pMbvvJra03gNDxxLc44Rj5YdHxn7U7gIUNc08RxtNgC6hPSMuZVIgz8AU
      Fc0YuBYML9fXrHytNoaZ0JXS2djGRdwRT0V30tK2qDfH8+vVr9vTBt3xmK6hPREXfB9odRDH6ZP9mn6R2vLLzF4ANwklha0L4+
      mwbPtdfV0Ss3wdkoUFF8CAwEAAAMgMB4wCwYDVR0PBAQDAgK0MA8GA1UdEwEB/wQFMAMBAQAwDQYJKoZIhvcNAQEFBQADggEBAJ+
      5ZaqzQmnVDrIQBkxiBXql/
      aRbQpetu5jLsJtHDxRd1d2YiEFJHwy4jnUefrvTUf7648oCQ5Ft0jYL5lYn7QQ3MjvsFKzBh2wVlqJJAhjjjwbzMoVdRxfY8Aftc
      ler9Tw/WvxiECvdgxB37iA+EGVTrUGuCYtY+GhawIKQMR5Ac2XBpfgPatbV9xAtCo4+M9I4jqhSx0GMwLjRHs4yELyH71w/
      Au3VlTYezf+dqof0aZ8c4wB0K3T4VeowMJJjwDe9Q3/j3sf/CjxvNhm1yYRUqUieZx7Vdf2+
      oKKqMczyAVz0Y8UX26lMnpMQn4Fv9b5m0vo5U2/ZgpiQ8cHxM=
    </data>
  </dict>
</plist>
```

Note: All account passwords need to be supplied in cleartext.

fdsetup enable -inputplist with public key data

base64 /path/to/filename.cer > /path/to/filename.txt

A screenshot of a macOS-style terminal window. The title bar at the top shows three colored window control buttons (red, yellow, green) on the left, a home icon followed by the text 'username — bash — 78x5' in the center, and a close button on the right. The terminal content shows two lines of text: 'computername:~ username\$ base64 /path/to/filename.cer > /path/to/filename.txt' and 'computername:~ username\$' with a cursor. A vertical scrollbar is visible on the right side of the terminal window.

```
computername:~ username$ base64 /path/to/filename.cer > /path/to/filename.txt
computername:~ username$
```

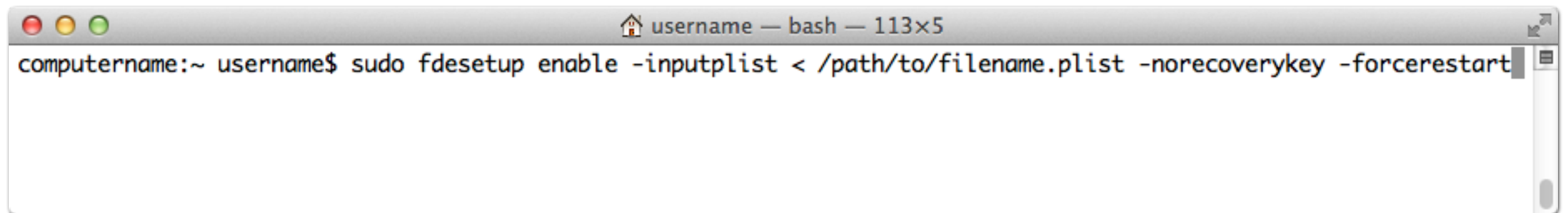
fdsetup enable -inputplist with public key data

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
  <dict>
    <key>Username</key>
    <string>username</string>
    <key>Password</key>
    <string>pass123</string>
    <key>AdditionalUsers</key>
    <array>
      <dict>
        <key>Username</key>
        <string>tom</string>
        <key>Password</key>
        <string>password</string>
      </dict>
      <dict>
        <key>Username</key>
        <string>harry</string>
        <key>Password</key>
        <string>password</string>
      </dict>
    </array>
    <key>Certificate</key>
    <data>
MIIDLjCCAahagAwIBAgIENsqyWTALBgqhkiG9w0BAQUwSTEFMB0GA1UEAwWRmlsZVZhdWx0IFJlY292ZXJ5IEtleTEemMCQGA1UE
DQwddHJvdXRvbnQtdm0zLnRhb21lY2h3b3Jrcy5uZXQwHhcNMjMwODI0MjEzNTI1WjBJMR8wHQYDVQDD
DBZGawxlVmF1bHJvdWVjb3ZlcnkgS2V5MSYwJAYDVQQNDB10cm91dG9udC12bTMudGFvbWVjaHdvcm5ldDCCASIwDQYJKoZI
hvcNAQEBBQADggEPADCCAQoCggEBAMJp04TYyH8HC3H4WwV0FPV48I1cdRjhBGi90YJX8cQrgWb72FTDCNbU+fd1ce1++
KALheMNCmh0IwPshQtUuU9ngpfRYI/L6KIgIaUW4bpUksHXZ0CIwSvfqD6+
np06wniS48KNPmnyHqxNeYyfU20wAITPznP7w0VjU7pMbvvJra03gNDxxLc44Rj5YdHxn7U7gIUNc08RxtNgC6hPSMuZVIgz8AU
Fc0YuBYML9fXrHytNoaZ0JXS2djGRdwRT0V30tK2qDFh8+vVr9vTBt3xmk6hPREXfB9odRDH6ZP9mn6R2vLLzF4ANwklha0L4+
mwbPtdfV0Ss3wdkoUFF8CAwEAAAMgMB4wCwYDVR0PBAQDAgK0MA8GA1UdEwEB/wQFMAMBAQAwDQYJKoZIhvcNAQEFBQADggEBAJ+
5ZaqzQmnVDrIQBkxiBXql/
aRbQpetu5jLsJtHDxRd1d2YiEFJHwy4jnUefrvTUf7648oCQ5Ft0jYL5lYn7QQ3MjvsFKzBh2wVlqJJAhjjjwbzMoVdRxfY8Aftc
ler9Tw/WvxiECvdgxB37iA+EGVTrUGuCYtY+GhawIKQMR5Ac2XBpfgPatbV9xAtCo4+M9I4jqhSx0GMwLjRHs4yELyH71w/
Au3VlTYezf+dqof0aZ8c4wB0K3T4VeowMJJjwDe9Q3/j3sf/CjxvNhmlyYRUqUieZx7Vdf2+
oKKqMczyAVz0Y8UX26lMnpMQn4Fv9b5m0vo5U2/ZgpiQ8cHxM=
    </data>
  </dict>
</plist>
```

Note: All account passwords need to be supplied in cleartext.

fdsetup enable -forcerestart

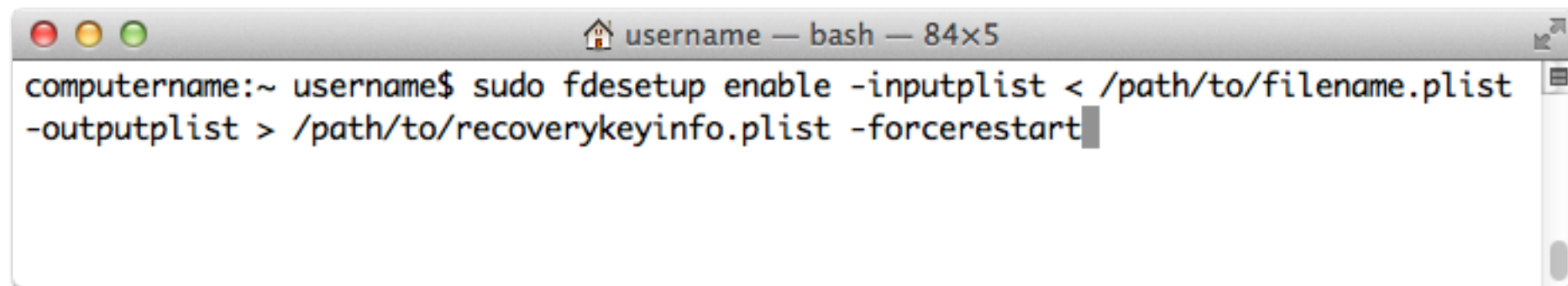
sudo fdsetup enable -inputplist < /path/to/filename.plist -norecoverykey -forcerestart

A screenshot of a macOS terminal window. The title bar shows a home icon, the text 'username — bash — 113x5', and standard window control buttons. The terminal content shows the prompt 'computername:~ username\$' followed by the command 'sudo fdsetup enable -inputplist < /path/to/filename.plist -norecoverykey -forcerestart'. A cursor is visible at the end of the command line.

```
computername:~ username$ sudo fdsetup enable -inputplist < /path/to/filename.plist -norecoverykey -forcerestart
```

fdsetup enable -forcerestart

*sudo fdsetup enable -inputplist < /path/to/filename.plist -outputplist > /
path/to/recoverykeyinfo.plist -forcerestart*

A screenshot of a macOS terminal window. The title bar shows a home icon, the text 'username — bash — 84x5', and standard window control buttons. The terminal content shows the prompt 'computername:~ username\$' followed by the command 'sudo fdsetup enable -inputplist < /path/to/filename.plist -outputplist > /path/to/recoverykeyinfo.plist -forcerestart'. The cursor is at the end of the command line.

```
computername:~ username$ sudo fdsetup enable -inputplist < /path/to/filename.plist  
-outputplist > /path/to/recoverykeyinfo.plist -forcerestart
```



Name

Password



Sleep



Restart



Shut Down

fdsetup disable



» Disables FileVault 2 encryption

fdsetup add

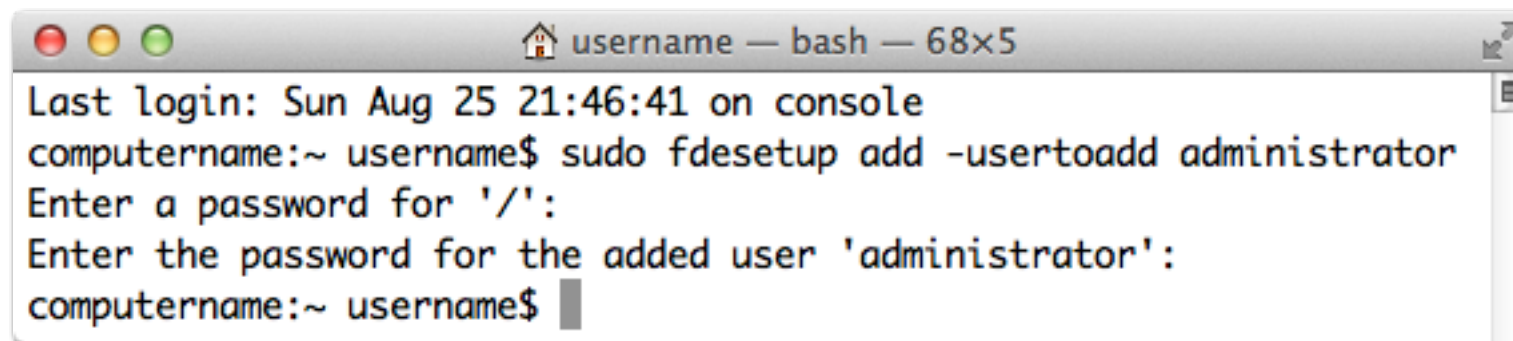


➤ Enables additional accounts after FileVault 2 encryption is complete

- Can enable multiple user accounts
- Can import user information

fdsetup add -usertoadd

sudo fdsetup add -usertoadd username



```
username — bash — 68x5
Last login: Sun Aug 25 21:46:41 on console
computername:~ username$ sudo fdsetup add -usertoadd administrator
Enter a password for '/':
Enter the password for the added user 'administrator':
computername:~ username$
```

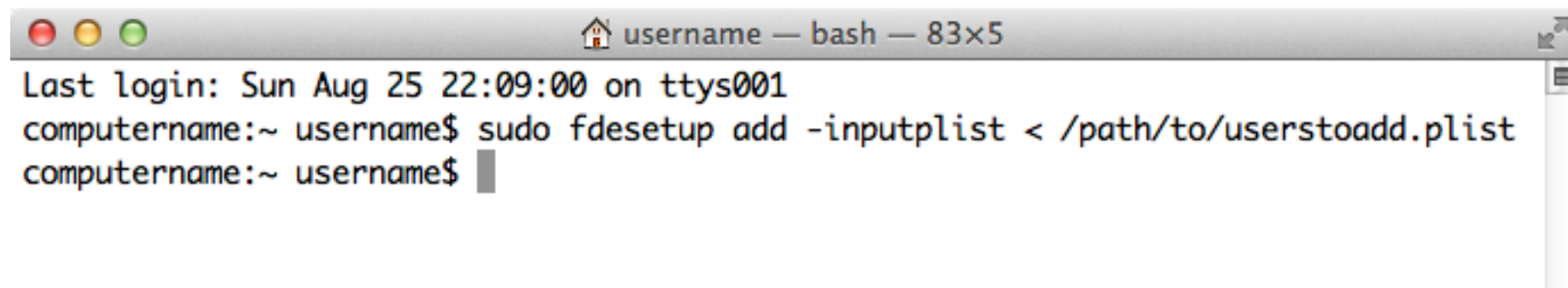
fdsetup add -inputplist

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
<key>Username</key>
<string>rtrouton</string>
<key>Password</key>
<string>password</string>
<key>AdditionalUsers</key>
<array>
  <dict>
    <key>Username</key>
    <string>fcheeryble</string>
    <key>Password</key>
    <string>password</string>
  </dict>
  <dict>
    <key>Username</key>
    <string>nnickleby</string>
    <key>Password</key>
    <string>password</string>
  </dict>
</array>
</dict>
</plist>
```

Note: All account passwords need to be supplied in cleartext.

fdsetup add -inputplist

sudo fdsetup add -inputplist /path/to/plistname.plist

A screenshot of a macOS terminal window. The title bar shows a home icon, the text 'username — bash — 83x5', and window control buttons. The terminal content shows a login message, a command being executed with sudo, and the prompt returning.

```

Last login: Sun Aug 25 22:09:00 on ttys001
computername:~ username$ sudo fdsetup add -inputplist < /path/to/userstoadd.plist
computername:~ username$

```

fdsetup list

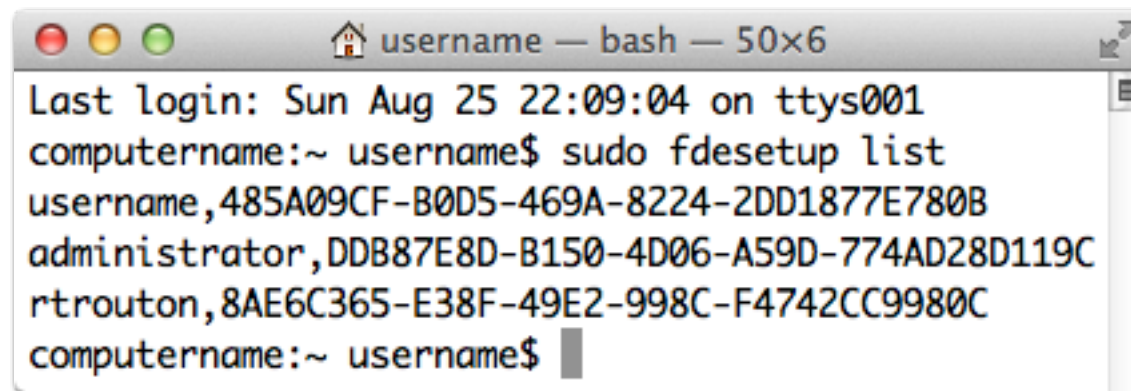


» Displays enabled accounts

- List includes the accounts' usernames and UUIDs

fdsetup list

sudo fdsetup list

A terminal window with a title bar that reads 'username — bash — 50x6'. The window contains the following text: 'Last login: Sun Aug 25 22:09:04 on ttys001', 'computername:~ username\$ sudo fdsetup list', 'username,485A09CF-B0D5-469A-8224-2DD1877E780B', 'administrator,DDB87E8D-B150-4D06-A59D-774AD28D119C', 'rtrouton,8AE6C365-E38F-49E2-998C-F4742CC9980C', and 'computername:~ username\$' followed by a cursor.

```
username — bash — 50x6
Last login: Sun Aug 25 22:09:04 on ttys001
computername:~ username$ sudo fdsetup list
username,485A09CF-B0D5-469A-8224-2DD1877E780B
administrator,DDB87E8D-B150-4D06-A59D-774AD28D119C
rtrouton,8AE6C365-E38F-49E2-998C-F4742CC9980C
computername:~ username$
```


fdsetup remove



➤ Removes accounts from the list of FileVault 2 enabled accounts

- Can disable using account username
- Can disable using account UUID

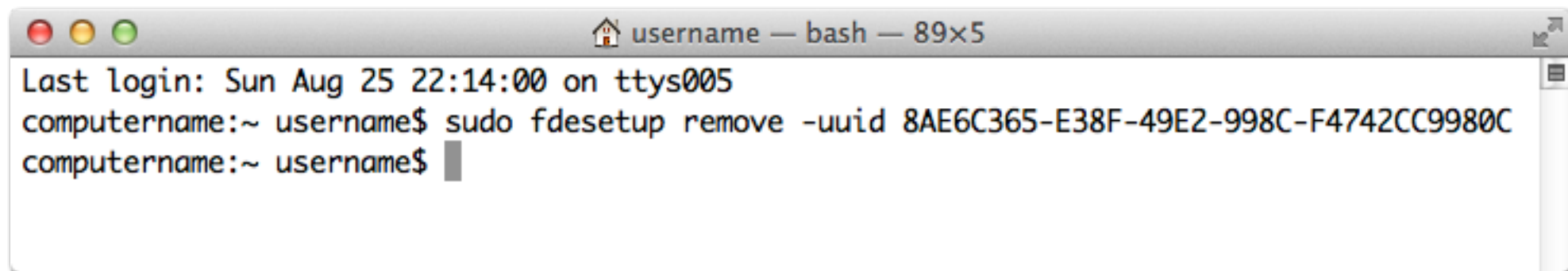
fdsetup remove -user

sudo fdsetup remove -user username

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
  <key>EnabledDate</key>
  <string>2012-07-08 08:05:54 -0400</string>
  <key>HardwareUUID</key>
  <string>00000000-0000-1000-8000-000C293DEFEC</string>
  <key>LVGUID</key>
  <string>B7990442-91D1-4F5E-8F04-DDAC7B3610F2</string>
  <key>LVUUID</key>
  <string>FF837400-B781-496E-8992-D5A11B5A1139</string>
  <key>PVUUID</key>
  <string>E0FD9F00-5D41-4597-A10B-73F0A463CC65</string>
  <key>RecoveryKey</key>
  <string>8MGZ-C7BL-ML2M-J6B4-HN35-ZDOA</string>
  <key>SerialNumber</key>
  <string>VMWVvk2fm2om0q0/em3zWsLr2g</string>
</dict>
</plist>
```

fdsetup remove -uuid

sudo fdsetup remove -uuid uuid_here

A screenshot of a macOS terminal window. The title bar shows a home icon, the text 'username — bash — 89x5', and window control buttons. The terminal content shows a login message, a command being executed with sudo, and the prompt returning to the user.

```
username — bash — 89x5
Last login: Sun Aug 25 22:14:00 on ttys005
computername:~ username$ sudo fdsetup remove -uuid 8AE6C365-E38F-49E2-998C-F4742CC9980C
computername:~ username$
```

fdsetup changerecovery

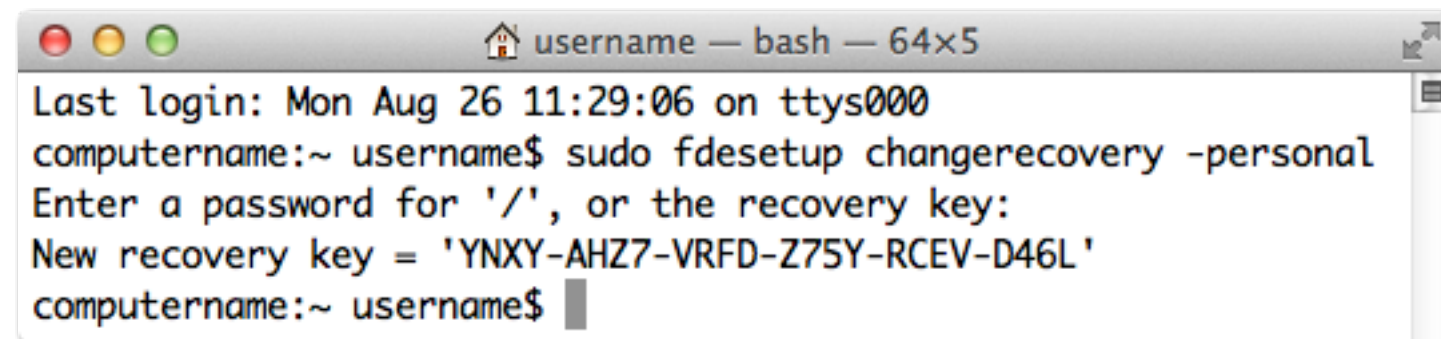


➤ Manage personal and institutional recovery keys

- Rotate existing recovery keys
- Add new personal or institutional recovery keys

fdsetup changerecovery -personal

sudo fdsetup changerecovery -personal

A terminal window with a title bar that reads 'username — bash — 64x5'. The window contains the following text: 'Last login: Mon Aug 26 11:29:06 on ttys000', 'computername:~ username\$ sudo fdsetup changerecovery -personal', 'Enter a password for '/', or the recovery key:', 'New recovery key = 'YNXY-AHZ7-VRFD-Z75Y-RCEV-D46L'', and 'computername:~ username\$' followed by a cursor.

```
username — bash — 64x5
Last login: Mon Aug 26 11:29:06 on ttys000
computername:~ username$ sudo fdsetup changerecovery -personal
Enter a password for '/', or the recovery key:
New recovery key = 'YNXY-AHZ7-VRFD-Z75Y-RCEV-D46L'
computername:~ username$
```

fdsetup changerecovery -personal -inputplist

```
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
<key>Password</key>
<string>password</string>
</dict>
</plist>
```

Note: All account passwords need to be supplied in cleartext.

fdsetup changerecovery -personal -inputplist



Note: All keychain passwords need to be supplied in cleartext.

fdsetup changerecovery -personal -inputplist

fdsetup(8) Mac OS X Manual Page

Apple Inc. developer.apple.com/library/mac/documentation Reader

Mac Developer Library

Apple Developer



OS X Man Pages

KeychainPath

The path to the private key keychain file if you are authenticating to certain comamnds.

KeychainPassword

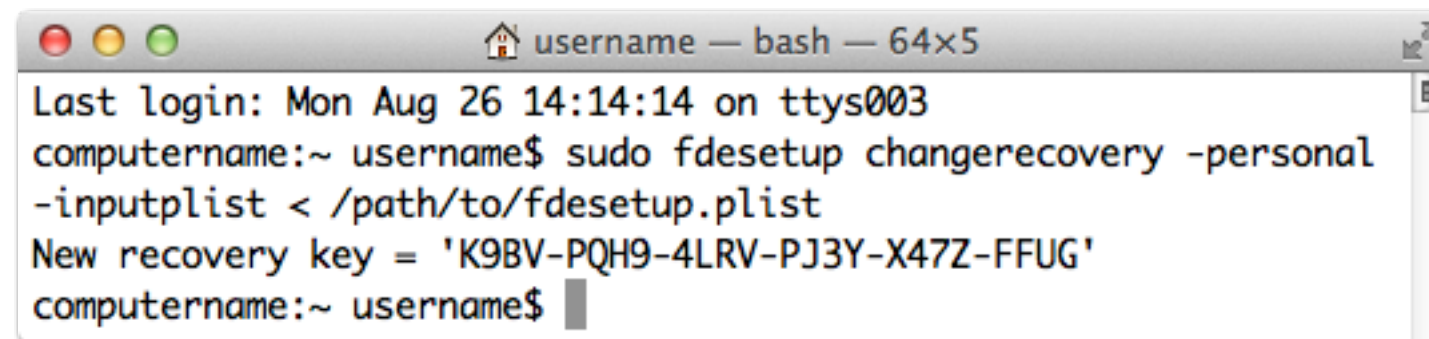
The password to the private key keychain.

Feedback

EXAMPLES

fdsetup changerecovery -personal -inputplist

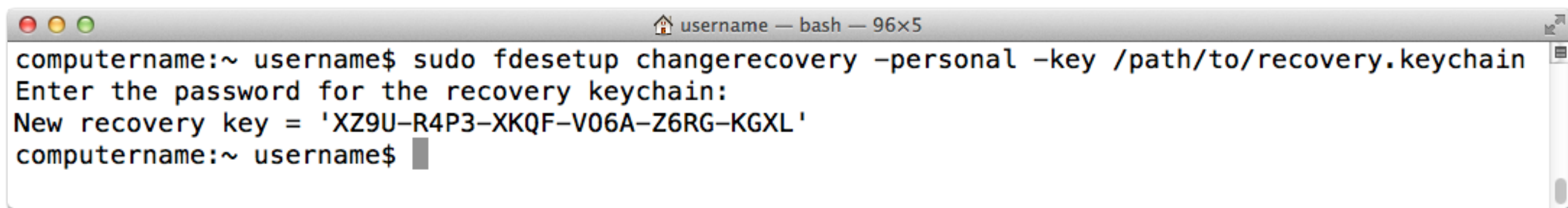
*sudo fdsetup changerecovery -personal
-inputplist < /path/to/filename.plist*

A screenshot of a macOS terminal window. The title bar shows a home icon, the text 'username — bash — 64x5', and standard window control buttons. The terminal content shows the last login time, the command being executed, and the resulting recovery key.

```
username — bash — 64x5
Last login: Mon Aug 26 14:14:14 on ttys003
computername:~ username$ sudo fdsetup changerecovery -personal
-inputplist < /path/to/fdsetup.plist
New recovery key = 'K9BV-PQH9-4LRV-PJ3Y-X47Z-FFUG'
computername:~ username$
```

fdsetup changerecovery -personal -key

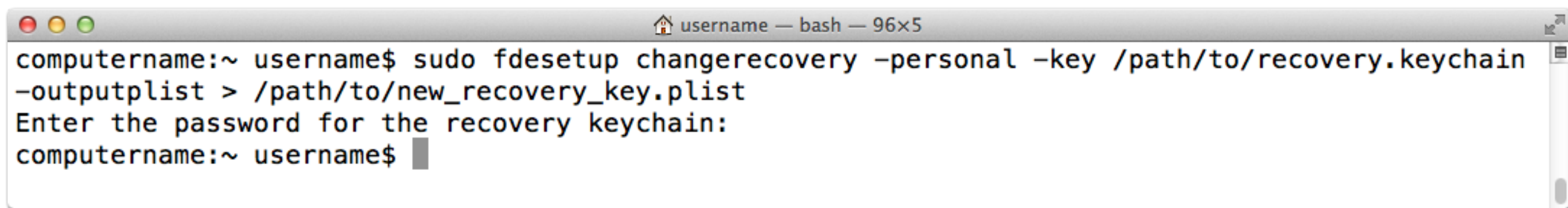
sudo fdsetup changerecovery -personal -key /path/to/recovery.keychain

A screenshot of a macOS terminal window. The title bar shows 'username — bash — 96x5'. The terminal text shows the command 'sudo fdsetup changerecovery -personal -key /path/to/recovery.keychain' being executed. It prompts for a password, then displays a new recovery key: 'XZ9U-R4P3-XKQF-V06A-Z6RG-KGXL'. The prompt returns to the shell.

```
computername:~ username$ sudo fdsetup changerecovery -personal -key /path/to/recovery.keychain
Enter the password for the recovery keychain:
New recovery key = 'XZ9U-R4P3-XKQF-V06A-Z6RG-KGXL'
computername:~ username$
```

fdsetup changerecovery -personal -key -outputplist

*sudo fdsetup changerecovery -personal -key /path/to/recovery.keychain -
outputplist > /path/to/new_recovery_key.plist*

A screenshot of a macOS terminal window. The title bar shows 'username — bash — 96x5'. The terminal text shows the command 'sudo fdsetup changerecovery -personal -key /path/to/recovery.keychain -outputplist > /path/to/new_recovery_key.plist' being entered. Below the command, it says 'Enter the password for the recovery keychain:' followed by a cursor. The prompt 'computername:~ username\$' is visible at the start of the line.

```
computername:~ username$ sudo fdsetup changerecovery -personal -key /path/to/recovery.keychain  
-outputplist > /path/to/new_recovery_key.plist  
Enter the password for the recovery keychain:  
computername:~ username$
```

fdesetup changerecovery -personal -key -outputplist

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
  <key>Change</key>
  <true/>
  <key>EnabledDate</key>
  <string>2013-12-20 13:51:58 -0500</string>
  <key>HardwareUUID</key>
  <string>00000000-0000-1000-8000-000C2991B2C4</string>
  <key>HasMasterKeychain</key>
  <true/>
  <key>RecoveryKey</key>
  <string>MLZA-NZTC-MVLM-082Q-Y8TW-F8FX</string>
  <key>SerialNumber</key>
  <string>VM401BlpPKGn</string>
</dict>
</plist>
```

fdsetup changerecovery -institutional

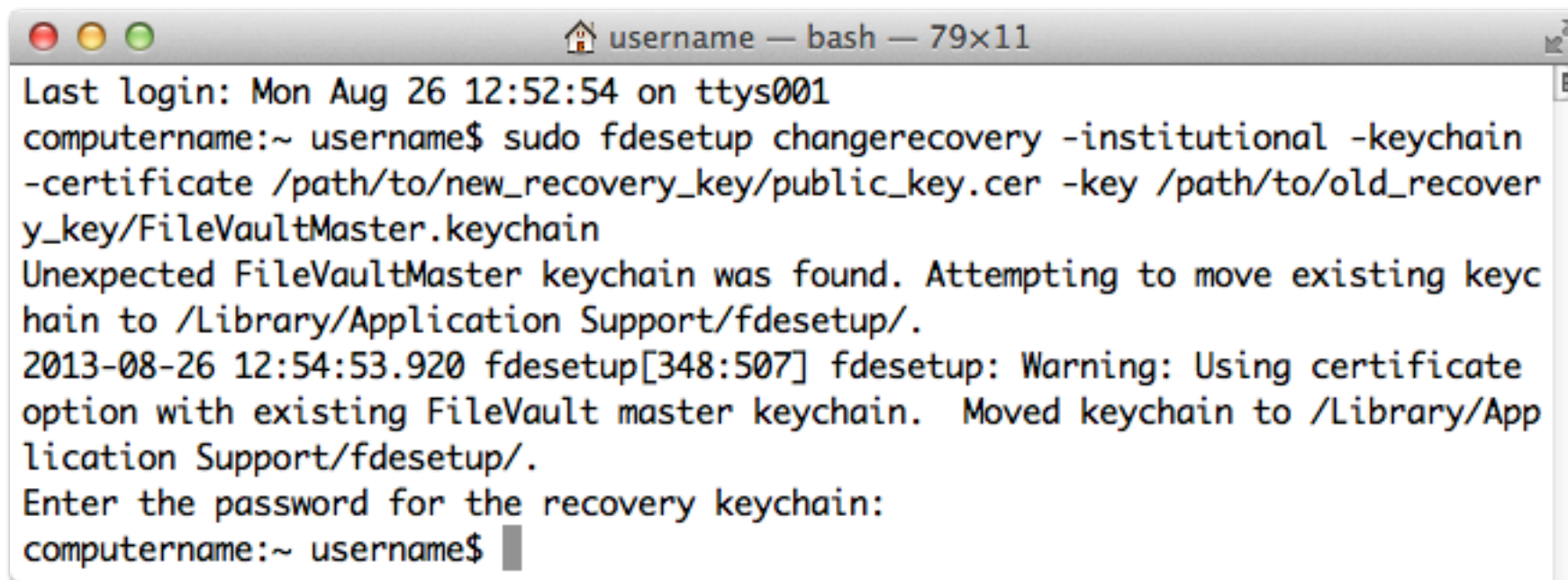
*sudo fdsetup changerecovery -institutional
-certificate /path/to/filename.cer*

A screenshot of a macOS terminal window. The title bar shows a home icon, the text 'username — bash — 69x10', and window control buttons. The terminal text shows a successful execution of the 'fdsetup' command with a warning about an existing keychain.

```
username ~$ sudo fdsetup changerecovery -institutional  
-certificate /path/to/new_recovery_key/new_public_key.cer  
Unexpected FileVaultMaster keychain was found. Attempting to move existing  
keychain to /Library/Application Support/fdsetup/.  
2013-08-27 15:33:49.628 fdsetup[1941:507] fdsetup: Warning: Using certificate  
option with existing FileVault master keychain. Moved keychain to  
/Library/Application Support/fdsetup/.  
Enter a password for '/':  
username ~$
```

fdsetup changerecovery -institutional -keychain

***sudo fdsetup changerecovery -institutional -keychain
-certificate /path/to/filename.cer
-key /path/to/filename.keychain***

A screenshot of a macOS terminal window. The title bar shows 'username — bash — 79x11'. The terminal output shows the user logging in, running the 'sudo fdsetup changerecovery' command with various flags, receiving a warning about an existing keychain, and being prompted for a password.

```
username — bash — 79x11
Last login: Mon Aug 26 12:52:54 on ttys001
computername:~ username$ sudo fdsetup changerecovery -institutional -keychain
-certificate /path/to/new_recovery_key/public_key.cer -key /path/to/old_recovery_key/FileVaultMaster.keychain
Unexpected FileVaultMaster keychain was found. Attempting to move existing keychain to /Library/Application Support/fdsetup/.
2013-08-26 12:54:53.920 fdsetup[348:507] fdsetup: Warning: Using certificate option with existing FileVault master keychain. Moved keychain to /Library/Application Support/fdsetup/.
Enter the password for the recovery keychain:
computername:~ username$
```


fdsetup changerecovery -institutional -inputplist

```
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
<key>Password</key>
<string>password</string>
<key>Certificate</key>
<data>
MIIDLjCCAhagAwIBAgIENSqyWTALBgkqhkiG9w0BAQUwSTEFMB0GA1UEAwWRmlsZVZhdWx0
IFJlY292ZXJ5IEtleTEmMCQGA1UEDQwddHJvdXRvbnQtdm0zLnRhb21lY2h3b3Jrcy5uZXQw
HhcNMTMwODI0MjEzNTI1WhcNMTQwODI0MjEzNTI1WjBJMR8wHQYDVQQDDDBZGaWxlVmF1bHQg
UmVjb3ZlcnkgS2V5MSYwJAYDVQQNDB10cm91dG9udC12bTMudGFvbWVjaHdvcmtzLm5ldDCC
ASIwDQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBAMJp04TYyH8HC3H4WWV0FPV48I1cdRjh
BGi90YJX8cQrgWb72FTDCNbU+fD1ce1++KALheMNCmh0IwPshQtUuU9ngpfRYI/
L6KIgIaUW4bpUksHXZ0CIwSvfqD6+
np06wniS48KNPmnyHqxNeYyfU20wAITPznP7w0VjU7pMbvvJra03gNDxxLc44Rj5YdHxn7U7
gIUNc08RxntNgC6hPSMuZVIgz8AUFc0YuBYML9fXrHytNoaZ0JXS2djGRdwRT0V30tK2qDFh
8+vVr9vTBt3xmk6hPREXfB9odRDH6ZP9mn6R2vLLzF4ANwklha0L4+
mwbPtdfVOSs3wdkoUFF8CAwEAAaMgMB4wCwYDVR0PBAQDAgK0MA8GA1UdEwEB/
wQFMAMBAQAwDQYJKoZIhvcNAQEFBQADggEBAJ+5ZaazQmnVDrIQBkxiBXql/
aRbQpetu5jLsJtHDxRd1d2YiEFJHwy4jnUefrvTUf7648oCQ5Ft0jYL5lYn7QQ3MjvsFKzBh
2wVlqJJAhjjjwbzMoVdRxfY8Aftcler9Tw/WvxiECvdgxB37iA+EGVTrRUGuCytY+
GhawIKQMR5Ac2XBpfgPatbV9xAtCo4+M9I4jqhSx0GMwLjRHs4yELyH71w/Au3VlTYezf+
dqof0aZ8c4wB0K3T4VeoWMJjjwDe9Q3/j3sf/CjxvNhm1yYRUqUieZx7Vdf2+
oKKqMczyAVz0Y8UX26lMnpMQn4Fv9b5m0vo5U2/ZgpiQ8cHxM=
</data>
</dict>
</plist>
```

Note: All account passwords need to be supplied in cleartext.

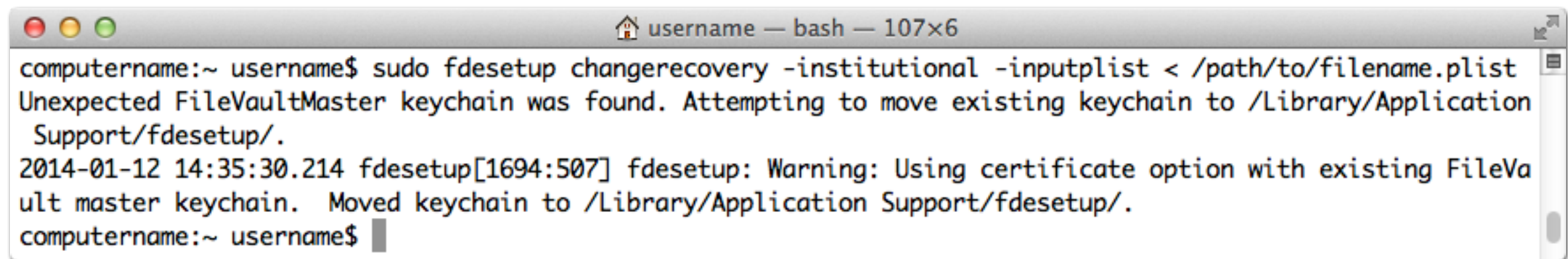
fdsetup changerecovery -institutional -inputplist

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
<key>KeychainPath</key>
<string>/path/to/filename.keychain</string>
<key>KeychainPassword</key>
<string>password</string>
<key>Certificate</key>
<data>
MIIDCDCCAfCgAwIBAgIEXKugDjALBgkqhkiG9w0BAQUwNjEfmB0GA1UEAwwWRm1sZVZhdWx0IFJlY292ZXJ5IEtleTETMBEGA1UEDQw
KZWJvbi1zaGlueTAeFw0xNDA3MDQyMzM1MjRaFw0xNTA3MDQyMzM1MjRaMDYxHzAdBgNVBAMMFkZpbGVWYXVsdCBSZWNvdmVyeSBLZX
kxEzARBgNVBA0MCmVib24tc2hpbmknwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQRNnxuc8ilkxDcDX+
wrm4ut1M8QErWjehdrVpCGA6ipCcyeYL4s8ZBgi/
ObI43q0vwgTvTVHrzzU46RIKGQcC5zYhvbvuFYIYjAgnVfaobW8xWEfRwSGihzpljlnCyvV1jQre4N6i/iIPBY8UR1svME+
1PEDUURgB/XXyhgWLIzu8V5RrUAkdLh3Hi0d+XctDs3sDigE69M+I0oThcgdN8tx3ZbGFUv+4vvMZH3bRc2Lc7goC44M5i+
dpdR1tc0xpcwxq4XYkkA/dcA+AauS95FoStVXgXB0CnfK5RneF5+
Aq0ibUHNhI2z5dIYvjFDPG3vjppsIZZdokVeCt0hSLlAgMBAAGjIDAeMAsgA1UdDwQEAWICtDAPBgNVHRMBAf8EBTADAQEAMA0GCSqG
SIb3DQEBBQUAA4IBAQCeeDPE7u++bYlrbPBWcySHRN8L/0Kqp5F554krJ06nFOFcwpgNBgZIyLzKx7gqYrJA3RHeT/yVysJ40/
4Z0ZJj6Sze6vIpPdahd5xL3QKr1A2ILBBKJtGBfgsevPfi9Gs4G/
PHb0LcEeT9mLm8gAlChK5V2HA00R3b31S5Zcw5rUNNxzSqSy4ZZzlnpY4WC7pzEDZ1KIUnIQZm1r3CkwdqzREzeW90o8VhnY3ypitNP
h+BmzbUks1zHccxJSCFp1cSZQw47nceqWtZkRcsLiSpwlpfXPY7wnrbJriPN9Yz0MFax5fXVWHefDXPy/lBCt9milfz30EGk0k0q/
AxD3
</data>
</dict>
</plist>
```

Note: All keychain passwords need to be supplied in cleartext.

fdsetup changerecovery -institutional -inputplist

*sudo fdsetup changerecovery -institutional
-inputplist < /path/to/filename.plist*

A screenshot of a macOS terminal window. The title bar shows a home icon, the text 'username — bash — 107x6', and window control buttons. The terminal text shows the command 'sudo fdsetup changerecovery -institutional -inputplist < /path/to/filename.plist' being executed. The output includes a warning about an existing FileVaultMaster keychain being moved and a timestamp '2014-01-12 14:35:30.214'.

```
computername:~ username$ sudo fdsetup changerecovery -institutional -inputplist < /path/to/filename.plist
Unexpected FileVaultMaster keychain was found. Attempting to move existing keychain to /Library/Application
Support/fdsetup/.
2014-01-12 14:35:30.214 fdsetup[1694:507] fdsetup: Warning: Using certificate option with existing FileVa
ult master keychain. Moved keychain to /Library/Application Support/fdsetup/.
computername:~ username$
```

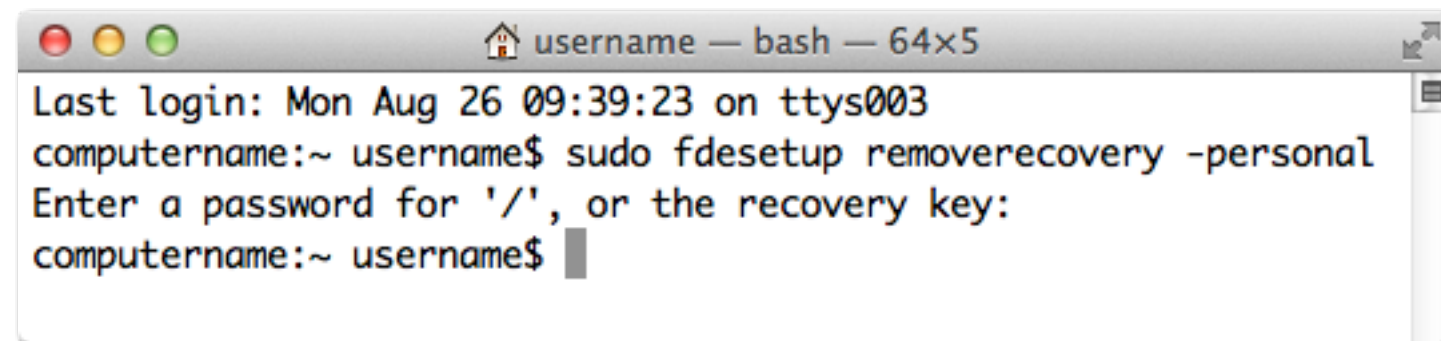
fdsetup removerecovery



➤ Removes the current institutional or personal recovery key

fdsetup removerecovery -personal

sudo fdsetup removerecovery -personal



```
username — bash — 64x5
Last login: Mon Aug 26 09:39:23 on ttys003
computername:~ username$ sudo fdsetup removerecovery -personal
Enter a password for '/', or the recovery key:
computername:~ username$
```

fdsetup removerecovery -personal -inputplist

```
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
<key>Password</key>
<string>password</string>
</dict>
</plist>
```

Note: All account passwords need to be supplied in cleartext.

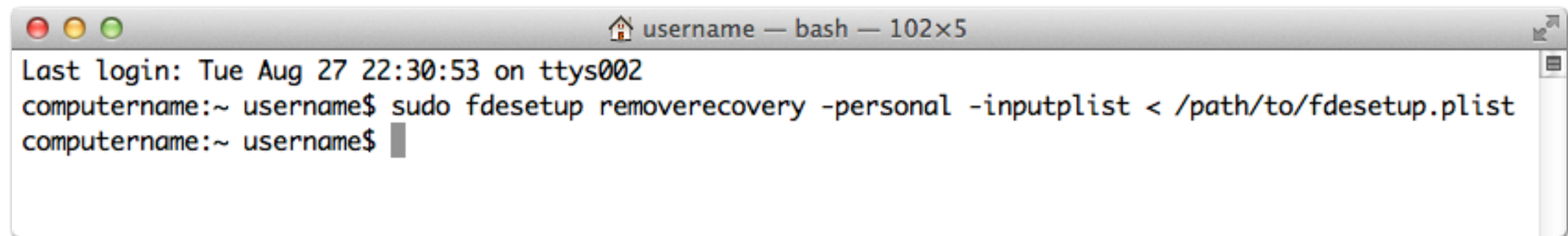
fdsetup removerecovery -personal -inputplist



Note: All keychain passwords need to be supplied in cleartext.

fdsetup removerecovery -personal -inputplist

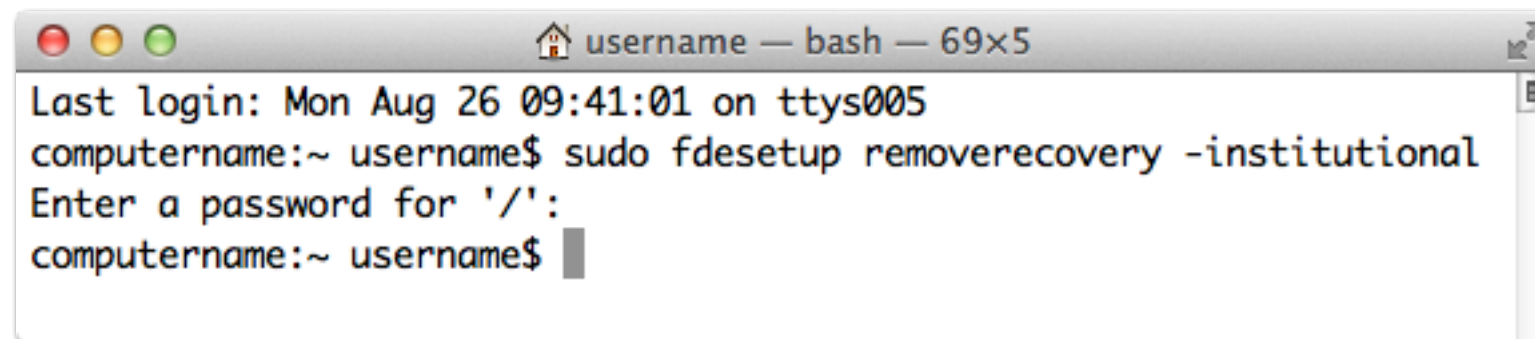
*sudo fdsetup removerecovery -personal
-inputplist < /path/to/filename.plist*

A screenshot of a macOS terminal window. The title bar shows a home icon, the text 'username — bash — 102x5', and window control buttons. The terminal content shows a login message, a command being executed with sudo, and the prompt returning to the user.

```
username — bash — 102x5
Last login: Tue Aug 27 22:30:53 on ttys002
computername:~ username$ sudo fdsetup removerecovery -personal -inputplist < /path/to/fdsetup.plist
computername:~ username$
```

fdsetup removerecovery -institutional

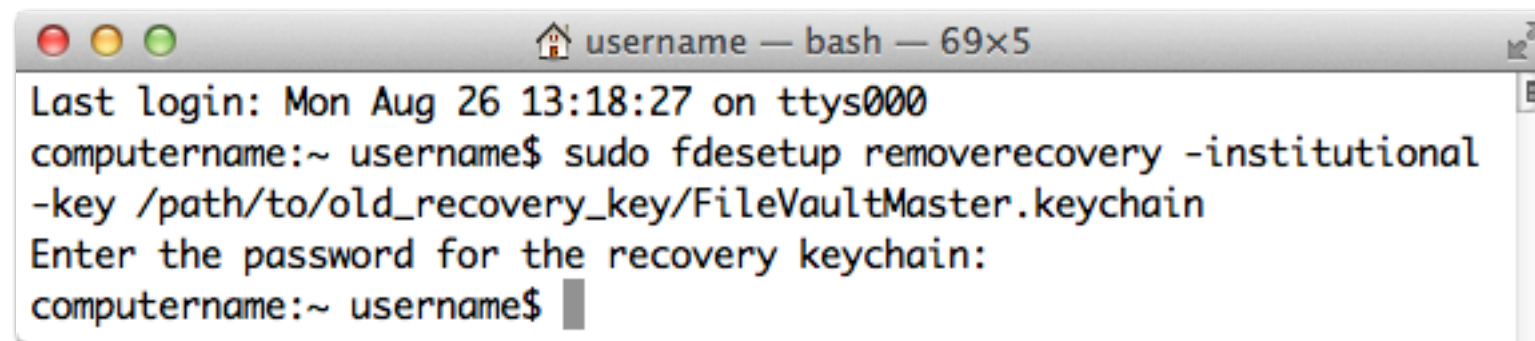
sudo fdsetup removerecovery -institutional

A terminal window with a title bar that reads 'username — bash — 69x5'. The window contains the following text: 'Last login: Mon Aug 26 09:41:01 on ttys005', 'computername:~ username\$ sudo fdsetup removerecovery -institutional', 'Enter a password for \'/\':', and 'computername:~ username\$' followed by a cursor. The window has standard macOS window controls (red, yellow, green buttons) and a zoom icon in the top right corner.

```
username — bash — 69x5
Last login: Mon Aug 26 09:41:01 on ttys005
computername:~ username$ sudo fdsetup removerecovery -institutional
Enter a password for '/':
computername:~ username$
```


fdsetup removerecovery -institutional -keychain

*sudo fdsetup removerecovery -institutional -keychain
-key /path/to/filename.keychain*

A screenshot of a macOS terminal window. The title bar shows a home icon, the text 'username — bash — 69x5', and window control buttons. The terminal content shows a login message, the command 'sudo fdsetup removerecovery -institutional -key /path/to/old_recovery_key/FileVaultMaster.keychain', a password prompt, and the command prompt ready for input.

```
username — bash — 69x5
Last login: Mon Aug 26 13:18:27 on ttys000
computername:~ username$ sudo fdsetup removerecovery -institutional
-key /path/to/old_recovery_key/FileVaultMaster.keychain
Enter the password for the recovery keychain:
computername:~ username$
```


fdsetup removerecovery -institutional -inputplist

```
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
<key>Password</key>
<string>password</string>
</dict>
</plist>
```

Note: All account passwords need to be supplied in cleartext.

fdsetup removerecovery -institutional -inputplist



Note: All keychain passwords need to be supplied in cleartext.

fdesetup sync



➤ Compares directory service account information with Mac's list of FileVault 2 enabled accounts

- Removes users that have been removed from the directory service.
- Does not add directory service accounts to list of FileVault 2 enabled accounts.

fdsetup authrestart



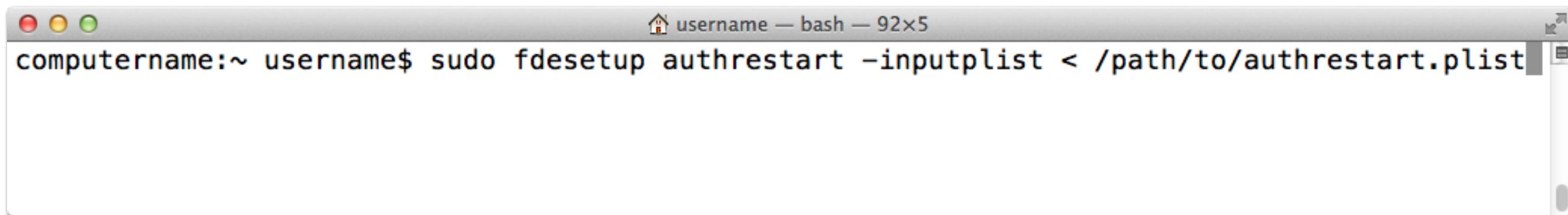
fdsetup authrestart -inputplist

```
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
<key>Password</key>
<string>password</string>
</dict>
</plist>
```

Note: All account passwords need to be supplied in cleartext.

fdsetup authrestart -inputplist

sudo fdsetup authrestart -inputplist < /path/to/filename.plist

A screenshot of a macOS terminal window. The title bar shows a home icon, the text 'username — bash — 92x5', and window control buttons. The terminal content shows the prompt 'computername:~ username\$' followed by the command 'sudo fdsetup authrestart -inputplist < /path/to/authrestart.plist'.

```
computername:~ username$ sudo fdsetup authrestart -inputplist < /path/to/authrestart.plist
```

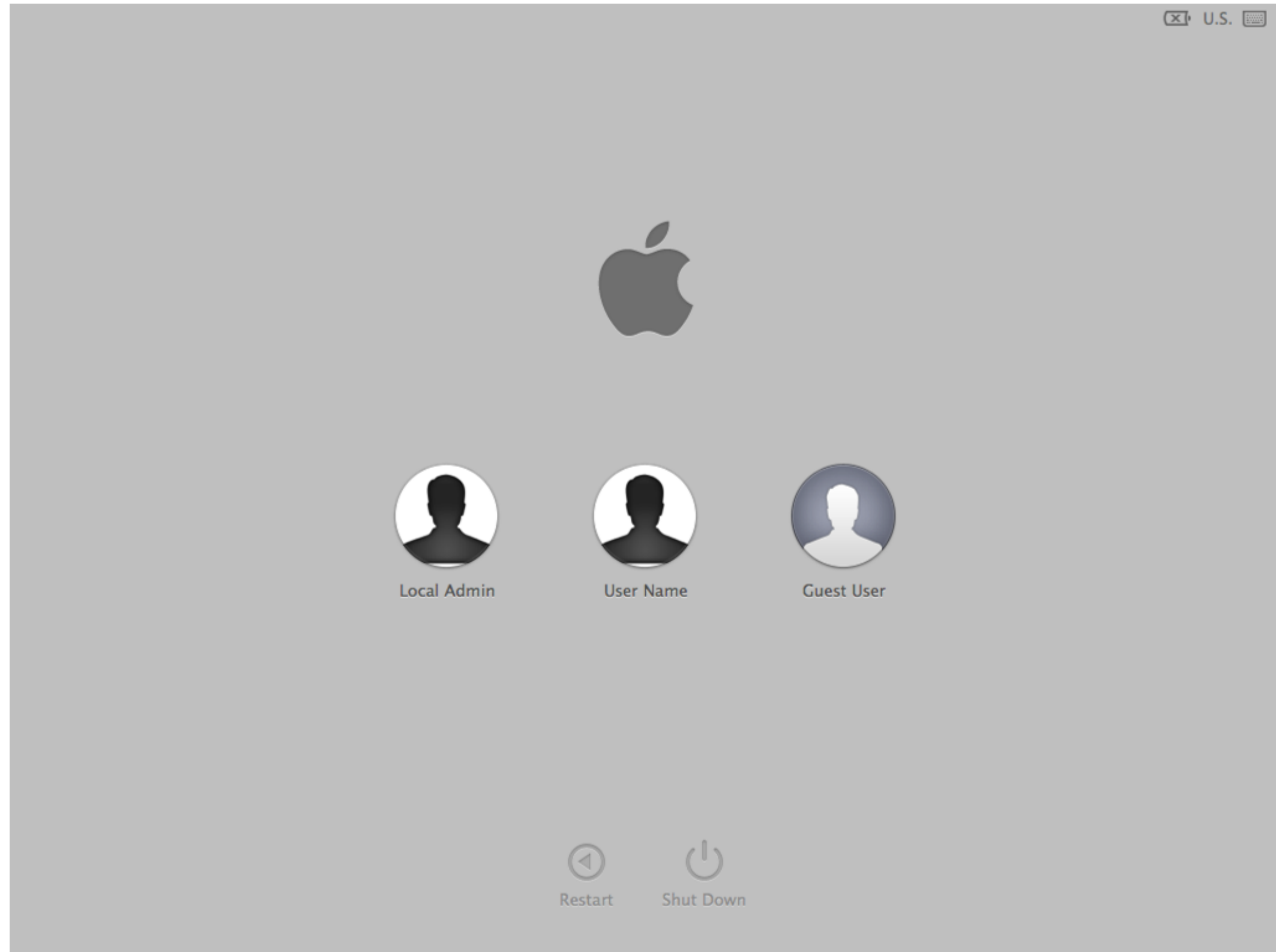

Limitations of FileVault 2

- FileVault 2 is an overall better solution than FileVault 1 is, but it does not necessarily cover all workplaces' requirements for full disk encryption.
 - Can't use remote management tools at the pre-boot login screen
 - Can't use authentication methods other than password/passphrases
 - Pre-boot login screen can't display username / password blanks

FileVault 2 and the Law

- For folks who need to satisfy regulatory requirements for encryption:
 - FileVault 2 is FIPS 140-2 Compliant for Mavericks.
 - FileVault 2's underlying low level encryption uses Apple's Common Crypto implementation
 - It is Apple's stated intention to continue its FIPS validation for shipping products.

FileVault 2's Guest User





Local Admin



User Name



Guest User



Restart

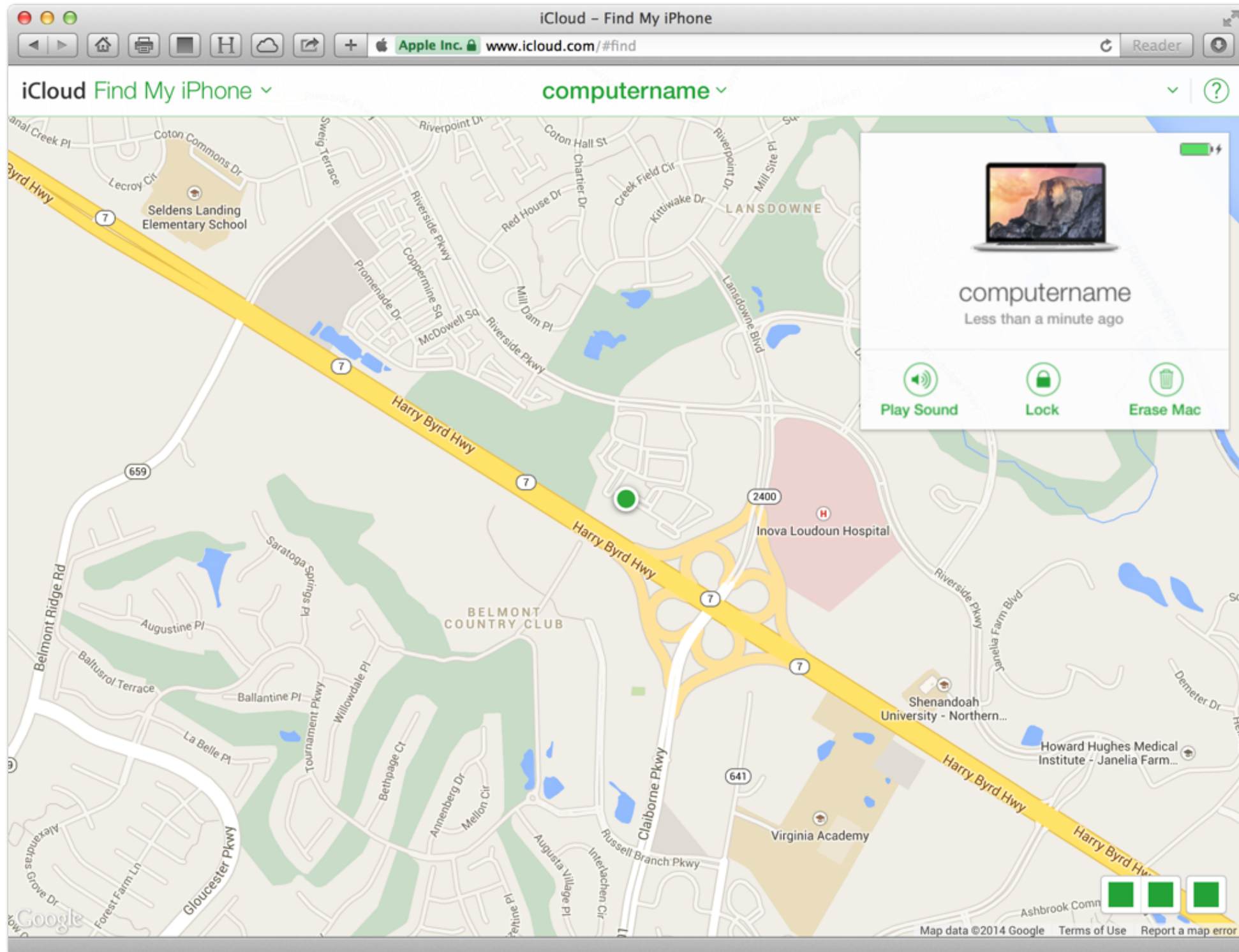


Shut Down

FileVault 2's Guest User



Remote Locking and Wiping





Sign in to iCloud

Apple ID

Password

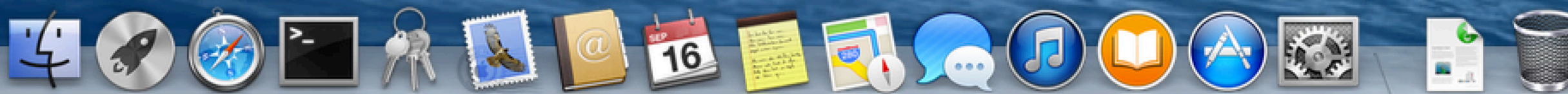


☐ Keep me signed in





Macintosh HD





Sign in to iCloud

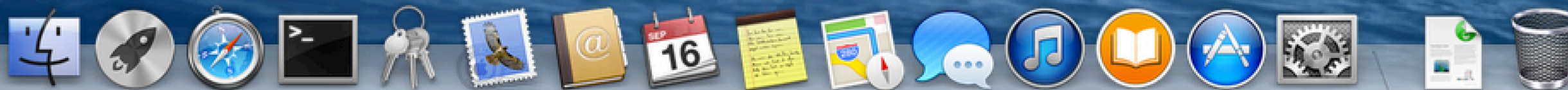
Apple ID

Password 

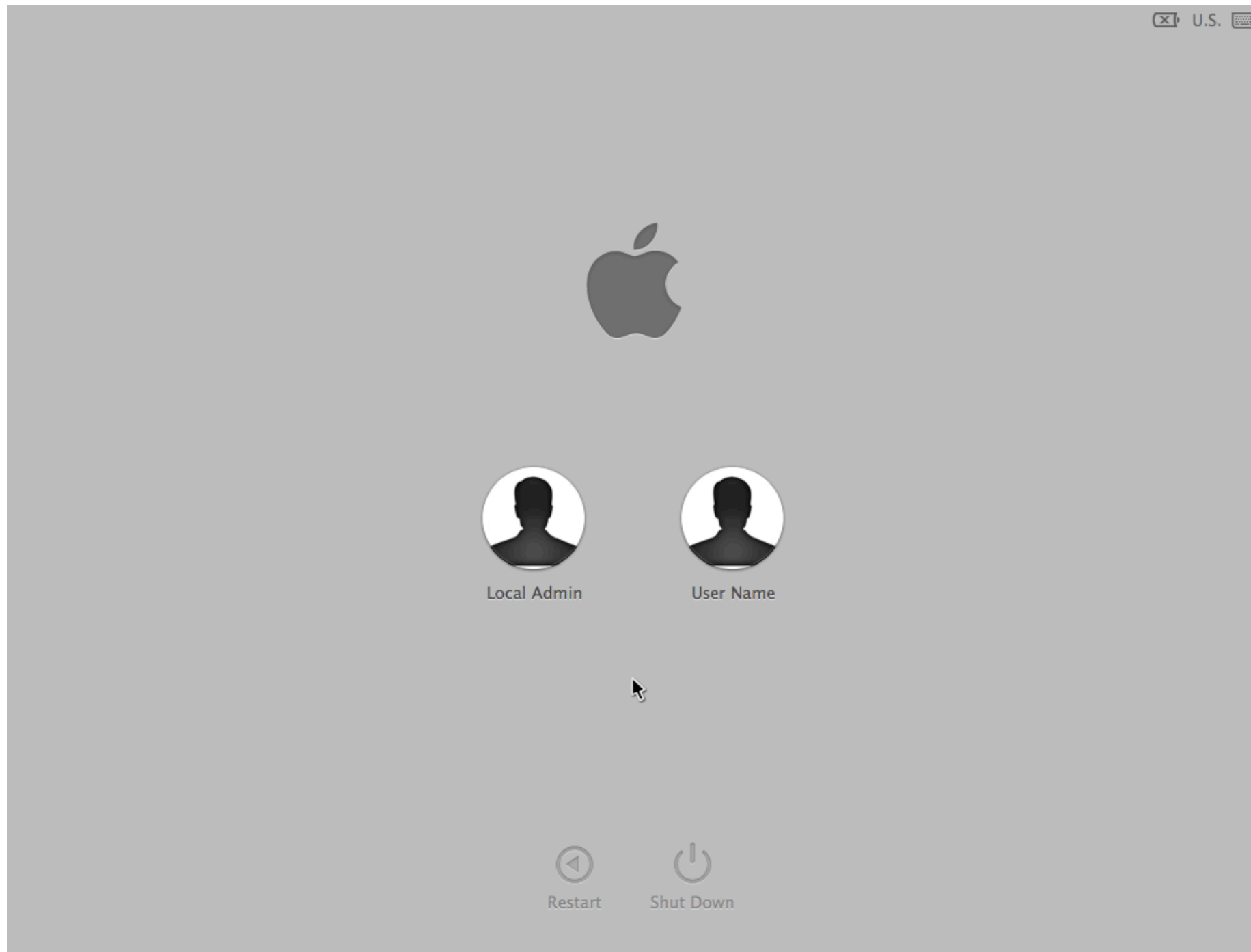
☐ Keep me signed in



Macintosh HD



Disabling FileVault 2's Automatic User Login



OS X: How to disable automatic login when FileVault is enabled

Languages English

OS X Mavericks lets you disable automatic login when FileVault is enabled using one of the methods described below.

Passwords

When disabling automatic login, two passwords are required while starting up:

- One password is needed to unlock the disk at the EFI Login Window.
- Another password is needed to log in at the regular Login Window.

These passwords can be the same if you are logging in as the user who unlocked the disk.

Command line

Use the following command to disable automatic login when FileVault is enabled:

```
sudo defaults write /Library/Preferences/com.apple.loginwindow DisableFDEAutoLogin  
-bool YES
```

Use the following command to allow automatic login when FileVault is enabled:

```
sudo defaults delete /Library/Preferences/com.apple.loginwindow DisableFDEAutoLogin
```

Profile Manager

You can use Profile Manager to create a Custom Settings payload with the following information.

- Preference Domain: com.apple.loginwindow
- Key: DisableFDEAutoLogin
- Type: Boolean
- Value: On (checked)



Local Admin



User Name



Restart



Shut Down

Tips for Improving FileVault 2 Performance

- FileVault 2 is able to encrypt and decrypt SSDs much faster than conventional spinning disk hard drives.
- Processors that have built-in AES-NI support provide additional performance improvements for FileVault 2-encrypted Macs.

FileVault 2
Decoded