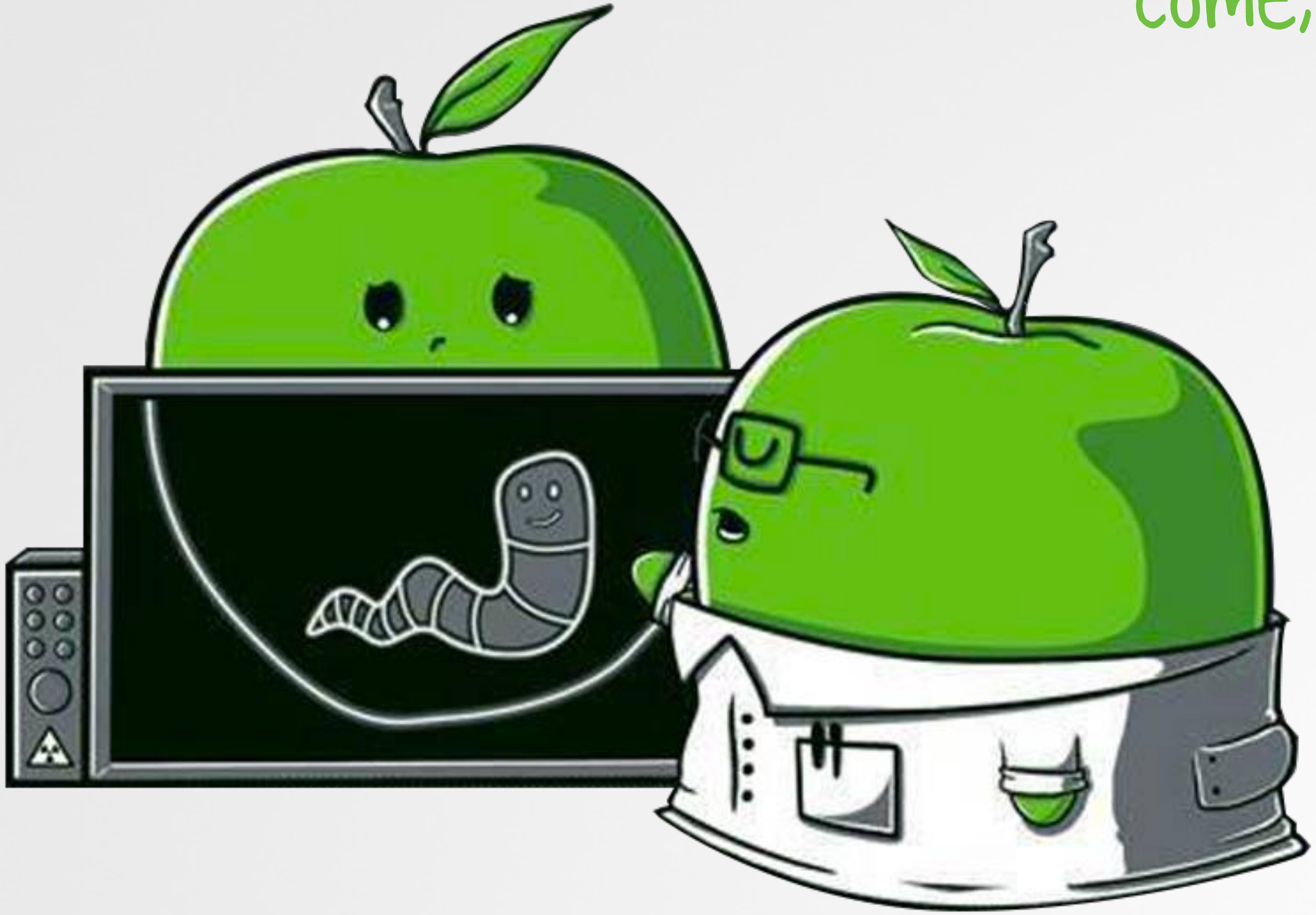
Exposing Gatekeeper

come, see, conquer!



@patrickwardle

Synack

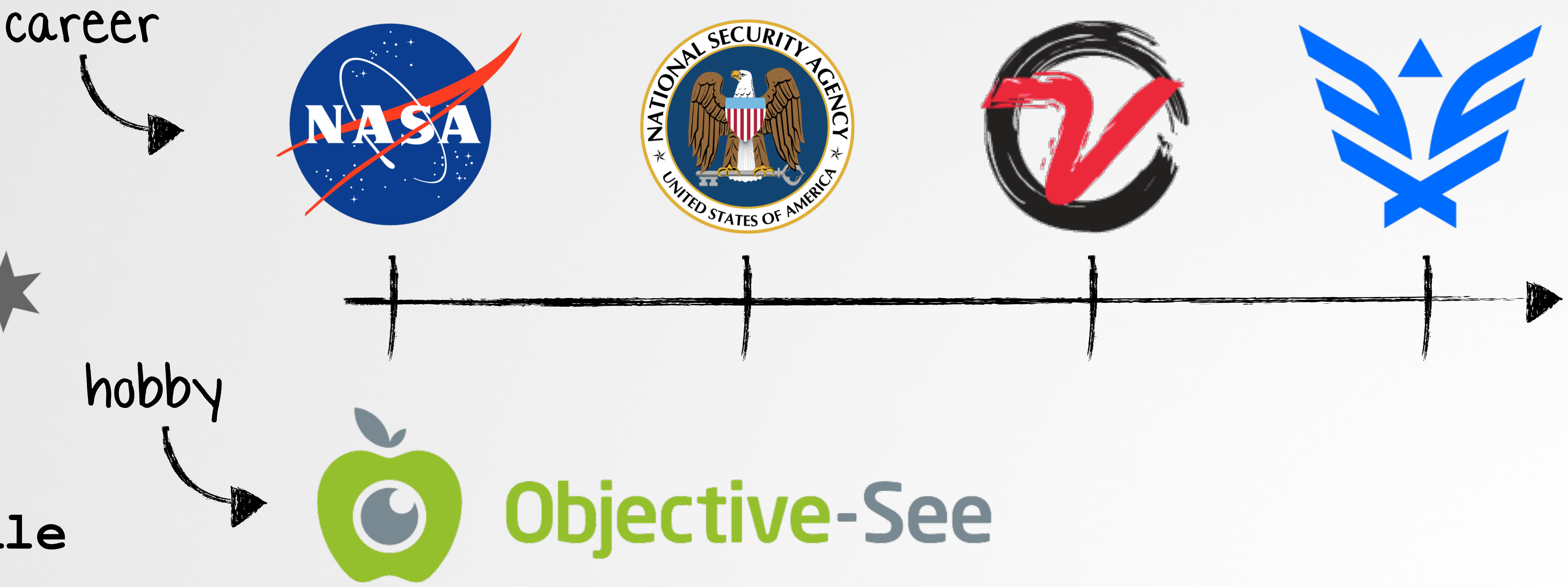
WHOIS

Synack

security for the 21st century

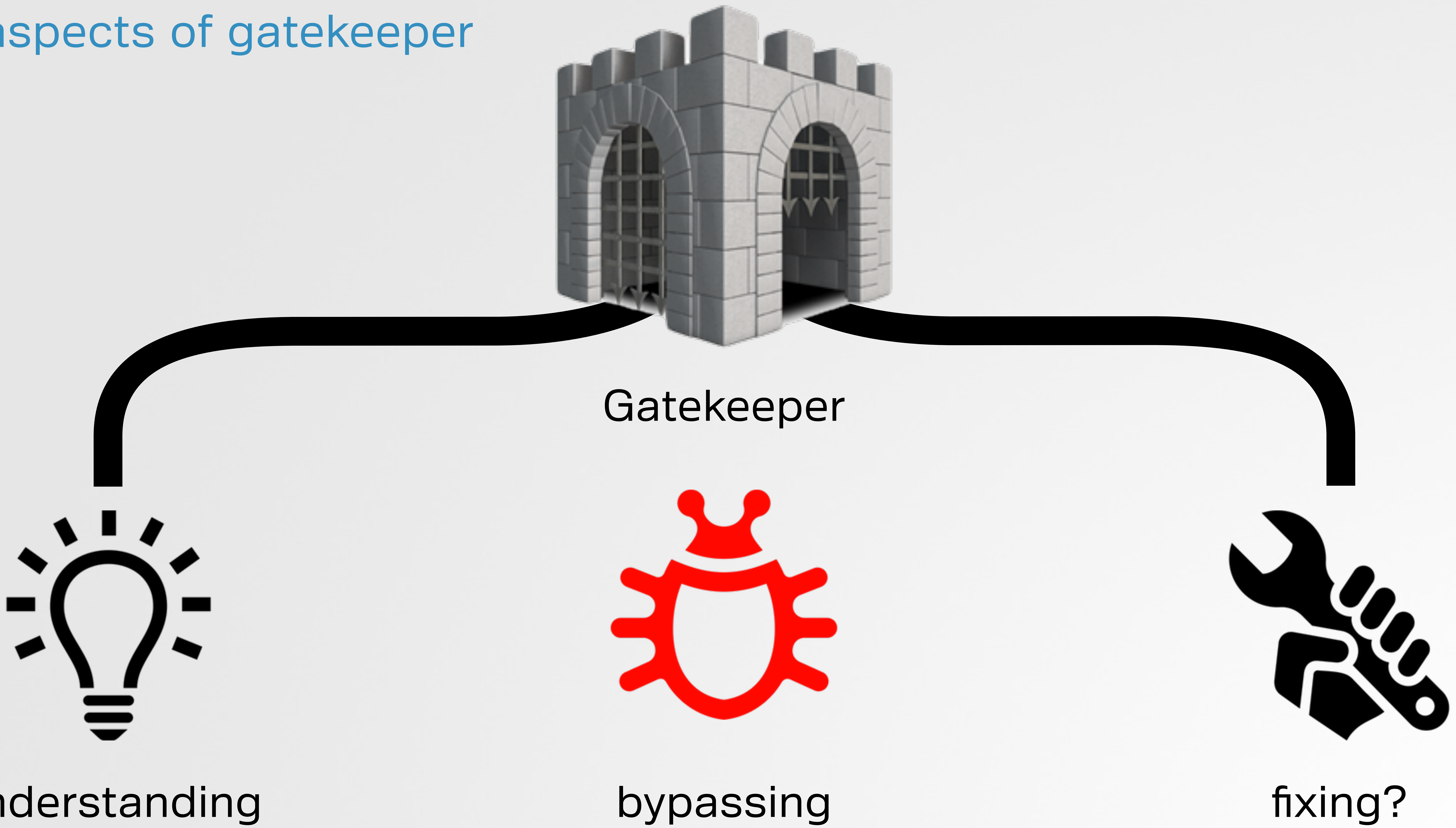
"leverages the best combination of humans and technology to discover security vulnerabilities in our customers' web apps, mobile apps, IoT devices and infrastructure endpoints"

@patrickwardle



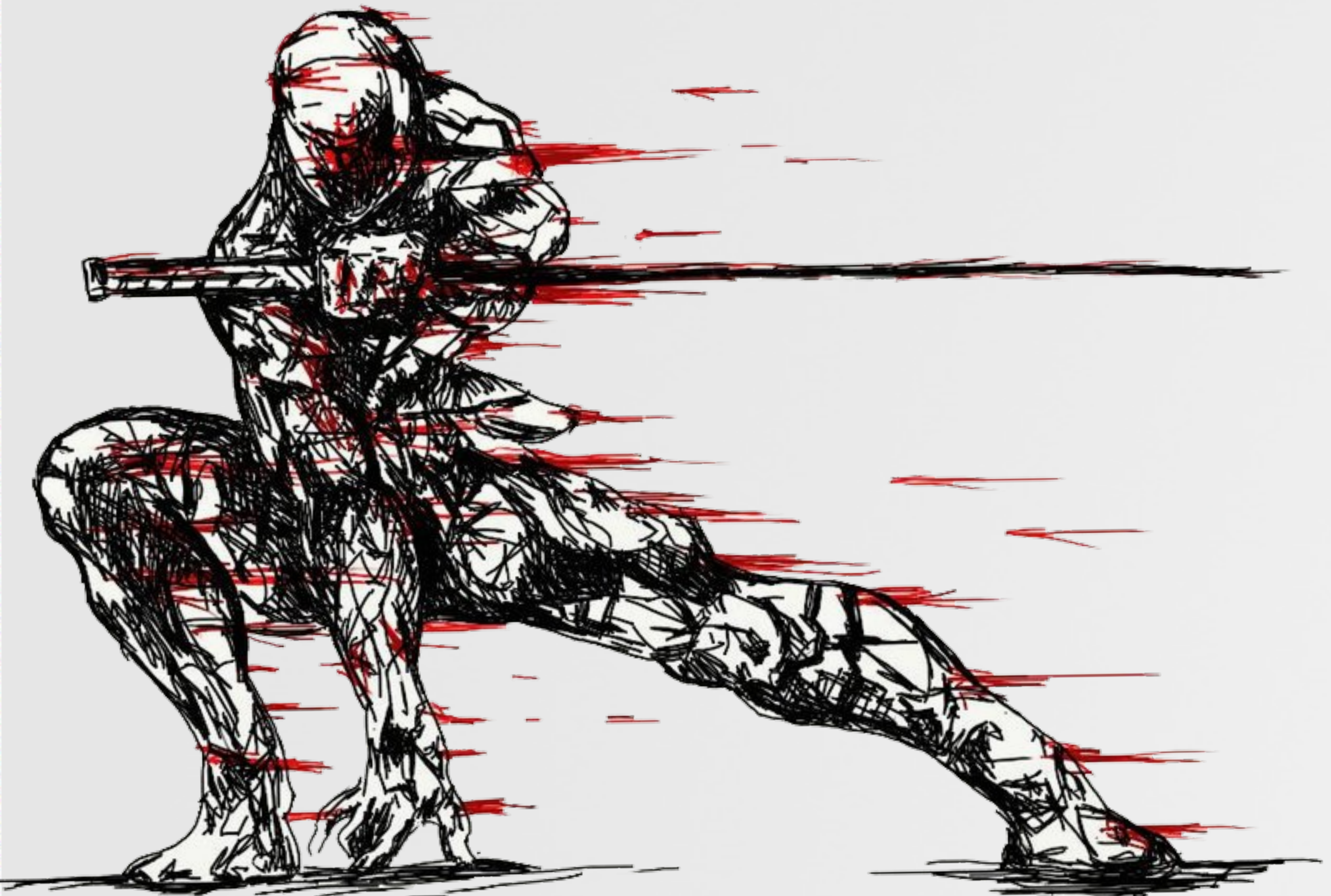
OUTLINE

all aspects of gatekeeper



UNDERSTANDING GATEKEEPER

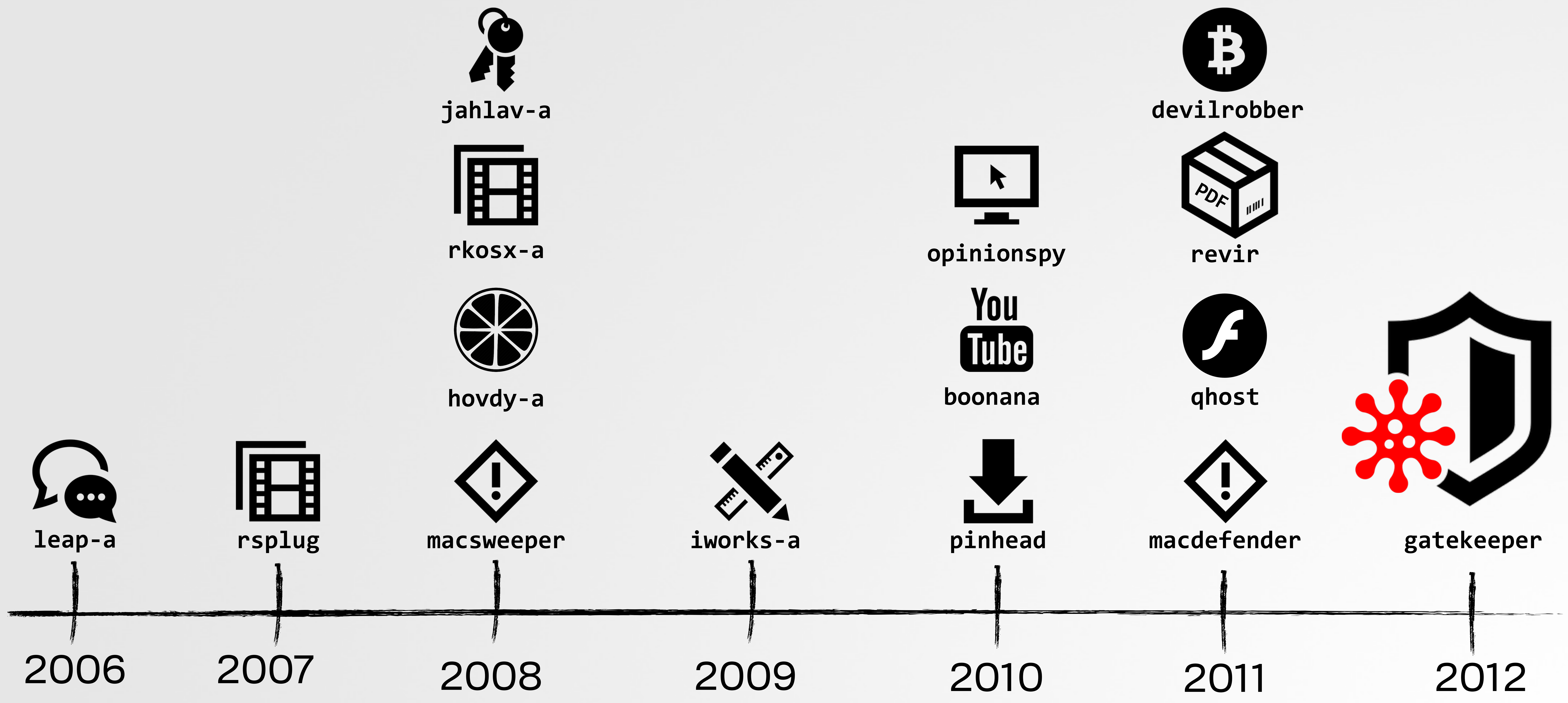
…under the hood



Life Before Gatekeeper

...os x trojans everywhere? everywhere!

countless OS X users infected

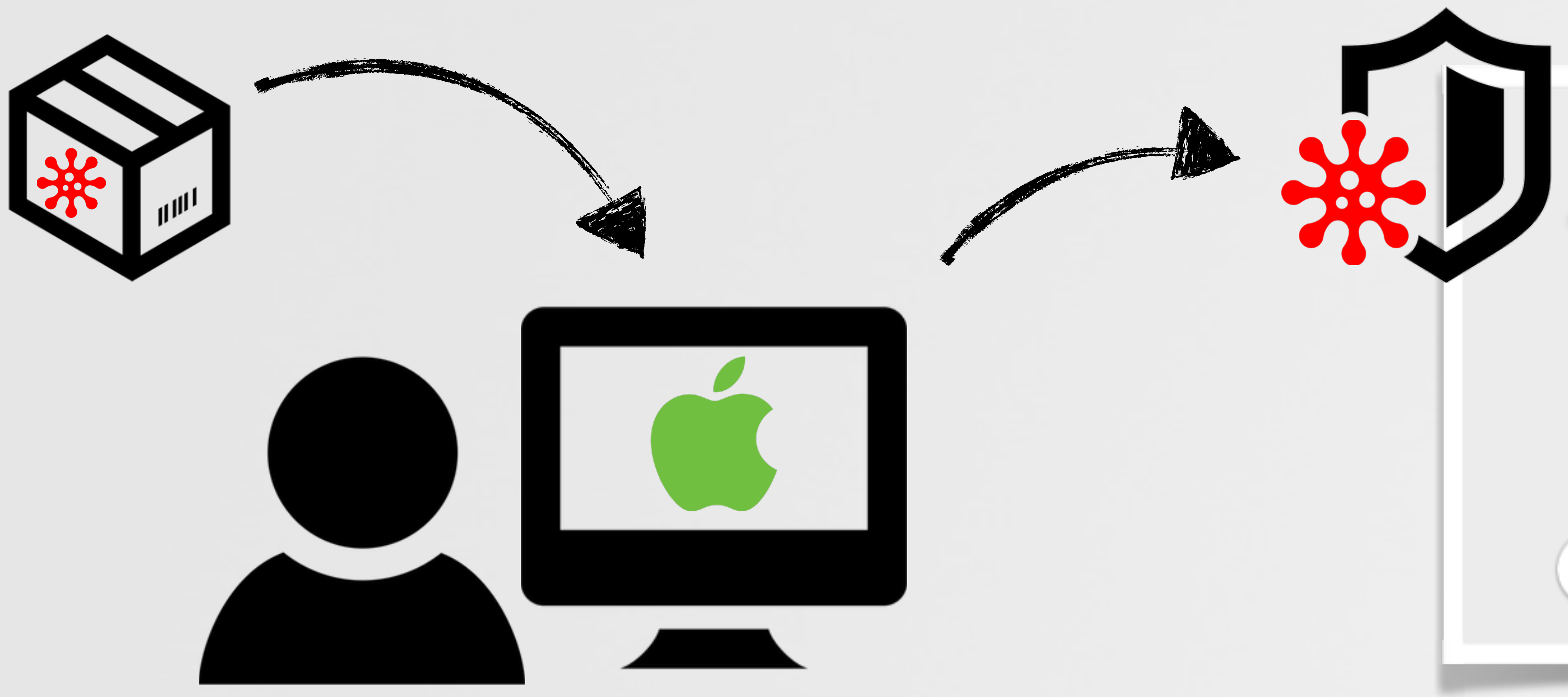


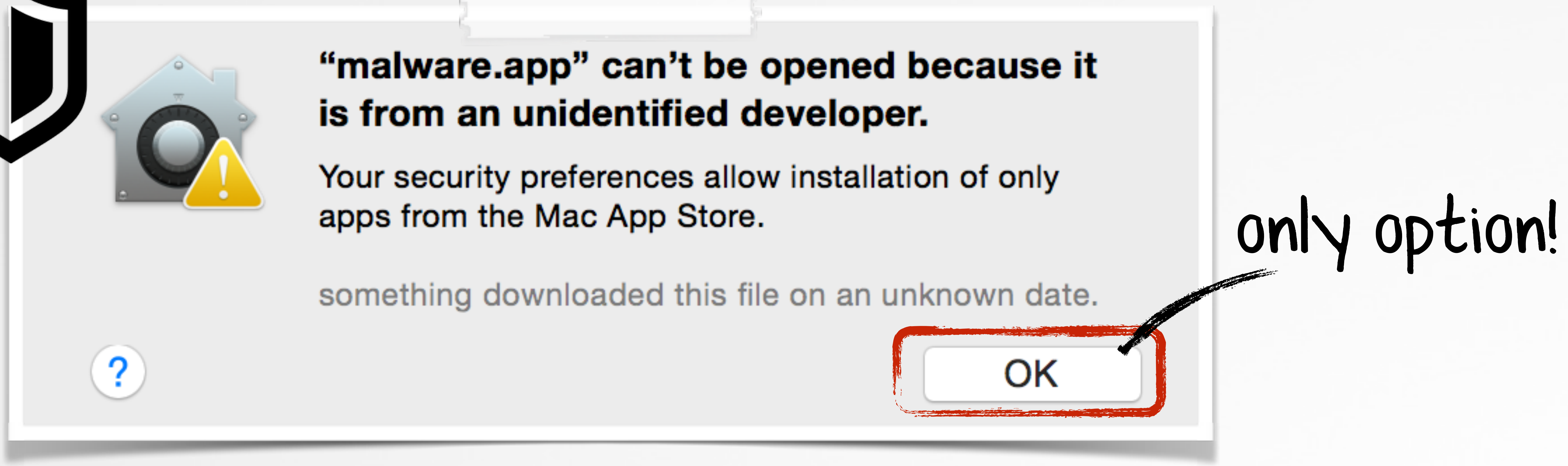
Gatekeeper Aims to Protect

as there is no patch for human stupidity ;)

Gatekeeper is a built-in anti-malware feature of OS X (10.7+)

"If a [downloaded] app was developed by an unknown developer—one with no Developer ID—or tampered with, Gatekeeper can block the app from being installed" -apple.com





"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.

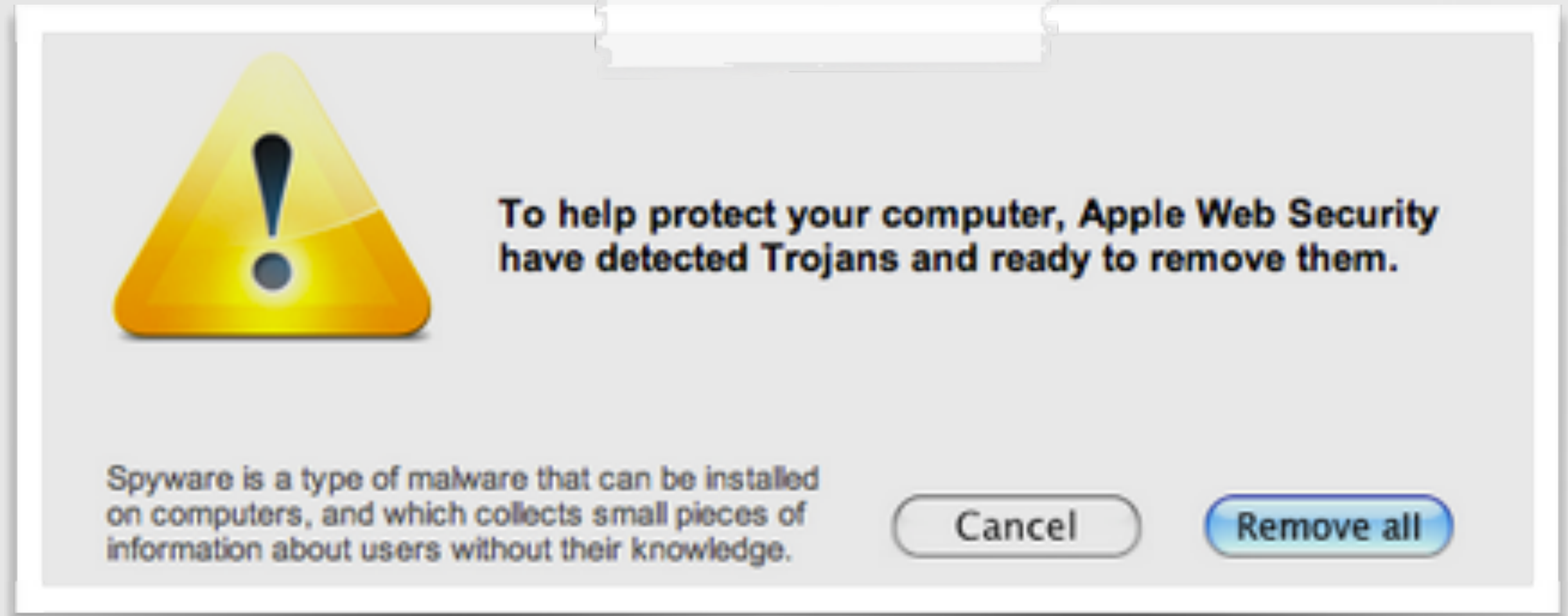
something downloaded this file on an unknown date.

→ **TL;DR block unauthorized code from the internet**

Gatekeeper Protect Users

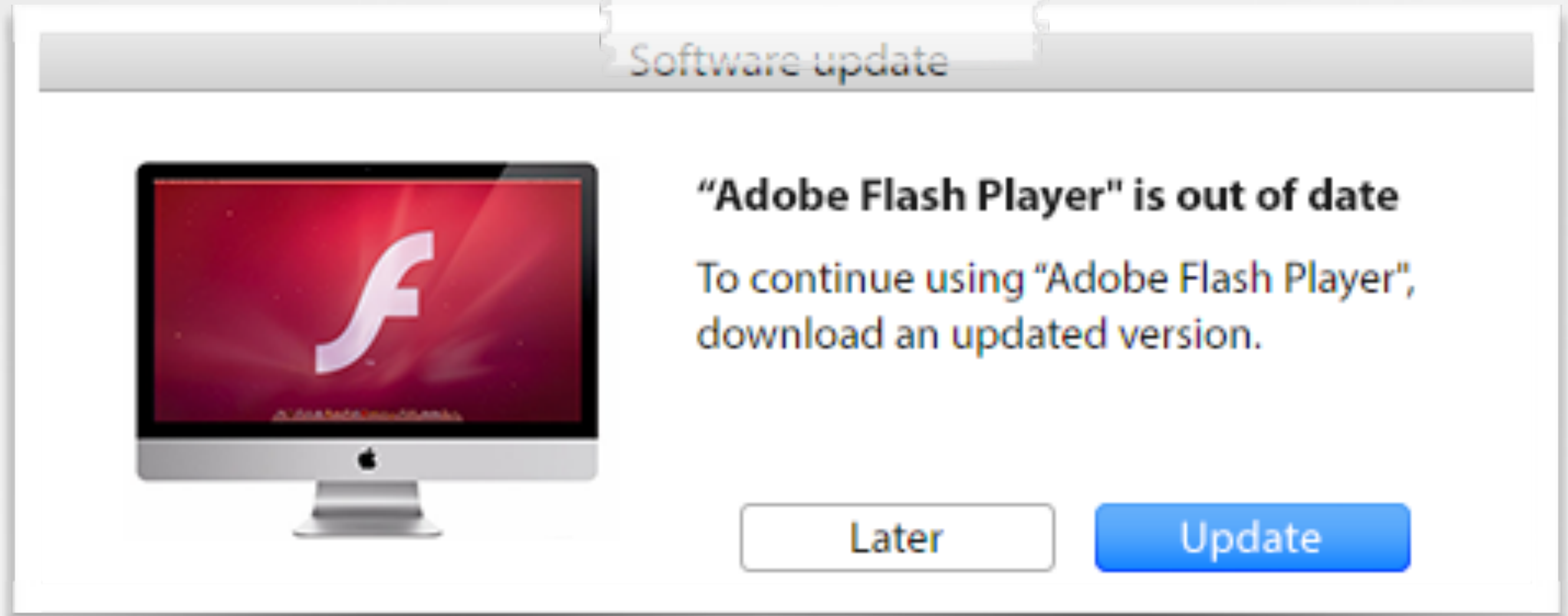
...from low-tech adversaries

"*Gatekeeper Slams the Door on Mac Malware Epidemics*" -tidbits.com


To help protect your computer, Apple Web Security have detected Trojans and ready to remove them.

Spyware is a type of malware that can be installed on computers, and which collects small pieces of information about users without their knowledge.

rogue "AV" products



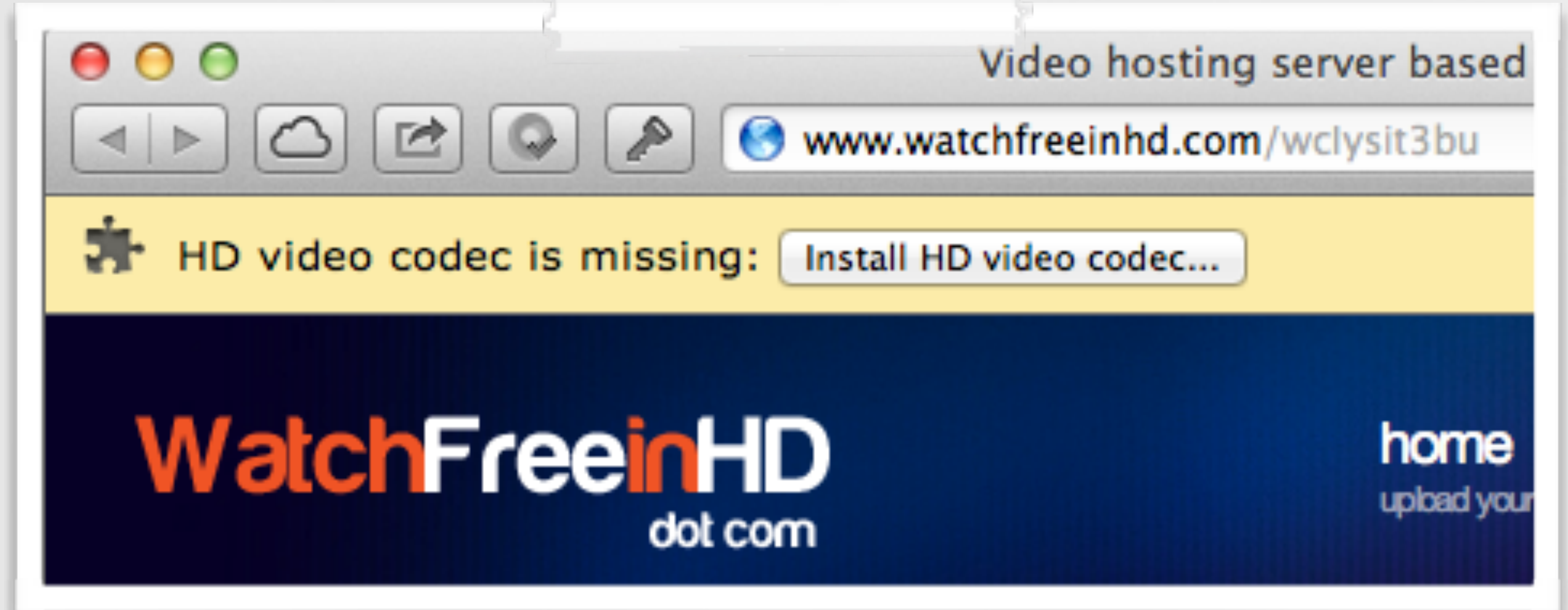
"Adobe Flash Player" is out of date

To continue using "Adobe Flash Player", download an updated version.

fake installers/updates

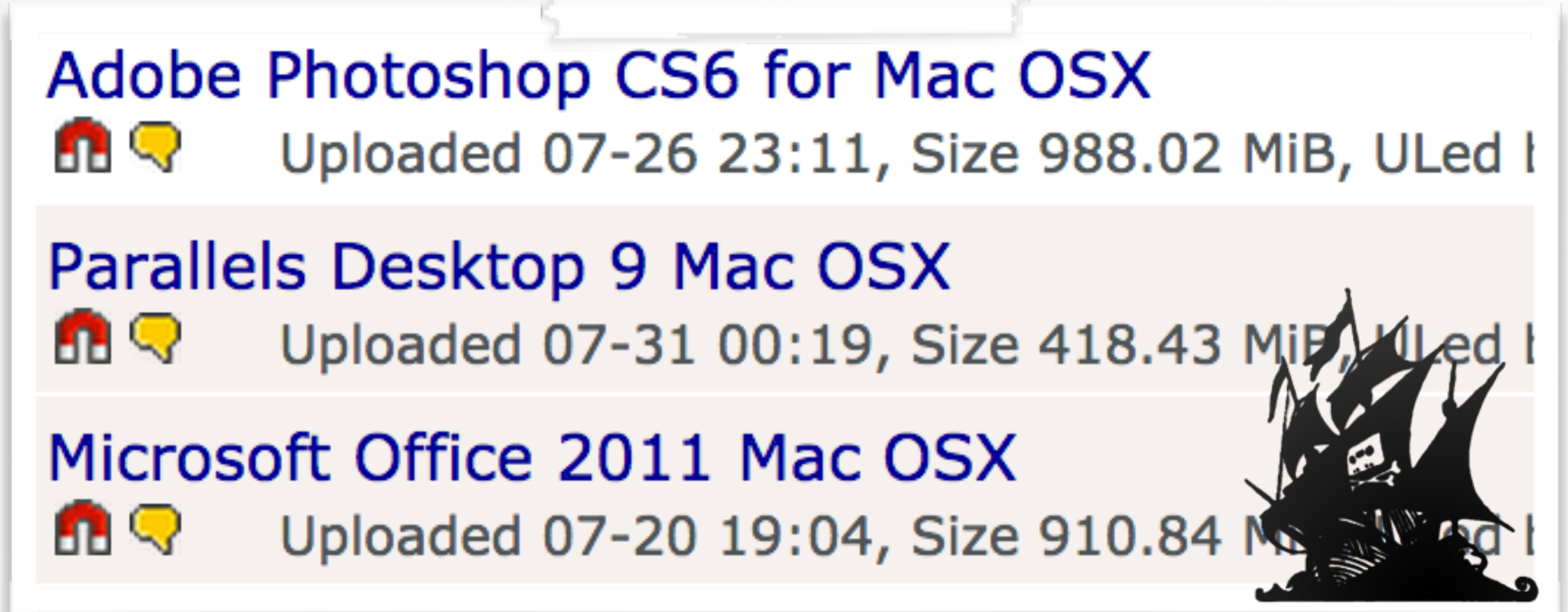
???

poor naive users!



HD video codec is missing: Install HD video codec...

fake codecs


Adobe Photoshop CS6 for Mac OSX
Uploaded 07-26 23:11, Size 988.02 MiB, ULed

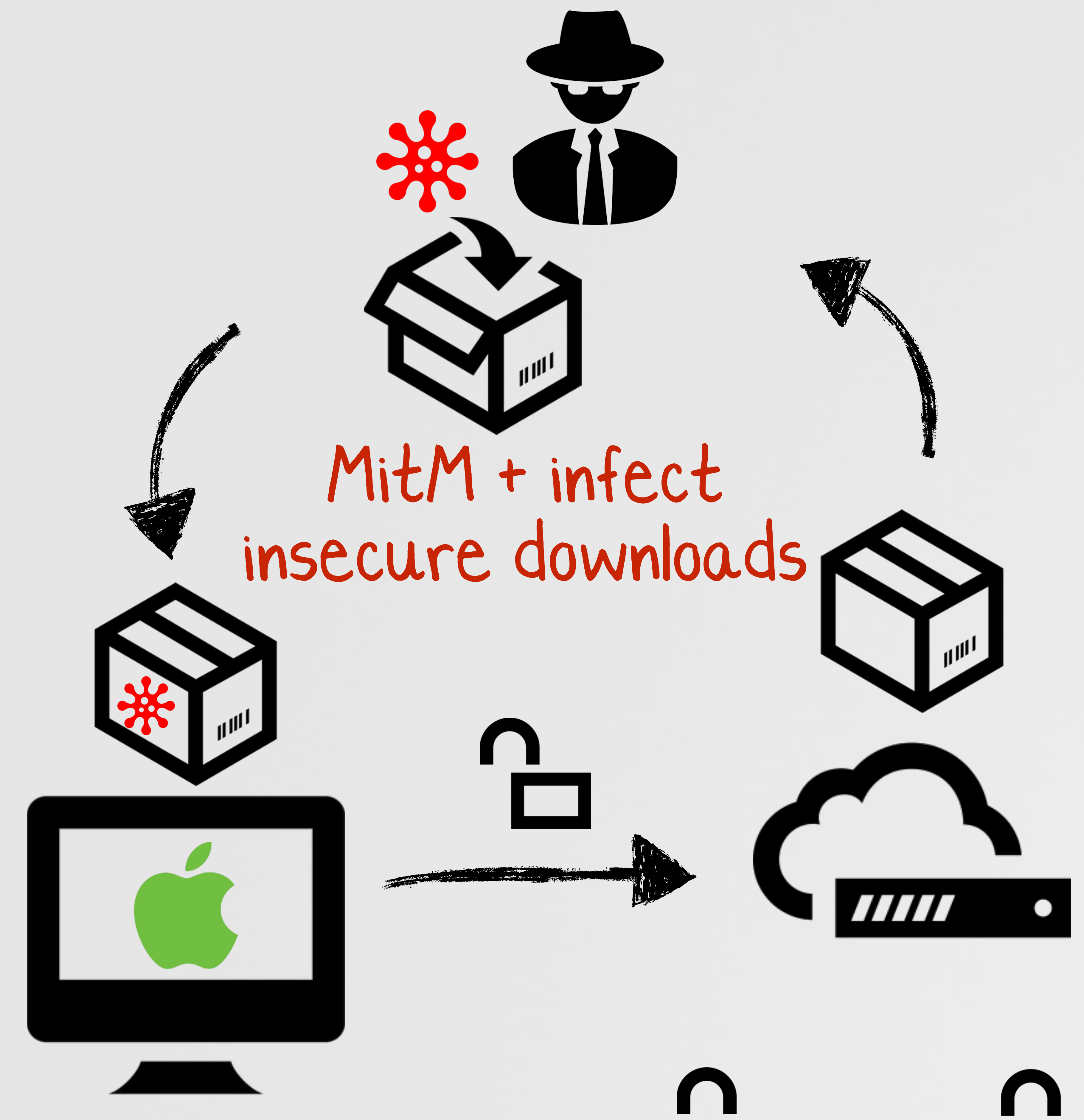
Parallels Desktop 9 Mac OSX
Uploaded 07-31 00:19, Size 418.43 MiB, ULed

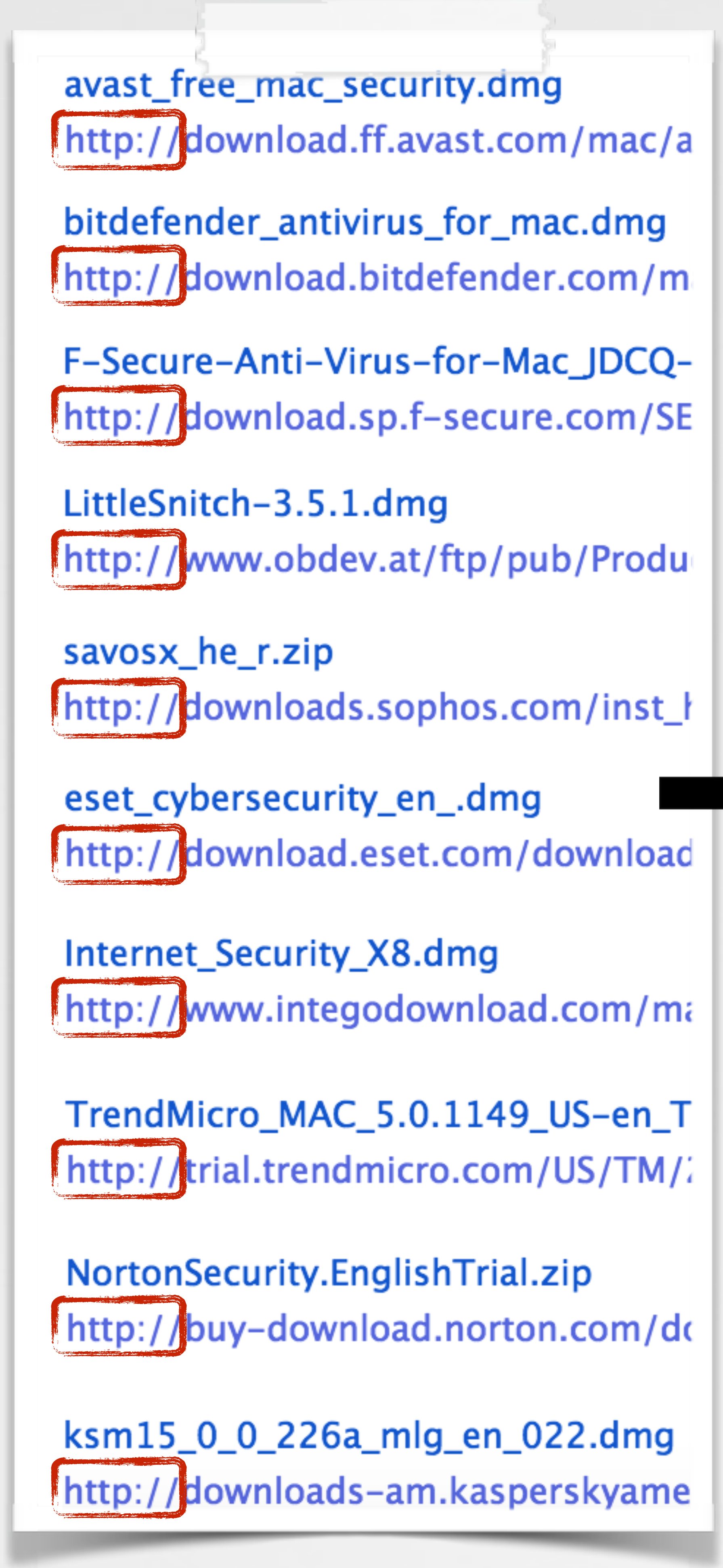
Microsoft Office 2011 Mac OSX
Uploaded 07-20 19:04, Size 910.84

infected torrents

Gatekeeper Protects Users

...from high-tech adversaries





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

bitdefender_antivirus_for_mac.dmg
http://download.bitdefender.com/m

F-Secure-Anti-Virus-for-Mac_JDCQ-
http://download.sp.f-secure.com/SE

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

savosx_he_r.zip
http://downloads.sophos.com/inst_

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

Internet_Security_X8.dmg
http://www.integodownload.com/ma

TrendMicro_MAC_5.0.1149_US-en_T
http://trial.trendmicro.com/US/TM/

NortonSecurity.EnglishTrial.zip
http://buy-download.norton.com/d

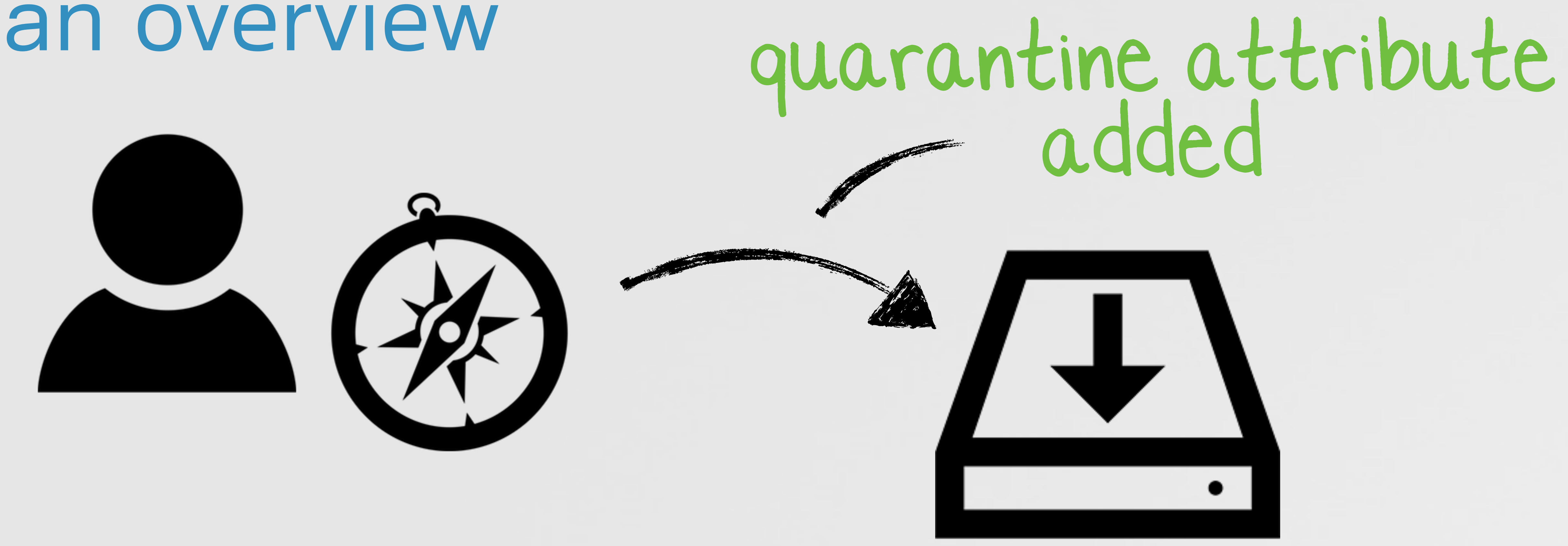
ksm15_0_0_226a_mlg_en_022.dmg
http://downloads-am.kasperskyame

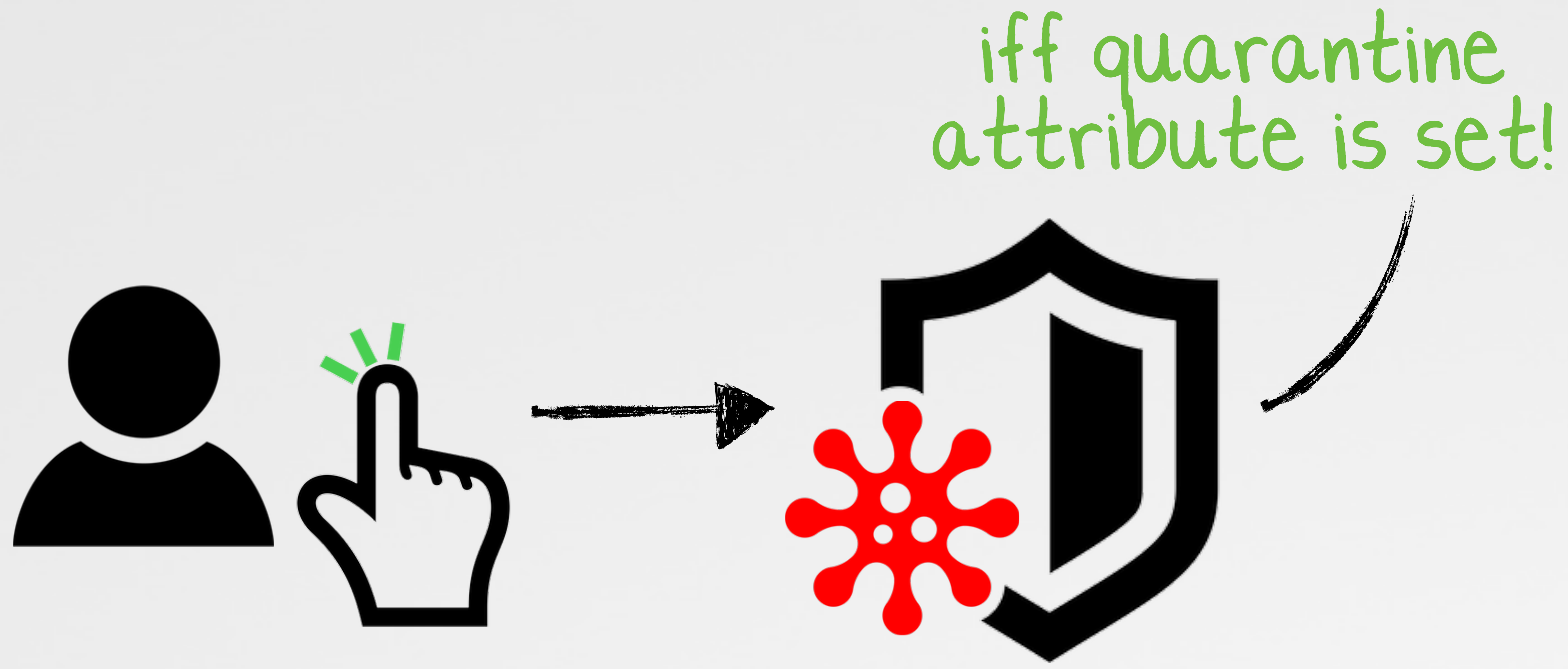
Q1 2015: all security software,
I downloaded -> served over HTTP :(

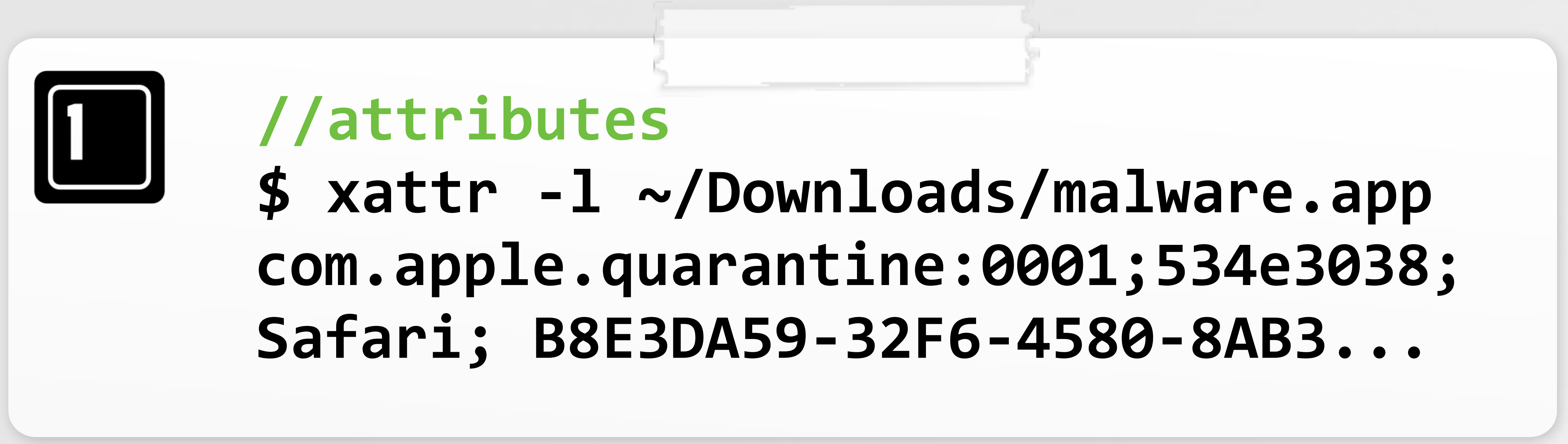
avast! intego Avira KASPERSKY lab Bitdefender LittleSnitch ClamXav Norton by Symantec eset Sophos F-Secure TREND MICRO

How Gatekeeper Works

an overview



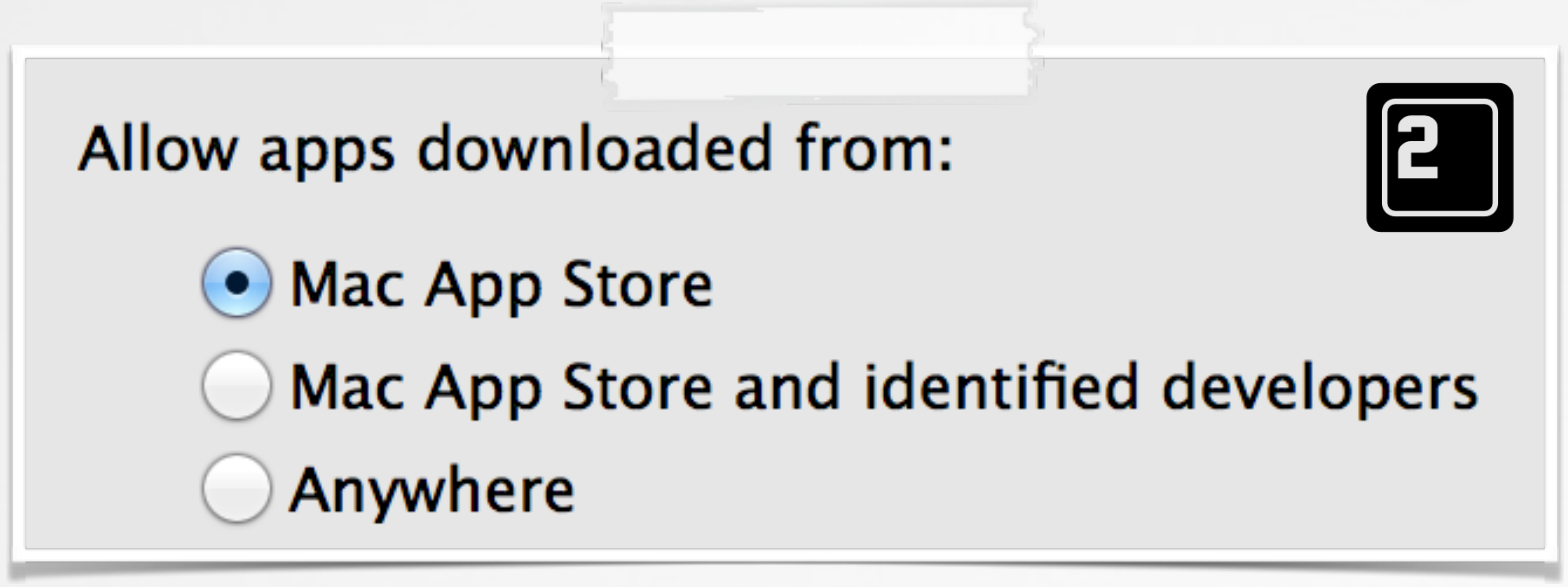




1

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

quarantine attributes

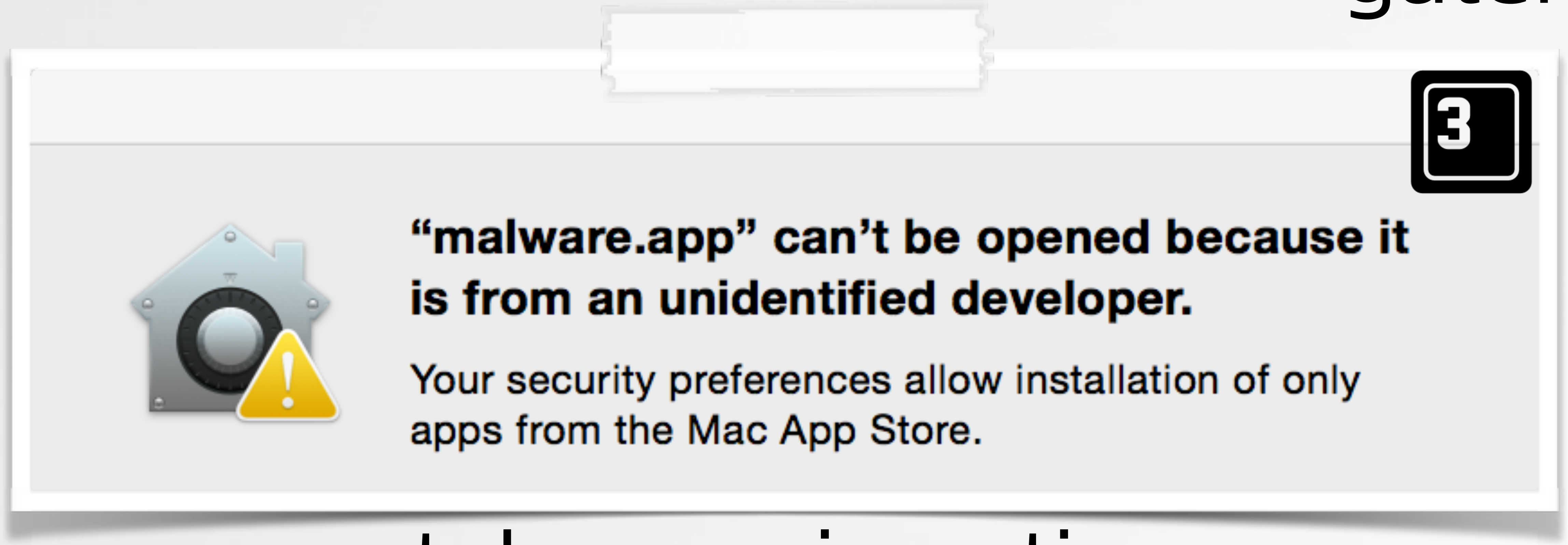


2

Allow apps downloaded from:

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

gatekeeper settings



3

“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.

gatekeeper in action

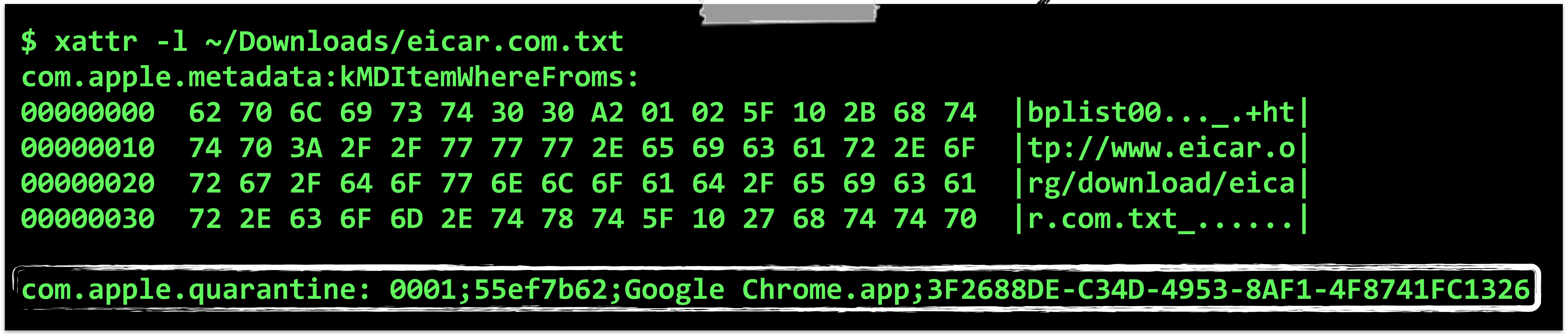
EXTENDED FILE ATTRIBUTES

simply put; file metadata

"Mac OS X & iOS Internals"
Jonathan Levin

extended attr. (com.apple.*)	brief details
FinderInfo	information for **Finder.app** (such as folder colors)
metadata	Spotlight data, such as download location & version info
quarantine	indicates that file is from an 'untrusted' source (internet)

dump w/ **xattr** command



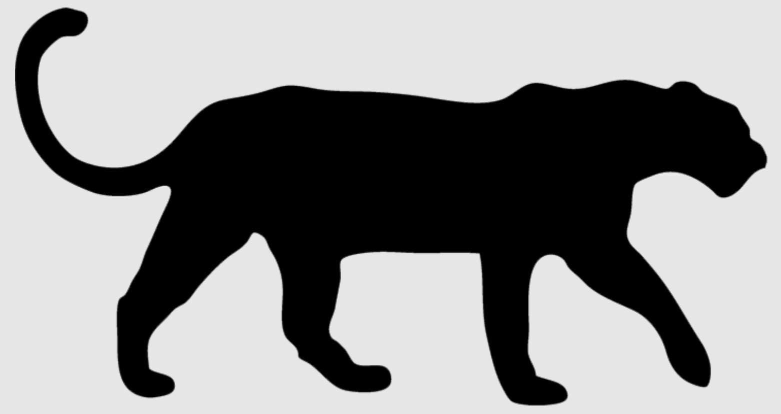
```
$ xattr -l ~/Downloads/eicar.com.txt
com.apple.metadata:kMDItemWhereFroms:
00000000  62 70 6C 69 73 74 30 30 A2 01 02 5F 10 2B 68 74  |bplist00..._.+ht|
00000010  74 70 3A 2F 2F 77 77 77 2E 65 69 63 61 72 2E 6F  |tp://www.eicar.o|
00000020  72 67 2F 64 6F 77 6E 6C 6F 61 64 2F 65 69 63 61  |rg/download/eica|
00000030  72 2E 63 6F 6D 2E 74 78 74 5F 10 27 68 74 74 70  |r.com.txt_......|

com.apple.quarantine: 0001;55ef7b62;Google Chrome.app;3F2688DE-C34D-4953-8AF1-4F8741FC1326
```

dumping quarantine attributes

Synack

'File Quarantine'

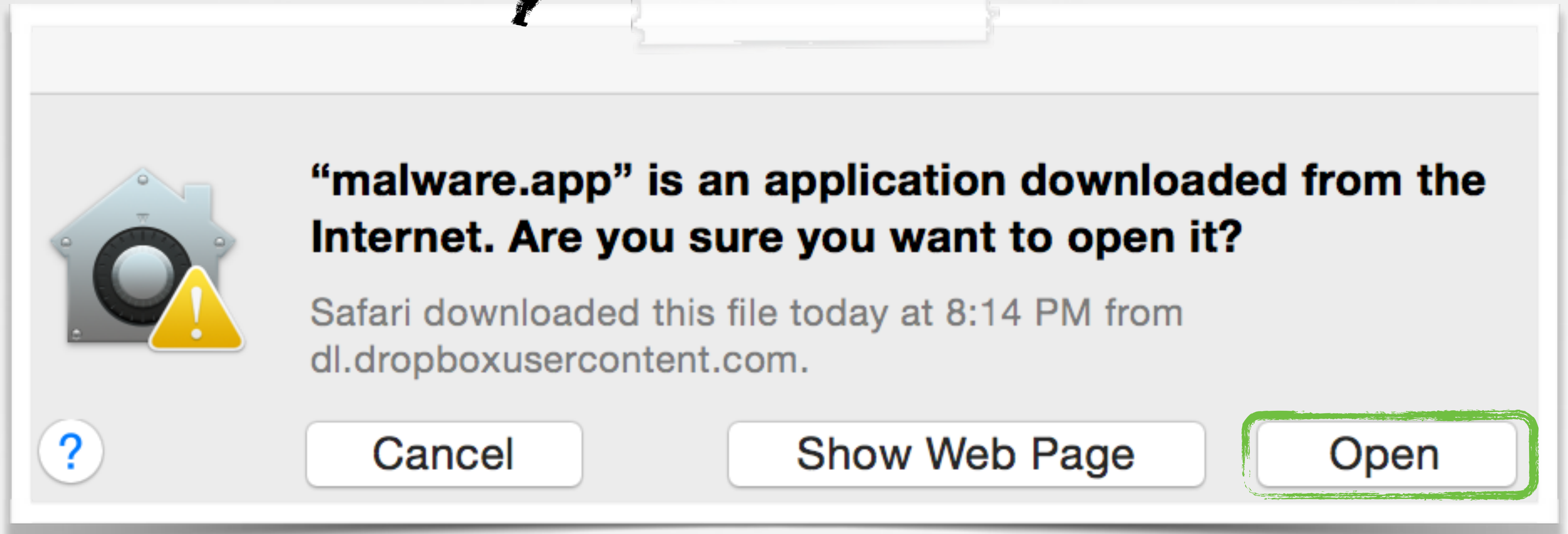
realized by the **com.apple.quarantine** file attribute



added in Leopard

"file from internet"

note; not gatekeeper



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

Safari downloaded this file today at 8:14 PM from dl.dropboxusercontent.com.

file quarantine in action

```
//dictionary for quarantine attributes
NSDictionary* quarantineAttributes = nil;

//get attributes
[fileURL getResourceValue:&quarantineAttributes
        forKey:NSURLQuarantinePropertiesKey error:NULL];
```

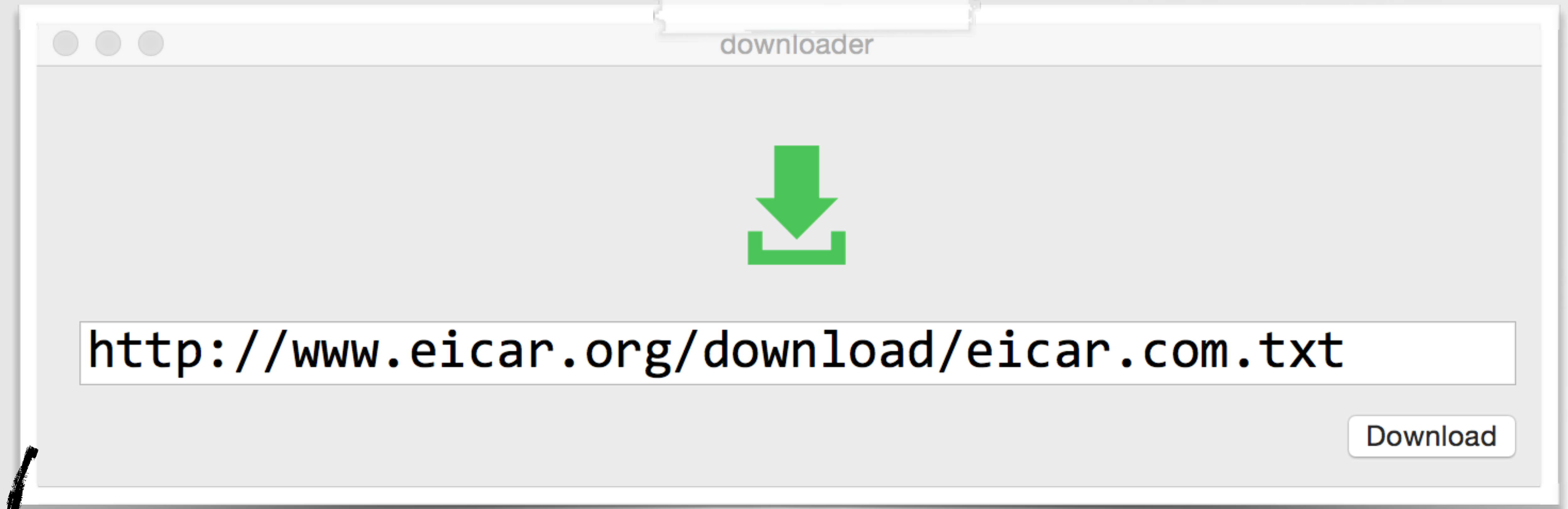
code to get attributes

dumping a file's **com.apple.quarantine** attribute

```
$ dumpAttrs ~/Downloads/eicar.com.txt
    LSQuarantineAgentBundleIdentifier = "com.google.Chrome";
    LSQuarantineAgentName = "Google Chrome.app";
    LSQuarantineDataURL = "http://www.eicar.org/download/eicar.com.txt";
    LSQuarantineEventIdentifier = "3F2688DE-C34D-4953-8AF1-4F8741FC1326";
    LSQuarantineOriginURL = "http://www.eicar.org/85-0-Download.html";
    LSQuarantineTimeStamp = "2015-09-09 00:20:50 +0000";
    LSQuarantineType = LSQuarantineTypeWebDownload;
```

SETTING THE QUARANTINE ATTRIBUTE

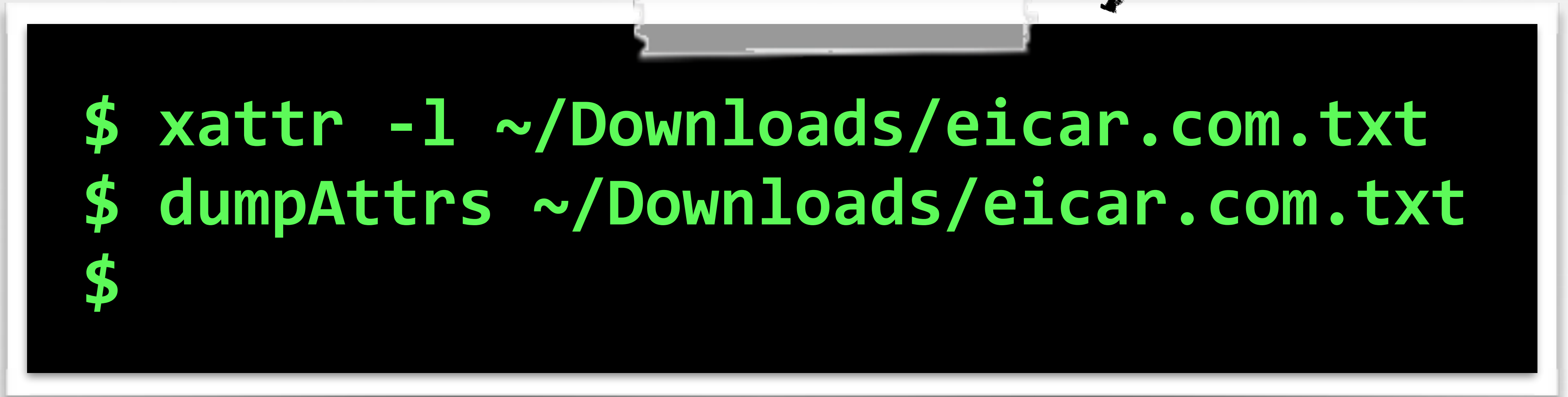
who done it!?



```
http://www.eicar.org/download/eicar.com.txt
```

custom downloader

none; huh?



```
$ xattr -l ~/Downloads/eicar.com.txt
$ dumpAttrs ~/Downloads/eicar.com.txt
$
```

any extended attributes?

```
//button handler: download file
-(IBAction)download:(id)sender
{
    //url
    NSURL *remoteFile = [NSURL URLWithString:self.textField.stringValue];

    //local file
    NSString* localFile = [NSString stringWithFormat:@"/tmp/%@", [remoteFile lastPathComponent]];

    //download & save to file
    [[NSData dataWithContentsOfURL:remoteFile] writeToFile:localFile atomically:NO];

    return;
}
```

custom downloader's source code

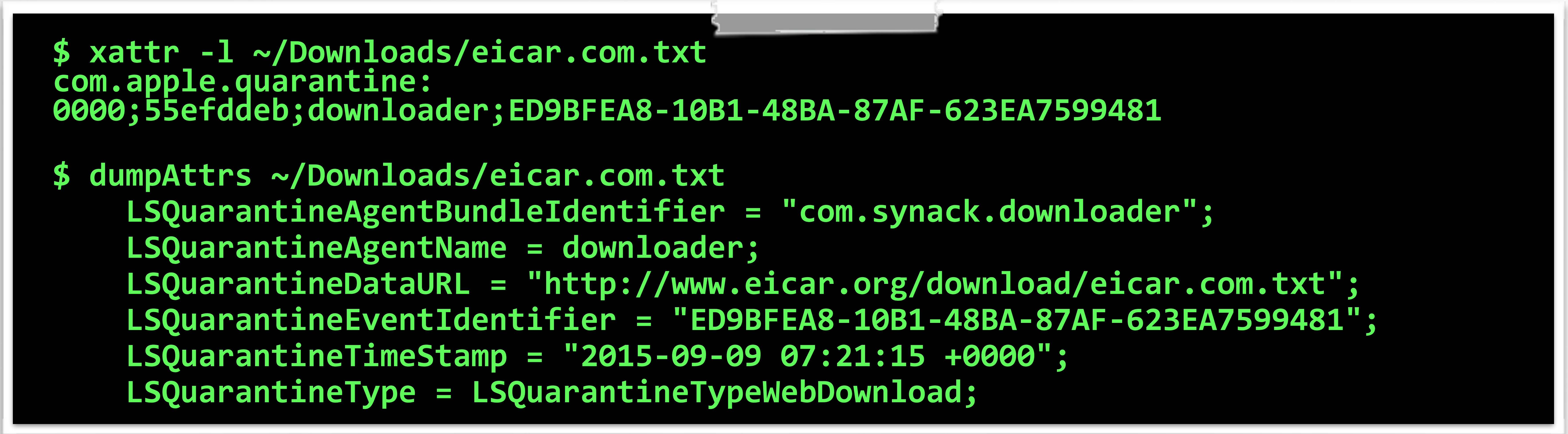
Setting The Quarantine Attribute

apps can manually add it

consts in LSQuarantine.h

```
-(void)setQAttr:(NSString*)localFile
{
    //quarantine attributes dictionary
    NSMutableDictionary* quarantineAttributes = [NSMutableDictionary dictionary];

    //add agent bundle id
    quarantineAttributes[kLSQuarantineAgentBundleIdentifierKey] = [[NSBundle mainBundle] bundleIdentifier];

    //add agent name
    quarantineAttributes[kLSQuarantineAgentNameKey] = [[[NSBundle mainBundle] infoDictionary] objectForKey:kCFBundleNameKey];

    ...

    //manually add quarantine attributes to file
    [[NSURL fileURLWithPath:localFile] setResourceValues:@{NSURLQuarantinePropertiesKey: quarantineAttributes} error:NULL];

    return;
}
```

code to set a file's quarantine attribute



```
$ xattr -l ~/Downloads/eicar.com.txt
com.apple.quarantine:
0000;55efddeb;downloader;ED9BFEA8-10B1-48BA-87AF-623EA7599481

$ dumpAttrs ~/Downloads/eicar.com.txt
    LSQuarantineAgentBundleIdentifier = "com.synack.downloader";
    LSQuarantineAgentName = downloader;
    LSQuarantineDataURL = "http://www.eicar.org/download/eicar.com.txt";
    LSQuarantineEventIdentifier = "ED9BFEA8-10B1-48BA-87AF-623EA7599481";
    LSQuarantineTimeStamp = "2015-09-09 07:21:15 +0000";
    LSQuarantineType = LSQuarantineTypeWebDownload;
```

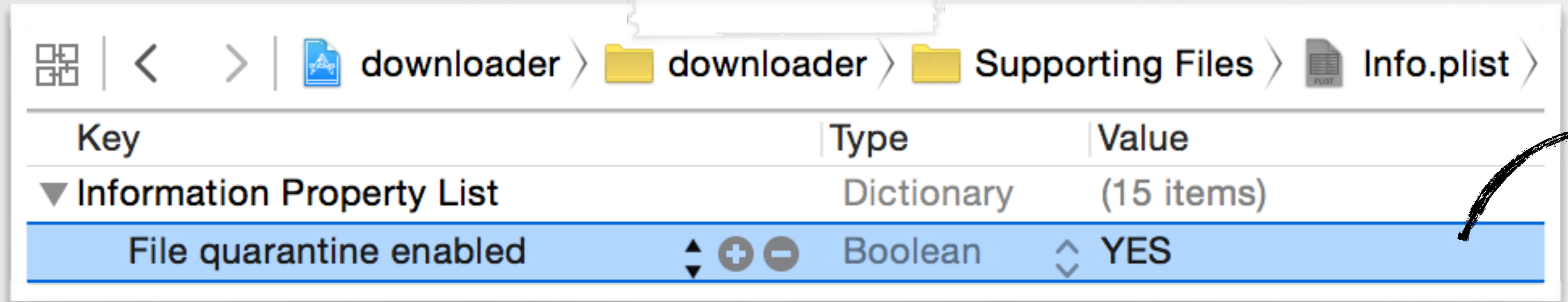
manually set, quarantine attribute

Synack

SETTING THE QUARANTINE ATTRIBUTE

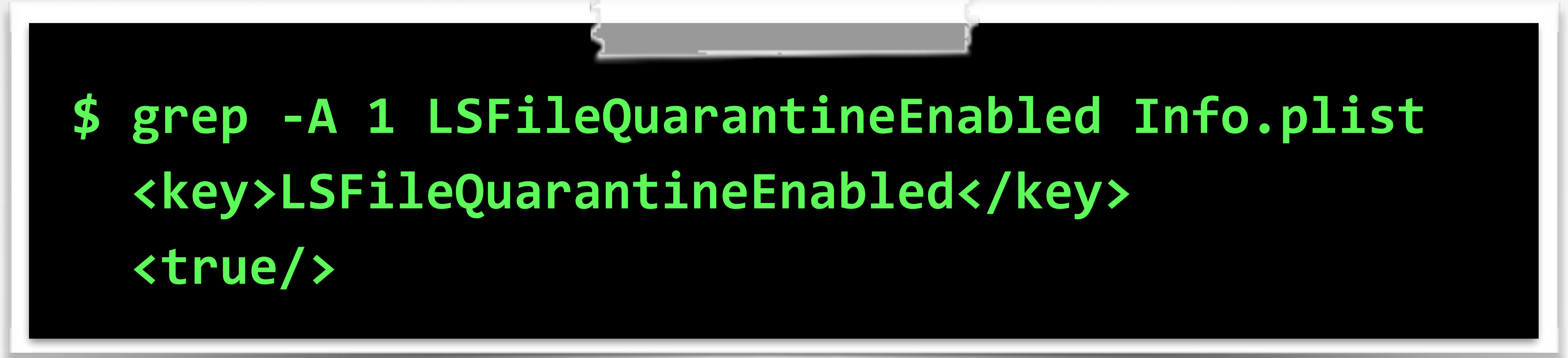
or, apps can generically tell the OS to add it

`Info.plist` keys: **`LSFileQuarantineEnabled`**
"When the value of this key is true, all files created by the application process will be quarantined by OS X" -apple.com



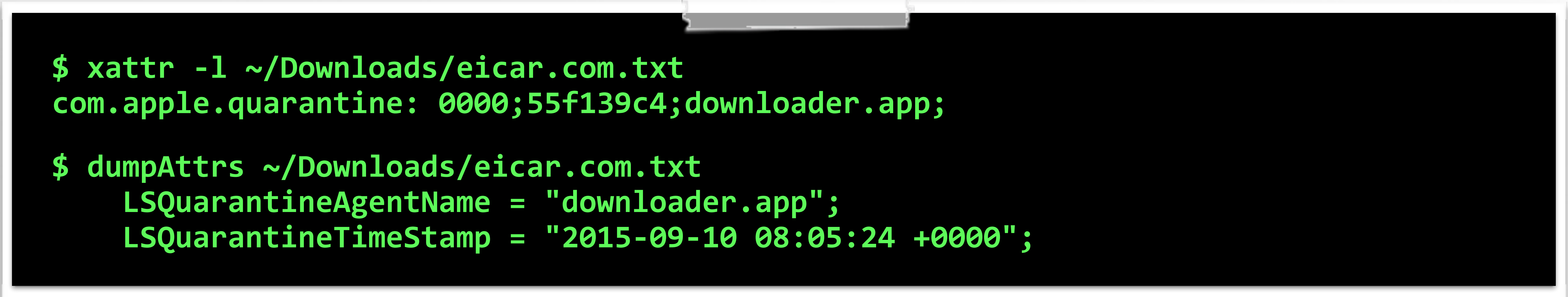
downloader 〉 downloader 〉 Supporting Files 〉 Info.plist 〉

Key	Type	Value
▼ Information Property List	Dictionary	(15 items)
File quarantine enabled	Boolean	YES



```
$ grep -A 1 LSFileQuarantineEnabled Info.plist
    <key>LSFileQuarantineEnabled</key>
    <true/>
```

app's **`Info.plist`** file updated (**`LSFileQuarantineEnabled`**)



```
$ xattr -l ~/Downloads/eicar.com.txt
com.apple.quarantine: 0000;55f139c4;downloader.app;

$ dumpAttrs ~/Downloads/eicar.com.txt
    LSQuarantineAgentName = "downloader.app";
    LSQuarantineTimeStamp = "2015-09-10 08:05:24 +0000";
```

automatically (OS) set, quarantine attribute

Gatekeeper in Action

an overview

Finder.app
1
LaunchServices framework
XPC request
Launchd
2
Quarantine.kext
3
CoreServicesUIAgent
XPC request
4
XProtect framework
5
?
OK
Synack

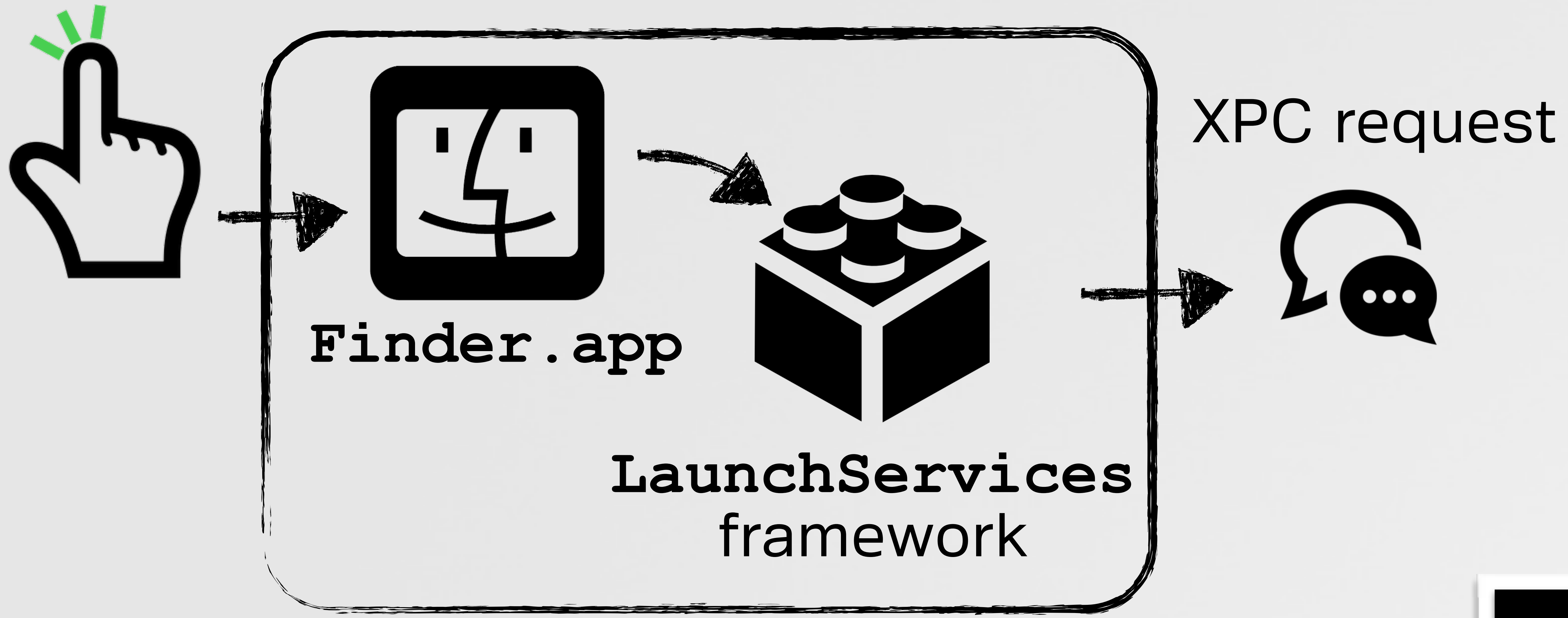
"malware.app" can't be opened because it is from an unidentified developer.

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

something downloaded this file on an unknown date.

1 Launching the Binary/App

handled by the **launchservices** framework



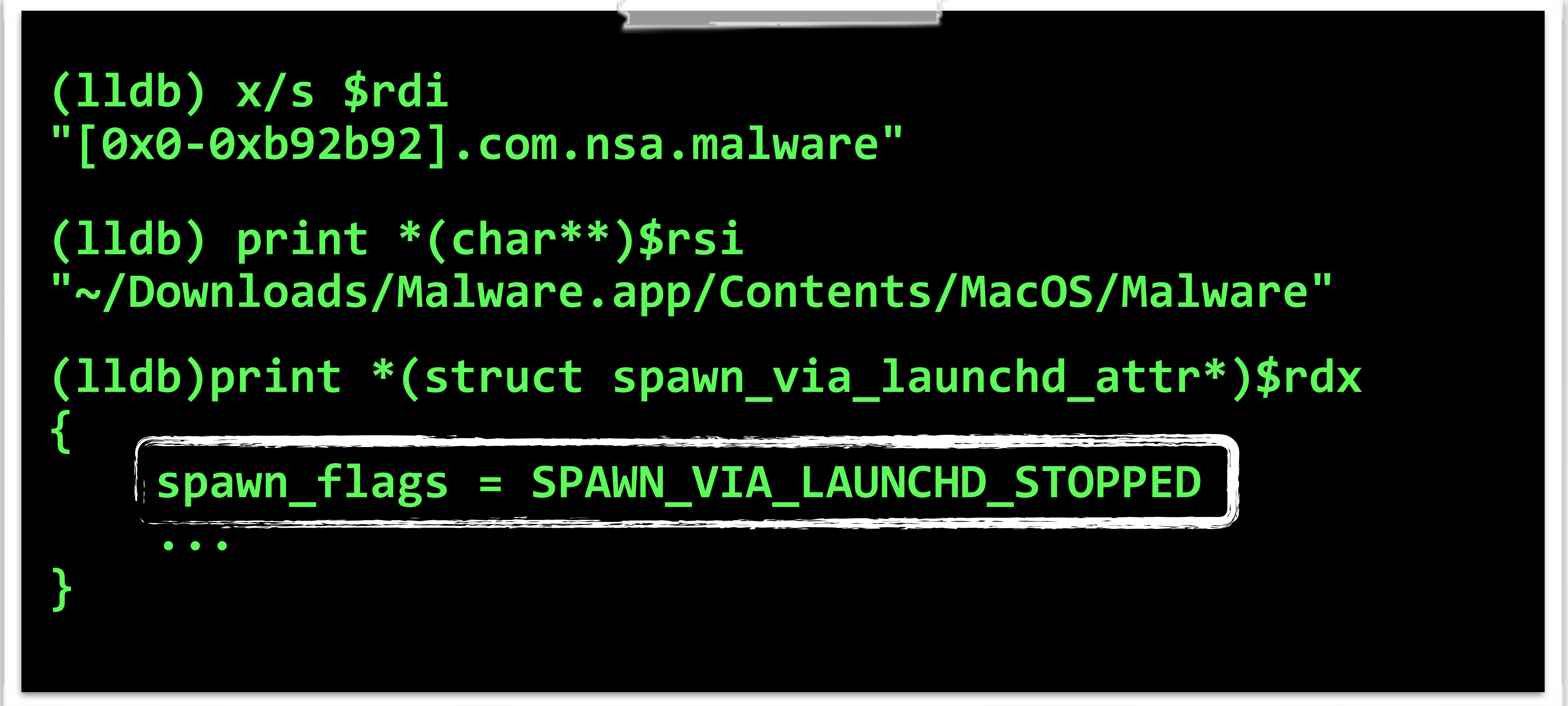
launch_priv.h

```
pid_t _spawn_via_launchd(
   const char *label,
   const char *const *argv,
   const struct spawn_via_launchd_attr *spawn_attrs,
   int struct_version
);
```

_spawn_via_launchd()

```
libxpc.dylib`_spawn_via_launchd
LaunchServices`LaunchApplicationWithSpawnViaLaunchD
LaunchServices`_LSLaunchApplication
LaunchServices`_LSLaunch
LaunchServices`_LSOpenApp
LaunchServices`_LSOpenStuffCallLocal
LaunchServices`_LSOpenStuff
LaunchServices`_LSOpenURLsWithRole_Common
LaunchServices`LSOpenURLsWithRole
```

call stack



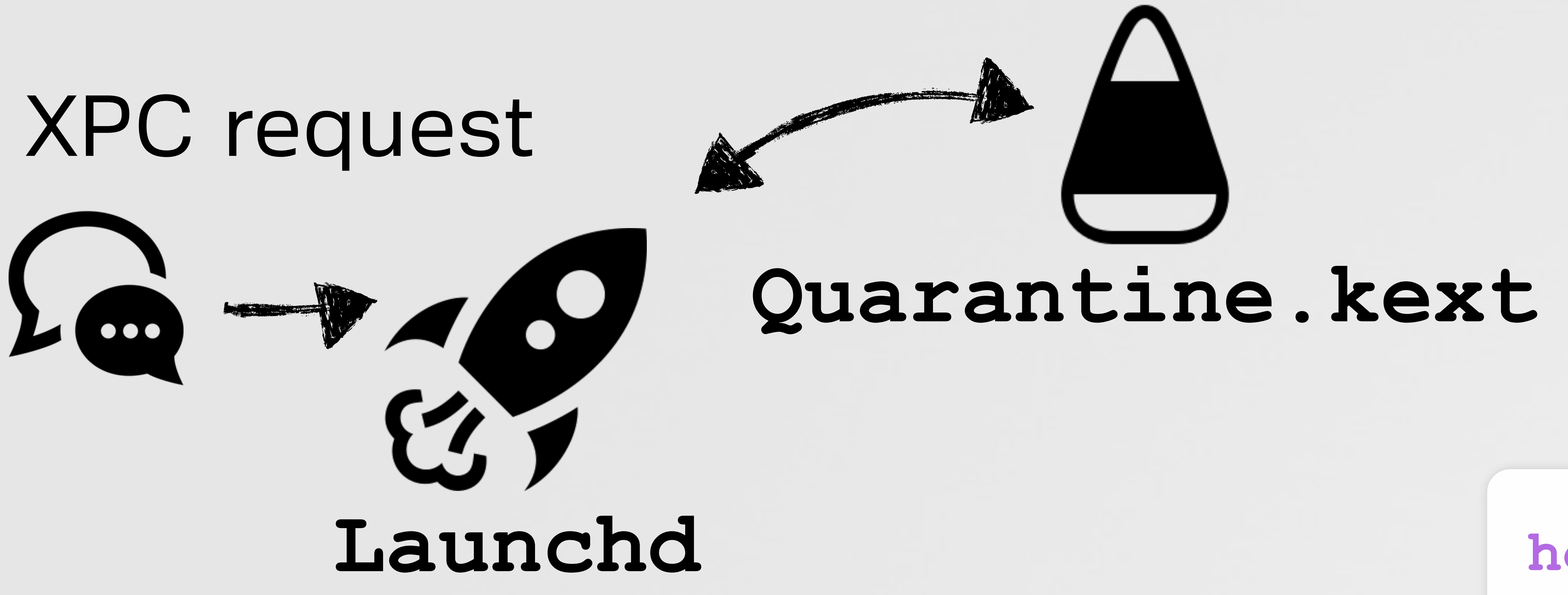
```
(lldb) x/s $rdi
"[0x0-0xb92b92].com.nsa.malware"

(lldb) print *(char**)$rsi
"~/Downloads/Malware.app/Contents/MacOS/Malware"

(lldb)print *(struct spawn_via_launchd_attr*)$rdx
{
    spawn_flags = SPAWN_VIA_LAUNCHD_STOPPED
    ...
}
```

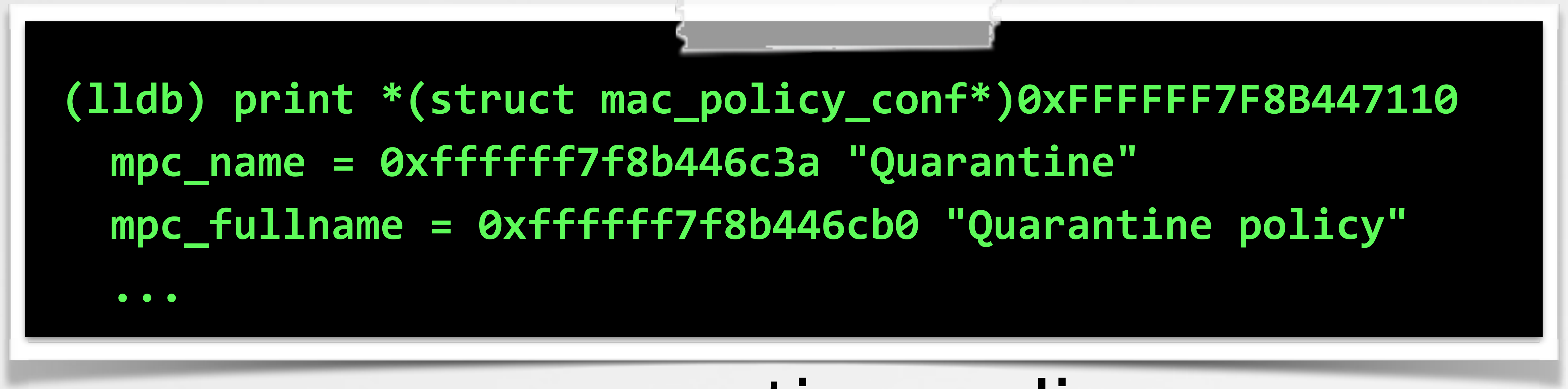
'spawn' attributes, etc.

Synack

2 Policy Enforcement with Quarantine.kext

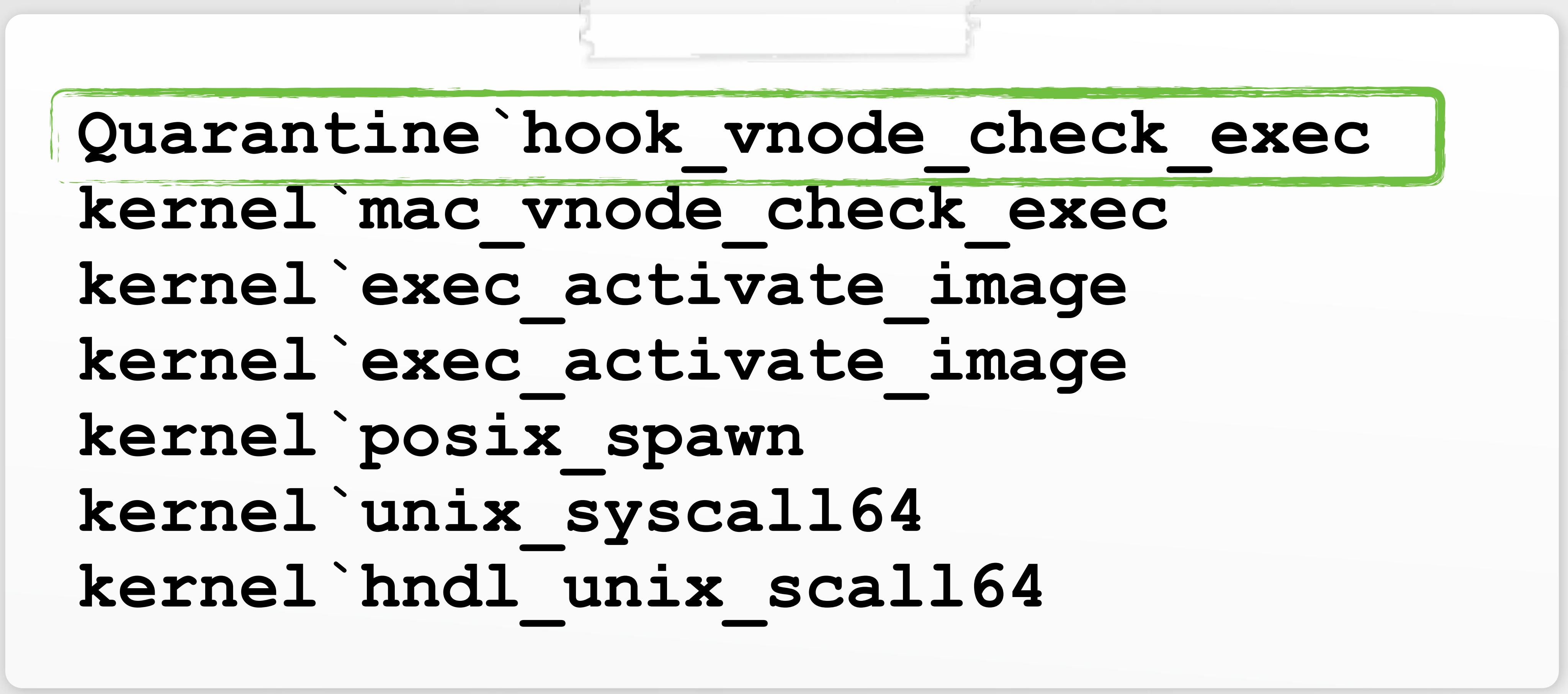
kernel-mode mac component





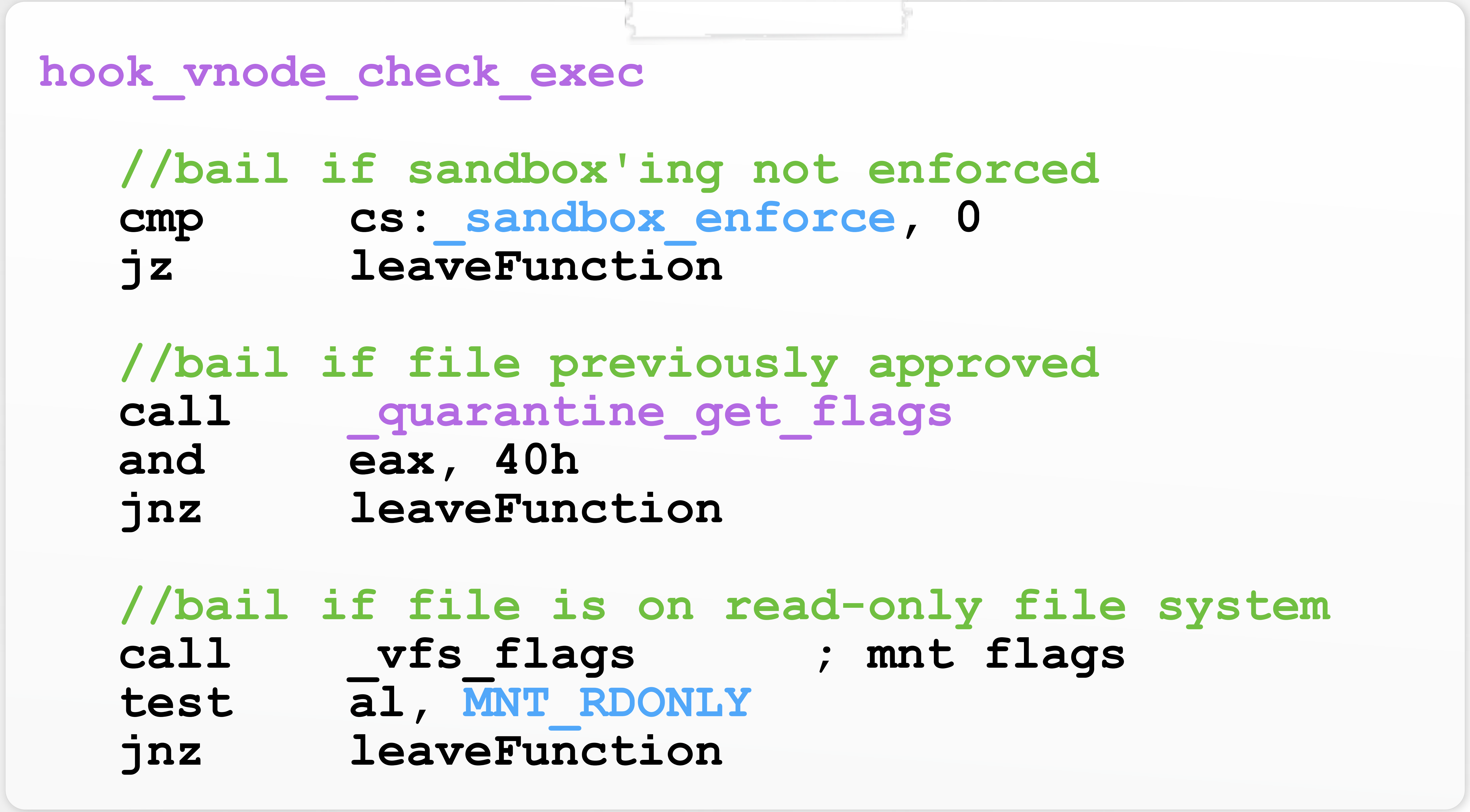
```
(lldb) print *(struct mac_policy_conf*)0xFFFFFF7F8B447110
  mpc_name = 0xffffff7f8b446c3a "Quarantine"
  mpc_fullname = 0xffffff7f8b446cb0 "Quarantine policy"
  ...
```

quarantine policy



```
Quarantine`hook_vnode_check_exec
kernel`mac_vnode_check_exec
kernel`exec_activate_image
kernel`exec_activate_image
kernel`posix_spawn
kernel`unix_syscall64
kernel`hndl_unix_scall64
```

call stack

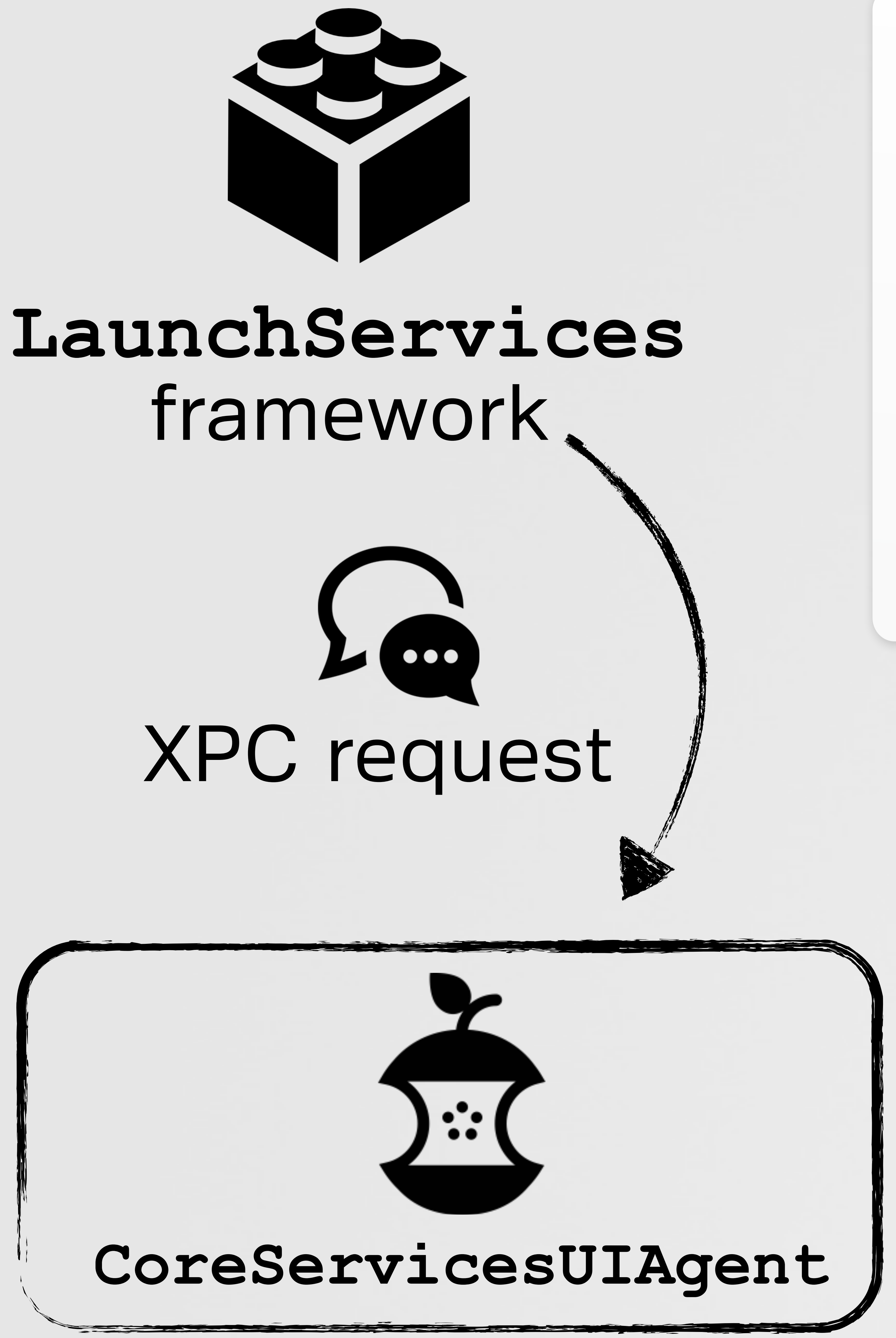


```
hook_vnode_check_exec

   //bail if sandbox'ing not enforced
   cmp       cs:_sandbox_enforce, 0
   jz        leaveFunction

   //bail if file previously approved
   call      _quarantine_get_flags
   and       eax, 40h
   jnz       leaveFunction

   //bail if file is on read-only file system
   call      _vfs_flags          ; mnt flags
   test      al, MNT_RDONLY
   jnz       leaveFunction
```

hook_vnode_check_exec

3 User Interaction via CoreServicesUIAgent

first, the xpc request



pseudo code

```
void ___LSAgentGetConnection_block_invoke(void * _block)
{
  rax = xpc_connection_create_mach_service("com.apple.coreservices.quarantine-resolver",
            dispatch_get_global_queue(0x0, 0x0), 0x0);

  xpc_connection_set_event_handler(rax, void ^(void * _block, void * arg1)
  {
        return;
  });

  xpc_connection_resume(rax);
  return;
}
```

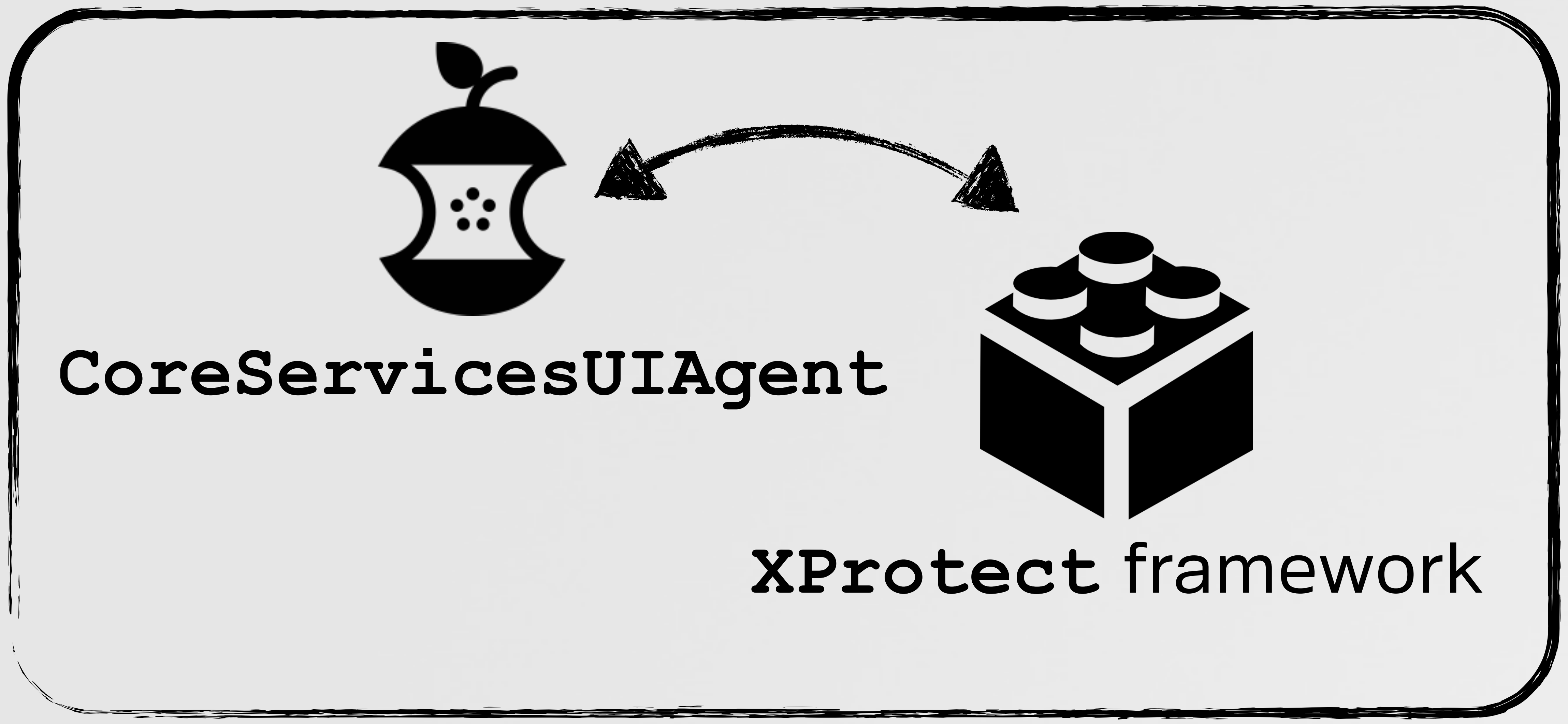
getting XPC connection to **CoreServicesUIAgent**

```
(lldb) po $rax
{
    LSQAllowUnsigned = 0;
    LSQAppPSN = 3621748;
    LSQAppPath = "/Users/patrick/Downloads/Malware.app";
    LSQAuthorization = <bed76627 c7cc0ae4 a6860100 00000000 ...
    LSQRiskCategory = LSRiskCategoryUnsafeExecutable;
}
```

XPC message contents

4 User Interaction via CoreServicesUIAgent

then, analysis via xprotect



```
(lldb) po $rdi
{
  FileURL = "file:///Users/patrick/Downloads/Malware.app";
  ShouldShowMalwareSubmission = 0;
  XProtectCaspianContext =     {
        "context:qtnflags" = 33;
        operation = "operation:execute";
  };
  XProtectDetectionType = 3;
  XProtectMalwareType = 2;
}
```

```
-[CSUIController handleIncomingXPCMessage:clientConnection:]
  -[GKQuarantineResolver resolve]
   -[GKQuarantineResolver malwareChecksBegin]
      -[GKQuarantineResolver malwareCheckNextItem]
          mov rdi, cs:classRef_XProtectAnalysis
          mov rsi, cs:selRef_alloc
          call     r15  ; _objc_msgSend
          mov      rdi, rax
          mov      rsi, cs:selRef_initWithURL_
          mov      rdx, r14 ;path to app
          call     r15  ; _objc_msgSend

        -[XProtectAnalysis
       beginAnalysisWithDelegate:didEndSelector:contextInfo:]
          +[WorkerThreadClass threadEntry:]
            mov      rdi, [rbp+staticCodeRef]
            lea      rdx, [rbp+signingInfo]
            xor      esi, esi ;flags
            call     _SecCodeCopySigningInformation
```

program control flow

XProtectMalwareType	meaning
0x2	unsigned
0x3	modified bundle
0x5	signed app
0x7	modified app

Synack

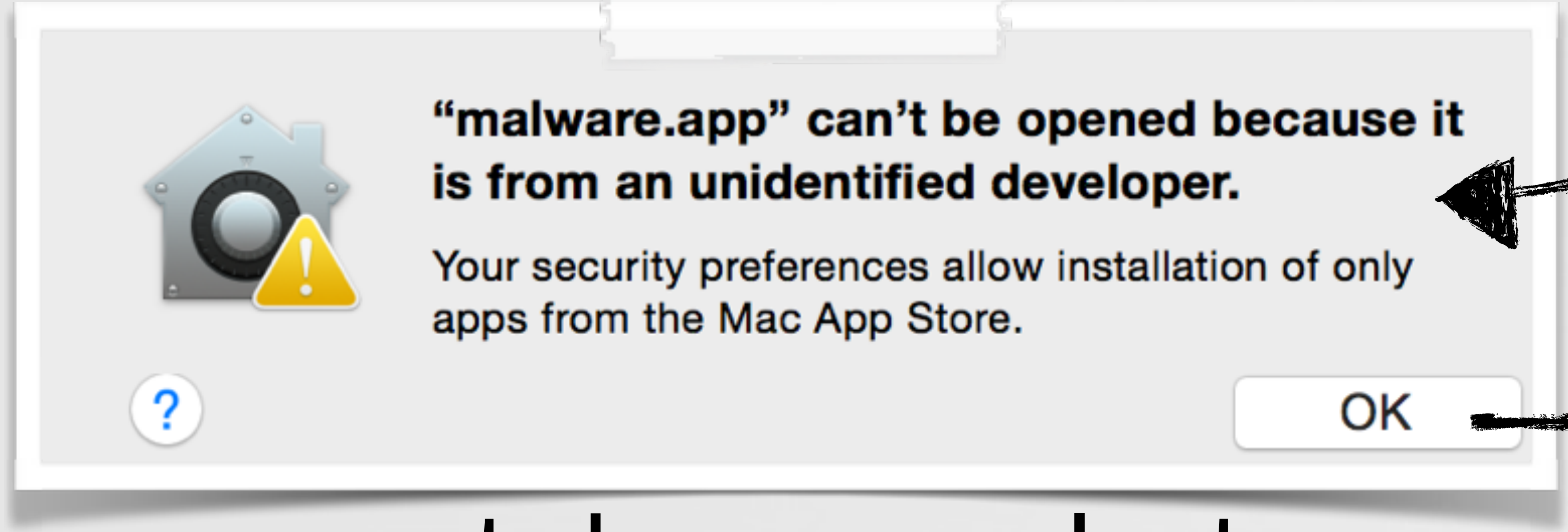
5 User Interaction via CoreServicesUIAgent

finally, display the alert

CoreServicesUIAgent

```
-[GKQuarantineResolver showGKAlertForPath:]
 -[GKQuarantineResolver alertForPath:malwareInfo:]

   mov     rax, _OBJC_IVAR_$_GKQuarantineResolver__allowUnsigned
   mov     rcx, [rbp+GKQuarantineResolver]
   cmp     byte ptr [rcx+rax], 0

   lea     rdi, cfstr_Q_headline_cas ; "Q_HEADLINE_CASPIAN_BAD_DISTRIBUTOR"

   mov     rdi, cs:classRef_NSAlert
   mov     rsi, cs:selRef_alloc
   call    r12  ; _objc_msgSend
```

alert customization


“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.

gatekeeper alert

```
mov     rsi, cs:selRef_deny
mov     rdi, r14
call    cs:_objc_msgSend_ptr

-[GKQuarantineResolver deny]
 -[GKQuarantineResolver denyWithoutSettingState]

   mov     rax, _OBJC_IVAR_$_GKQuarantineResolver__appASN
   mov     rsi, [rdi+rax]
   mov     edi, 0FFFFFFFEh
   mov     edx, 2
   call    __LSKillApplication
```

application termination

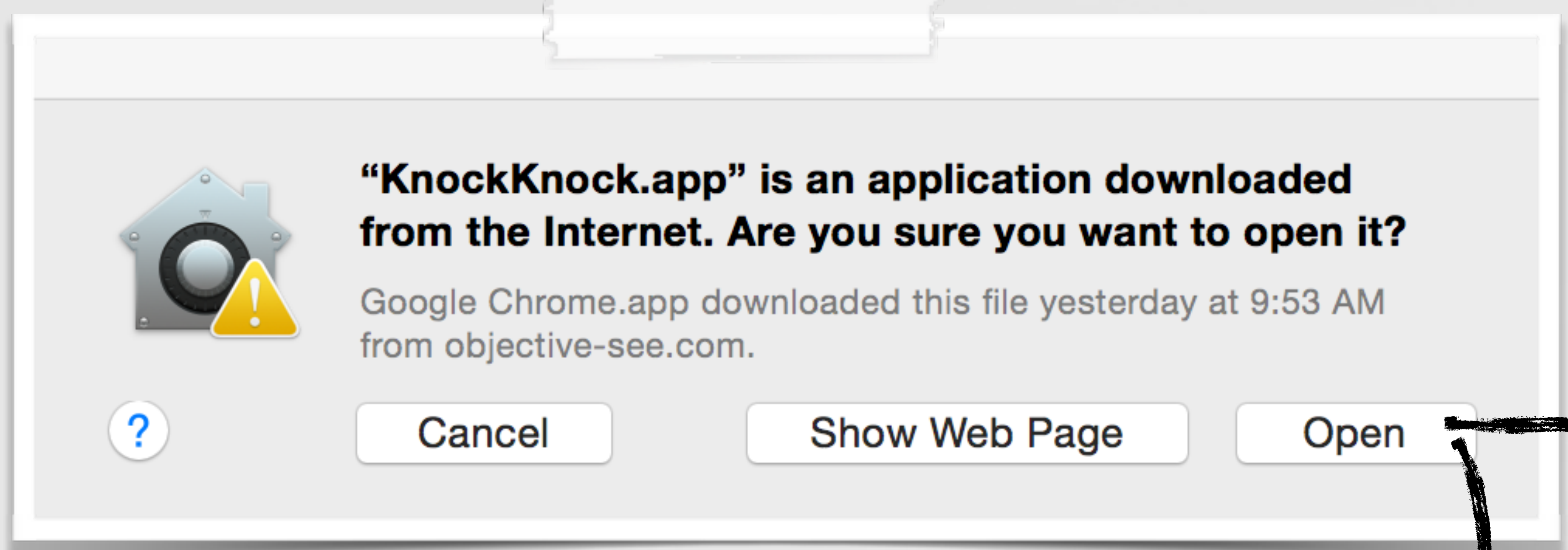
```
$ less QuarantineHeadlines.strings
<key>Q_HEADLINE_CASPIAN_BAD_DISTRIBUTOR</key>
<string>
  “%@” can’t be opened because it is from an unidentified developer.
</string>
<key>Q_HEADLINE_CASPIAN_BLOCKED</key>
<string>
  “%@” can’t be opened because it was not downloaded from the Mac App Store.
</string>
```

alert strings (QuarantineHeadlines.strings)

Synack

What if the App Conforms & is Allowed by the User?

quarantine attributes updated, then application resumed



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

Google Chrome.app downloaded this file yesterday at 9:53 AM from objective-see.com.

quarantine alert

```
-[GKQuarantineResolver
approveUpdatingQuarantineTarget:recursively:volume:]

call       __qtn_file_get_flags
or         eax, 40h
mov        rdi, [rbp+var_B8]
mov        esi, eax
call       __qtn_file_set_flags
```

updating quarantine attributes

```
mov     rsi, [r13+r14+0]
mov     rax, __kLSApplicationInStoppedStateKey_ptr
mov     rdx, [rax]
mov     edi, 0FFFFFFFEh
xor     r8d, r8d
mov     rcx, rbx
call    __LSSetApplicationInformationItem

;on error
lea     rsi, "Unable to continue stopped application"
mov     edi, 4
xor     eax, eax
mov     edx, ecx
call    logError
```

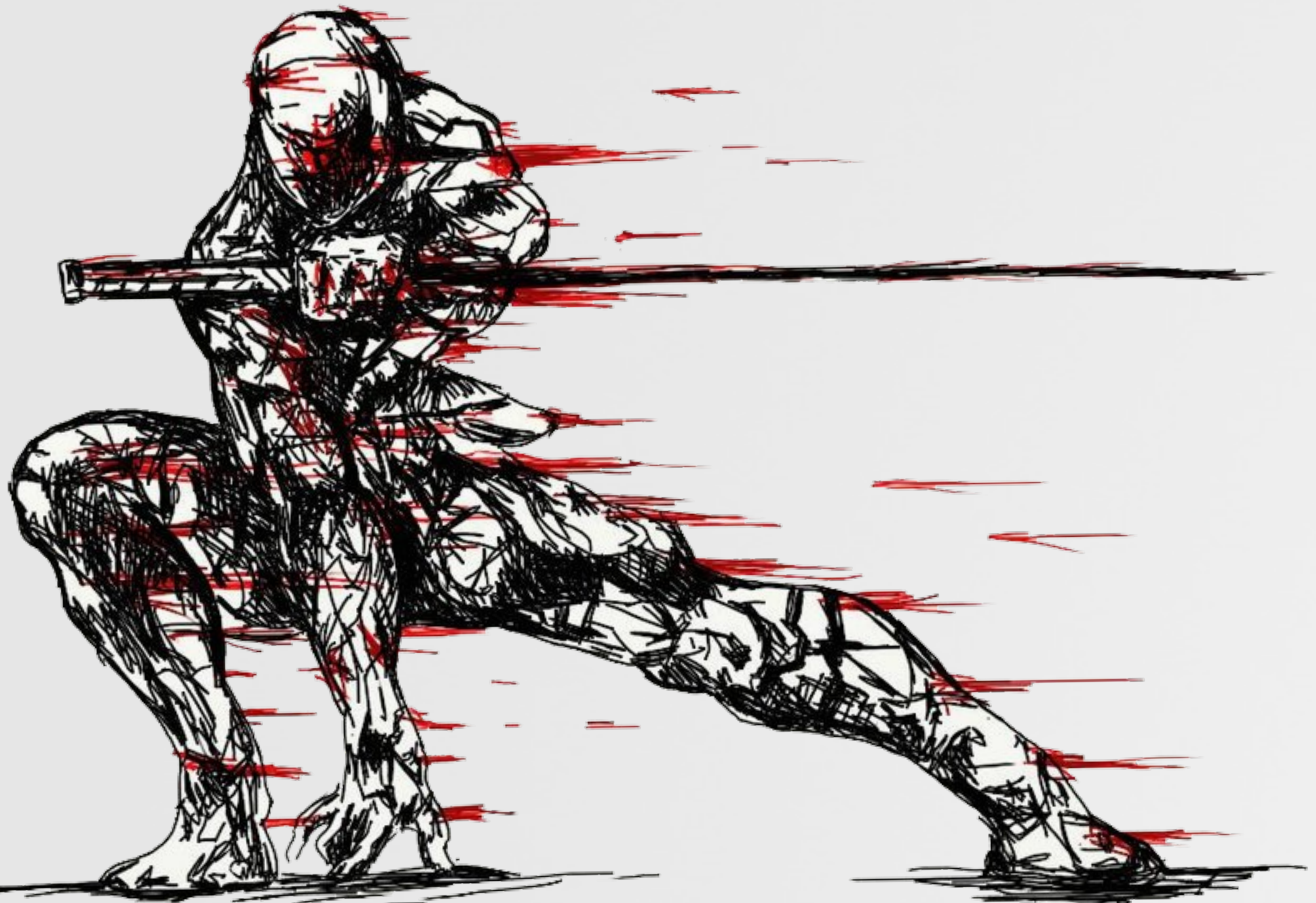
resuming application

```
$ xattr -l ~/Downloads/KnockKnock.app/Contents/MacOS/KnockKnock
com.apple.quarantine: 0001 55f3313d;Google\x20Chrome.app;FBF45932...

$ xattr -l ~/Downloads/KnockKnock.app/Contents/MacOS/KnockKnock
com.apple.quarantine: 0041 55f3313d;Google\x20Chrome.app;FBF45932...
```

before & after

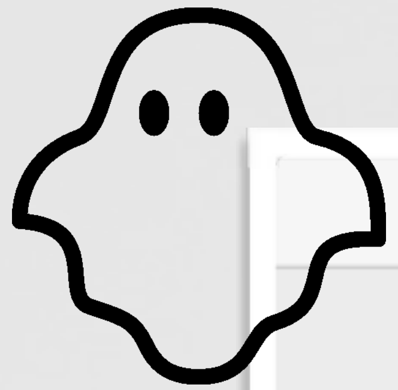
BYPASSING GATEKEEPER

unsigned code allowed!?

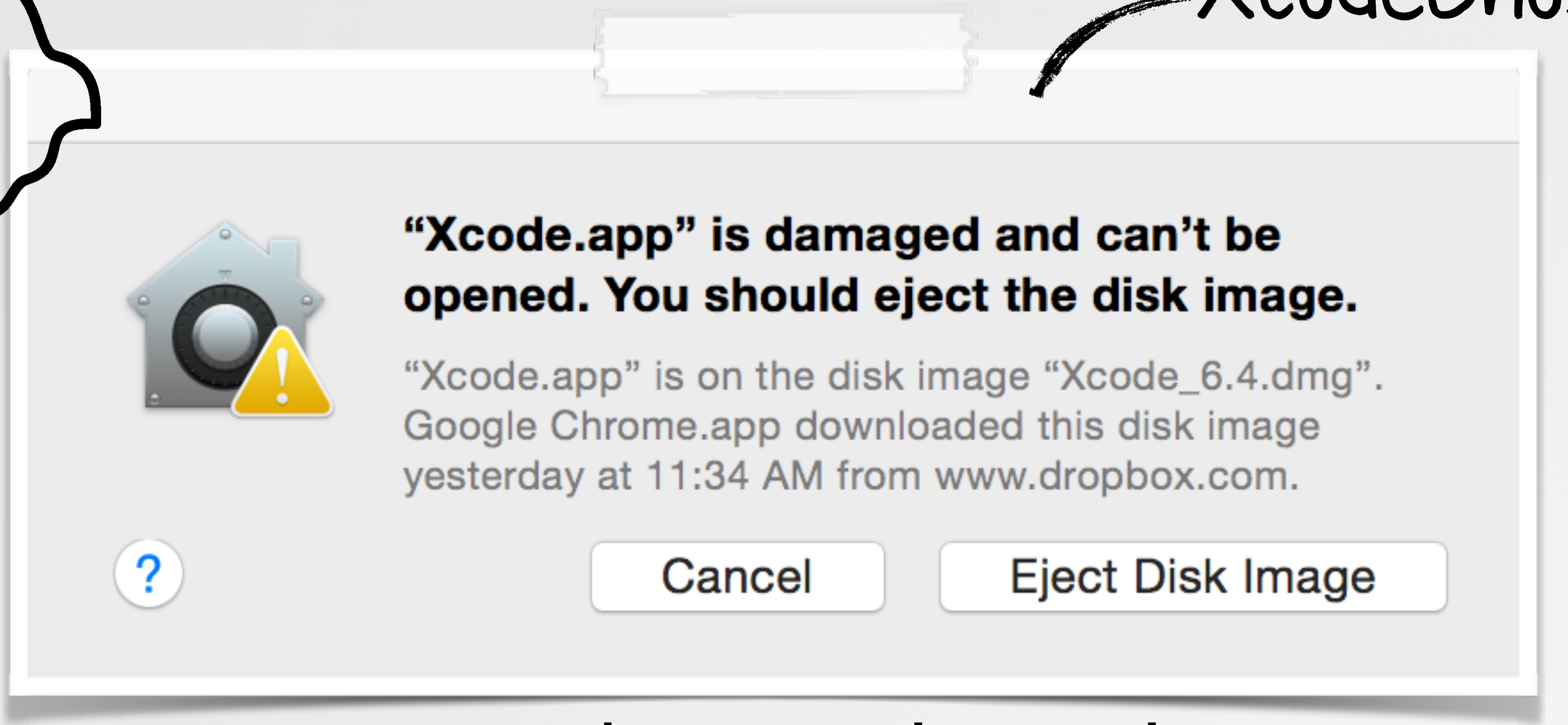


Recall; Gatekeeper Aims to Protect

...unauthorized code should be blocked!



XcodeGhost


"Xcode.app" is damaged and can't be opened. You should eject the disk image.

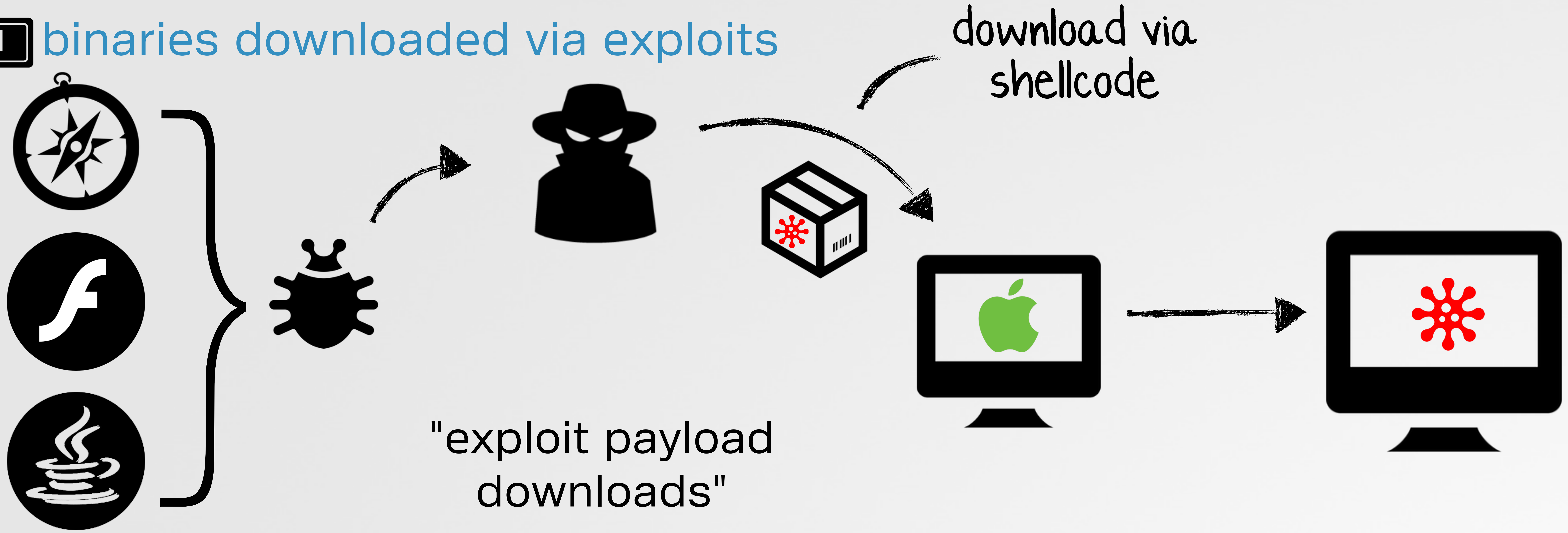
"Xcode.app" is on the disk image "Xcode_6.4.dmg". Google Chrome.app downloaded this disk image yesterday at 11:34 AM from www.dropbox.com.

gatekeeper in action

block unauthorized code from the internet

Gatekeeper Shortcomings

1 binaries downloaded via exploits

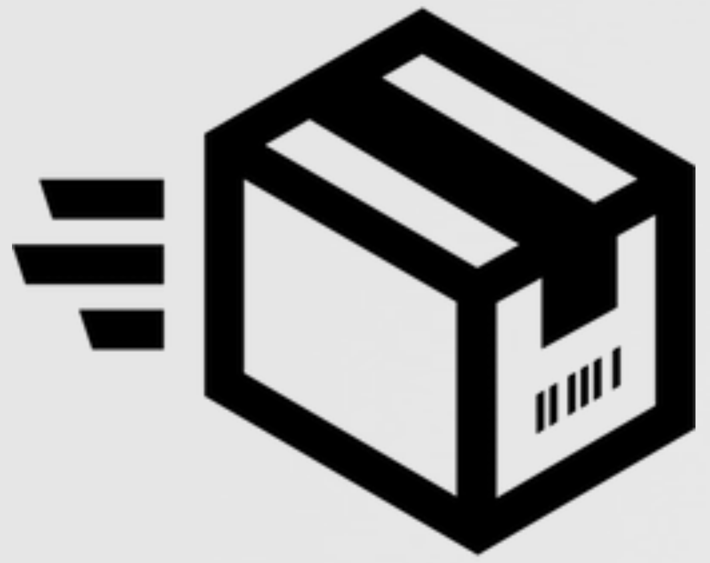


"malware that comes onto the system through vulnerabilities...bypass quarantine entirely. The infamous Flashback malware, for example, used Java vulnerabilities to copy executable files into the system. Since this was done behind the scenes, out of view of quarantine, those executables were able to run without any user interactions" -www.thesafemac.com

Gatekeeper Shortcomings

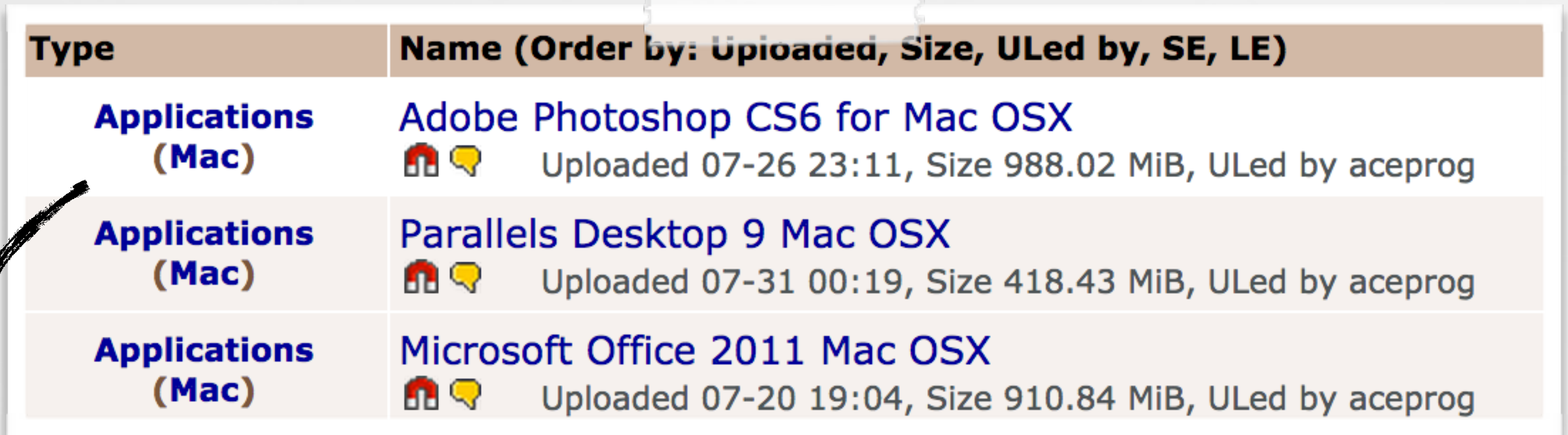
2 downloading app, must 'support' quarantine attribute

attribute added?



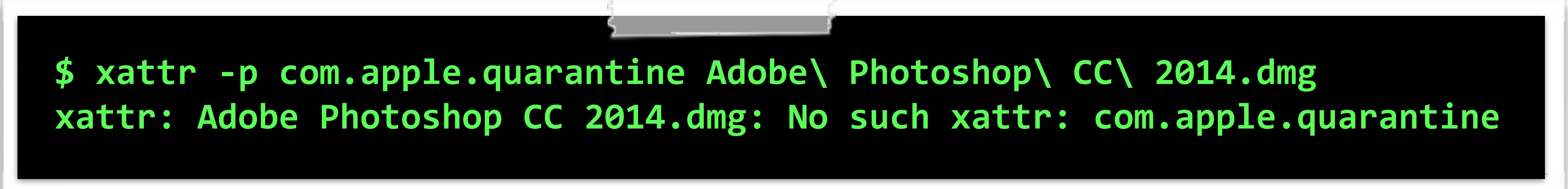
→ virus BULLETIN vb201410-iWorm.pdf

uTorrent



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

iWorm infected applications



```
$ xattr -p com.apple.quarantine Adobe\ Photoshop\ CC\ 2014.dmg
xattr: Adobe Photoshop CC 2014.dmg: No such xattr: com.apple.quarantine
```

no quarantine attribute :(

"the quarantine system relies on the app being used for downloading doing things properly. Not all do, and this can result in the quarantine flag not being set on downloaded files" -www.thesafemac.com

Synack

Gatekeeper Bypasses

allowing unsigned code to execute

2014 → CVE 2014-8826 (patched) → malicious jar file

required java

2015 → CVE 2015-3715 (patched) → dylib hijacking

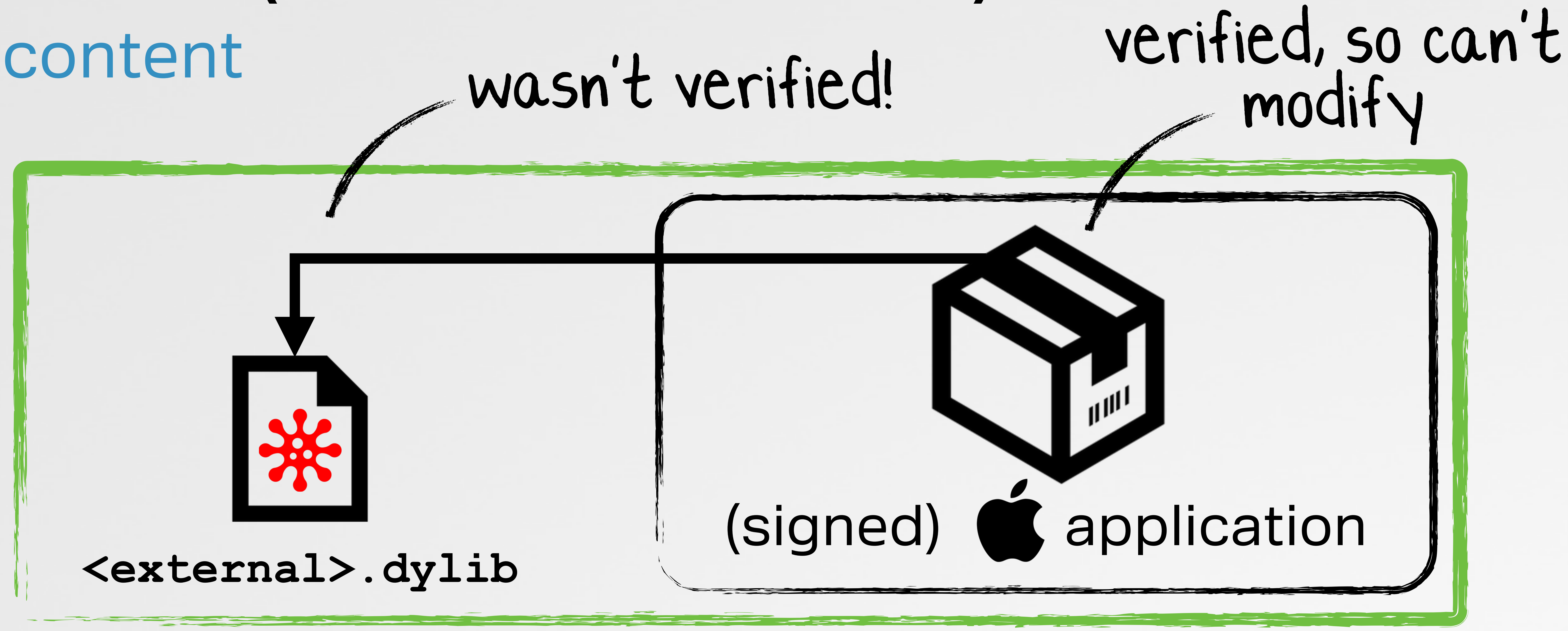
today → unpatched → "runtime shenanigans"

default OS install

GATEKEEPER BYPASS 0X1 (CVE 2015-3715)

(dylib) hijacking external content

gatekeeper **only** verified the app bundle!



.dmg/.zip layout

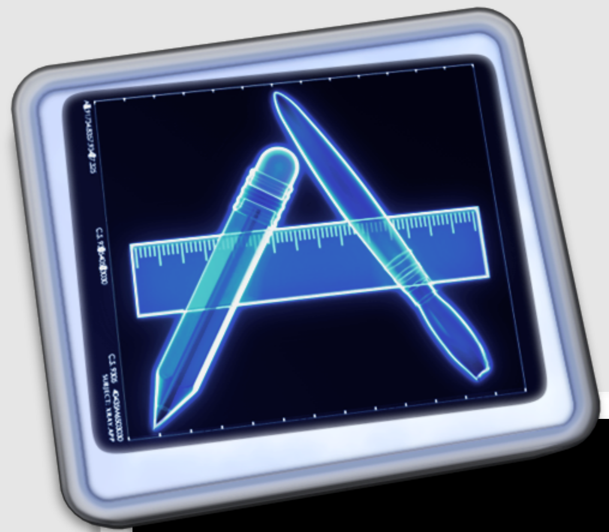
1. find an signed app that contains an **external, relative dependency** to a hijackable dylib
2. create a .dmg/.zip with the necessary folder structure (i.e. placing the malicious dylib in the **externally** referenced location)
3. host online or inject

→ virus BULLETIN white paper **www.virusbtn.com/dylib**

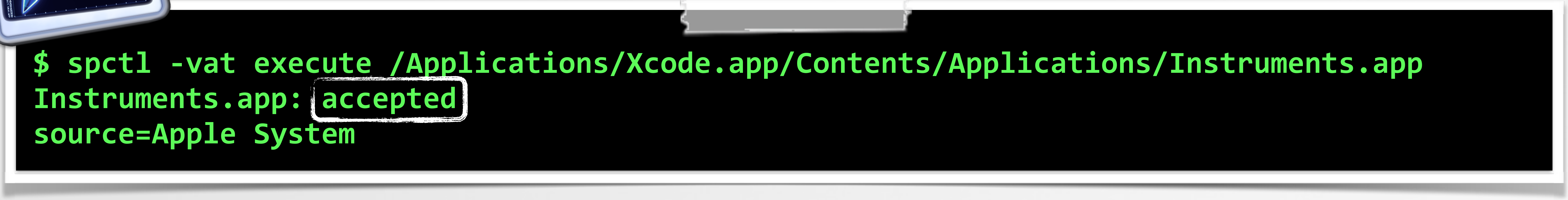
Synack

GATEKEEPER BYPASS 0X1 (CVE 2015-3715)

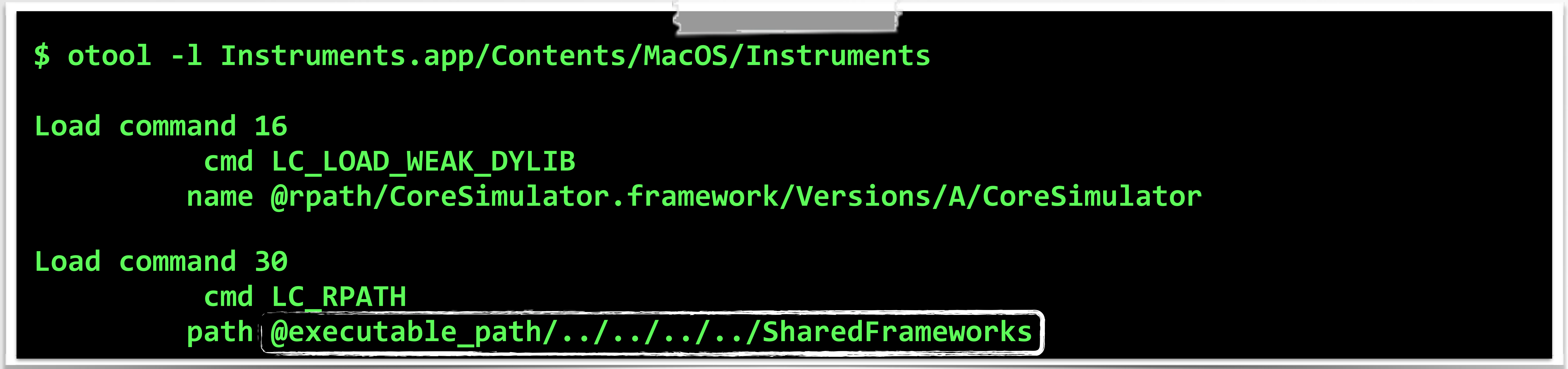
1 a signed app that contains an external dependency to hijackable dylib



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
```

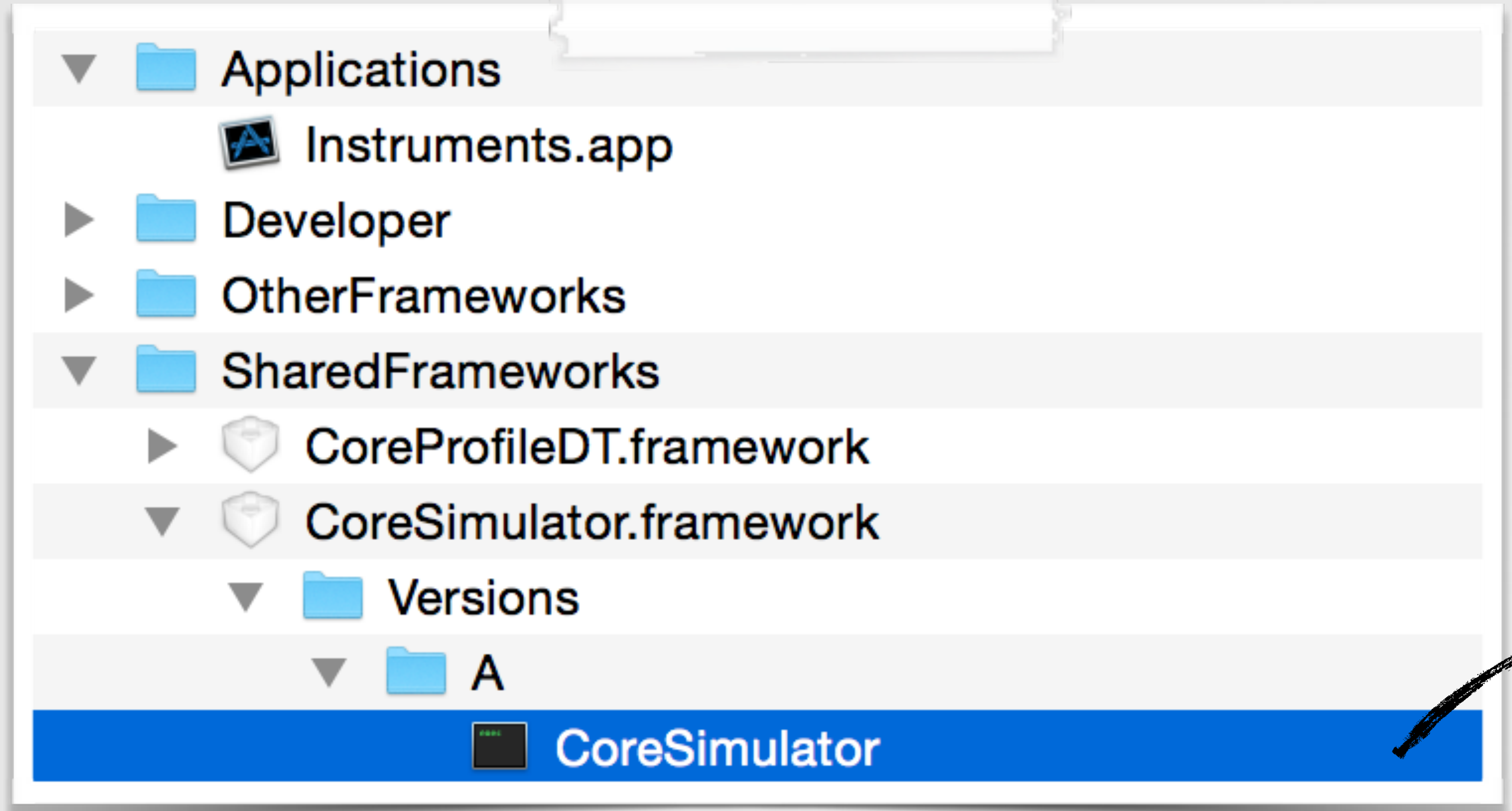


```
$ otool -l Instruments.app/Contents/MacOS/Instruments

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 0x1 (CVE 2015-3715)

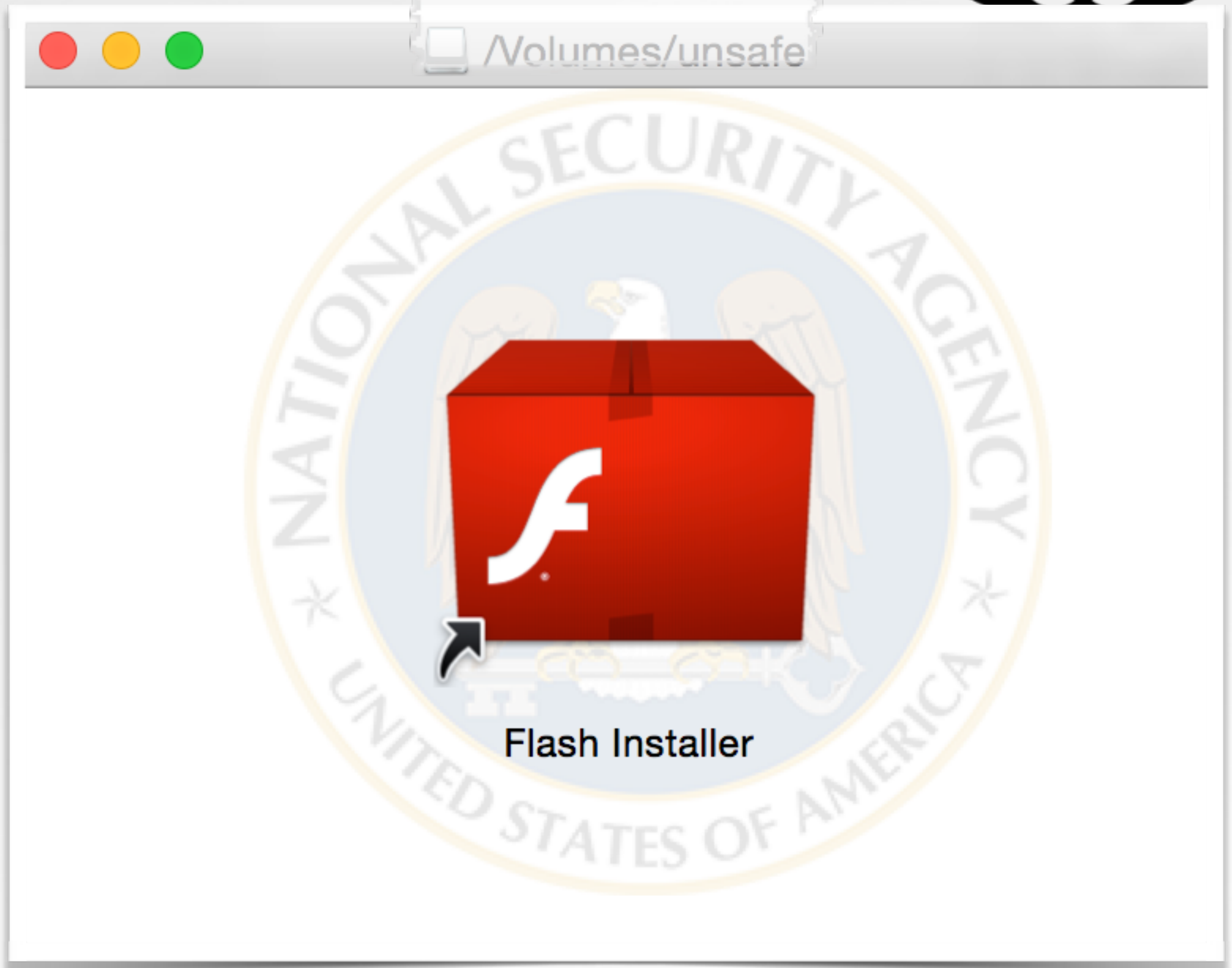
2 create a .dmg with the necessary layout


- Applications
  - Instruments.app
- Developer
- OtherFrameworks
- SharedFrameworks
  - CoreProfileDT.framework
  - CoreSimulator.framework
    - Versions
      - A
        - CoreSimulator

required directory structure

'clean up' the .dmg

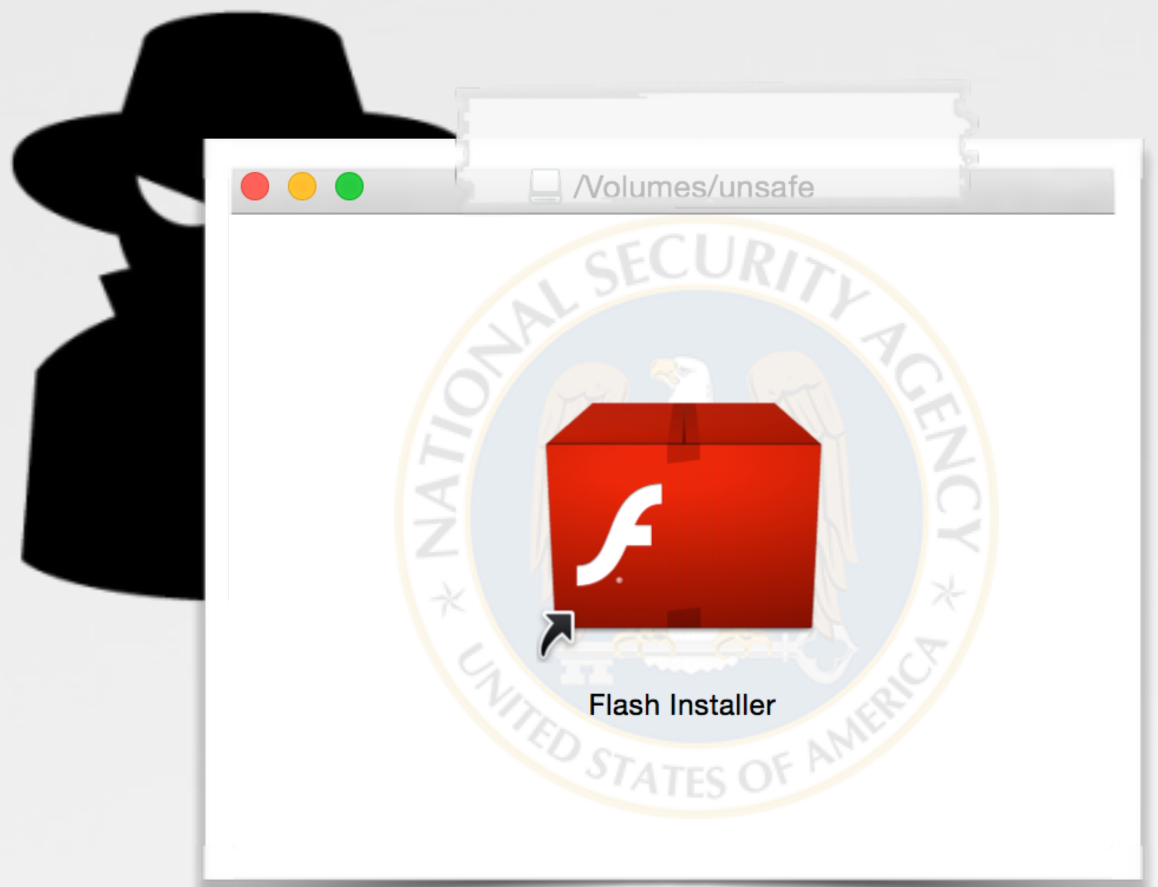
- hide files/folder
- set top-level alias to app
- change icon & background
- make read-only

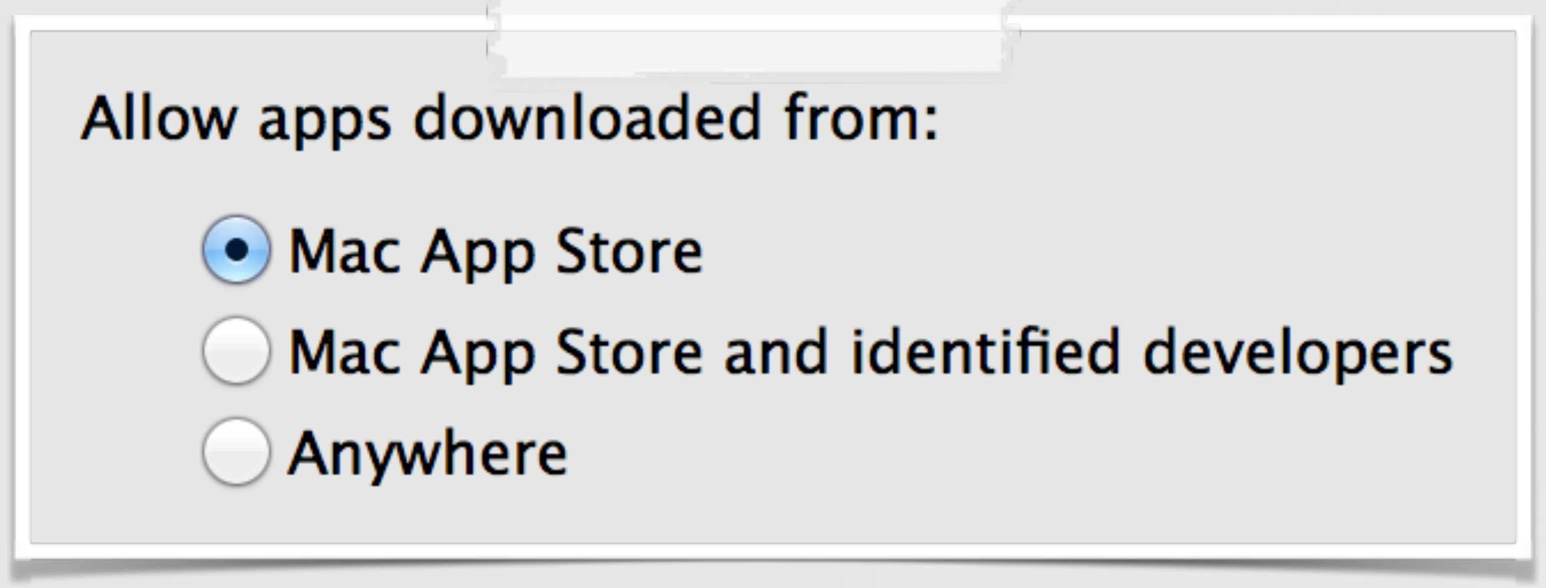


(deployable) malicious .dmg

Gatekeeper Bypass 0x1 (CVE 2015-3715)

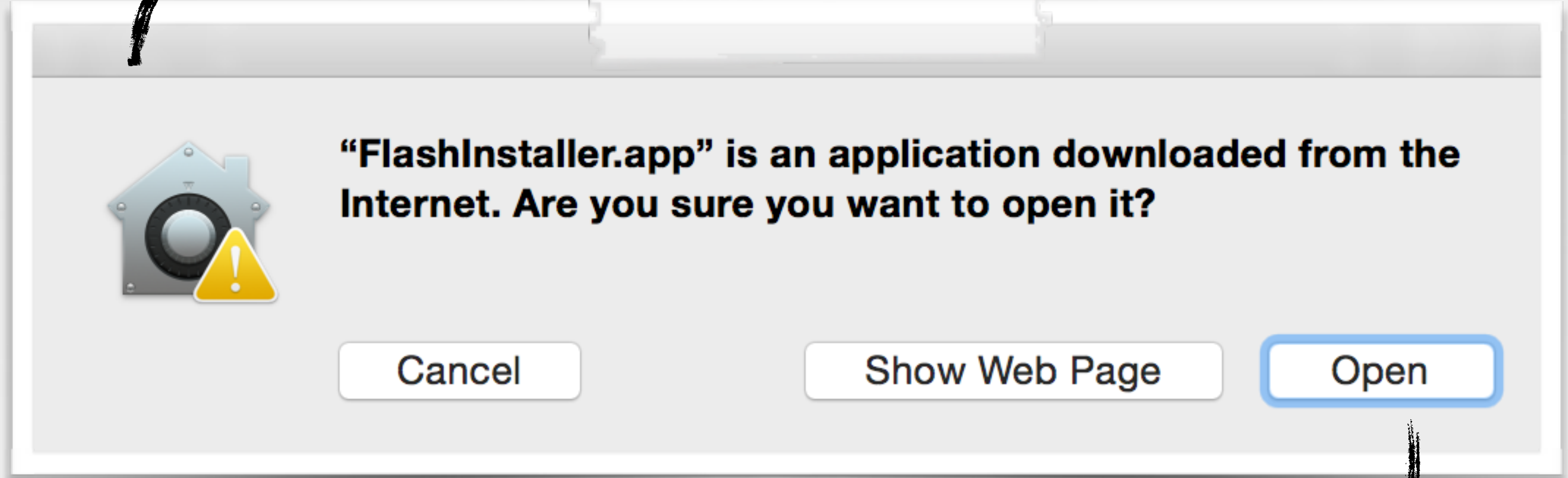
3 host online or inject into downloads




Allow apps downloaded from:
[● Mac App Store]
○ Mac App Store and identified developers
○ Anywhere

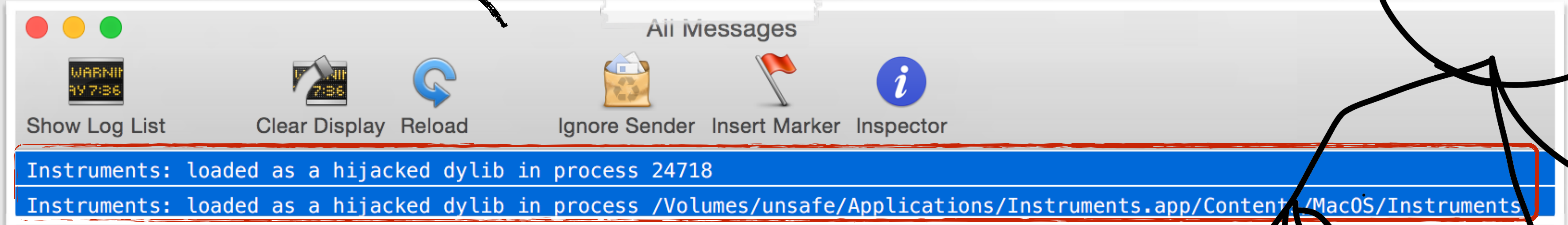
gatekeeper setting's (maximum)

quarantine popup (anything downloaded)


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

quarantine alert

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



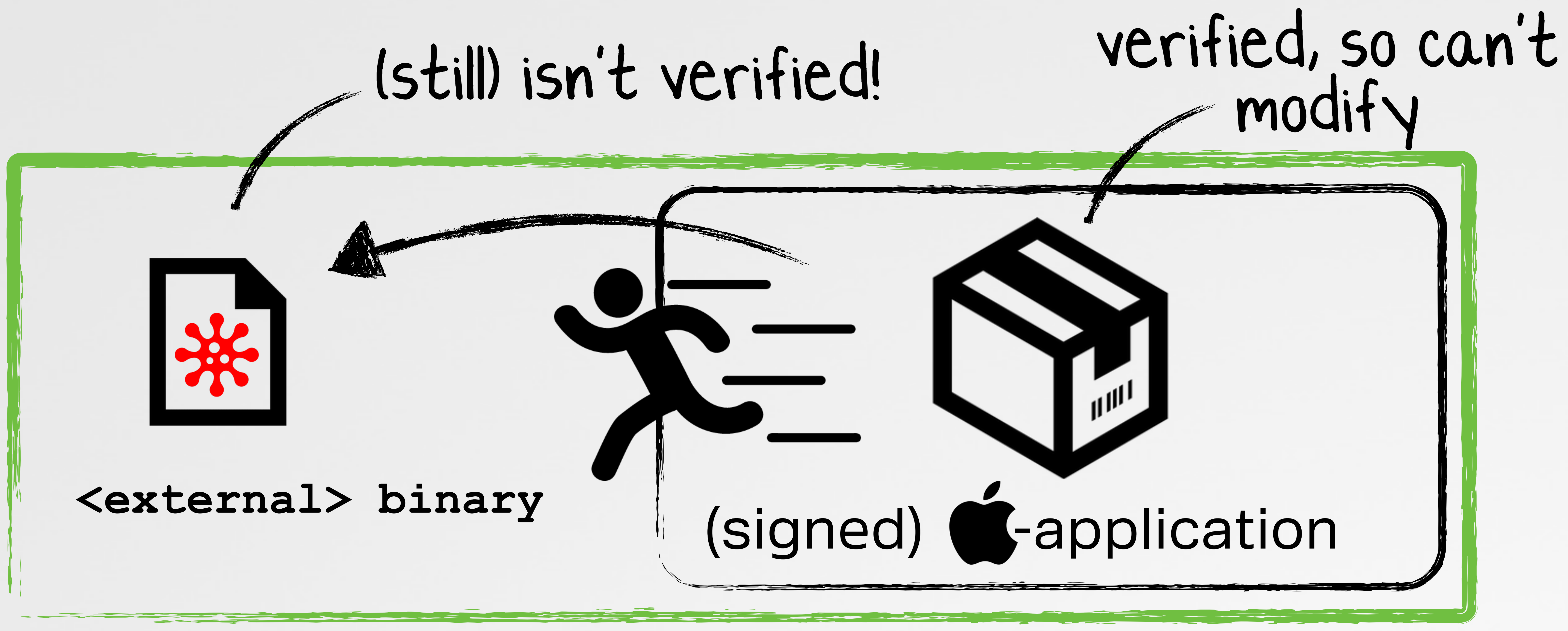
```
Instruments: loaded as a hijacked dylib in process 24718
Instruments: loaded as a hijacked dylib in process /Volumes/unsafe/Applications/Instruments.app/Contents/MacOS/Instruments
```

gatekeeper bypass :)

Gatekeeper Bypass 0x2

runtime shenanigans

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



.dmg/.zip layout

1. find any signed app that **at runtime**, loads and executes a **relatively external** binary
2. create a .dmg/.zip with the necessary folder structure (i.e. placing the malicious binary in the **externally** referenced location)
3. host online/inject into insecure downloads

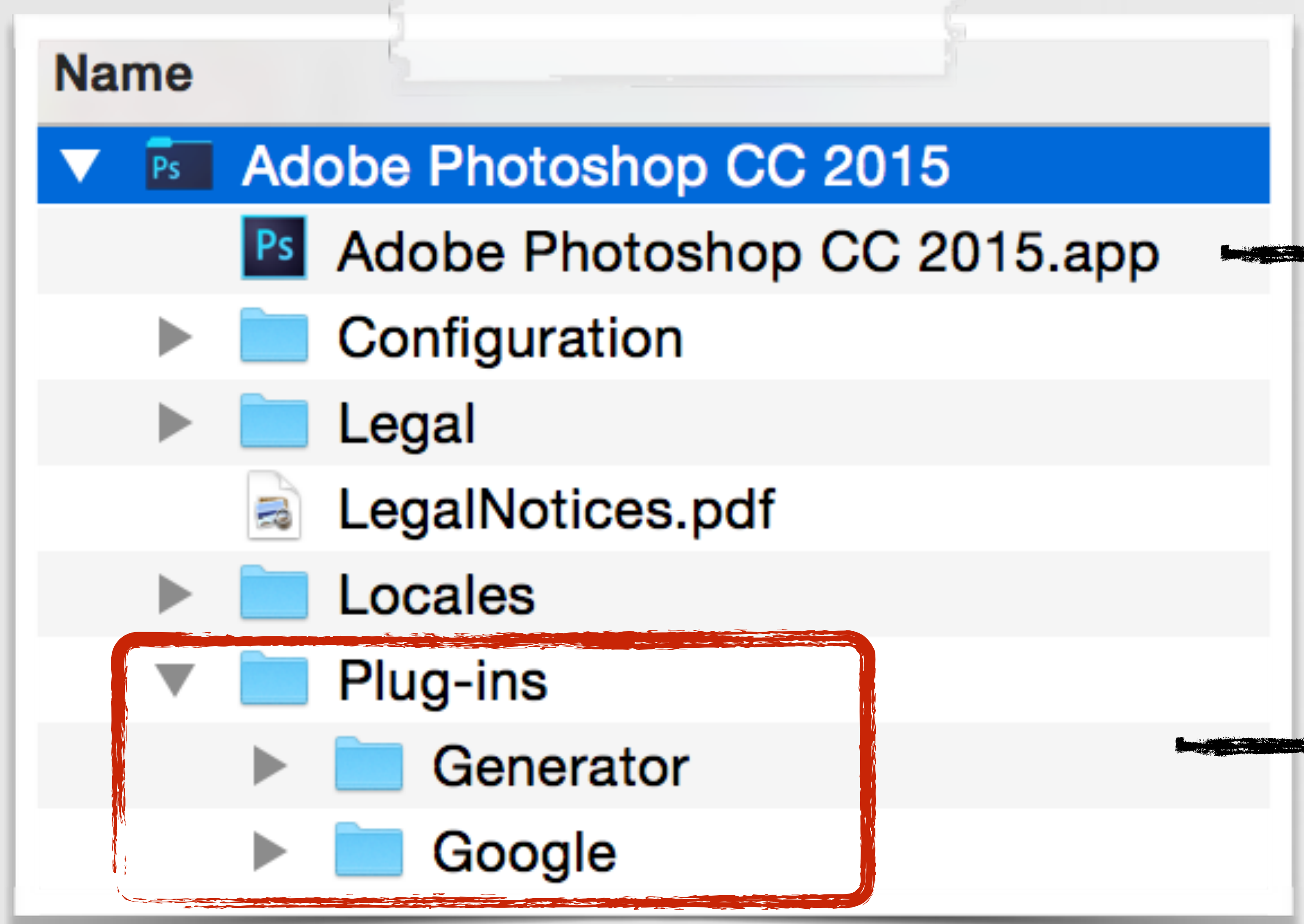
Gatekeeper Bypass 0x2

example 1: Adobe (Photoshop, etc)

3rd party plugins, etc.
-> go outside the bundle!

Q: Can I add/modify files in my signed (app) bundle?

A: "*This is no longer allowed. If you must modify your bundle, do it before signing. If you modify a signed bundle, you must re-sign it afterwards. Write data into files outside the bundle*" -apple.com


Name
- Adobe Photoshop CC 2015
  - Adobe Photoshop CC 2015.app
  - Configuration
  - Legal
  - LegalNotices.pdf
  - Locales
  - Plug-ins
    - Generator
    - Google

Adobe Photoshop

app bundle validates!

not validated

```
NSString* pluginDir = APPS_DIR + @"../Plug-ins";

for(NSString* plugins in pluginDir)
{
    //load plugin dylib!
}
```

plugin loading pseudo code

movie time! { iWorm vs. gatekeeper

Synack

Adobe Photoshop 2014
Adobe Photoshop 2015
Security & Privacy
Search
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...]

KnockKnock (UI)
KnockKnock
Start Scan

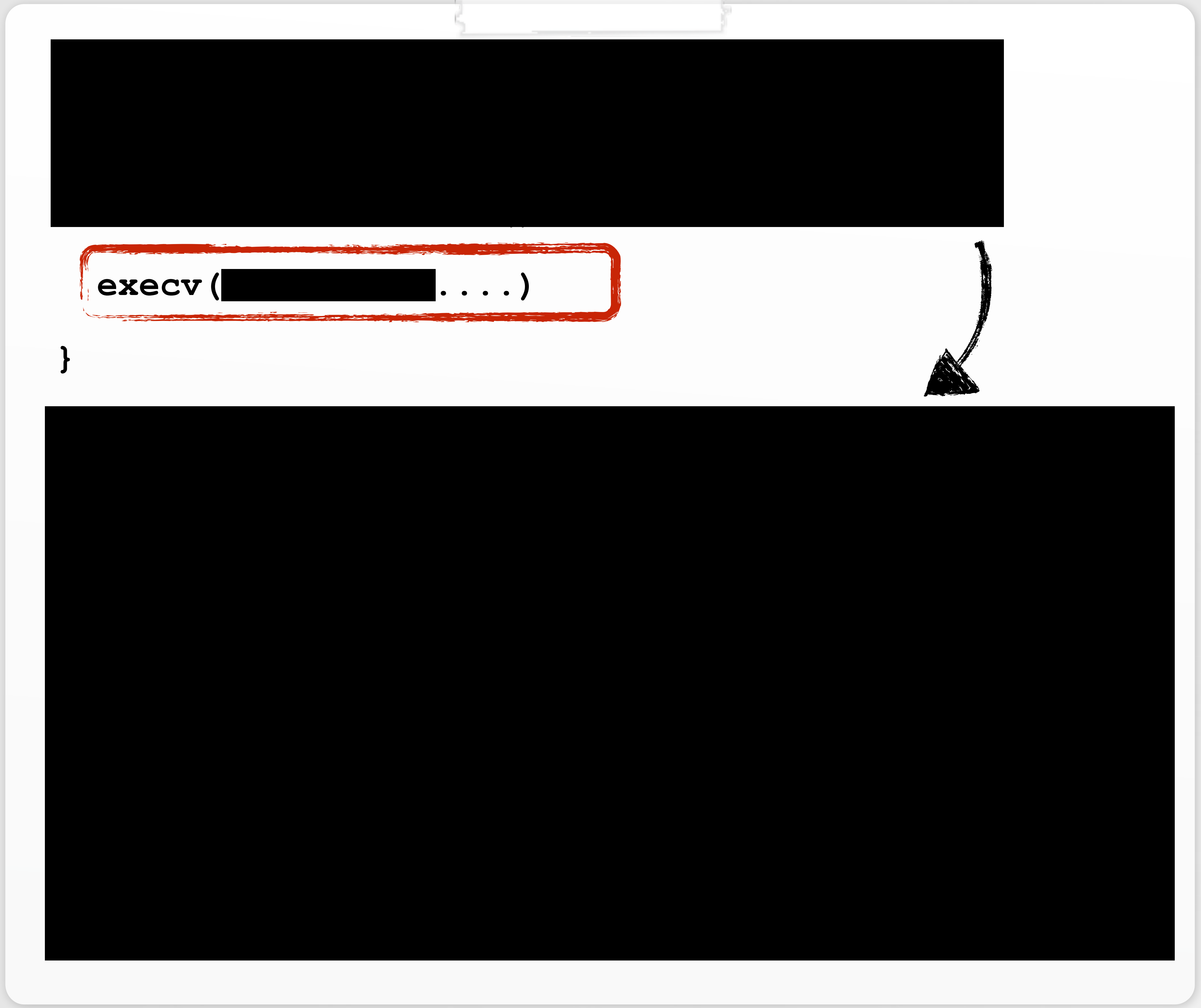
Authorization Plugins	registered custom authorization bundles	0
Browser Extensions	plugins/extensions hosted in the browser	0
Cron Jobs	current users cron jobs	0
Launch Items	daemons and agents loaded by launchd	6
Library Inserts	dylibs inserted via DYLD_INSERT_LIBRARIES	0
Login Items	items started when the user logs in	0
Spotlight Importers		1

		virustotal
check-aliases	/usr/libexec/postfix/check-aliases.sh /System/Library/LaunchDaemons/org.postfix.newaliases.plist	0/56
AdobeUpdateDaemon	/Library/Application Support/Adobe/Adobe Desktop Common/ElevationManager/AdobeUpdateDaemon /Library/LaunchDaemons/com.adobe.adobeupdatedaemon.plist	?
vmware-tools-daemon	/Library/Application Support/VMware Tools/vmware-tools-daemon /Library/LaunchDaemons/com.vmware.launchd.tools.plist	0/57
UpdaterStartupUtility	/Library/Application Support/Adobe/OOBE/PDApp/UWA/UpdaterStartupUtility /Library/LaunchAgents/com.adobe.AAM.Updater-1.0.plist	0/56
Creative Cloud	/Applications/Utilities/Adobe Creative Cloud/ACC/Creative Cloud.app/Conten.../Creative Cloud /Library/LaunchAgents/com.adobe.AdobeCreativeCloud.plist	0/56
vmware-tools-daemon	/Library/Application Support/VMware Tools/vmware-tools-daemon /Library/LaunchAgents/com.vmware.launchd.vmware-tools-userd.plist	0/57

```
users-Mac:~ user$ xattr -p com.apple.quarantine ~/Downloads/Adobe\ Photoshop\ CC*
```

Gatekeeper Bypass 0x2

example 2: Apple ([redacted])

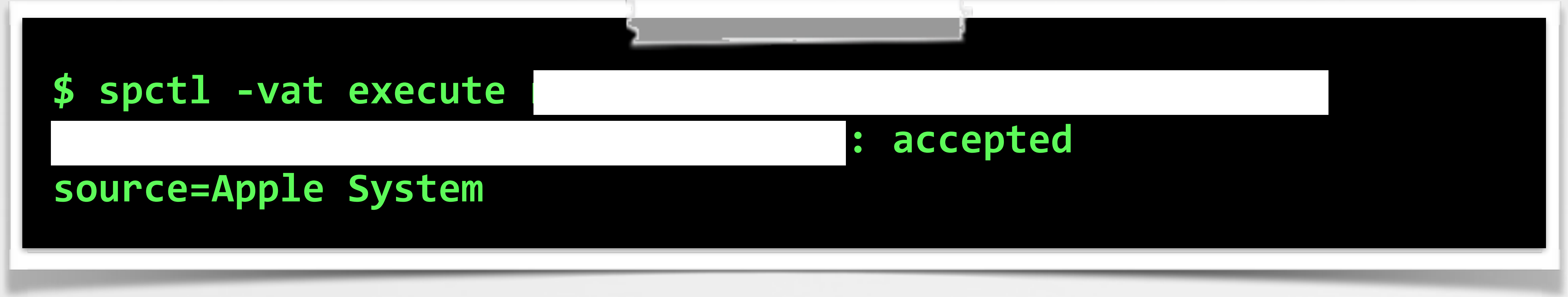


[redacted]

```
execv([redacted]....)
}
```

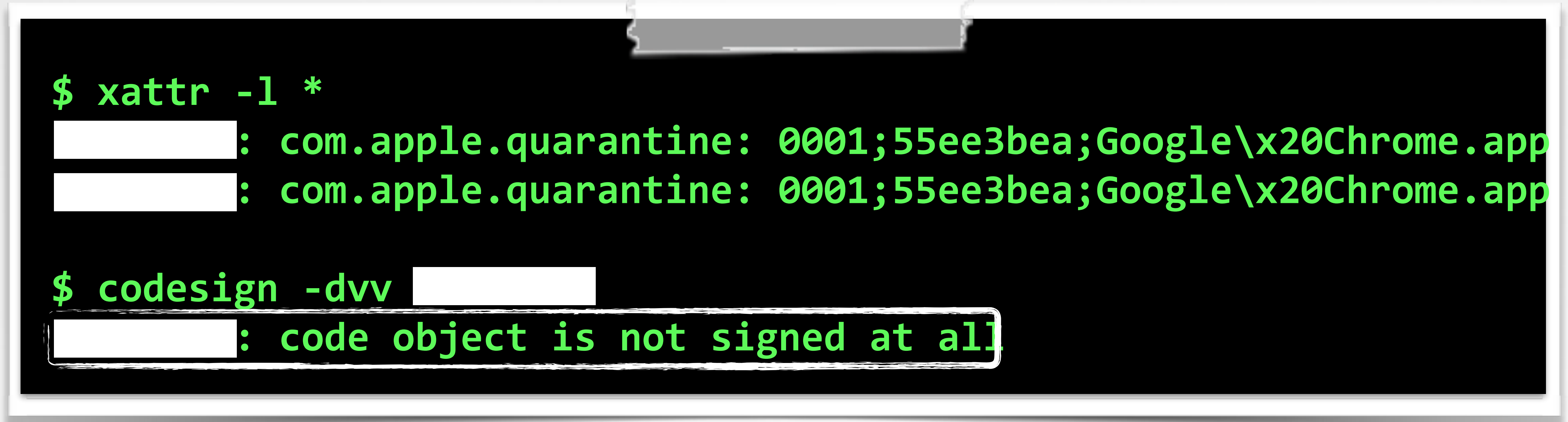
[redacted]

[redacted]'s pseudo code



```
$ spctl -vat execute [redacted]
[redacted]: accepted
source=Apple System
```

gatekeeper, happy with [redacted]

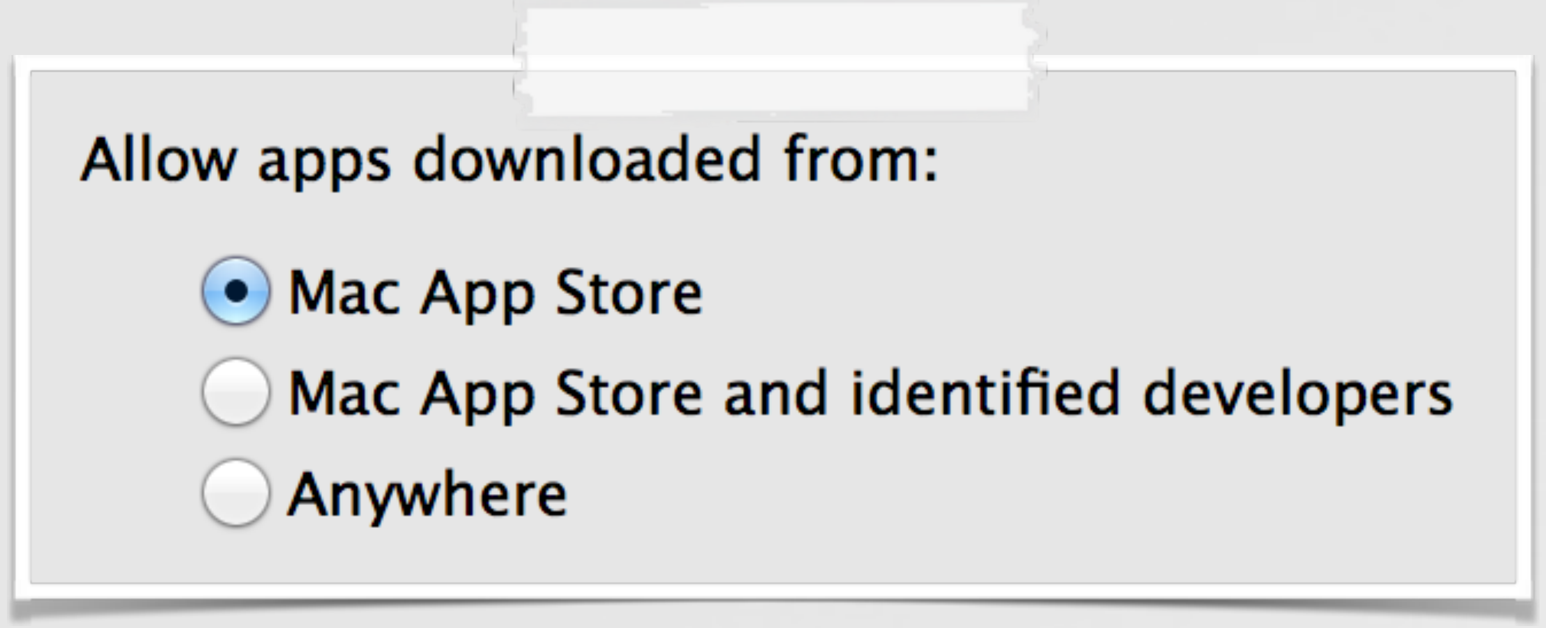


```
$ xattr -l *
[redacted]: com.apple.quarantine: 0001;55ee3bea;Google\x20Chrome.app
[redacted]: com.apple.quarantine: 0001;55ee3bea;Google\x20Chrome.app

$ codesign -dvv [redacted]
[redacted]: code object is not signed at all
```

...but [redacted] is unsigned

Gatekeeper Bypass 0x2

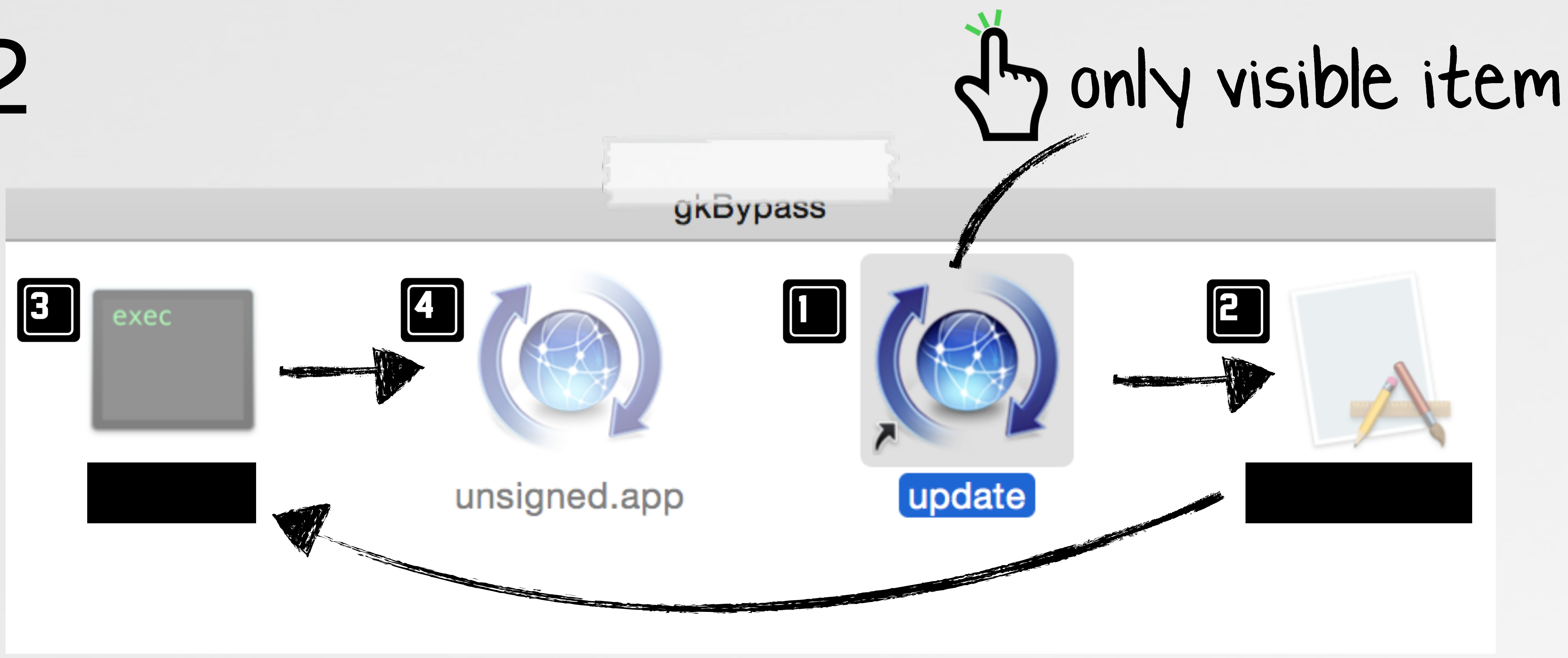
example 2: Apple ([redacted])



Allow apps downloaded from:

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

gatekeeper setting's (max.)



.dmg setup

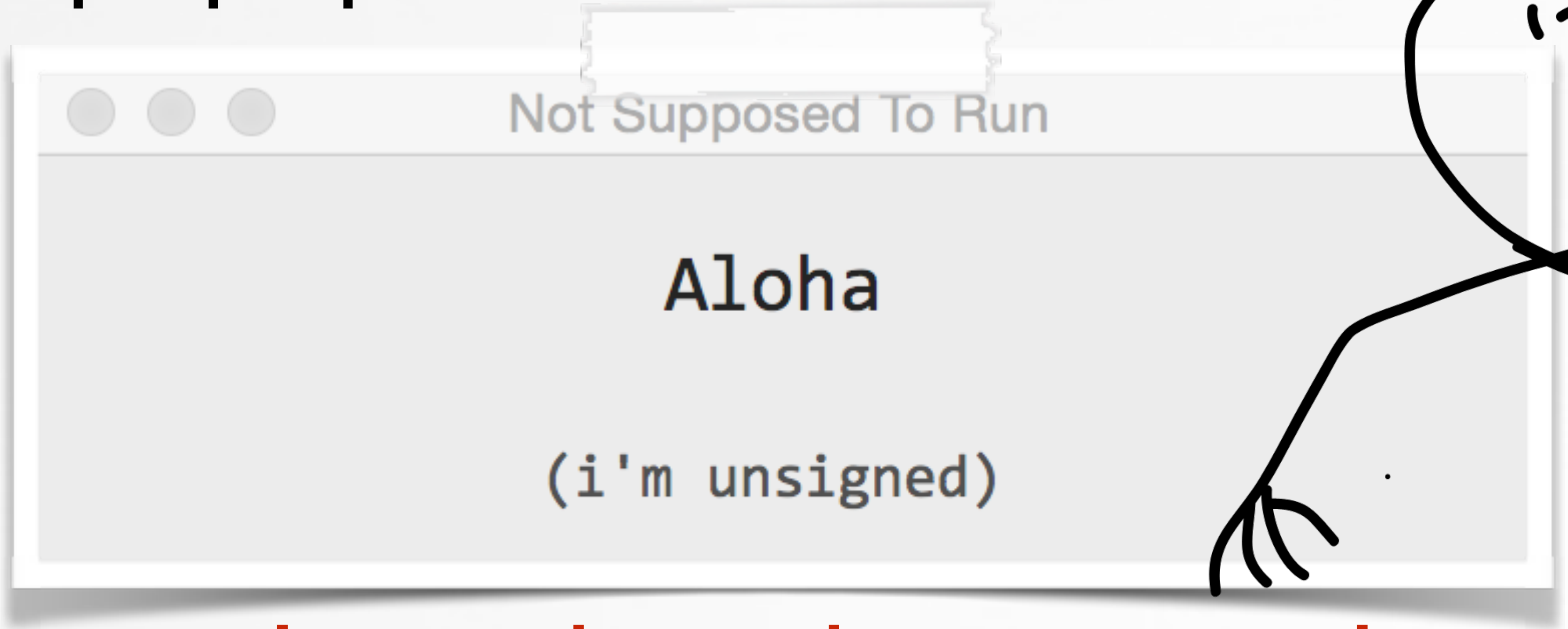
1 alias to [redacted]
...name & icon attacker controlled

2 apple-signed [redacted]
.app extension prevents **Terminal.app** popup

3 unsigned [redacted]
command-line executable

4 unsigned application

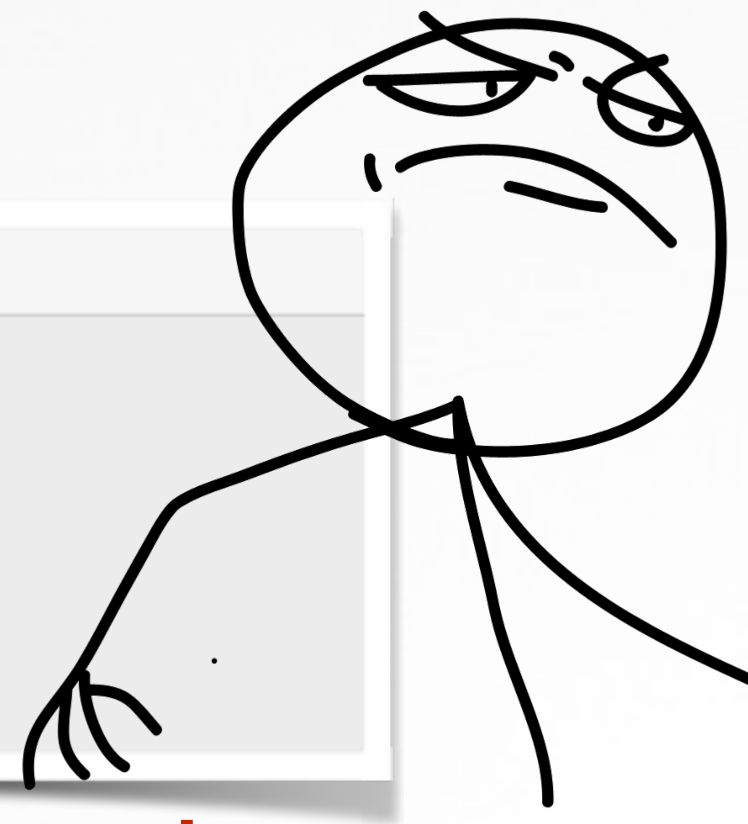
hide



Not Supposed To Run

Aloha

(i'm unsigned)

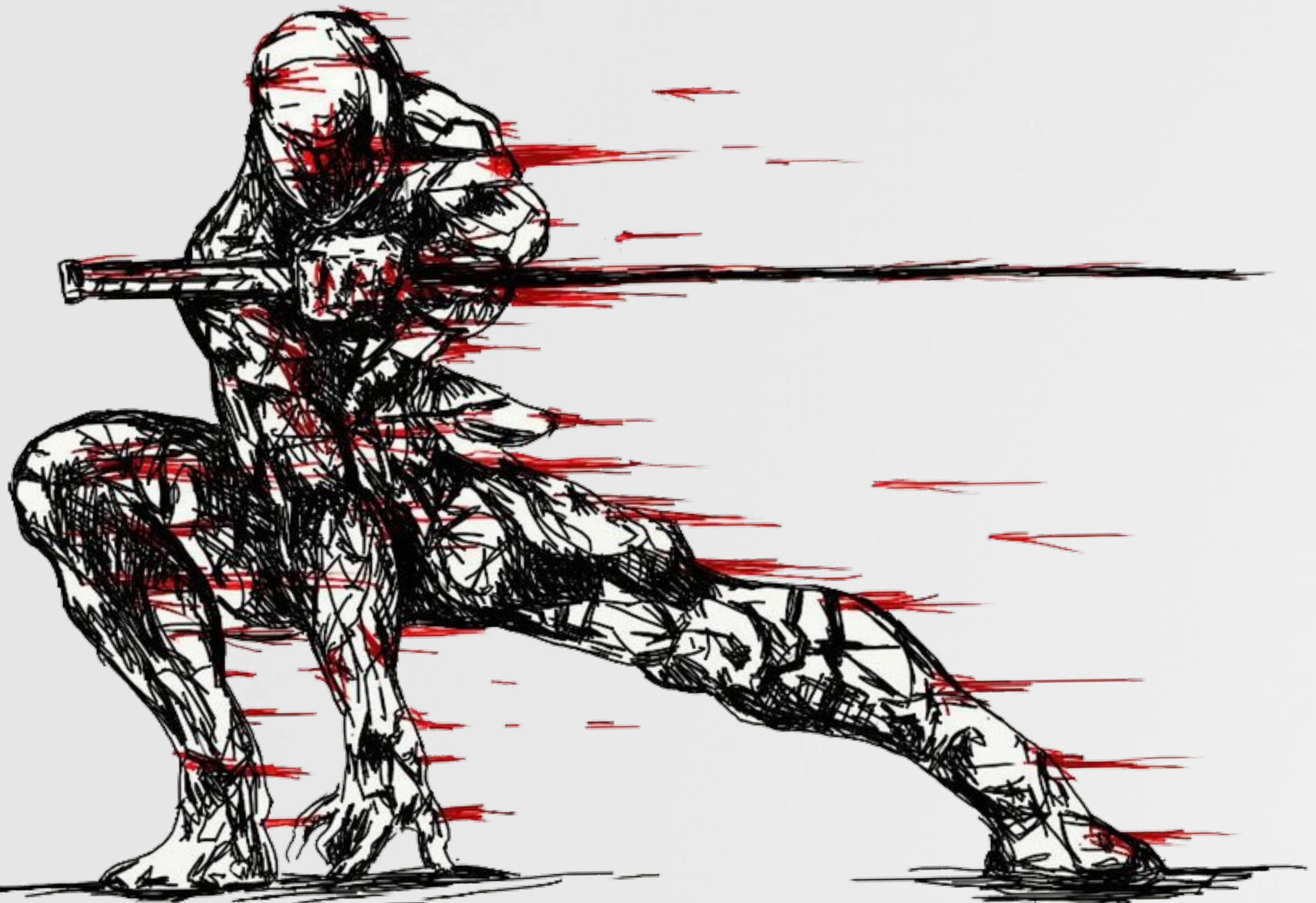


unsigned code execution

Synack

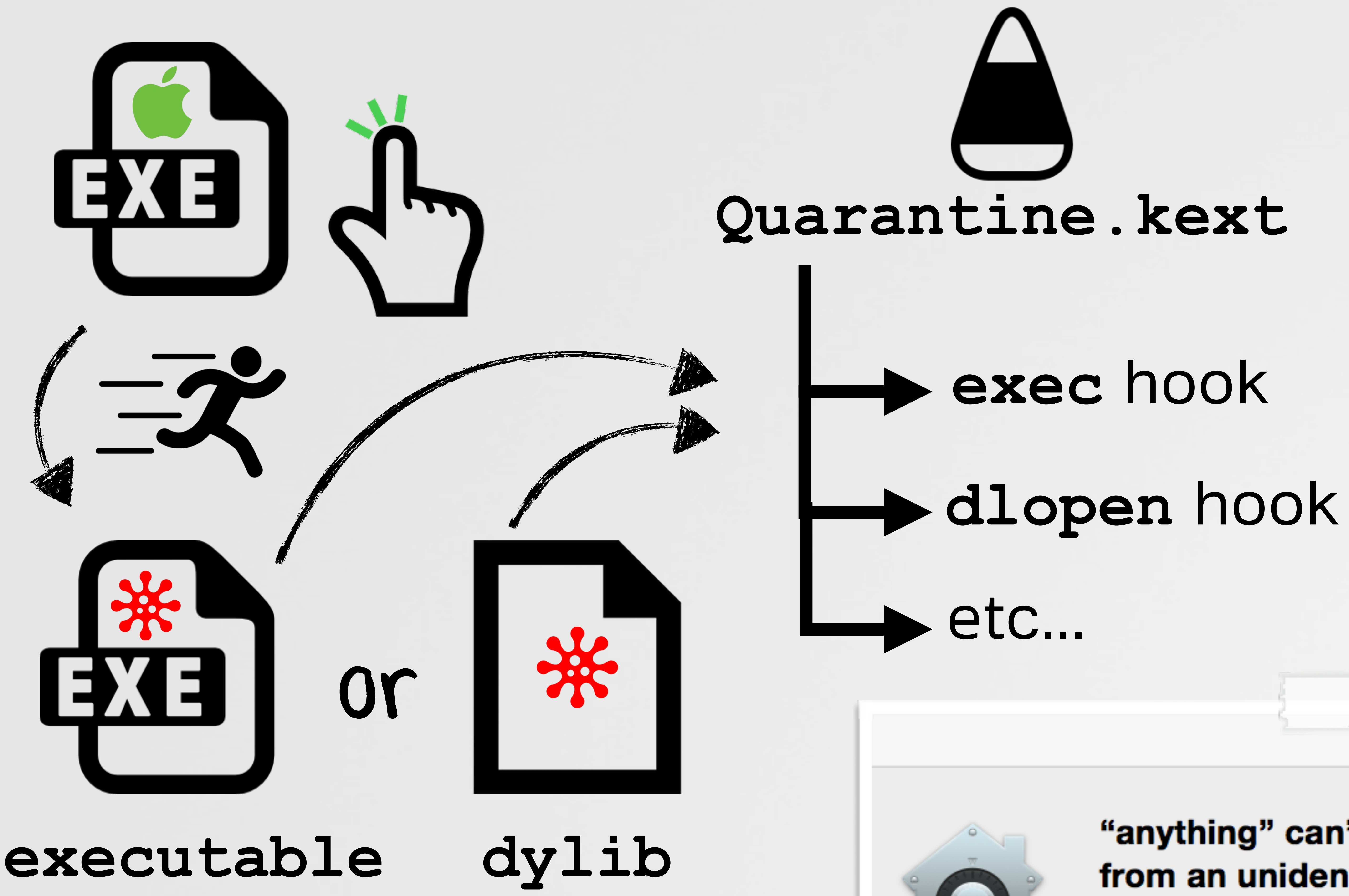
FIXING GATEKEEPER

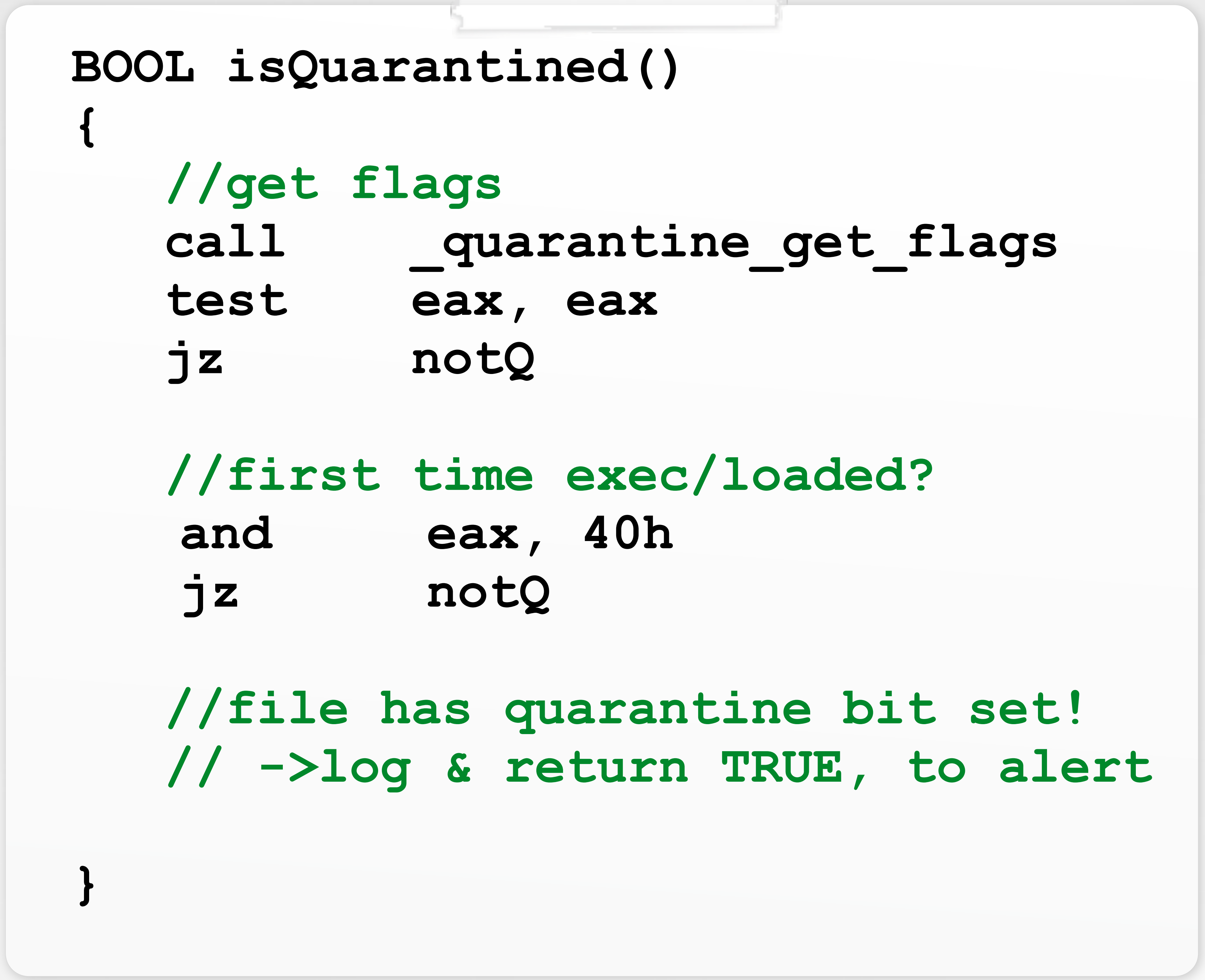
suggestion; runtime validation?



Validate All Binaries At Runtime!

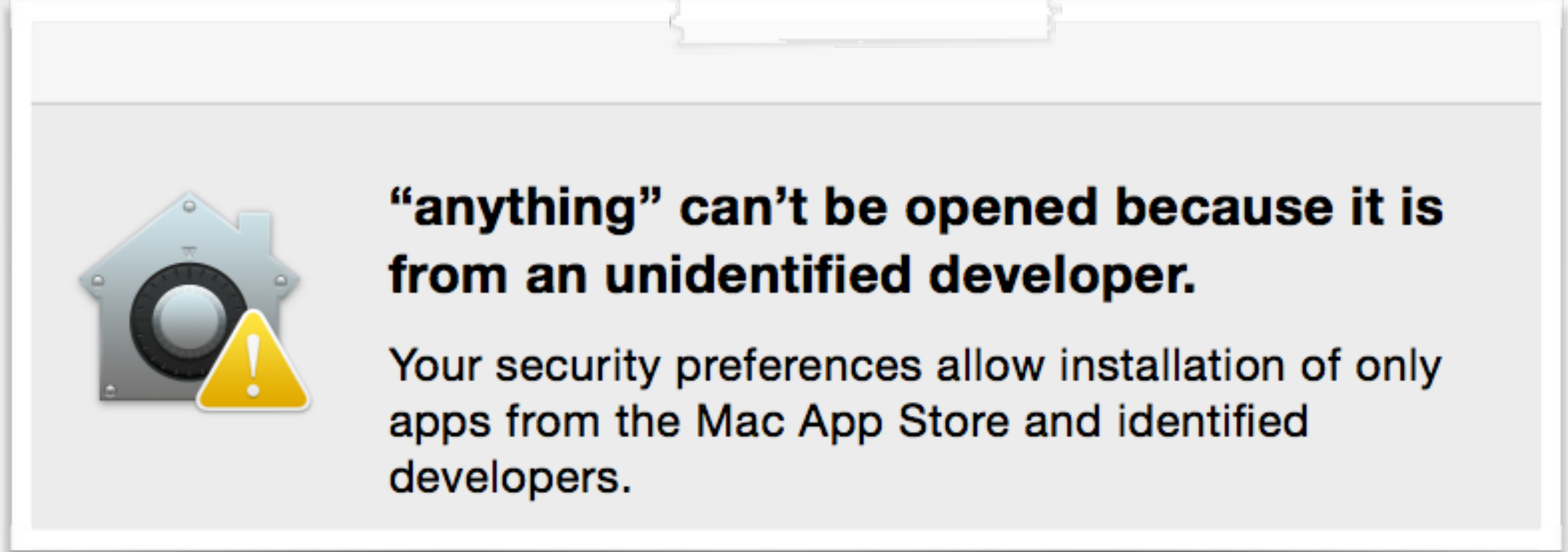
a suggestion to thwart runtime bypasses?





```
BOOL isQuarantined()
{
   //get flags
   call     _quarantine_get_flags
   test     eax, eax
   jz       notQ

   //first time exec/loaded?
    and      eax, 40h
    jz       notQ

   //file has quarantine bit set!
   // ->log & return TRUE, to alert

}
```

checking quarantine attribute



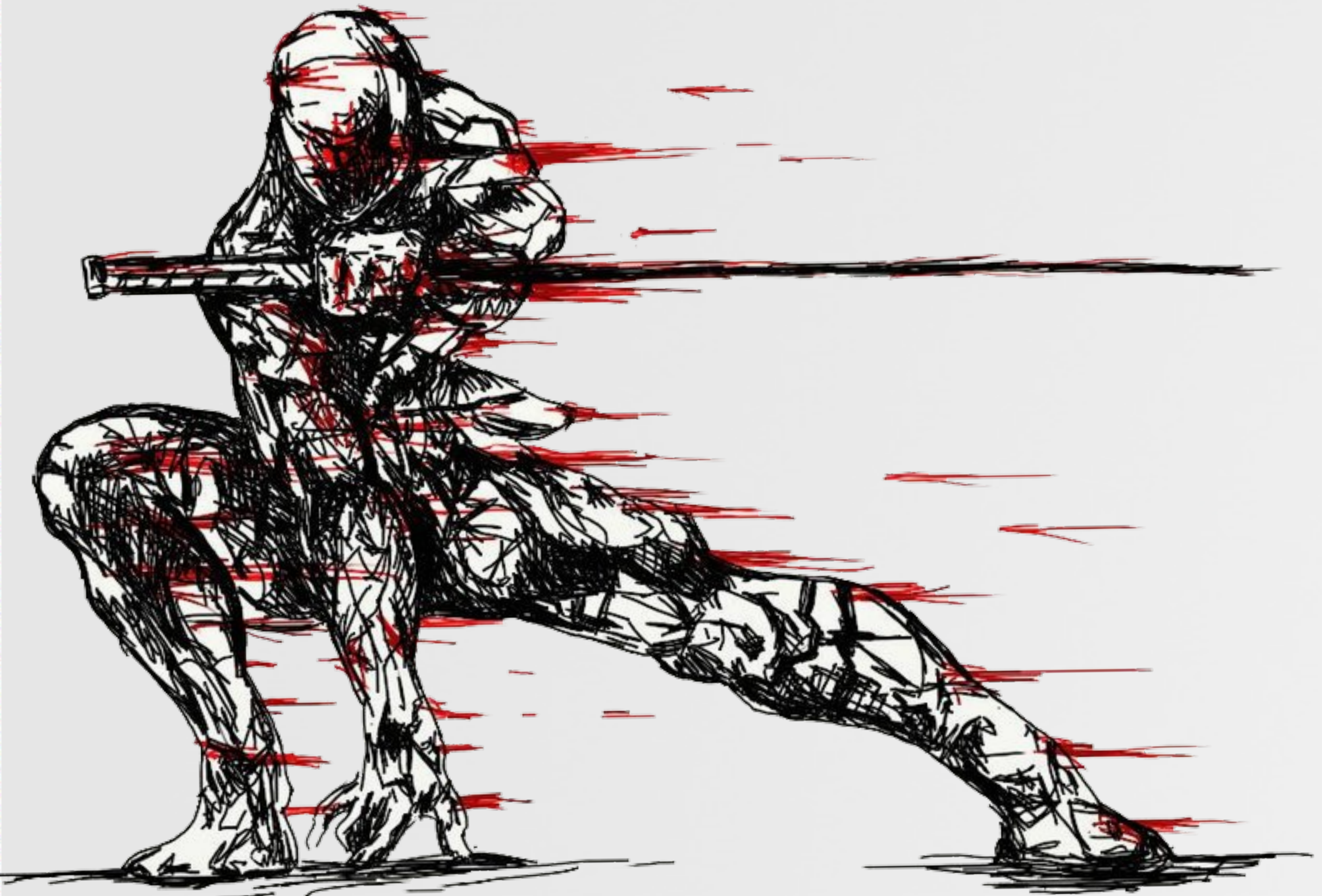
“anything” can’t be opened because it is from an unidentified developer.

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

alert on all unauthorized code

CONCLUSIONS

wrapping it up



GATEKEEPER

theory (or, apple marketing)

protects naive OS X users
from attackers

"*omg, my mac is so secure, no need for AV*" -mac users

GATEKEEPER

the unfortunate reality

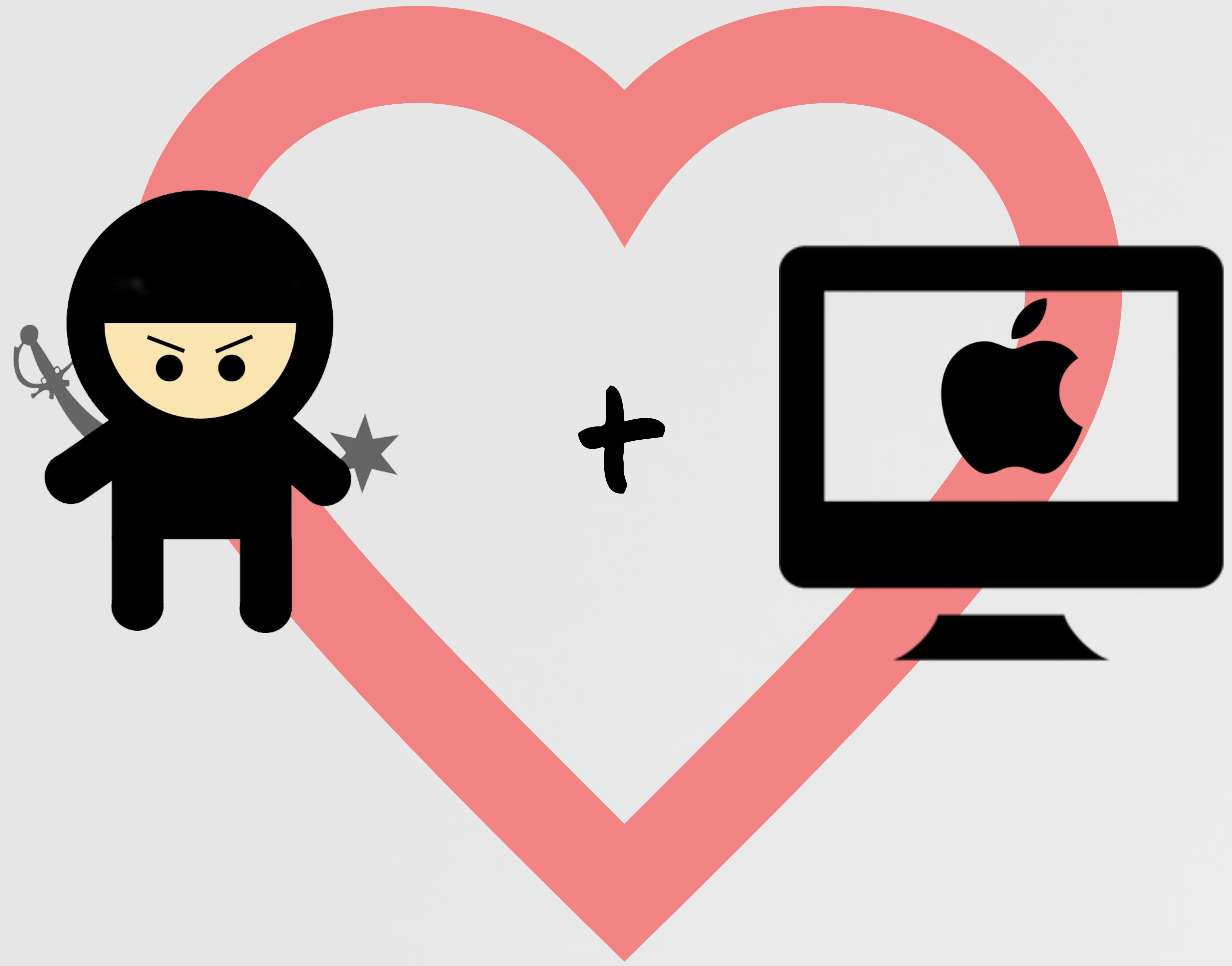
protects naive OS X users
from lame attackers

& false sense
of security?

highly recommend;
3rd-party security tools

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 :)

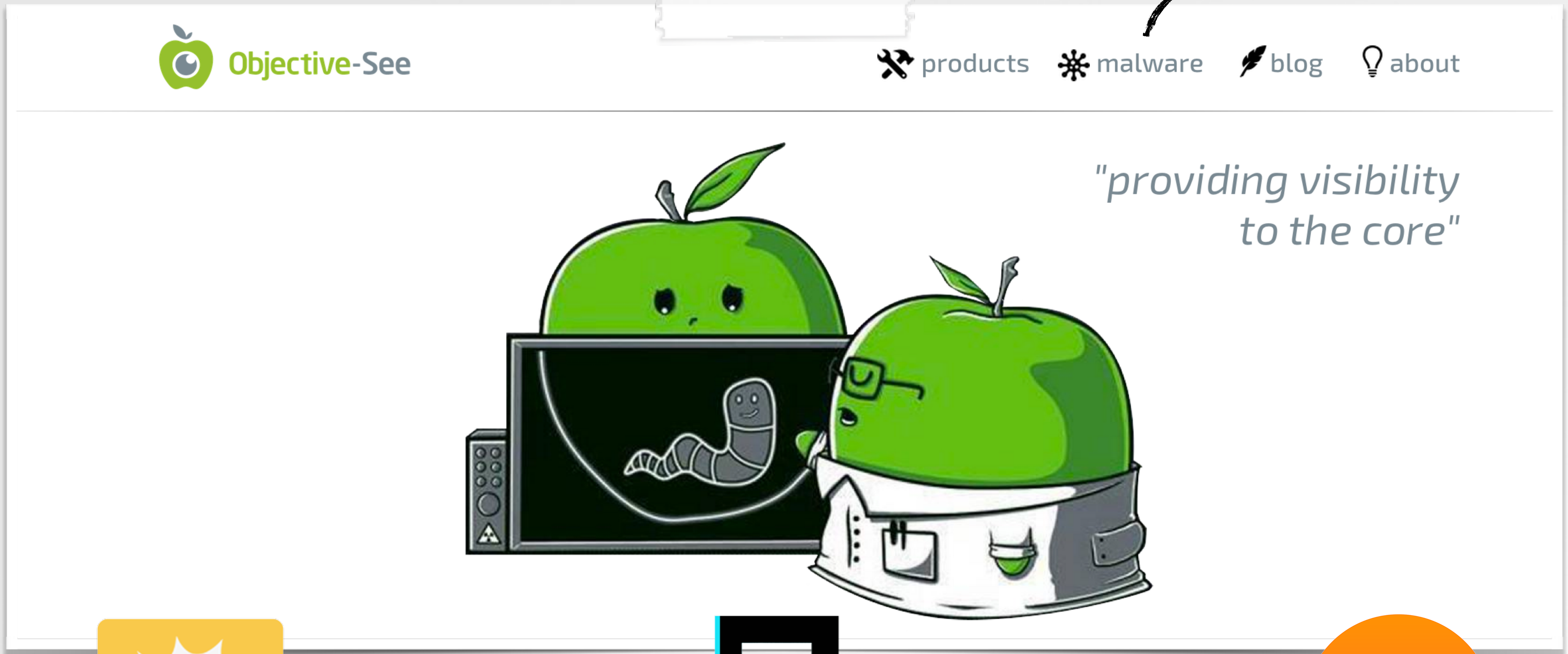
i beg to differ!

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

OBJECTIVE-SEE

free security tools & malware samples

os x malware samples



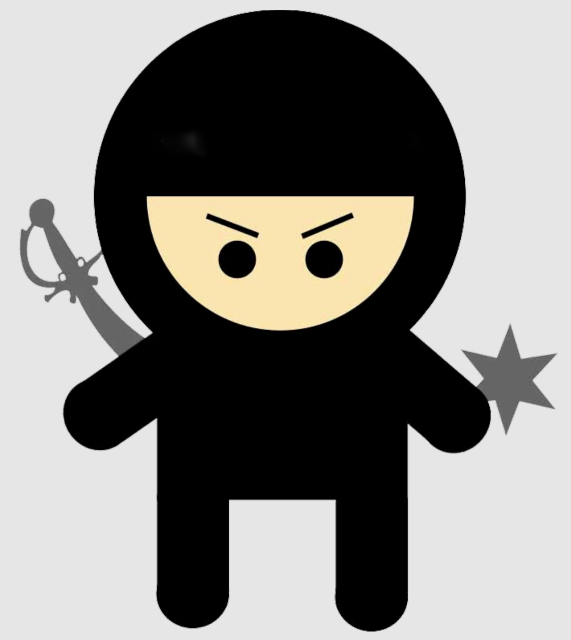
KnockKnock

BlockBlock

TaskExplorer

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

- http://wirdou.com/2012/02/04/is-that-bad-doctor/
- thezooom.com
- http://th07.deviantart.net/fs70/PRE/f/2010/206/4/4/441488bcc359b59be409ca02f863e843.jpg

- iconmonstr.com
- flaticon.com

resources

- thesafemac.com
- "Mac OS X & iOS Internals", Jonathan Levin
- https://securosis.com/blog/os-x-10.8-gatekeeper-in-depth

Synack