Creating Neutered Administrators

Session M214

Macworld San Francisco 2005

Dave Pooser
Alford Media Services
ACSA 10.3

Mike Sebastian
Splash of Color
ACSA 10.3

Apple Certifications for 10.3.x

Course Names	Certification Level	Length (days)
Mac OS X Help Desk Essentials	ACHDS	3
Mac OS X Server Essentials	ACTC (with Help Desk Essentials)	4
System Administration of Mac OS X Clients	Apple Certified System Administrator	5 days each course
System Administration using Mac OS X Server		

train.apple.com • 800-848-6398

What We'll Cover

- Securing the system
  - Setting an Open Firmware password
  - Restricting single-user boot
  - Adding a hidden administrator account

And more…

- Adding capabilities to standard users
  - Editing `/etc/authorization`
  - Changing permissions
  - Editing `/etc/sudoers`

Why You Need To Know This

Murphy's Laws of System Administration:

- Everything a user CAN change, he WILL change
- Any changes will break something important
  - At the last second, so you have no time to fix it
  - As far away from you as possible
  - The first things broken will be the tools you need to fix systems remotely

And somehow, it's still YOUR fault!

Why You Need To Know This

- Apple doesn't offer granular permissions
- Two options:
  - Local administrators rule their own boxes
  - Standard users can't even change time zone
- We're looking for a middle ground…

Where this is useful

- Road warriors– laptop users need control over network, time zone, and similar
- Remote sites/branch offices– may not have IT staff on hand
- Management– sometimes the folks who sign the checks want to feel independent

Securing the system

- Setting up an Open Firmware password
  - Prevents users' changing boot device
  - Prevents booting in single user mode
  - Prevents startup in Target Disk mode
- Easily defeated; just add or remove RAM

Demo 1

- Add an Open Firmware password using Apple's Open Firmware Password 1.02
  - Get it at <http://docs.info.apple.com/article.html?artnum=120095>
  - See <http://docs.info.apple.com/article.html?artnum=106482> for details
- Bonus geek points– set at the OF prompt!

Securing the system

- Restricting single-user boot
  - By default, Command–S gives root access, no password required
    - User can edit configuration files such as `/etc/sudoers` to gain privileges
- Wouldn't you rather require a password? (Or disable single-user mode entirely?)

Demo II

- Edit `/etc/ttys` configuration file
  - (AFTER backing it up!)
  - Replace “secure” with “insecure” throughout file
  - Now can’t boot single-user mode without `/etc/master.passwd` root password– which doesn’t exist by default

Securing the system

- Do you WANT `/etc/master.passwd` to contain a root password?
  - If not, can't boot single-user mode at all
  - If so, password hash decryptable offline

Demo III

- Add `/etc/master.passwd` root password:
  - Use openssl passwd command:
    `openssl passwd -salt xx password`
    - xx= any two random characters
    - password = 8-char password (use unique!)
  - Enter result into `/etc/master.passwd`
    - Replace asterisk after “root:” in file

Securing the system

- Adding a hidden administrator account
  - A “back door” if primary admin cracked
  - Hidden to avoid user confusion
  - Can be disguised as (unused) system user to minimize chance of detection
    - e.g. mailman, cyrus or postfix users
  - Easily detected in NetInfo Manager

Demo IV

- Use NetInfo Manager to delete user “cyrus”
- Create new user “cyrus” via Accounts pane
- Edit user “cyrus” with NetInfo Manager:
  - change UID to 98; change GID to 80
  - change home to `/var/imap`
  - delete SharedDir

Demo IV

- Using Terminal:

```
sudo mv /Users/cyrus /var/imap
```

```
sudo chown -R 98 /var/imap
```

- Log out
- On login, use down arrow to select user; then Option-Enter to get to user/password entry blanks
- Log in as cyrus

Upgrading users

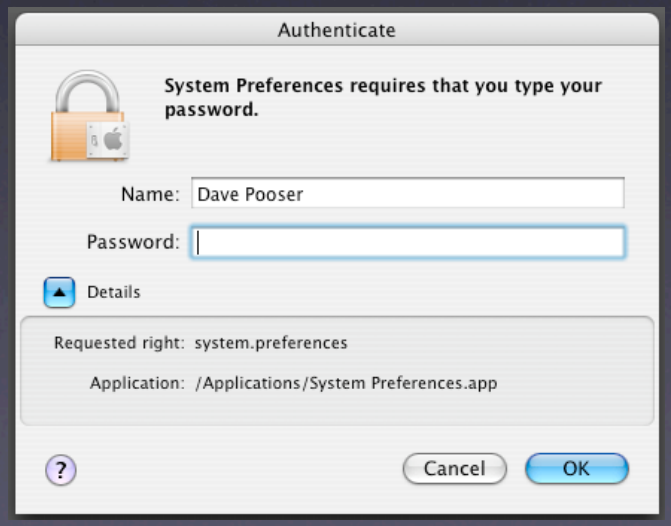
- Create a group for users who'll have some administrative rights
  - Include administrators!
- Reassign some admin group privileges to this powerusers group

Upgrading users

- `/etc/authorization` is a collection of rights and rules
  - For example, the right *system.burn* matches the rule *allow*; by default anyone can burn CDs/DVDs
  - Open `/etc/authorization` with Property List Editor (from Xcode) to view all rights and rules

Upgrading users

- To expand users, first identify the capabilities needed
- The Authenticate dialog box hides that information under Details; hit the disclosure triangle to see
  - For instance, to unlock a preference pane the requested right is system.preferences



Demo V

- Using NetInfo Manager
  - Duplicate the admin group
  - Change the name from “admin copy” to “powerusers” and the GID to any unused GID <500
  - Add the users you wish to enhance

Demo V

- Make `/etc/authorization` editable
- Open `/etc/authorization` with Property List Editor
- Find system.preferences and change the group value from “admin” to “powerusers”
- Save changes and set `/etc/authorization` permissions back to root:admin rw-r--r--

Demo V

- Log back in as enhanced user to verify access to… all system panes?
  - Including Accounts, so you can make yourself an administrator…
  - And Startup Disk, so you can boot off another drive…

"Danger, Will Robinson!"

Re-restricting users

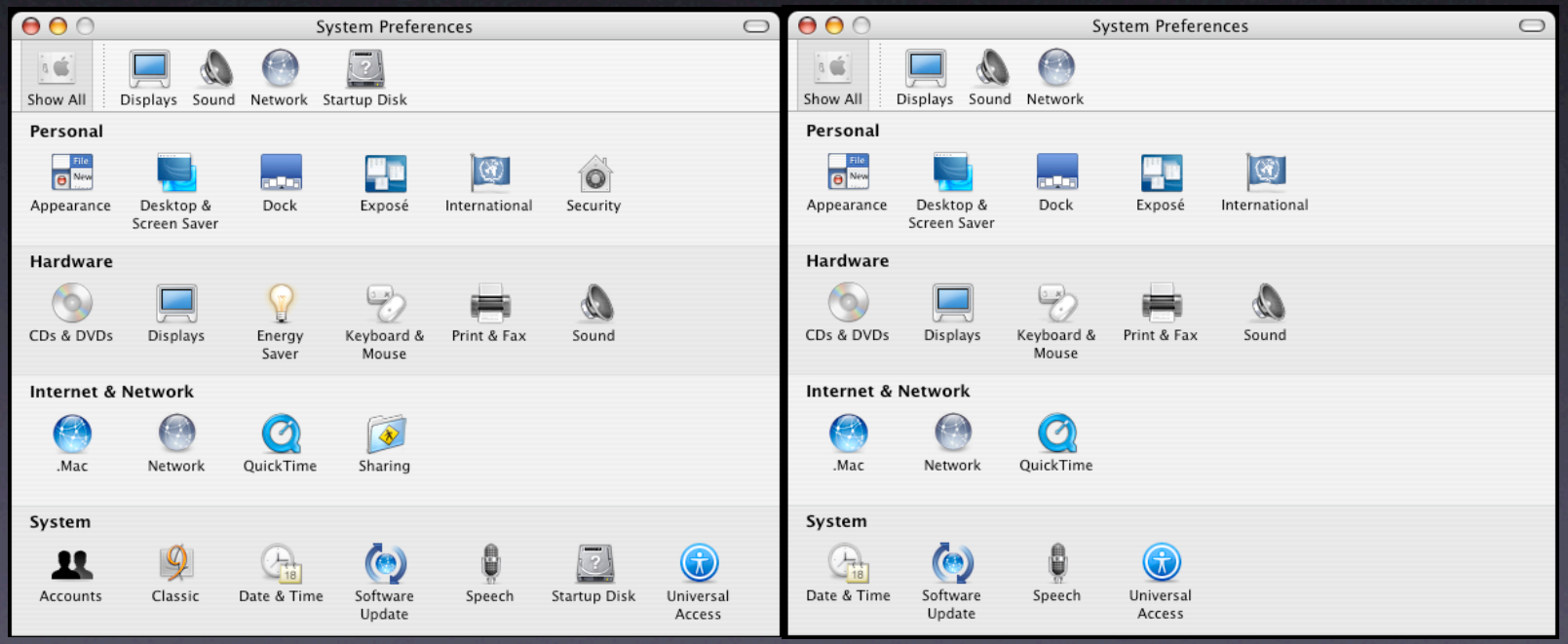
- With the system.preferences right an all or nothing change, we need another way to lock the user out of some preference panes
  - Time to change permissions!
    - But be careful, running Repair Permissions will undo all this work…
- Dangerous panes: Accounts, Classic, Energy Saver, Security, Sharing and Startup Disk

Demo VI

- Using Terminal, navigate to `/System/Library/PreferencePanes`
- Type `ls -l` to see that each pane is a directory owned by root with group wheel and permissions rwxr-xr-x
- Use `sudo chgrp -R admin` on a pane to change the group and `chmod -r o-x` to make non-admins unable to see or open the pane

Demo VI

- Log out and log in as the user to see the change…



Permissions Tweaks

- By default, /Applications admin-writeable
  - Make /Applications group “powerusers” to allow drag-and-drop installs
    - (Microsoft Office, OmniWeb…)
    - Why not ~/Applications? Possible version conflicts
    - Licensing compliance may be problem

Permissions Tweaks

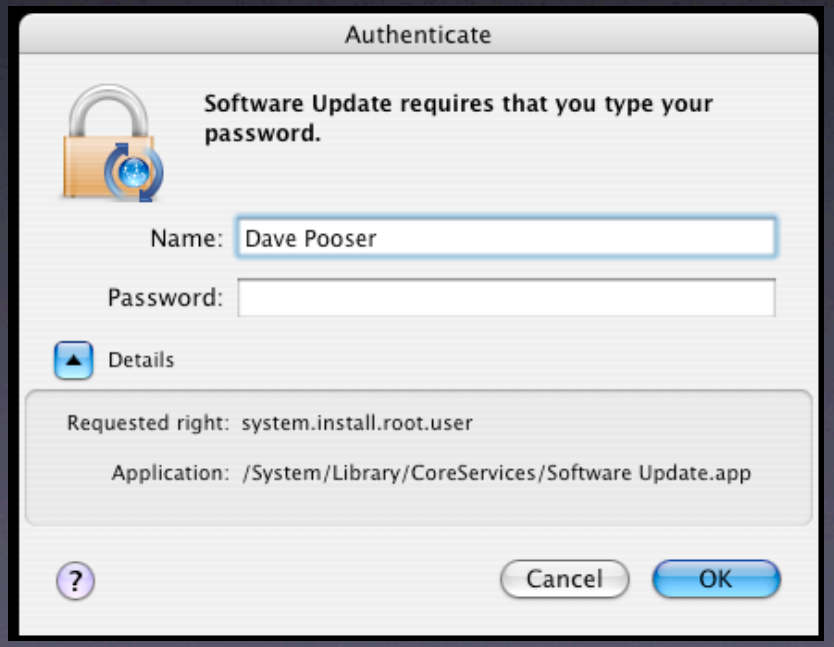
- /Library more dangerous
  - Any script in /Library/StartupItems runs as root
  - Change group on subdirectories only
    - Application Support, Fonts, Preferences
    - Create StartupItems and set ownership root:wheel

Editing `/etc/sudoers`

- /etc/sudoers is a list of users and groups allowed to run commands as root
  - Can allow some users to run any command (by default the admin group)
  - Can also allow users to run a specific list of commands…

Editing `/etc/sudoers`

- Example: You want power users to be able to run Software Update
- The Software Update GUI requires admin privileges
  - Specifically the *system.install.root.user* right…



Editing `/etc/sudoers`

- So why not edit `/etc/authorization` to give powerusers access to that right?
  - Because then they can install any package
    - As root
    - Including pre/postflight scripts

In other words, they could run any script they chose *as root*.

Editing `/etc/sudoers`

- Instead, edit `/etc/sudoers` to give the powerusers group permission to run `softwareupdate`
  - But be careful!
    - Use full path: `/usr/bin/softwareupdate`
    - Make sure the parent directory and the binary are only writeable by root

Demo VII

- Use `sudo visudo` to edit `/etc/sudoers`
  - Feel free to change your editor first: `export EDITOR=/usr/bin/pico`
- Add a line as follows:

```
%powerusers  ALL=NOPASSWD: /usr/sbin/softwareupdate
```

- Translation: Members of the group powerusers can `sudo` to run `/usr/sbin/softwareupdate` as root without a password

Demo VII

- Log out and log back in as the enhanced user
- Open Terminal and type:
  `sudo /usr/sbin/softwareupdate -i -a`
  - Translation: Run Software Update and install all updates
- Can also be created as a one-line script; make it a .command file to have a double-clickable option for Terminal-phobic users

Synopsis

- Secure the system– Open Firmware is key
- Create a powerusers group as admin-lite
- Give powerusers rights as needed
- Restrict dangerous prefpanes with `chmod`
- Use `/etc/sudoers` for specific functions

Thank You!

Creating Neutered Administrators

Session M214

Macworld San Francisco 2005

Dave Pooser
Alford Media Services
ACSA 10.3

Mike Sebastian
Splash of Color
ACSA 10.3