Configuration Management for Mac OS X: It's just Unix, Right?

David G. Pullman
Janet Bass
National Institute of Standards
And Technology (NIST)

Configuration Management

DISA
Security Technical
Implementation Guides

Getting the Configuration to the Mac

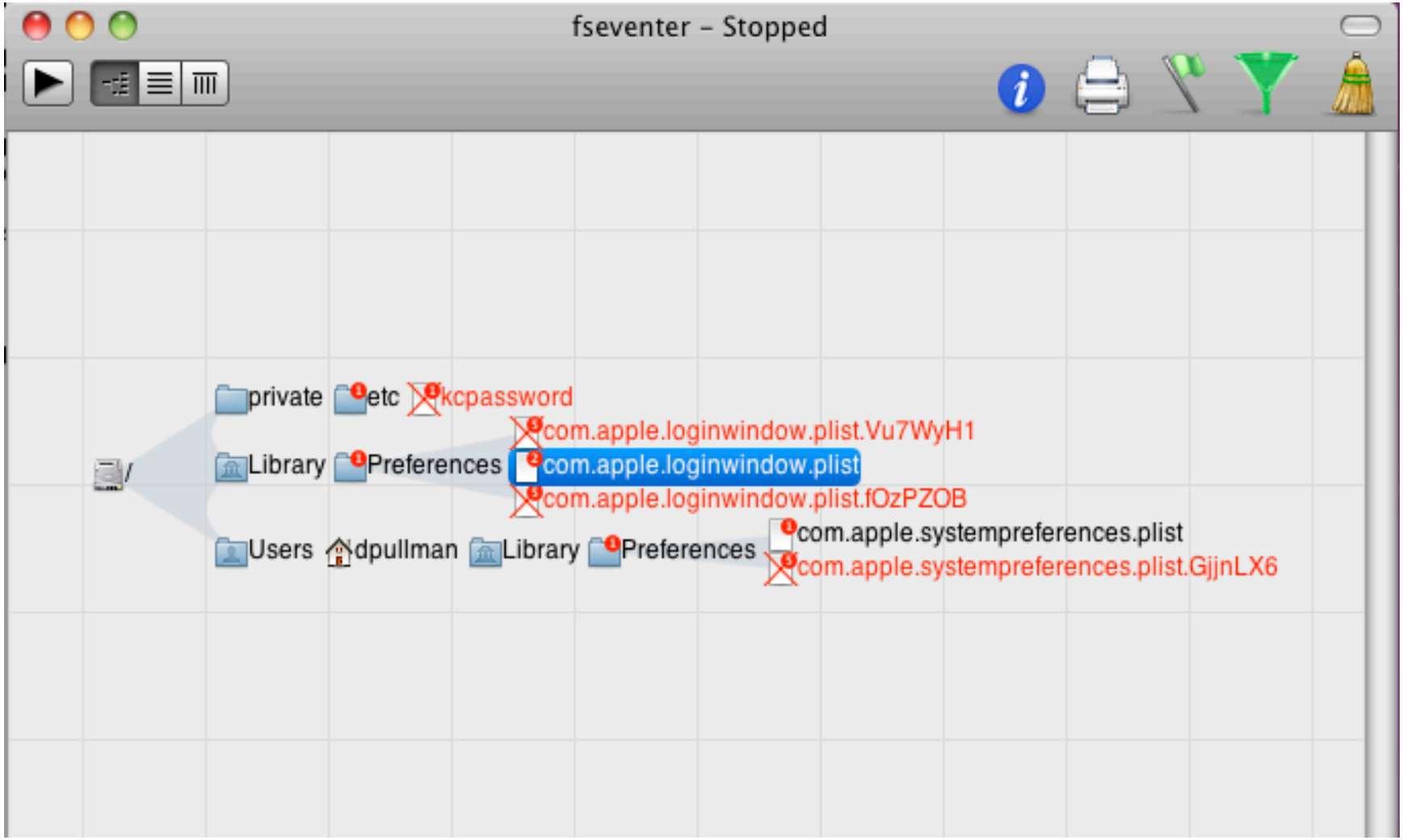
- One time script
  - Doesn't maintain the configuration
- Secure config guides and Applescripts (System Preferences)
  - Settings sometimes not effective
  - Some are per user settings
- We needed to find where the system preferences were held…

System preferences: plists

- Apple's ".conf" files
- Some are found in the config guides…

- Where are they?

We found the file!



Check it out…

```
less /Library/Preferences/com.apple.loginwindow.plist
"/Library/Preferences/com.apple.loginwindow.plist" may be a binary file.
See it anyway?
bplist00<D8>^A^B^C^D^E^F^G^H
^K^L^M^N^L^R_^P^YOptimizerLastRunForSystem
\lastUserName_^P^XOptimizerLastRunForBuild_^P^WMCXLaunchAfterUserLoginXlas
tUser_^P^UMCXLaunchOnUserLogout\SHOWFULLNAME_^P^PRetriesUntilHint^R
^F^D^@Xdpullman^R^AEG   XloggedIn<D1>^O^LXdpullman
^P^@^H^Y5B]
w<80><98><A5><B8><BD><C6><CB><CC><D5><D8><E1><E2><E3>^@^@^@^@^@^@^A^A^@^@^
@^@^@^@^@^S^@^@^@^@^@^@^@^@^@^@^@^@^@^@<E5>
/Library/Preferences/com.apple.loginwindow.plist (END)
```

…not so good

Working with plists

- Property List Editor: Nice GUI Editor…
- PlistBuddy: CLI: read and write values…
- plutil: CLI: convert format, run lint…

- defaults: command line access to the plists!

Reading the plist

```
file /Library/Preferences/com.apple.loginwindow.plist
/Library/Preferences/com.apple.loginwindow.plist: Apple binary
property list

defaults read /Library/Preferences/com.apple.loginwindow
{
    MCXLaunchAfterUserLogin = 1;
    MCXLaunchOnUserLogout =     {
        dpullman = 1;
    };
    OptimizerLastRunForBuild = 21317408;
    OptimizerLastRunForSystem = 168166400;
    RetriesUntilHint = 0;
    SHOWFULLNAME = 1;
    autoLoginUser = dpullman;
    lastUser = loggedIn;
    lastUserName = dpullman;
}
```

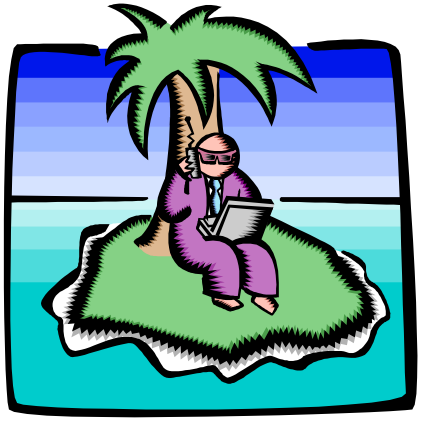
Reading the plist entries

```
defaults read /Library/Preferences/com.apple.loginwindow
autoLoginUser
dpullman

defaults delete /Library/Preferences/com.apple.loginwindow
autoLoginUser

defaults read /Library/Preferences com.apple.loginwindow
autoLoginUser
2010-11-02 19:25:08.924 defaults[5631:903]
The domain/default pair of (com.apple.loginwindow, autoLoginUser)
does not exist
```

Writing plist values…

```
defaults write /Library/Preferences/com.apple.Bluetooth
ControllerPowerState –int 0

defaults write /Users/$user/Library/Preferences/ByHost/
com.apple.ImageCaptureExtension2.$uuid shared –bool FALSE

defaults write /Library/Preferences/com.apple.loginwindow
MasterPasswordHint ''

defaults write /Library/Preferences/SystemConfiguration/
com.apple.nat NAT –dict Enabled –int 0

defaults write /private/var/db/dslocal/nodes/Default/users/root
authentication_authority –array ';DisabledUser;;ShadowHash;'
```

In our opinion, Apple programmers work on an island theory



Problems with plists

- Getting the right settings, sometimes multiple settings
- Sometimes the settings wouldn’t work
- Even if you set them, the user can just change them back

Disable Bluetooth - Linux

Example

```
service bluetooth stop
chkconfig bluetooth off
```

Disable Bluetooth - OSX

Example

```
launchctl unload –w /System/Library/
LaunchDaemon/com.apple.blued.plist

defaults write /Library/Preferences/
com.apple.Bluetooth ControllerPowerState 0

networksetup –setnetworkserviceenabled
bluetooth off

dscl /Local/MCX mcxset /Computers/localhost
com.apple.MCXBluetooth DisableBluetooth
always –bool 1
```

Prepare to Launch!

- launchctl – the interface to launchd
- loads and unloads daemons/agents
- Resource reporting and control and more

Un-Launch!

```
launchctl unload -w /System/Library/
LaunchDaemon/com.apple.blued.plist
```

Checking for a disabled launcher

```
defaults read /System/Library/
LaunchDaemons/com.apple.blued Disabled
1
```

Leopard!

Checking for a disabled launcher

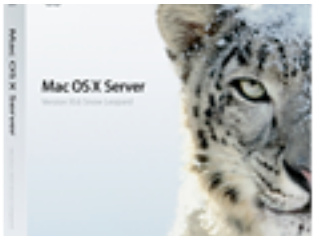
```
defaults read /var/db/launchd.db/com.apple.launchd/
overrides com.apple.blued
{
    Disabled = 1;
}
```

Snow Leopard!

Problems with lauchctl

- Similar to problems with plist…
- Awkward to check if a service is enabled or disabled
- The user can turn them back on…

Locking it down…



OSX Server
Workgroup Manager

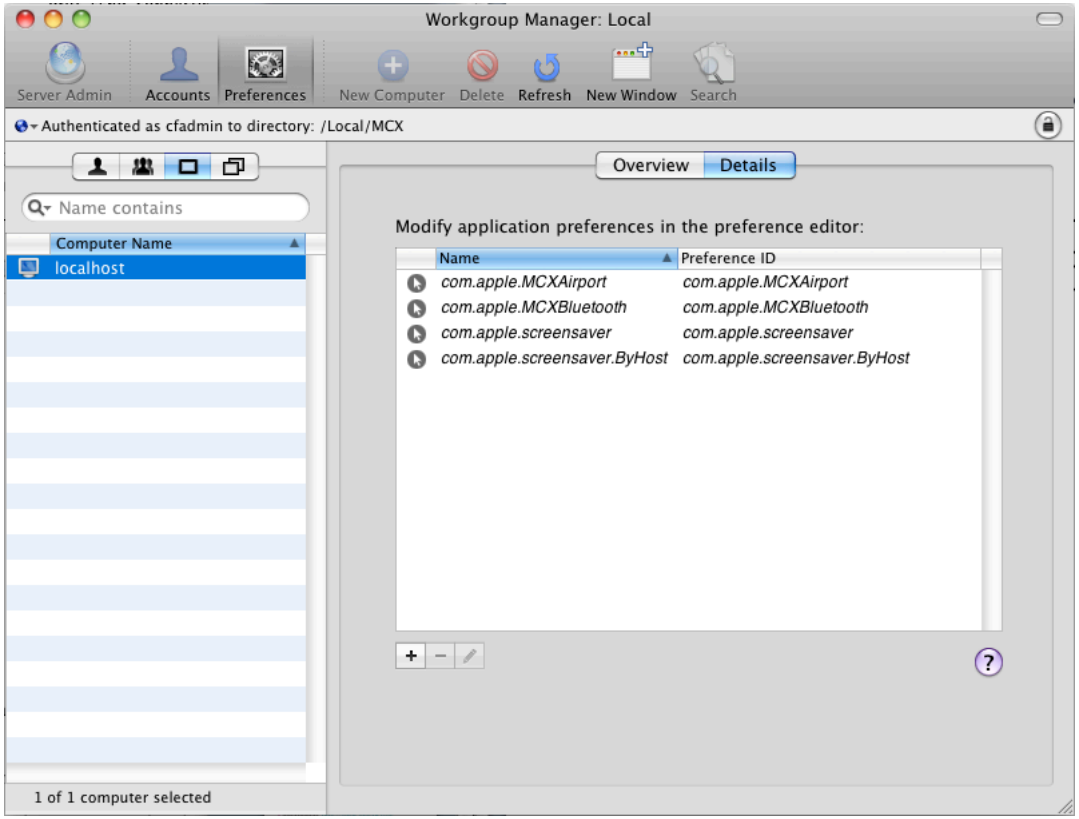
dscl: the Directory Service Command Line!

Locking it down…Locally

```
dscl /Local/MCX mcxread /Computers/localhost
com.apple.MCXBluetooth DisableBluetooth
State: always
Value: 1

dscl /Local/MCX mcxset /Computers/localhost
com.apple.MCXBluetooth DisableBluetooth always –bool 1
```

Getting there…but not very far yet

- Only some controls are available in MCX
- The same type and structure variation as plists



It's just UNIX, Right?

- As much as anything else these days!
- Plists are a common preference control…
  - once you get used to the variations!
- Launchd is a combination of init, inet, cron…
  - launchctl could use a little more functionality
- MCX and dscl provide secure configuration…
  - for the items it can control
- Cfengine
  - Metalanguage not applicable
  - Modules work (Perl!)

References

- DISA STIGs: http://iase.disa.mil/stigs/checklist
- fseventer: http://www.fernlightning.com
- Mac OS X Security Configuration Guides: http://www.apple.com/support/security/guides
- Cfengine: http://www.cfengine.org
- Information: http://www.afp548.com
- Information: http://www.mactech.com
- Greg Neagle's Blog: http://managingosx.wordpress.com
- Information: http://www.macenterprise.org

...and many more...