Fixing without moving

Unix to the rescue

Someone's ruined our Web server, but we're nowhere near it. How are we going to fix it? Fortunately, Unix can come to the rescue with SSH and SFTP.

Words? Pictures? Whatever

Secure file transfer

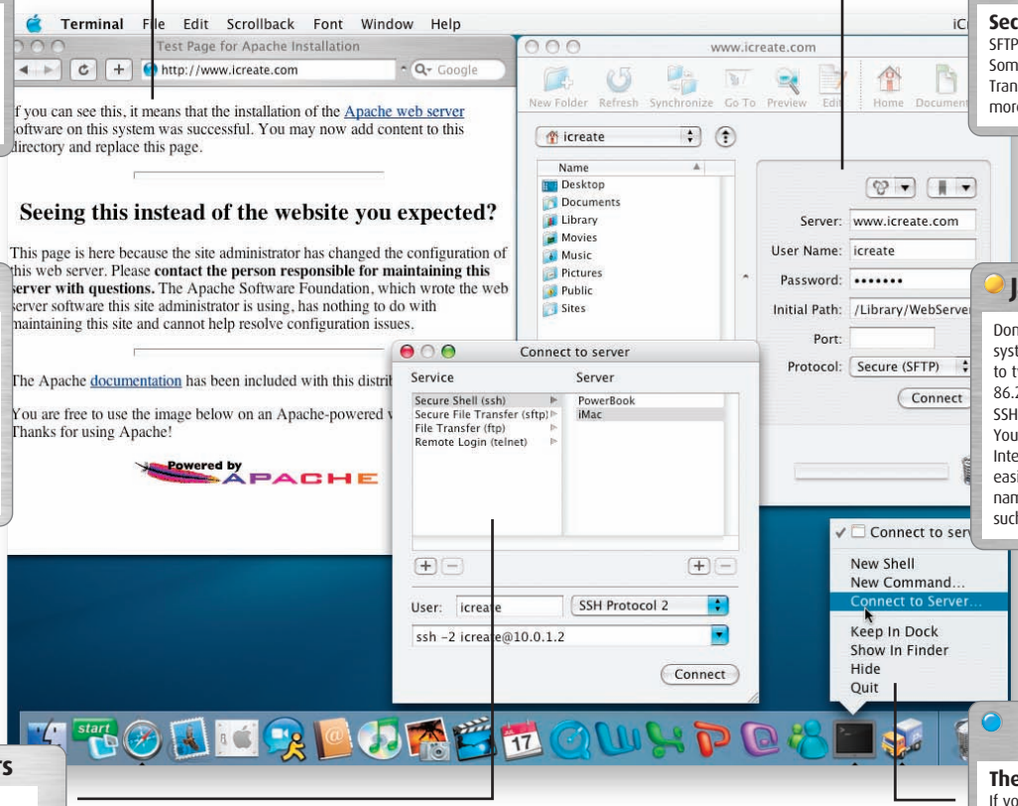
SFTP isn't just for the command line. Some FTP programs, such as Transmit, will also let you use this more secure file transfer system.

Jargon Buster: SSH

SSH – or Secure Shell – is a way for you to log into any OS X Mac that you can find on a network. It uses Public Key Infrastructure (the same system that online retailers and banks use) to ensure all your communications are secure. You can also use SSH to access Unix and Linux systems, and vice versa.

Jargon Buster: DNS

Domain name service (DNS) is the system that ensures we don't have to type Internet addresses, such as 86.23.45.126, into Web browsers or SSH to get to the sites we want. You could memorise your Mac's Internet address to find it, but an easier way is to get your own DNS name for free through a company such as **www.dyndns.org**



Safer than the others

Out with the old...

SSH and SFTP replace less secure, older systems. But Terminal will still let you connect with Telnet and FTP if you want.

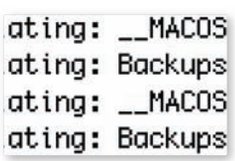
Inside is even easier

The magic of Rendezvous

If you want to access a Mac on your network, Rendezvous lets you automatically find it from Terminal's Connect to Server window.

Tutorial: Remotely and securely control your Mac

We're going to learn the two basics of controlling your Mac from afar: the secure shell (SSH) and secure FTP (SFTP)



Task: Control your Mac from anywhere in the world

Difficulty: Expert

Time needed: 30 minutes

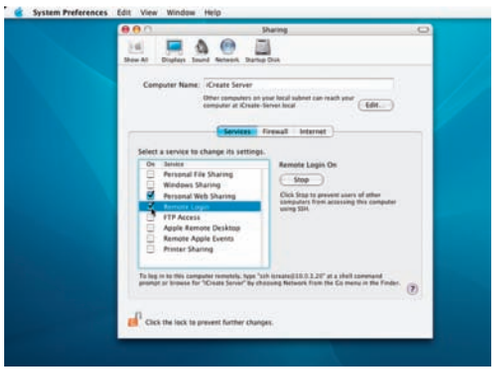
Disaster has struck at iCreate Consultants. The junior Web developer has just upgraded the Web server and completely replaced the company's Web pages with the default pages. Even worse, you're on holiday and nowhere near the backup server. How can you fix the problem?

Fortunately, the Mac has a way for you to control it from anywhere in the world, provided you have Internet access. Called Remote Login (or SSH to be technical), it enables you to log in to your Mac completely securely and use Unix commands to control it. These commands are exactly the same as those you would use on your Mac through Terminal, so you don't have to learn anything new. You can even SSH to one machine and then SSH from that machine to another. Clever, huh?

Even better than that, SSH comes with a replacement for the standard FTP server, called SFTP. FTP is a way for you to transfer files between computers over the Internet. However, it isn't very secure, since it will send your password unencrypted. Anyone intercepting communications with your Mac would be able to find out your password and do anything you could do, including copy and delete files. Fortunately, SFTP is as safe as houses and enables you to look anywhere on your Mac that you're allowed to, unlike FTP. You also need to enable FTP in the Sharing pane of your Systems Preference, but SFTP is activated when you turn on Remote Login, so that's one fewer checkbox to remember.

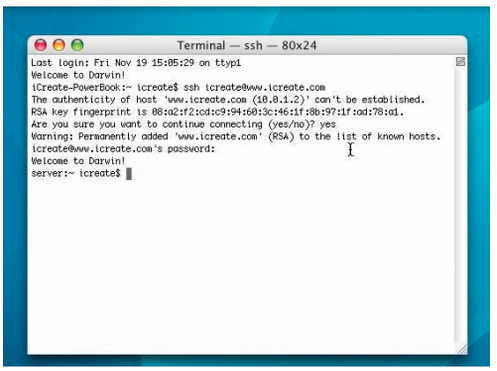
What does all this mean? Well, even when you're on the other side of the world, you can nip into Starbucks with your AirPort-enabled iBook and log into your home Mac to grab that important file you left behind, update it with the latest security patch or even to back up your iBook in case you lose it. Read on to find out how the whole thing works.

Step-by-step | OS X | Using Remote Login with Terminal



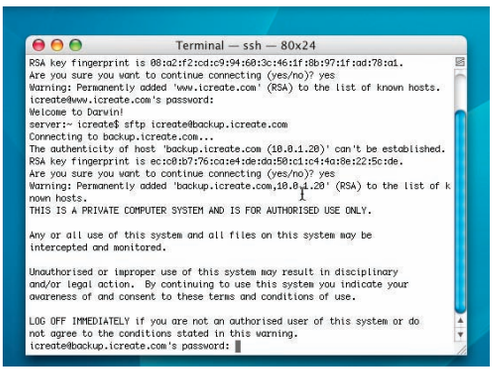
1: Turn on remote login

Before you leave on your holiday, remember to click in the 'Remote Login' checkbox in the Sharing System Preferences pane. That will activate SSH.



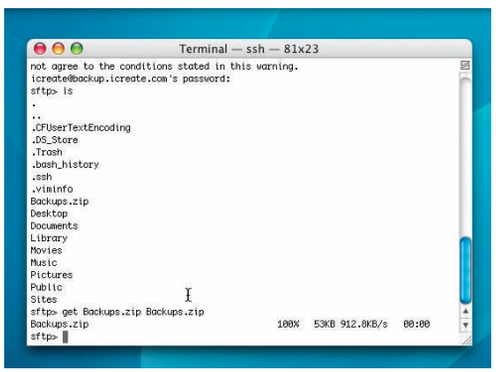
2: Log in to the Web server

Your ID on the web server is 'icreate', so type 'ssh icreate@www.icreate.com' in Terminal to access it. You'll be asked if you trust it, so say 'yes'.



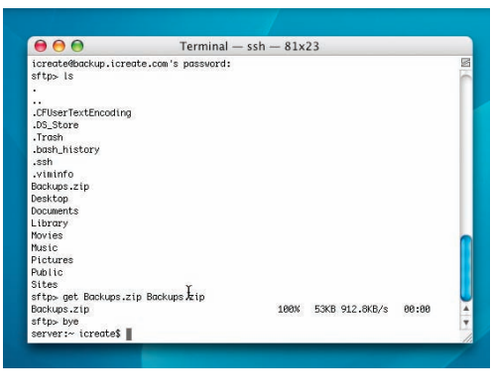
3: Get the backups

Enter your password to finish logging in. The backups are on another server with Remote Login enabled, so you can use SFTP to get the files.



4: Grab the files...

List the files with 'ls'. The command to download a file is 'get', followed by its filename and what we want to call it on our Mac ('put' will upload).

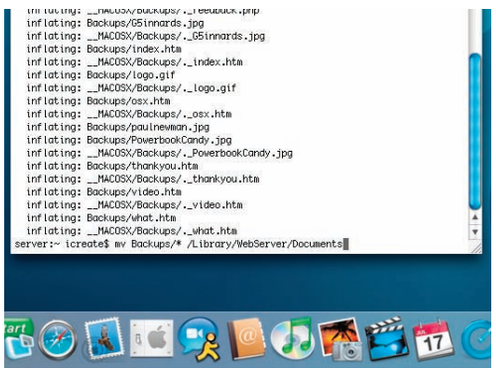


5: ...and run

Since we don't want to do anything else, all that's needed now is to log out of the backup server using the command 'bye'.

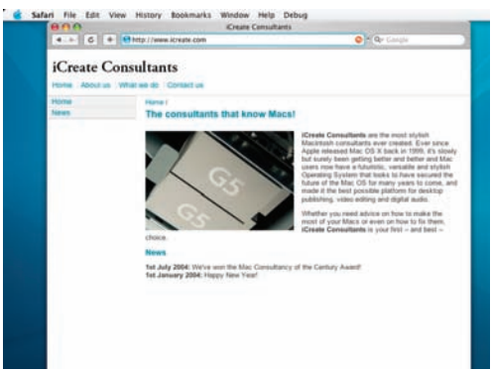
6: Big time

The files are inside a Mac OS X archive. Use the 'unzip' command to extract them. An extra directory of Mac-specific stuff is created, but you can ignore it.



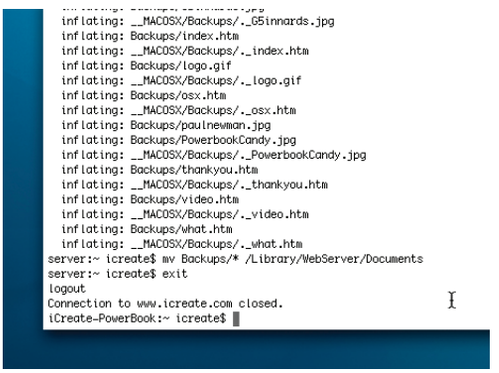
7: Replace the bad files

We can use the 'mv' command now to start replacing all the files in the Web server directory with those from the backup.



8: Check it worked

When this task is complete, it's a good idea to make sure everything's the way it should be by calling up the Web page in Safari.



9: Take your leave

Since everything's fine, log out of the Web server using the command 'exit'. You'll be back at the Terminal prompt of your own computer.