Securing Mac OS X

Paul Day, pd(at)csse.uwa.edu.au

“Keep others out - With Mac OS X, you may never need to worry about security again.”

Top 10 reasons to upgrade Apple web-site
Introduction

Background

• Mac OS X made a major transition from Classic to X.
  - Introduced Unix in the form of FreeBSD, NeXT and the Mach/Darwin Kernel
• One of the more secure Unix installations by default, but still plenty of drawbacks.
Overview

Aims of this talk:
• Discuss methods of hardening OS X from a:
  - Local user perspective
  - Network perspective
• Point out vulnerabilities in recommendations and existing technologies.

Similar work - why bother?

• Tried not to let paranoia result in recommendations with little/no benefit but large inconvenience.
• Included recommendations, and discussed vulnerabilities others have over-looked.
• Simplified and reduced background and semi-relevant information.
Local Security

Local security?

Methods to harden security within Mac OS X from a local user perspective:

- With local physical access to the machine via its console, OR
- With interactive local access to the machine via methods such as Secure Shell (SSH) or Apple Remote Desktop (ARD).
Login Window

- Enable it
- Disable auto-login
- Insert login window message
- Change your password

Enable it and disable auto-login

- Uncheck “Automatically log in as:” in the Accounts System Preferences pane.
- Disabling Fast User Switching if not used (recent/current vulnerability in it)
Login Window
Enable it and disable auto-login

- “Disable automatic login” in the Security System Preferences Pane

Login Window
Insert a message

/Library/Preferences/com.apple.loginwindow.plist

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple Computer//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
  <key>DisableConsoleAccess</key>
  <true/>
  <key>LoginwindowText</key>
  <string>Authorized users only.</string>
</dict>
Login Window

Change your password

- Use the Accounts system preferences pane

Screensaver

- Enable the screensaver
- Require a password to exit it
- Enable active corners
**Screensaver**

**Enable it and lock it**

- Enable it in the Screensaver preferences pane.
- Turn on “Require a password” to exit the screensaver in the Security pane.

**Screensaver**

**Active corners**

- “Hot Corners” in Screensaver pane
- Use it to instantly lock the machine with a mouse movement
Keychain

- Have it automatically lock access after a time-out.
- Change its password so it’s not the same as your login password.
- Recognise security drawbacks of Keychain.

Keychain

Automatically lock

Load Keychain -> Edit -> Change Keychain Settings
Keychain
Change the password

Edit -> Change Password for Keychain

Keychain
Insecurities

- Keychain does not use mlock()
  - Memory can be swapped out of physical memory, and often is
  - Memory can include your passwords
  - Mac OS X does not use encrypted swap

- The result: an attacker could get your Keychain password by stealing your machine or gaining root access.

- The solution: memorise your passwords.
Patching

- Enable Apple Software Update
- Update Fink and Darwin Ports from cron
- Enable any third party software update features

Patching

Apple Software Update

- Use the Software Updates preferences pane.
- Choose check for updates and daily.
Patching Apple Software Update

- Can also be run from the command line or cron with:
  `/usr/sbin/softwareupdate -ia`
- and scheduled to run with:
  `/usr/sbin/softwareupdate -schedule on`

Patching - Fink

```
/sw/bin/fink -y selfupdate
/sw/bin/fink -y selfupdate-cvs
/sw/bin/fink -y update-all
/sw/bin/fink -y scanpackages
/sw/bin/fink -y index
/sw/bin/fink -y cleanup
/sw/bin/apt-get -y update
/sw/bin/apt-get -y install fink
/sw/bin/apt-get -y upgrade
/sw/bin/apt-get -y dist-upgrade
/sw/bin/apt-get -y clean
/sw/bin/apt-get -y autoclean
/sw/bin/apt-get -y check
```
Patching - Darwin Ports

• A little pointless as there’s no port upgrade.

# Change this location to whatever your ports build directory is
cd /opt/darwinports/dports/
/opt/local/bin/port_clean
/opt/local/bin/portindex
# Again, change this to be your ports build directory
cd /opt/darwinports/base
cvs -z3 update -dP
./configure
make clean && nice make
make install
# Port upgrade isn’t implemented in Darwin Ports yet...
# But here it is in case it gets put in one day
/opt/local/bin/port upgrade

Patching
Third party software

• If you can auto-patch it, enable it. eg, Microsoft AutoUpdate
File encryption

- Options for file encryption:
  - FileVault
  - Encrypted sparse disk images
  - OpenSSL
  - GnuPG

- Recognise security drawbacks of FileVault and Encrypted images

File encryption

FileVault

- Turn it on from Security preferences pane.
- May take a while.
File encryption

Encrypted disk images

- Create using Disk Utility
- Enable Encryption
- Choose Sparse image

File encryption - OpenSSL

- To encrypt a file using openssl and the (128bit) blowfish encryption algorithm:
  
  openssl bf -salt -in <plain file> -out <encrypted file>

- Then securely remove the original file:
  
  rm -fP <input file>

- Finally, decrypt the file back:
  
  openssl bf -d -in <encrypted file> -out <plain file>

- See paper for scripts to nicely en/decrypt entire directories.
File Encryption
Gnu Privacy Guard

- To encrypt using your GPG key:
  
gpg -r <your key’s name> --encrypt-files <filename>

- To decrypt using your GPG
  
gpg -r <your key’s name> --decrypt-files <filename.gpg > filename

- Can also:
  - Do symmetric (password-only) encryption like SSL
  - Use scripts for nice en/decryption of entire directories
  - GPG uses mlock()

File encryption
Insecurities

- SecurityAgent doesn’t use mlock() either. Passwords for FileVault and Encrypted disk images cached cleartext on disk.
- FileVault doesn’t rm -P or shred files after encrypting. May be easy to retrieve.
OpenFirmware Password

• Enable it with Apple’s GUI:
  

• Enable it from OPFW:
  
  $ <power-button>
  option-apple-o-f
  password
  <enter your password>
  setenv security-mode command

OpenFirmware password

Ways around it

• Change amount of RAM and reset PRAM three times (option-apple-p-r) to clear password.
• OPFW can’t hash the password - is stored as plain ASCII hex codes:
  
  nvram security-password
**Disable FireWire DMA**

- FireWire gives direct access to memory
  - Could be used to slurp entire memory contents to an attacker’s machine.
  - Could be used to crash screensaver
- Enabling OpenFirmware password disabled FireWire DMA
- Doesn’t appear to have much of a performance impact.

**Single User authentication**

- By default, Mac OS X drops straight to a root shell on a single-user start-up.
- Force password authentication with:
  `/etc/ttys` - change “secure” to “insecure”
- Generate a password with:
  `openssl passwd -salt <xy> <password>`
  `/etc/master.passwd` - insert password hash next to “root”
Disable Safari auto-open

- Automatically opens various files
- Untick “Open ‘safe’ files after downloading”.

Removing users

- Remove other normal users
- Ensure system accounts are disabled from interactive login
Removing users
Normal users

• Remove excess normal users from Accounts preferences pane.

Removing users
System accounts

• Remove password entries from system users using NetInfo Manager.
• Check for other system users.
• Be careful deleting anything!
**Fix file permissions**

- Use Apple’s Disk Utility to scan and fix file permission errors
- Scan for other susceptible file permissions

**Fix file permissions**

*Apple’s diskutil*

- Use the Disk Utility GUI, or
- Use the command line diskutil utility:
  ```bash
  /usr/sbin/diskutil repairPermissions
  ```
Fix file permissions
Scanning for unusual files

• To list all setuid/gid:
  ```bash
  find / -type f \( -perm -4000 -o -perm -2000 \) \-exec ls -al {} \; 2>/dev/null
  ```

• To list all world writable files:
  ```bash
  find / -type f \( -perm -2 \) \-exec ls -al {} \; 2>/dev/null
  ```

• To list all world writable directories:
  ```bash
  find / -type d \( -perm -2 \) \-exec ls -ald {} \; 2>/dev/null
  ```

• To list all un-owned files:
  ```bash
  find / -nouser -o -nogroup \-exec ls -al {} \; 2>/dev/null
  ```

Remove Classic

• If you don’t use it, remove it.
  ```bash
  rm -rf the following as root:
  /System/Library/PreferencePanes/Classic.prefPane/
  '/System/Library/Classic/'/System/Library/Core Services/Classic Startup.app/
  '/System/Library/UserTemplate/English.lproj/Desktop/Desktop (Mac OS 9)/'
  '/System Folder/
  '/Mac OS 9 Files/
  '/Applications (Mac OS 9)'```
Securing Bluetooth

- Secure Bluetooth using Bluetooth preferences pane:
  - Turn it off
  - Put in hidden/invisible mode
  - Turn on authentication
  - Turn on encryption
  - Do not auto-accept files
  - Disable file shares
  - Do not pair with unknown devices

Network Security
**Network Security?**

Methods to harden security within Mac OS X from:
- An external perspective - eg, the Internet or anything remote to your Mac connecting across a network.

**Disabling Services**

- Disable and understand services supplied by:
  - Sharing preferences pane
  - xinetd
  - /etc/hostconfig
  - SystemStarter
- Check for remaining network services
Disabling Services
Sharing

- Check sharing preferences pane for enabled services.
- Untick unused services.
- Basic description on RHS of pane.
- Detailed description on next slide.

Disabling Services
Sharing services

<table>
<thead>
<tr>
<th>Apple Service</th>
<th>Internet Service</th>
<th>Software</th>
</tr>
</thead>
<tbody>
<tr>
<td>Personal File Sharing</td>
<td>AFP(overTCP)</td>
<td>AppleFileServer</td>
</tr>
<tr>
<td>Windows Sharing</td>
<td>SMB/CIFS</td>
<td>Samba</td>
</tr>
<tr>
<td>Personal Web Sharing</td>
<td>HTTP</td>
<td>Apache</td>
</tr>
<tr>
<td>Remote Login</td>
<td>SSH</td>
<td>OpenSSH</td>
</tr>
<tr>
<td>FTP access</td>
<td>FTP</td>
<td>tnftpdd</td>
</tr>
<tr>
<td>Apple Remote Desktop</td>
<td>ARD</td>
<td>ARD Helper</td>
</tr>
<tr>
<td>Remote Apple Events</td>
<td>EPPC</td>
<td>AEServer</td>
</tr>
<tr>
<td>Printer Sharing</td>
<td>LPR/printer</td>
<td>CUPS</td>
</tr>
</tbody>
</table>
Disabling Services

`xinetd`

- Some Sharing services use `xinetd`.
- Plenty of other useless services in `xinetd`.
- Check what’s left enabled with:
  
grep disable /etc/xinetd.d/* | grep no

Disabling Services

`/etc/hostconfig`

- Many SystemStarter service scripts source `/etc/hostconfig`
- Probably only need:
  
  CUPS=-YES-
  
  NETINFOSERVER=-AUTOMATIC-
- Possibly also `NTPD`, if not - use `ntpd` in `cron`.
- See next slides for descriptions.
### Disabling Services

**hostconfig services**

<table>
<thead>
<tr>
<th>Service</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AFPSERVER</td>
<td>Apple File Serving, over TCP for “Personal File Sharing”</td>
</tr>
<tr>
<td>AUTHSERVER</td>
<td>Apple NetInfo Authentication service</td>
</tr>
<tr>
<td>AUTOMOUNT</td>
<td>Automatic mounting of NFS mount-points (not to be confused with amd)</td>
</tr>
<tr>
<td>CUPS</td>
<td>Local printing services</td>
</tr>
<tr>
<td>IPFORWARDING</td>
<td>IP routing for other clients</td>
</tr>
<tr>
<td>IPV6</td>
<td>IP version 6 protocol support</td>
</tr>
<tr>
<td>MAILSERVER</td>
<td>The postfix SMTP mail server</td>
</tr>
<tr>
<td>NETINFOSERVER</td>
<td>Bind to a NetInfo server for directory and authentication access</td>
</tr>
<tr>
<td>NFSLOCKS</td>
<td>Network File System file locking support</td>
</tr>
<tr>
<td>NISDOMAIN</td>
<td>Bind to a NIS domain server for authentication</td>
</tr>
<tr>
<td>ARDGENT</td>
<td>Apple Remote Desktop server</td>
</tr>
</tbody>
</table>

---

### Disabling Services

**hostconfig services**

<table>
<thead>
<tr>
<th>Service</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RPCSERVER</td>
<td>Remote Procedure Call support for numerous Unix services, such as NFS</td>
</tr>
<tr>
<td>TIMESYNC</td>
<td>Run NTPd to maintain constant time synchronisation</td>
</tr>
<tr>
<td>QTSSERVER</td>
<td>Apple QuickTime Streaming Server modules</td>
</tr>
<tr>
<td>WEBSERVER</td>
<td>The Apache web-server for “Personal Web Sharing”</td>
</tr>
<tr>
<td>SMBSERVER</td>
<td>Windows file sharing using Samba</td>
</tr>
<tr>
<td>DNSERVER</td>
<td>BIND DNS server</td>
</tr>
<tr>
<td>COREDUMPS</td>
<td>Writes a core dump to disk in the case of a kernel panic</td>
</tr>
<tr>
<td>VPNSERVER</td>
<td>Apple’s VPN service daemon (LT2P and PPTP)</td>
</tr>
<tr>
<td>CRASHREPORTER</td>
<td>Apple’s crash logging service</td>
</tr>
<tr>
<td>XGRIDSERVER</td>
<td>Act as a server for Apple’s grid computing software, xgrid</td>
</tr>
<tr>
<td>ARDGENT</td>
<td>Act as a client for Apple’s grid computing software, xgrid</td>
</tr>
<tr>
<td></td>
<td>Act as a server for Apple’s grid computing software, xgrid</td>
</tr>
</tbody>
</table>
Disabling Services

SystemStarter

• Check /Library/StartupItems/, /System/Library/StartupItems, /etc/mach_init.d for other services that don’t use hostconfig.

• Examples:
  - nfsiod (NFS client daemon)
  - AMD (Apple auto-mount service)

Disabling Services

Checking for any remaining services

• Look for any remaining network services with:
  /usr/sbin/lsof | grep LISTEN
Disabling directory access methods

- If you’re not using a directory service, disable it.
- If using LDAP, make sure you uncheck “Use DHCP-supplied LDAP server”.
- See next slide for description of Directory services.

Disabling directory access Descriptions

<table>
<thead>
<tr>
<th>Directory Access method</th>
<th>Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>Active Directory</td>
<td>Windows 2000 domain file sharing and authentication</td>
</tr>
<tr>
<td>AppleTalk</td>
<td>Apple’s legacy protocol for discovering file and print services</td>
</tr>
<tr>
<td>BSD Flat File and NIS</td>
<td>/etc flat files and Unix Network Information Service (NIS) or Yellow Pages (yp) directory and authentication</td>
</tr>
<tr>
<td>LDAPv3</td>
<td>LDAP directory access and authentication</td>
</tr>
<tr>
<td>NetInfo</td>
<td>Apple’s directory access and authentication</td>
</tr>
<tr>
<td>Rendezvous</td>
<td>Apple multicast protocol for file, print, chat, music and other network services</td>
</tr>
<tr>
<td>SLP</td>
<td>Service Location Protocol – open standard file and print server discovery</td>
</tr>
<tr>
<td>SMB</td>
<td>Windows workgroup file and print sharing/serving</td>
</tr>
</tbody>
</table>
Configuring a firewall

- Three methods of enabling an ipfw firewall:
  - Sharing preferences pane
  - Using a third party firewall application
  - Script and SystemStarter
- Monitor the firewall

Configuring a firewall
Sharing

- Enable a firewall from Sharing preferences pane.
- Very basic but:
  - Simple to use
  - Better than nothing
- Auto-adds 0/0 rule for enabled services
Configuring a firewall
Third party apps

- Many third party firewall apps that act as a front-end to ipfw:
  - Commercial
  - Shareware
  - Freeware
- Search the web or your favourite Mac software site.

Configuring a firewall
SystemStarter shell script

- Add a directory firewall to StartupItems
- Create StartupParameters.plist
- Create a script firewall which will run at boot
- Add to IPservices: Requires =("firewall");
- See paper for full details and an example of a firewall script.
- Plenty ipfw FAQs on the web.
Configuring a firewall Monitoring

- The final rule of the script should be something like:
  
  `ipfw add deny log all from any to any`

- This will log all packets before being dropped. The output from the firewall logging can be viewed with:
  
  `/usr/bin/tail -f /var/log/system.log | grep ipfw`

Kernel tweaking

- Add network kernel variable settings to `/etc/sysctl.conf` to:
  - Verbose firewall logging
  - Limit ICMP
  - Don’t accept or transmit ICMP redirects
  - Don’t accept source routing
  - Stop broadcast ECHO response
  - Stop other broadcast probes
  - TCP delayed ack off
  - Turn off forwarding/routing
  - Turn on strong/randomized TCP sequencing

- Details on next slide and paper.
Kernel tweaking

• Variables to set to achieve this:
  
  net.inet.ip.fw.verbose = 1
  net.inet.ip.fw.verbose_limit = 65535
  net.inet.icmp.icmplim = 1024
  net.inet.icmp.drop_redirect = 1
  net.inet.icmp.log_redirect = 1
  net.inet.ip.redirect = 0
  net.inet.ip.sourceroute = 0
  net.inet.ip.accept_sourceroute = 0
  net.inet.icmp.bmcastecho = 0
  net.inet.icmp.maskrepl = 0
  net.inet.tcp.delayed_ack = 0
  net.inet.ip.forwarding = 0
  net.inet.tcp.strict_rfc1948 = 1

Securing SSH

• Lock down sshd_config.
• Use SSH keys for authentication instead of passwords.
• Forward X11 through SSH.
• Tunnel other IP protocols through SSH rather than connecting directly.
Securing SSH
Locking down sshd

- In /etc/sshd_config:
  ```
  #Protocol 2,1 (to)
  Protocol 2
  #PermitRootLogin yes (to)
  PermitRootLogin no
  Subsystem sftp /usr/libexec/sftp-server” (to)
  #Subsystem sftp /usr/libexec/sftp-server
  ```

Securing SSH
Keys instead of a password

- Edit /etc/sshd_config:
  ```
  #PasswordAuthentication yes (to)
  PasswordAuthentication no
  ```

- Generate a key on the remote machine:
  ```
  ssh-keygen -b 4096 -t dsa -C "Key for user@host Nov 2004”
  ```

- Put ~/.ssh/id_dsa.pub from remote machine into ~/.ssh/authorized_keys on the Mac.
Securing SSH
Tunnelling X11

• In /etc/sshd_config:
  
  #X11Forwarding no
  
  (to)
  
  X11Forwarding yes

• And then, from the client machine:

  ssh –X –l username <remote Mac>

• Xauthority and DISPLAY are automatically set.

Securing SSH
Tunnelling other IP services

• You can also use SSH to tunnel an insecure service through it.

  • eg, rather than opening firewall and connecting directly to VNC server (TCP/5900), make a tunnel using SSH:

    ssh –N –L 5900:localhost:5900 <remote Mac>

  • Now connect VNC to 127.0.0.1:5900
Questions?

Paper

The paper which accompanies this talk:

Feedback, corrections and additions welcome:
- pd(at)csse.uwa.edu.au