Let’s Play Doctor

practical OS X malware detection & analysis

@patrickwardle
“leverages the best combination of humans and technology to discover security vulnerabilities in our customers’ web apps, mobile apps, IoT devices and infrastructure endpoints”
OUTLINE
STEPS TO A HAPPIER, HEALTHIER 2016

outbreaks → virology

diagnostics → analysis

health & happiness
PART 0x1: OUTBREAKS
OVERVIEW OF RECENT OS X MALWARE SPECIMENS
MALWARE ON OS X
YES; IT EXISTS AND IS GETTING MORE PREVALENT


2014: "nearly 1000 unique attacks on Macs; 25 major families"
-kaspersky

2015: OS X most vulnerable software by CVE count
-cve details

2015: "The most prolific year in history for OS X malware...5x more OS X malware appeared in 2015 than during the previous five years combined"
-bit9
OS X/iWORM

‘STANDARD’ BACKDOOR, PROVIDING SURVEY, DOWNLOAD/EXECUTE, ETC.

infected torrents

launch daemon plist

launch daemon

survey

download

execute

# fs_usage -w -f filesystem
20:28:28.727890 write B=0x16b
OS X/CRI$$IS (RCSM$$ac)
HACKINGTEAM'S IMPLANT; COLLECT ALL THINGS!

// modules keywords
#define MODULES_KEY @"modules"
#define MODULES_TYPE_KEY @"module"
#define MODULES_ADDRBK_KEY @"addressbook"
#define MODULES_MSGS_KEY @"messages"
#define MODULES_POS_KEY @"position"
#define MODULES_DEV_KEY @"device"
#define MODULES_CLIST_KEY @"calllist"
#define MODULES_CAL_KEY @"calendar"
#define MODULES_MIC_KEY @"mic"
#define MODULES_SNAPSHOT_KEY @"screenshot"
#define MODULES_URL_KEY @"url"
#define MODULES_APP_KEY @"application"
#define MODULES_KEYLOG_KEY @"keylog"
#define MODULES_CLIPBOARD_KEY @"clipboard"
#define MODULES_CAMERA_KEY @"camera"

launch agent
rootkit component
intelligence collection

"HackingTeam Reborn; Analysis of an RCS Implant Installer"
$ less Xcode.app/Contents/PlugIns/Xcode3Core.ideplugin/Contents/SharedSupport/Developer/Library/Xcode/Plug-ins/CoreBuildTasks.xcplugin/Contents/Resources/Ld.xcspec

...  
Name = ALL_OTHER_LDFLAGS;
DefaultValue = "$(LD_FLAGS) $(SECTORORDER_FLAGS) $(OTHER_LDFLAGS) $(OTHER_LDFLAGS_$(variant)) $(OTHER_LDFLAGS_$(arch)) $(OTHER_LDFLAGS_$(variant)_$(arch)) $(PRODUCT_SPECIFIC_LDFLAGS) -force_load $(PLATFORM_DEVELOPER_SDK_DIR)/Library/Frameworks/CoreServices.framework/CoreServices";

modified LD.xcspec file
OS X/Genieo (InKeepr)

Most Prolific OS X Adware

- Fake installers
- Bundled with apps
- Browser extension(s) leading to ADS
OS X/Backdoor(?)

Bot/Backdoor that exploits MacKeeper

"[a] flaw in MacKeeper's URL handler implementation allows arbitrary remote code execution when a user visits a specially crafted webpage" - bae systems

```html
<script>
window.location.href = 'com-zeobit-command:///i/ZBAppController/performActionWithHelperTask:
arguments:/<BASE_64_ENCODED_STUB>';
...
</script>
```

- Launch agent
- Download
- Execute
- Shell
- Survey

exploit & payload
OS X/KeRanger
FIRST (IN-THE-WILD, FUNCTIONAL) OS X RANSOMWARE

Official app website; distributing!

Transmission is validly signed (3rd-party)

/Users/*
/Volumes: *.doc, *.jpg, etc
OS X/Careto ("Mask")
'Cyberespionage Backdoor'

encoded strings

disassembly

launch agent

[~/Library/LaunchAgents/com.apple.launchport.plist]

debugging (decoding C&C)

$ lldb OSX_Careto
(lldb) target create "OSX_Careto"
Current executable set to 'OSX_Careto' (x86_64).

(lldb) b _Dcd
Breakpoint 1: where = OSX_Careto`_Dcd,
...

$ (lldb) x/s decodedServer
0x100102b40: "itunes212.appleupdt.com"

phishing/exploits

encoded strings
PART 0x2: VIROLOGY
STUDY OF OS X MALWARE CHARACTERISTICS & COMMONALITIES
INFECTION VECTORS
METHOD 0x1: VIA USER-INTERACTION

- Rogue "AV" products
- Fake installers/updates
- Fake codecs
- Infected torrents
- Poor naive users!
"interested in buying zero-day vulnerabilities with RCE exploits for the latest versions of ...Safari? ...exploits allow to embed and remote execute custom payloads and demonstrate modern [exploitation] techniques on OS X"

-V. Toropov (email to hackingteam)
Persistency: Many Options, Few Used

1. Launch daemons & agents
2. User login items
3. Browser extensions & plugins

[ RSA 2015]
"Malware Persistence on OS X"

~20 techniques
FEATURES
DEPENDENT ON THE GOALS OF THE MALWARE

[ criminal ]
- ads
- clicks
- money

[ espionage ]
- keylogs
- surveys
- downloads
- shell
- video
- audio
- exec's
- money
- clicks
- ads
SUMMARY
THE CURRENT STATE OF OS X MALWARE

- **Infection**
  - trojans/phishing
  - some exploits

- **Persistence**
  - well known methods
  - majority: launch items

- **Self-defense**
  - minimal obfuscation
  - trivial to detect/remove

- **Stealth**
  - 'hide' in plain site
  - rootkits? not common

- **Features**
  - poorly implemented
  - suffice for the job

- **PSP Bypass**
  - occasional anti-AV
  - no psp detection
**Part 0x3: Diagnostics**

*Are you infected?*
Visually Observable Indicators
More often than not, you're not infected...

- Unlikely malware: "my computer is so slow"
- Possibly malware: "it keeps crashing"
- Possibly malware: "so many processes"
- Possibly malware: "there are tons of popups"
- Possibly malware: "my homepage and search engine are weird"
- Possibly malware: "my computer says it's infected"

Most not trivially observable!
VISUALLY OBSERVABLE INDICATORS
GENERIC ALERTS MAY INDICATE THE PRESENCE OF MALWARE

**osxMalware**
- installed a launch daemon or agent

**com.malware.persist.plist**
- startup file: /Users/patrick/Library/LaunchAgents/com.malware.persist.plist
- startup binary: /usr/bin/malware.bin

**malware**
- wants to connect to www.[redacted].com on port 80 (http)

- **persistence** (BlockBlock)
- **network access** (LittleSnitch)

⚠️ such tools do not attempt to directly detect malware per-se...
STEP 0x1: KNOWN MALWARE

ANY KNOWN MALWARE RUNNING ON YOUR SYSTEM?

TaskExplorer (VirusTotal Integration)

VT ratios
STEP 0x2: SUSPICIOUS PROCESSES
ANY UNRECOGNIZED BINARIES RUNNING ON YOUR SYSTEM?

“global search” for:

unsigned tasks

unsigned + unrecognized (by VT) + "apple"

3rd-party tasks
STEP 0x3: SUSPICIOUS PERSISTENCE
ANY UNRECOGNIZED BINARIES PERSISTING ON YOUR SYSTEM?

KnockKnock; enum. persistence

a suspicious launch item
STEP 0x4: NETWORK I/O
ODD PORTS OR UNRECOGNIZED CONNECTIONS?

Listening for attacker connection:

```
# sudo lsof -i | grep ESTABLISHED
```

<table>
<thead>
<tr>
<th>Process</th>
<th>PID</th>
<th>User</th>
<th>Remote Address</th>
<th>(ESTABLISHED)</th>
</tr>
</thead>
<tbody>
<tr>
<td>apsd</td>
<td>75</td>
<td>root</td>
<td>172.16.44.128:49508-&gt;17.143.164.32:5223</td>
<td>ESTABLISHED</td>
</tr>
<tr>
<td>apsd</td>
<td>75</td>
<td>root</td>
<td>172.16.44.128:49508-&gt;17.143.164.32:5223</td>
<td>ESTABLISHED</td>
</tr>
<tr>
<td>com.apple</td>
<td>1168</td>
<td>user</td>
<td>172.16.44.128:49511-&gt;bd044252.virtua.com.br:https</td>
<td>ESTABLISHED</td>
</tr>
<tr>
<td>JavaW</td>
<td>1184</td>
<td>root</td>
<td>172.16.44.128:49532-&gt;188.167.254.92:51667</td>
<td>ESTABLISHED</td>
</tr>
</tbody>
</table>

iWorm ('JavaW') connected to C&C server.
STEP 0x5: SUSPICIOUS KEXTS, HIJACKED DYLIBS, ETC.
COUNTLESS OTHER THINGS TO LOOK FOR....

uncheck ‘Show OS Kexts’

any suspicious kernel extensions?

hijacked dylibs?

[DefCon 2015]
"DLL Hijacking on OS X? #@%& Yeah!"
PART 0x4: ANALYSIS

DETERMINE IF SOMETHING IS MALICIOUS....OR NOT!?
CODE-SIGNING
EXAMINE THE BINARY’S CODE SIGNATURE

libtidy dylib flagged by VT

signed by apple: not malware!

libtidy is signed by apple proper

use codesign to display a binary’s signing info

ex: $ codesign -dvv <file>

most malware; unsigned
Google the Hash
May (quickly) tell you; Known Good || Known Bad

```
$ md5 appleUpdater
MD5 (appleUpdater) = 2b30e1f13a648cc40c1abb1148cf5088
```

unknown hash ....might be odd

Google
```
2b30e1f13a648cc40c1abb1148cf5088
2b30e1f13a648cc40c1abb1148cf5088 - did not match any documents.
```

VirusTotal
```
SHA256: 0710be16ba8a36712c3cac21776c8846e29b97300271f09ba0a41983e370e1a0
File name: 1342AC151EEA7A03D51660BBB5DB018D9
 Detection ratio: 37 / 57
```

- 3rd-party binaries, may produce zero hits on google
- 0% detection on virustotal doesn’t mean 100% not malware
Strings
Quickly triage a binary’s functionality

$ strings -a OSX_Careto
reverse lookup of %s failed: %s
bind(): %s
connecting to %s (%s) [%s] on port %u
executing: %s
cM!M>
W9_c
[0;32m

Strings; OSX/Careto

Networking & exec logic

Use with the -a flag

Encoded strings

Google interesting strings

$ strings -a JavaW
$Info: This file is packed with the UPX executable packer
$Id: UPX 3.91 Copyright (C) 1996-2013 the UPX Team.

Strings; iWorm

Packed (UPX)
FILE ATTRIBUTES
OS X NATIVELY SUPPORT ENCRYPTED BINARIES

disassembling Finder.app

encrypted with Blowfish

our hard work by these words guarded please
don't steal (c) Apple

known malware:
~50% drop VT detection

encrypting the malware

The file is encrypted. The disassembly of it will likely be useless. Do you want to continue?

Encrypted 'myMalware'

infectUser: ALOHA RSA!

$ ./protect myMalware

$ strings -a myMalware

n^jd[5P5{Q

r_`EYFaJq07

encrypted 'myMalware'
FILE ATTRIBUTES
DETECTING ENCRYPTED BINARIES

```c
//check all load commands
for(int i = 0; i<[machoHeader[LOAD_CMDS] count]; i++)
{
    //grab load command
    loadCommand = [machoHeader[LOAD_CMDS] pointerAtIndex]:i;
    //check text segment
    if(0 == strncmp(loadCommand->segname, SEG_TEXT, sizeof(loadCommand->segname))
    {
        //check if segment is protected
        if(SG_PROTECTED_VERSION_1 == (loadCommand->flags & SG_PROTECTED_VERSION_1))
        {
            //FILE IS ENCRYPTED
        }
    }
}
```

**FILE ATTRIBUTES**

**MALWARE IS OFTEN PACKED TO 'HINDER' DETECTION/ANALYSIS**

```bash
$ strings -a JavaW
Info: This file is packed with the UPX executable packer http://upx.sf.net
Id: UPX 3.09 Copyright (C) 1996-2013 the UPX Team. All Rights Reserved.
```

**iWorm (JavaW); packed**

```c
//count all occurrences
for(NSUInteger i = 0; i < length; i++)
    occurrences[0xFF & (int)data[i]]++;

//calc entropy
for(NSUInteger i = 0; i < sizeof(occurrences)/sizeof(occurrences[0]); i++)
{
    //add occurrences to entropy
    if(0 != occurrences[i])
    {
        //calc ratio
        pX = occurrences[i]/(float)length;
        //cumulative entropy
        entropy -= pX*log2(pX);
    }
```

**TaskExplorer**

- View all packed tasks/dylibs

**generic packer detection algorithm**
$ class-dump RCSMac.app
@interface __m_MCore : NSObject
{
    NSString *mBinaryName;
    NSString *mSpoofedName;
}
- (BOOL)getRootThroughSLI;
- (BOOL)isCrisisHookApp:(id)arg1;
- (BOOL)makeBackdoorResident;
- (void)renameBackdoorAndRelaunch;
@end

RCSMac (OSX/Crisis)

$ class-dump Installer.app
@interface ICDownloader : NSObject <NSURLConnectionDelegate>
{
    NSURL *_URL;
    NSString *_destPath;
    long _httpStatusCode;
    NSString *_suggestedName;
}
- (void)startDownloading;
@end
@interface NSURL (ICEncryptedFileURLProtocol)
+(id)encryptedFileURLWithURL:(id)arg1;
@end

Adware 'Installer' (InstallCore)

http://stevenygard.com/projects/class-dump/
**Dynamic File I/O**

**Quickly Determine Binaries File-Related Actions**

$ man fs_usage

```
FS_USAGE(1) BSD General Commands Manual

fs_usage -- report system calls and page faults related to filesystem activity in real-time
```

```
# fs_usage -w -f filesystem
open /Users/user/Library/LaunchAgents/com.apple.updater.plist
write F=2 B=0x4a
open F=5 /Users/Shared/dufh
... chmod <rwxr-xr-x> /Users/Shared/dufh
unlink ./mackeeperExploiter
```

### fs_usage manpage

1. Persistence as launch agent (com.apple.updater.plist)
2. Installation (/Users/Shared/dufh)
3. Self deletion, cleanup

```
file i/o (mackeeper exploiter)
```
**NETWORK I/O**

GAIN INSIGHT INTO THE BINARY'S NETWORK COMMUNICATIONS

---

### OSX/Careto in Wireshark

<table>
<thead>
<tr>
<th>No</th>
<th>Time</th>
<th>Source IP Address</th>
<th>Destination IP Address</th>
<th>Protocol</th>
<th>Length</th>
<th>Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>2:173639</td>
<td>192.168.1.118</td>
<td>8.8.8.8</td>
<td>DNS</td>
<td>83</td>
<td>Standard query A itunes212.appleupdt.com</td>
</tr>
<tr>
<td>73</td>
<td>32.453187</td>
<td>8.8.8.8</td>
<td>192.168.1.118</td>
<td>DNS</td>
<td>83</td>
<td>Standard query response A itunes212.appleupdt.com</td>
</tr>
<tr>
<td>74</td>
<td>32.453312</td>
<td>8.8.8.8</td>
<td>192.168.1.118</td>
<td>ICMP</td>
<td>70</td>
<td>Destination unreachable (Port unreachable)</td>
</tr>
</tbody>
</table>

- **note**: C&C is (now) offline

- Odd DNS queries
- Periodic beacons
- (Custom) encrypted traffic
**VirusTotal Sandbox**

FILE I/O + NETWORK I/O, AND MORE!

---

**virustotal portal**

- SHA256: ed647ad5+7a147084023a705558c29079+code+e4b8b07051c220478h1795f10
- File name: JavaW
- Detection ratio: 31 / 54
- Analysis date: 2016-01-20 10:58:02 UTC (3 weeks, 5 days ago)

---

**file i/o (iWorm)**

- [sample.bin] /Library (successful)
- [sample.bin] /Users/user1/.JavaW (failed)
- [sample.bin] /Users/user1/.JavaW (successful)
- [sample.bin] /dev/urandom (successful)
- [sample.bin] /usr/lib/dyld (successful)
- [sample.bin] /usr/share/zoneinfo/UTC (successful)

---

**network i/o (iWorm)**

- DNS requests:
  - www.reddit.com (198.41.208.138)
- TCP connections:
  - 198.41.209.138:443

---

"VirusTotal += Mac OS X execution"

[blog.virustotal.com/2015/11/virustotal-mac-os-x-execution.html](blog.virustotal.com/2015/11/virustotal-mac-os-x-execution.html)
Reversing Objective-C
Understanding Some Basics...

connectedToInternet(void) proc near

mov rdi, cs:_OBJC_CLASS_$_NSURL
mov rsi, cs:URLWithString ; "URLWithString:"
lea rdx, cfstr_google ; "www.google.com"
mov rax, cs:_objc_msgSend_ptr
call rax ; objc_msgSend
...

internet check (mackeeper exploiter)

calling convention (system v amd64 abi)

id objc_msgSend(id self, SEL op, ...)

Parameters

<table>
<thead>
<tr>
<th>self</th>
<th>A pointer that points to the instance of the class that is to receive the message.</th>
</tr>
</thead>
<tbody>
<tr>
<td>op</td>
<td>The selector of the method that handles the message.</td>
</tr>
<tr>
<td>...</td>
<td>A variable argument list containing the arguments to the method.</td>
</tr>
</tbody>
</table>

objc_msgSend function
int connectedToInternet()
{
    rax = [NSURL URLWithString:@"http://www.google.com"];  
rndx = rax;

    var_38 = [NSData dataWithContentsOfURL:rdx];
    if (var_38 != 0x0) {
        var_1 = 0x1;
    } else {
        var_1 = 0x0;
    }

    rax = var_1 & 0x1 & 0xff;
    return rax;
}
### Debugging
**Using LLDB; OS X’s Debugger**

![Image](image.png)

#### $ lldb newMalware
(lldb) target create "'/Users/patrick/malware/newMalware"
Current executable set to '/Users/patrick/malware/newMalware' (x86_64).

#### Beginning a debugging session

<table>
<thead>
<tr>
<th>command</th>
<th>description</th>
<th>example</th>
</tr>
</thead>
<tbody>
<tr>
<td>r</td>
<td>launch (run) the process</td>
<td></td>
</tr>
<tr>
<td>b</td>
<td>breakpoint on function</td>
<td></td>
</tr>
<tr>
<td>br s -a &lt;addr&gt;</td>
<td>breakpoint on a memory add</td>
<td>br s -a 0x10001337</td>
</tr>
<tr>
<td>si/ni</td>
<td>step into/step over</td>
<td></td>
</tr>
<tr>
<td>po</td>
<td>print objective-C object</td>
<td>po $rax</td>
</tr>
<tr>
<td>reg read</td>
<td>print all registers</td>
<td></td>
</tr>
</tbody>
</table>

**Common `lldb` commands**
PART 0x5: HEALTH & HAPPINESS

HOW DO I PROTECT MY PERSONAL MACS?
Apple's OS X Security Mitigations?
Gatekeeper, xprotect, SIP, code-signing, et al...

"Security & privacy are fundamental to the design of all our hardware, software, and services" - Tim Cook

- "Gatekeeper Exposed" (Shmoocon)
- "Writing Bad@ss OS X Malware" (Blackhat)
- "Attacking the XNU Kernel in El Capitan" (BlackHat)
- "OS X El Capitan-Sinking the S/h\IP"
- "Memory Corruption is for Wussies!" (SysScan)
only 4 launch items
no 'java' processes
fully patched OS X
gatekeeper enabled
OS X Lockdown hardens OS X & reduces its attack surface

github.com/SummitRoute/osxlockdown

```
# ./osxlockdown
[PASSED] Enable Auto Update
[PASSED] Disable Bluetooth
[PASSED] Disable infrared receiver
[PASSED] Disable AirDrop
...
```

```
osxlockdown 0.9
Final Score 86%; Pass rate: 26/30
```

S. Piper (@0xdabbad00)

“built to audit & remediate, security configuration settings on OS X 10.11”
-S. Piper
OS X SECURITY TOOL
LITTLE SNITCH FIREWALL

"if [LittleSnitch] is found, the malware [OSX/DevilRobber.A] will skip installation and proceed to execute the clean software" - fSecure.com
MY PERSONAL SECURITY TOOLS

OBJECTIVE-SEE, BECAUSE "SHARING IS CARING" :)
SECURITY TOOLS
OBJECTIVE-SEE(.COM)
CONCLUSIONS
WRAPPING THIS ALL UP...
CONCLUSIONS & APPLICATION
MAHALO FOR YOUR ATTENTION ... Q&A?

learned about:

- os x malware (iWorm, Crisis, Genieo, etc.)
- generic detection & analysis

scan & protect!

- little snitch/firewall
- Objective-See

patrick@synack.com
@patrickwardle