Mac OS X Security

Rick Hill
UC Davis, Engineering, Webmaster

Jackie Simmons
UC Davis, Engineering, Deans’ Office

Paul Waterstraat
UC Davis, Geology, Systems Manager
Mac OS X Security

“Trust No One”
Fox Mulder - The X Files

“Computer security is the result of a long unbroken chain of diligence.”
Paul Waterstraat

- Network Security
- Server Security
- Wired & Wireless Security
Enabling and Disabling Services

- Internal Monitoring of Services
  - *Sharing and Server Settings*
  - `netstat`
  - `host`
  - `lsof`

- External Monitoring of Services
  - `nmap`
Mac OS X client

Out of the box, Mac OS X is pretty secure.

If you check the Sharing System Preference you’ll find things disabled by default.
Mac OS X Server

Server Settings.app can show you the status of the common services that Apple has supplied with Mac OS X Server. Out of the box, if you don’t enable any services your machine is pretty secure.
Mac OS X

But how do I know there aren’t other ports or daemons or service running that leave my machine vulnerable to the network?

On the next few slides I’ll describe a couple of tools you can use on the machine itself and from an external machine to determine which ports are open and who is connecting to them.
Use the `netstat` command to list the active and pending TCP/IP connections between your machine and the network. This is a useful tool if you suspect that someone is breaking into your computer or using your computer to break into other computers.

The command’s output displays the host and port number of each end of the connection, and the connection’s current status.
Use the terminal command `netstat` to learn on which ports your computer is listening for connections.

```
epaul% /usr/sbin/netstat -a -f inet | grep LISTEN
```

Active Internet connections
Proto Recv-Q Send-Q Local Address          Foreign Address        (state)
tcp4       0      0  localhost.ipp          *.*                    LISTEN
tcp4       0      0  localhost.1033         *.*                    LISTEN

Two ports are open. The **ipp** port is the CUPS printing software and port **1033** is netinfo. Both client and server look like this with nothing enabled.
Compare that with a Sun Workstation running Solaris 8

<table>
<thead>
<tr>
<th>Local Address</th>
<th>Remote Address</th>
<th>Swind</th>
<th>Send-Q</th>
<th>Rwind</th>
<th>Recv-Q</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>localhost.4999</td>
<td><em>.</em></td>
<td>0</td>
<td>0</td>
<td>24576</td>
<td>0</td>
<td>LISTEN</td>
</tr>
<tr>
<td>*.sunrpc</td>
<td><em>.</em></td>
<td>0</td>
<td>0</td>
<td>24576</td>
<td>0</td>
<td>LISTEN</td>
</tr>
<tr>
<td>*.ftp</td>
<td><em>.</em></td>
<td>0</td>
<td>0</td>
<td>24576</td>
<td>0</td>
<td>LISTEN</td>
</tr>
<tr>
<td>*.telnet</td>
<td><em>.</em></td>
<td>0</td>
<td>0</td>
<td>24576</td>
<td>0</td>
<td>LISTEN</td>
</tr>
<tr>
<td>*.shell</td>
<td><em>.</em></td>
<td>0</td>
<td>0</td>
<td>24576</td>
<td>0</td>
<td>LISTEN</td>
</tr>
<tr>
<td>*.login</td>
<td><em>.</em></td>
<td>0</td>
<td>0</td>
<td>24576</td>
<td>0</td>
<td>LISTEN</td>
</tr>
<tr>
<td>*.exec</td>
<td><em>.</em></td>
<td>0</td>
<td>0</td>
<td>24576</td>
<td>0</td>
<td>LISTEN</td>
</tr>
<tr>
<td>*.exec</td>
<td><em>.</em></td>
<td>0</td>
<td>0</td>
<td>24576</td>
<td>0</td>
<td>LISTEN</td>
</tr>
<tr>
<td>*.uucp</td>
<td><em>.</em></td>
<td>0</td>
<td>0</td>
<td>24576</td>
<td>0</td>
<td>LISTEN</td>
</tr>
<tr>
<td>*.finger</td>
<td><em>.</em></td>
<td>0</td>
<td>0</td>
<td>24576</td>
<td>0</td>
<td>LISTEN</td>
</tr>
<tr>
<td>*.time</td>
<td><em>.</em></td>
<td>0</td>
<td>0</td>
<td>24576</td>
<td>0</td>
<td>LISTEN</td>
</tr>
<tr>
<td>*.echo</td>
<td><em>.</em></td>
<td>0</td>
<td>0</td>
<td>24576</td>
<td>0</td>
<td>LISTEN</td>
</tr>
<tr>
<td>*.discard</td>
<td><em>.</em></td>
<td>0</td>
<td>0</td>
<td>24576</td>
<td>0</td>
<td>LISTEN</td>
</tr>
<tr>
<td>*.daytime</td>
<td><em>.</em></td>
<td>0</td>
<td>0</td>
<td>24576</td>
<td>0</td>
<td>LISTEN</td>
</tr>
<tr>
<td>*.chargen</td>
<td><em>.</em></td>
<td>0</td>
<td>0</td>
<td>24576</td>
<td>0</td>
<td>LISTEN</td>
</tr>
<tr>
<td>*.32771</td>
<td><em>.</em></td>
<td>0</td>
<td>0</td>
<td>24576</td>
<td>0</td>
<td>LISTEN</td>
</tr>
<tr>
<td>*.32772</td>
<td><em>.</em></td>
<td>0</td>
<td>0</td>
<td>24576</td>
<td>0</td>
<td>LISTEN</td>
</tr>
<tr>
<td>*.32773</td>
<td><em>.</em></td>
<td>0</td>
<td>0</td>
<td>24576</td>
<td>0</td>
<td>LISTEN</td>
</tr>
<tr>
<td>*.32774</td>
<td><em>.</em></td>
<td>0</td>
<td>0</td>
<td>24576</td>
<td>0</td>
<td>LISTEN</td>
</tr>
<tr>
<td>*.fs</td>
<td><em>.</em></td>
<td>0</td>
<td>0</td>
<td>24576</td>
<td>0</td>
<td>LISTEN</td>
</tr>
<tr>
<td>*.32775</td>
<td><em>.</em></td>
<td>0</td>
<td>0</td>
<td>24576</td>
<td>0</td>
<td>LISTEN</td>
</tr>
<tr>
<td>*.printer</td>
<td><em>.</em></td>
<td>0</td>
<td>0</td>
<td>24576</td>
<td>0</td>
<td>LISTEN</td>
</tr>
<tr>
<td>*.lockd</td>
<td><em>.</em></td>
<td>0</td>
<td>0</td>
<td>24576</td>
<td>0</td>
<td>LISTEN</td>
</tr>
<tr>
<td>*.dtspc</td>
<td><em>.</em></td>
<td>0</td>
<td>0</td>
<td>24576</td>
<td>0</td>
<td>LISTEN</td>
</tr>
<tr>
<td>*.8888</td>
<td><em>.</em></td>
<td>0</td>
<td>0</td>
<td>24576</td>
<td>0</td>
<td>LISTEN</td>
</tr>
<tr>
<td>*.32777</td>
<td><em>.</em></td>
<td>0</td>
<td>0</td>
<td>24576</td>
<td>0</td>
<td>LISTEN</td>
</tr>
<tr>
<td>*.32778</td>
<td><em>.</em></td>
<td>0</td>
<td>0</td>
<td>24576</td>
<td>0</td>
<td>LISTEN</td>
</tr>
<tr>
<td>*.32779</td>
<td><em>.</em></td>
<td>0</td>
<td>0</td>
<td>24576</td>
<td>0</td>
<td>LISTEN</td>
</tr>
<tr>
<td>*.27000</td>
<td><em>.</em></td>
<td>0</td>
<td>0</td>
<td>24576</td>
<td>0</td>
<td>LISTEN</td>
</tr>
<tr>
<td>*.33945</td>
<td><em>.</em></td>
<td>0</td>
<td>0</td>
<td>24576</td>
<td>0</td>
<td>LISTEN</td>
</tr>
<tr>
<td>*.898</td>
<td><em>.</em></td>
<td>0</td>
<td>0</td>
<td>24576</td>
<td>0</td>
<td>LISTEN</td>
</tr>
<tr>
<td>*.5987</td>
<td><em>.</em></td>
<td>0</td>
<td>0</td>
<td>24576</td>
<td>0</td>
<td>LISTEN</td>
</tr>
<tr>
<td>*.34209</td>
<td><em>.</em></td>
<td>0</td>
<td>0</td>
<td>24576</td>
<td>0</td>
<td>LISTEN</td>
</tr>
<tr>
<td>*.6000</td>
<td><em>.</em></td>
<td>0</td>
<td>0</td>
<td>24576</td>
<td>0</td>
<td>LISTEN</td>
</tr>
</tbody>
</table>
Now, I open a web page http://security.ucdavis.edu, and an ssh session with quartz.geology.ucdavis.edu. Use **netstat** to see what connections are now open.

```
epaul% /usr/sbin/netstat -a -f inet

Active Internet connections
Proto Recv-Q Send-Q  Local Address          Foreign Address        (state)
tcp4  0      0  ammonite.geology.49205 quartz.geology.u.ssh   ESTABLISHED
tcp4  0      0  ammonite.geology.49203 192.35.210.223.http    ESTABLISHED
tcp4  0      0  ammonite.geology.49202 a192-35-210-200..http  ESTABLISHED
tcp4  0      0  ammonite.geology.49201 des067.ucdavis.e.http  ESTABLISHED
tcp4  0      0  ammonite.geology.49199 des067.ucdavis.e.http  ESTABLISHED
tcp4  0      0  ammonite.geology.49198 des067.ucdavis.e.http  ESTABLISHED
tcp4  0      0  localhost.1033         localhost.968          ESTABLISHED
tcp4  0      0  localhost.968          localhost.1033         ESTABLISHED
tcp4  0      0  localhost.ipp          *.*                    LISTEN
```
netstat

First, here’s the ssh connection to quartz:

<table>
<thead>
<tr>
<th>Proto</th>
<th>Recv-Q</th>
<th>Send-Q</th>
<th>Local Address</th>
<th>Foreign Address</th>
<th>(state)</th>
</tr>
</thead>
<tbody>
<tr>
<td>tcp4</td>
<td>0</td>
<td>0</td>
<td>ammonite.geology.49205</td>
<td>quartz.geology.u.ssh</td>
<td>ESTABLISHED</td>
</tr>
</tbody>
</table>

Next are the connection that result from opening the web page at security.ucdavis.edu

<table>
<thead>
<tr>
<th>Proto</th>
<th>Recv-Q</th>
<th>Send-Q</th>
<th>Local Address</th>
<th>Foreign Address</th>
<th>(state)</th>
</tr>
</thead>
<tbody>
<tr>
<td>tcp4</td>
<td>0</td>
<td>0</td>
<td>ammonite.geology.49203</td>
<td>192.35.210.223.http</td>
<td>ESTABLISHED</td>
</tr>
<tr>
<td>tcp4</td>
<td>0</td>
<td>0</td>
<td>ammonite.geology.49202</td>
<td>a192-35-210-200..http</td>
<td>ESTABLISHED</td>
</tr>
<tr>
<td>tcp4</td>
<td>0</td>
<td>0</td>
<td>ammonite.geology.49201</td>
<td>des067.ucdavis.e.http</td>
<td>ESTABLISHED</td>
</tr>
<tr>
<td>tcp4</td>
<td>0</td>
<td>0</td>
<td>ammonite.geology.49200</td>
<td>des067.ucdavis.e.http</td>
<td>ESTABLISHED</td>
</tr>
<tr>
<td>tcp4</td>
<td>0</td>
<td>0</td>
<td>ammonite.geology.49199</td>
<td>des067.ucdavis.e.http</td>
<td>ESTABLISHED</td>
</tr>
<tr>
<td>tcp4</td>
<td>0</td>
<td>0</td>
<td>ammonite.geology.49198</td>
<td>des067.ucdavis.e.http</td>
<td>ESTABLISHED</td>
</tr>
</tbody>
</table>

But where’s security.ucdavis.edu? The **host** command shows us that it’s an alias for des067.ucdavis.edu

```
epaul% /usr/bin/host security.ucdavis.edu
security.ucdavis.edu is a nickname for des067.ucdavis.edu
des067.ucdavis.edu has address 169.237.11.67
```
You may have noticed that `netstat` displays only 22 characters of host and port information.

<table>
<thead>
<tr>
<th>Proto</th>
<th>Recv-Q</th>
<th>Send-Q</th>
<th>Local Address</th>
<th>Foreign Address</th>
<th>(state)</th>
</tr>
</thead>
<tbody>
<tr>
<td>tcp4</td>
<td>0</td>
<td>0</td>
<td>ammonite.geology.49203</td>
<td>192.35.210.223.http</td>
<td>ESTABLISHED</td>
</tr>
<tr>
<td>tcp4</td>
<td>0</td>
<td>0</td>
<td>ammonite.geology.49202</td>
<td>a192-35-210-200..http</td>
<td>ESTABLISHED</td>
</tr>
</tbody>
</table>

Use `-n` to show addresses as numbers.

```
epaul% /usr/sbin/netstat -a -f inet -n
```

Active Internet connections

<table>
<thead>
<tr>
<th>Proto</th>
<th>Recv-Q</th>
<th>Send-Q</th>
<th>Local Address</th>
<th>Foreign Address</th>
<th>(state)</th>
</tr>
</thead>
<tbody>
<tr>
<td>tcp4</td>
<td>0</td>
<td>0</td>
<td>169.237.93.100.49205</td>
<td>169.237.93.2.22</td>
<td>ESTABLISHED</td>
</tr>
<tr>
<td>tcp4</td>
<td>0</td>
<td>0</td>
<td>169.237.93.100.49218</td>
<td>192.35.210.200.80</td>
<td>ESTABLISHED</td>
</tr>
<tr>
<td>tcp4</td>
<td>0</td>
<td>0</td>
<td>169.237.93.100.49216</td>
<td>192.35.210.223.80</td>
<td>ESTABLISHED</td>
</tr>
<tr>
<td>tcp4</td>
<td>0</td>
<td>0</td>
<td>169.237.93.100.49215</td>
<td>169.237.11.67.80</td>
<td>ESTABLISHED</td>
</tr>
<tr>
<td>tcp4</td>
<td>0</td>
<td>0</td>
<td>169.237.93.100.49214</td>
<td>169.237.11.67.80</td>
<td>ESTABLISHED</td>
</tr>
<tr>
<td>tcp4</td>
<td>0</td>
<td>0</td>
<td>169.237.93.100.49213</td>
<td>169.237.11.67.80</td>
<td>ESTABLISHED</td>
</tr>
<tr>
<td>tcp4</td>
<td>0</td>
<td>0</td>
<td>169.237.93.100.49212</td>
<td>169.237.11.67.80</td>
<td>ESTABLISHED</td>
</tr>
</tbody>
</table>

Use the host command to look up names.

```
epaul% /usr/bin/host 192.35.210.200
```

200.210.35.192.IN-ADDR.ARPA domain name pointer
a192-35-210-200.deploy.akamaitechnologies.com
The file `/etc/services` is a comprehensive list of well known network services and the ports on which they run.
Here’s another netstat display

```
epaul% /usr/sbin/netstat -f inet -a
Active Internet connections (including servers)
Proto Recv-Q Send-Q  Local Address          Foreign Address        (state)
tcp4  0      0  theoffice.geolog.ssh terpsichore.geol.49152 ESTABLISHED
tcp4  0      0  theoffice.geolog.49174 palladium.geolog.49770 ESTABLISHED
tcp4  0      0  theoffice.geolog.49174 onyx.geology.ucd.49162 ESTABLISHED
tcp4  0      0  theoffice.geolog.49174 dunite.geology.u.49228 ESTABLISHED
tcp4  0      0  theoffice.geolog.49174 amazonite.geolog.49158 ESTABLISHED
tcp4  0      0  theoffice.geolog.afpov amazonite.geolog.49153 ESTABLISHED
tcp4  0      0  theoffice.geolog.49174 neuromancer.geol.49156 ESTABLISHED
tcp4  0      0  theoffice.geolog.afpov smilodon.geology.49154 ESTABLISHED
```

We see an ssh connection and AppleShareIP (afpovertcp) connections, but what’s running on port 49174? A grep for 49174 in /etc/services finds nothing.

```
epaul% /usr/bin/grep ssh /etc/services
ssh    22/tcp  #Secure Shell Login
ssh    22/udp  #Secure Shell Login

epaul% /usr/bin/grep 49174 /etc/services
epaul%
```
Some of the ports on which the computer is listening have no corresponding entries in /etc/services and so are listed only by number. But we can use `lsof, list open files`, to show what process is listening on port 49174.

```
epaul% /usr/bin/sudo /usr/sbin/lsof -i:49174

COMMAND    PID USER   FD   TYPE      DEVICE SIZE/OFF NODE NAME
mmserver 536 root    7u  inet 0x03d64a6c      0t0  TCP *:49174 (LISTEN)
mmserver 536 root   12u  inet 0x03d6175c      0t0  TCP
  theoffice.geology.ucdavis.edu:49174->chalcedony.geology.ucdavis.edu:49441
    (ESTABLISHED)
mmserver 536 root   13u  inet 0x03d0e1dc      0t0  TCP
  theoffice.geology.ucdavis.edu:49174->neuromancer.geology.ucdavis.edu:49156
    (ESTABLISHED)
```

We see that mmserver, the MeetingMaker Server, is running on port 49174.
Add entries to /etc/services

You can add your own entries for services you use to the /etc/services file. Such as...

<table>
<thead>
<tr>
<th>Service</th>
<th>Port/TCP</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>automount</td>
<td>1023/udp</td>
<td>#Automounter</td>
</tr>
<tr>
<td>itunes</td>
<td>3689/tcp</td>
<td>#iTunes</td>
</tr>
<tr>
<td>ichat</td>
<td>5298/tcp</td>
<td>#iChatAgent</td>
</tr>
<tr>
<td>mmsserver</td>
<td>49174/tcp</td>
<td>#Meeting Maker server</td>
</tr>
</tbody>
</table>

Then netstat will display the service name instead of the port number. Also use service names with lsof.

```
epaul% /usr/sbin/netstat -a -f inet  
Proto Recv-Q Send-Q Local Address Foreign Address (state)
tcp4   0    33304 neuromancer.geol.itune ammonite.geology.49163 ESTABLISHED
tcp4   0      0 neuromancer.geol.itune ammonite.geology.49157 ESTABLISHED ...
epaul% sudo /usr/sbin/lsof -i:itunes
COMMAND    PID USER     FD  TYPE DEVICE SIZE/OFF NODE NAME
iTunes      421 epaul   14u  inet 0x02779f4c 0t0  TCP *:itunes (LISTEN)
iTunes      421 epaul   17u  inet 0x03089cbc 0t0  TCP
            neuromancer.geology.ucdavis.edu:itunes->
            ammonite.geology.ucdavis.edu:49171 (ESTABLISHED)
iTunes      421 epaul   18u  inet 0x02774acb 0t0  TCP
            neuromancer.geology.ucdavis.edu:itunes->
            ammonite.geology.ucdavis.edu:49157 (ESTABLISHED)
```
Use nmap from an external host (e.g. your desktop) to scan your servers for open ports.

epaul% /sw/bin/nmap -sT amber.geology.ucdavis.edu
Starting nmap V. 3.00 ( www.insecure.org/nmap/ )
Interesting ports on amber.geology.ucdavis.edu (169.237.93.62):
(The 1572 ports scanned but not shown below are in state: closed)
<table>
<thead>
<tr>
<th>Port</th>
<th>State</th>
<th>Service</th>
</tr>
</thead>
<tbody>
<tr>
<td>7/tcp</td>
<td>open</td>
<td>echo</td>
</tr>
<tr>
<td>9/tcp</td>
<td>open</td>
<td>discard</td>
</tr>
<tr>
<td>13/tcp</td>
<td>open</td>
<td>daytime</td>
</tr>
<tr>
<td>19/tcp</td>
<td>open</td>
<td>chargen</td>
</tr>
<tr>
<td>21/tcp</td>
<td>open</td>
<td>ftp</td>
</tr>
<tr>
<td>23/tcp</td>
<td>open</td>
<td>telnet</td>
</tr>
<tr>
<td>37/tcp</td>
<td>open</td>
<td>time</td>
</tr>
<tr>
<td>79/tcp</td>
<td>open</td>
<td>finger</td>
</tr>
<tr>
<td>111/tcp</td>
<td>open</td>
<td>sunrpc</td>
</tr>
<tr>
<td>512/tcp</td>
<td>open</td>
<td>exec</td>
</tr>
<tr>
<td>513/tcp</td>
<td>open</td>
<td>login</td>
</tr>
<tr>
<td>514/tcp</td>
<td>open</td>
<td>shell</td>
</tr>
<tr>
<td>515/tcp</td>
<td>open</td>
<td>printer</td>
</tr>
<tr>
<td>32771/tcp</td>
<td>open</td>
<td>sometimes-rpc5</td>
</tr>
<tr>
<td>32779/tcp</td>
<td>open</td>
<td>sometimes-rpc21</td>
</tr>
</tbody>
</table>
Remote operating system guess: Solaris 8 early access beta through actual release
Uptime 15.757 days (since Tue May 20 15:35:09 2003)

Nmap run completed -- 1 IP address (1 host up) scanned in 54 seconds
Not to Sound Paranoid, But...

How do you know that your copies of netstat, lsof, top, Process Viewer, and other apps have not been hacked so that they hide the ports and processes that the hacker is using?

Programs like **Tripwire** can be installed to note changes in your system files/applications and notify you of any changes.

http://sourceforge.net/projects/tripwire

Brian Hill has written **CheckMate** that implements a tripwire-like algorithm for Mac OS X.

http://personalpages.tds.net/~brian_hill
CheckMate installs as a System Preference

[Image of the CheckMate interface showing schedule options for manual, at startup, every hour, every day, every week, and every month. There are options for notification when files are found, with checkboxes for showing alert dialog, logging to system.log, and sending email. The mail will be sent to security@geology.ucdavis.edu.]
CheckMate

Export a copy of your files and checksums and put this somewhere other than the computer you've scanned.
CheckMate

You can do manual scans as well as automated ones.
Eavesdropping and Snooping

No matter how tightly secured your computer configuration may be, once your information begins traveling over the network it’s vulnerable to eavesdropping. A bad guy doesn’t need to attack your computer directly to gain access to your sensitive information. No talent, programming, or networking experience is required to eavesdrop on a network. The software works like this:

1. You connect to the network
2. You start it.
3. You tell it to record passwords or packets.
4. You take your recorded passwords and go.
Shared, Switched, or Wireless
All Networks Are Vulnerable!

Session hijacking was previously very difficult. **Ettercap** can be used to implement a man-in-the-middle attack without its user having any knowledge of networking or spoofing. Ettercap can sniff packets between any two hosts on a switched LAN. It can log passwords for TELNET, FTP, POP, SMB, SSH1, HTTP, X11, IMAP4, SNMP, IRC, SOCKS5, ICQ…

Beginning with Mac OS X Server 10.2, AppleShare connections can be automatically tunnelled via ssh, providing full, end-to-end, encryption of the traffic.
Enabling SSH connections on server
Connecting to the File Server

Connect to the file server "Geoshare" as:

- Guest
- Registered User

Name: EPaul
Password:

Options...  Cancel  Connect
Default Connection

Connecting to "Geoshare" using:
Two-Way Encrypted Password

Preferences

- Add Password to Keychain
- Allow Clear Text Password
- Warn when sending password in Clear Text
- Allow Secure Connections using SSH

Save Preferences

Change Password... Cancel OK
Preferred Connection
Secure AFP Connection
If you succeed in making a secure connection you should briefly see the following dialog box after you click OK and before you are presented with the list of mountable volumes on the remote AFP server.
For client-to-client connections you can create your own SSH tunnels manually.

Assuming the remote AFP server is running SSH and you have a username and password for the server, you can tunnel AppleShare traffic from port 548 on the server to port 10548 on your local machine using ssh.

```
epaul% ssh <username@afp-server> -L 10548:127.0.0.1:548
```
Encrypted AppleShare

Then use the Connect to Server… in the Finder to securely access your files and information using afp://127.0.0.1:10548
Use Vapor to Create Tunnels

Vapor can be used to create tunnels for a variety of protocols without using the command line.
http://www.afp548.com/Software/Vapor
Resources

Mac OS X Maximum Security

John Ray & William C. Ray
Sams, 1st Edition

Practical Unix & Internet Security

Simon Garfinkel, Gene Spafford & Alan Schwartz
O’Reilly, 3rd Edition

“DD’s Ultimate Guide to Mac OS X Security”

Contact Information

Rick Hill
rrhill@ucdavis.edu

Jackie Simmons
jdsimmons@ucdavis.edu

Paul Waterstraat
waterstraat@geology.ucdavis.edu
computing.geology.ucdavis.edu/security/macosx