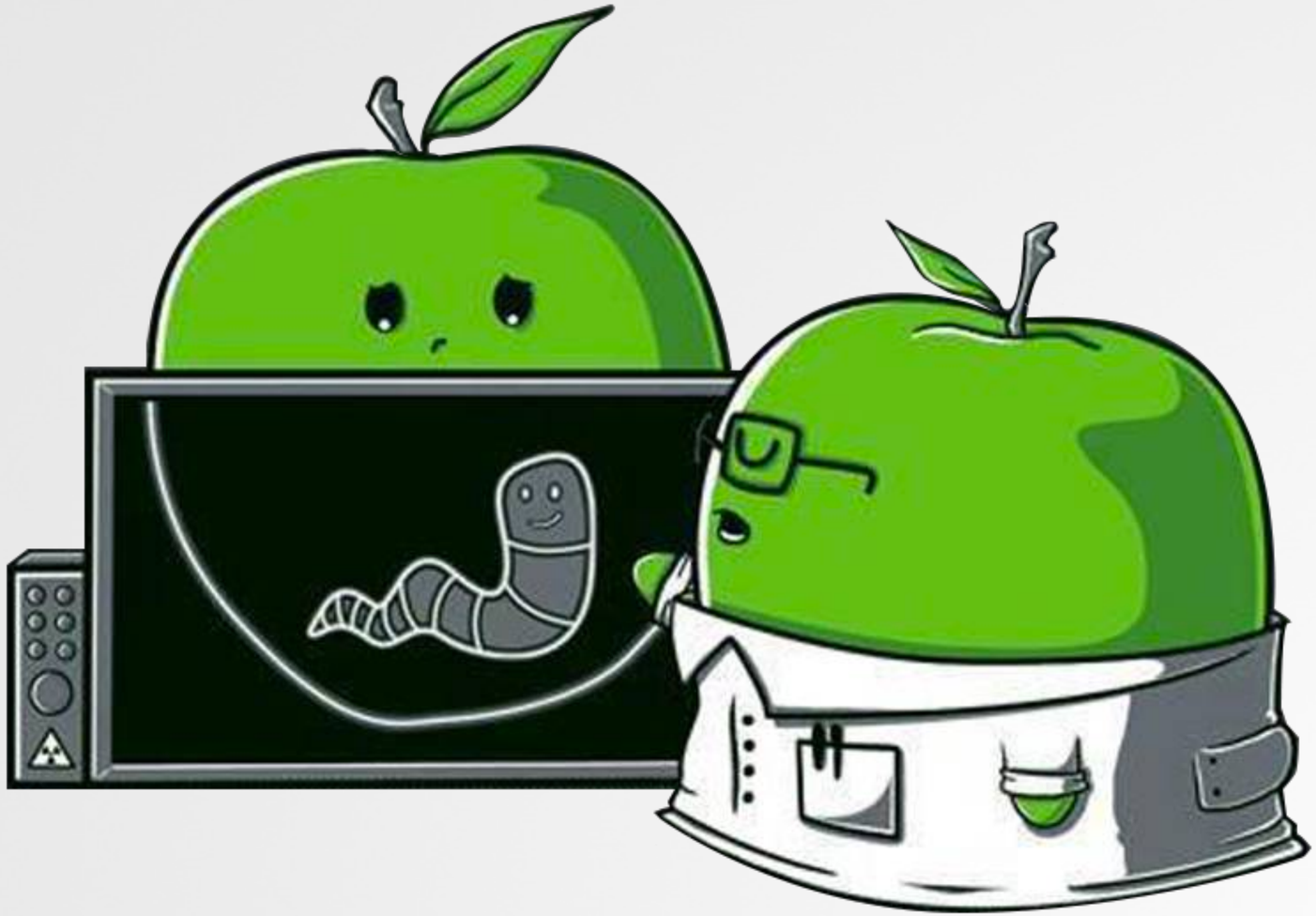
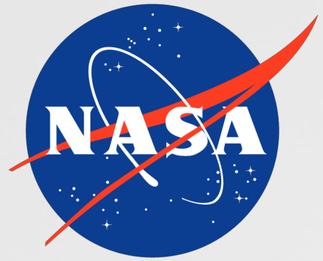
Protecting the Garden of Eden



WHOIS

@patrickwardle



nasa

nsa

synack

digita

Objective-See

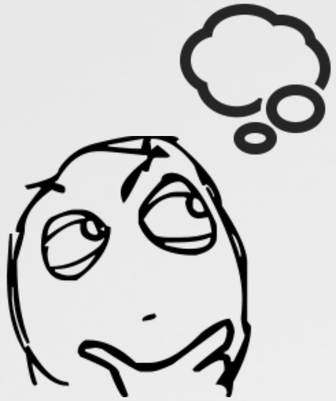
digita security

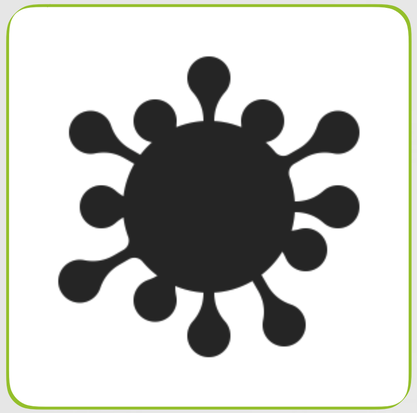
cybersecurity solutions for the mac enterprise

Outline

PART I: threats

PART II: protections





malware

bugs

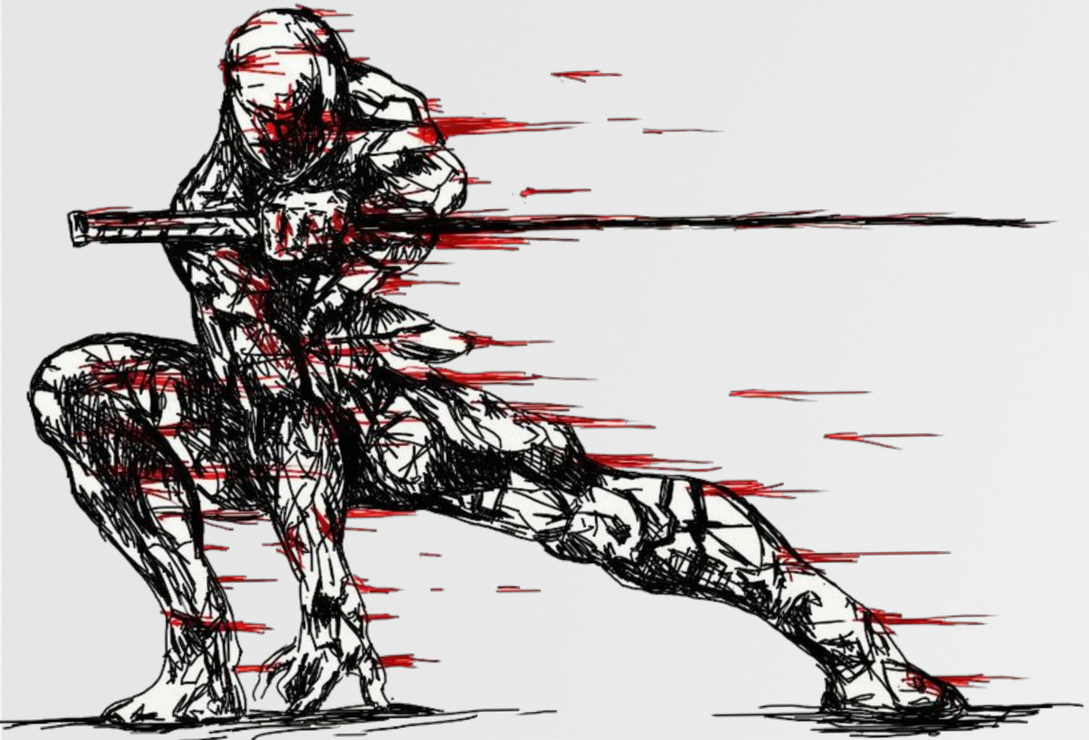
mac

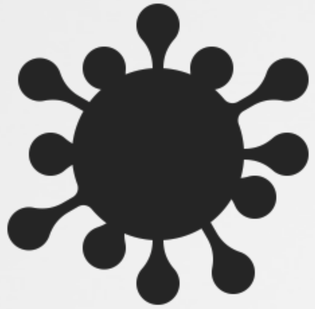
3rd-party

(mac) musings

Mac Malware

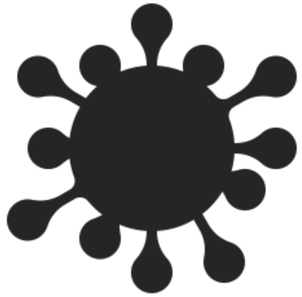
history & current trends





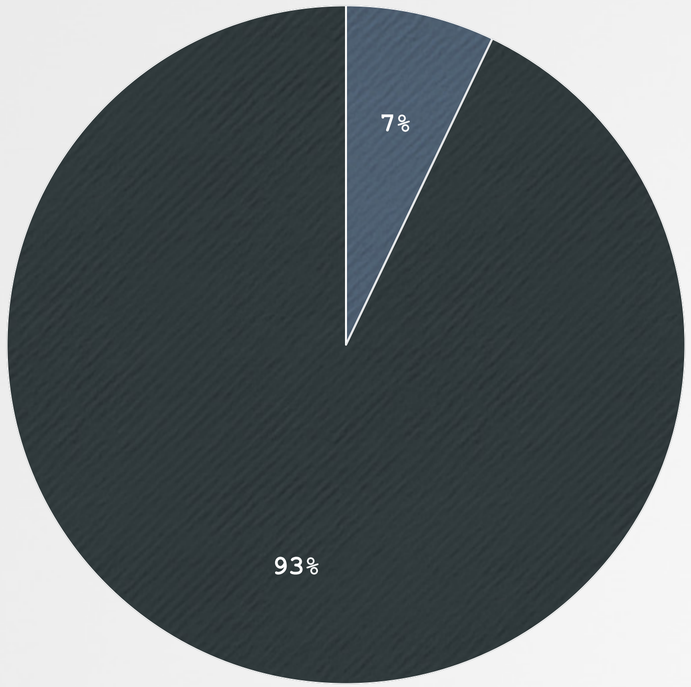
Digita Security

The Reality



Macs are no more resistant to malware than their (modern) Windows counterparts.

ok, then why not more mac malware!?

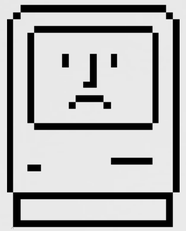


only 7.1% of PC Market (Q2 2018, Gartner)

Apple vs. Malware/Vulnerabilities

apple.com (2012)

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



1982

'first' in the wild virus infected apple II's

2014

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

2015

OS X most vulnerable software by CVE count -cve details

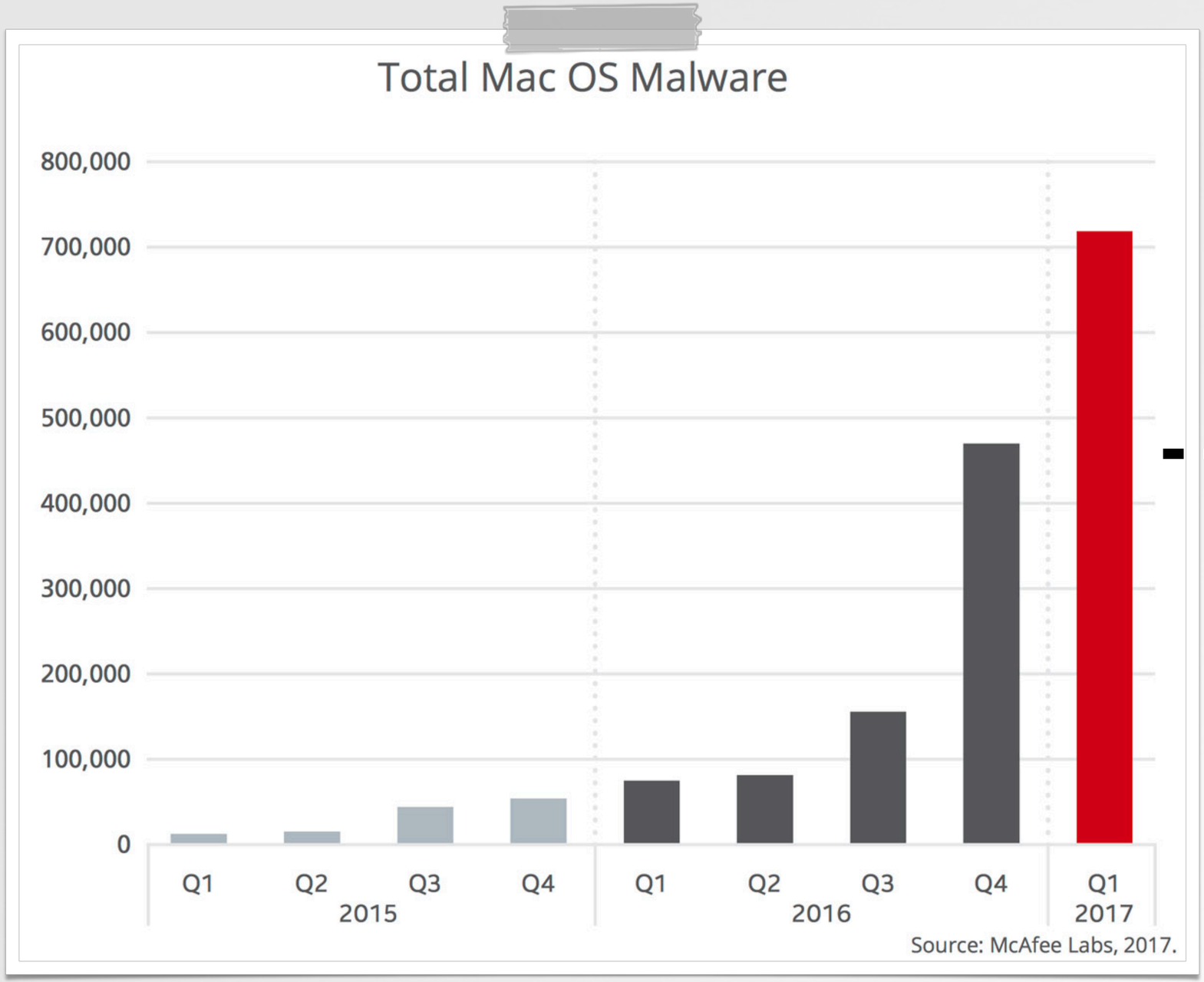
2015

"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

2017/2018

Mac-specific malware increased by 270% in 2017 compared with 2016, and four new Mac threats were detected in the first two months of 2018 -malwarebytes

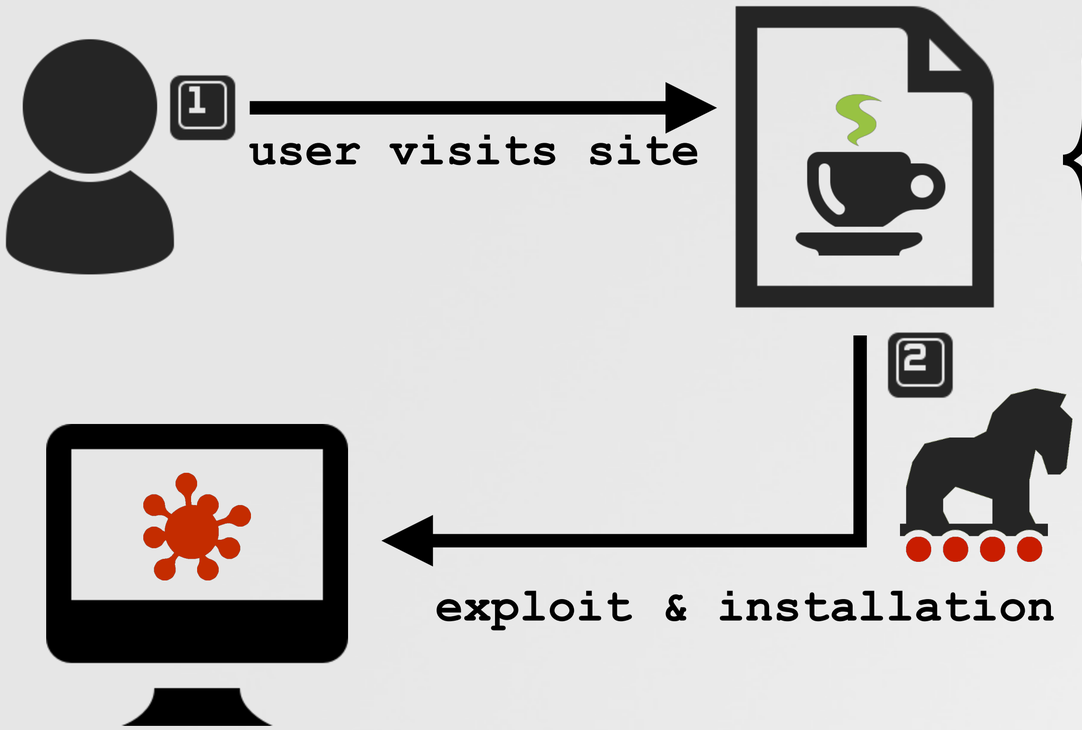
Apple vs. Malware



note:
includes adware

OSX.Flashback (2012)

infecting over 1/2 million macs, for 'black hat' SEO



automatic exploitation via java vulnerability (CVE-2012-0507)

apple failed to release a patch for 2 months :(

- - -▶ 700,000 infections, including 250+ in cupertino

"Flashback is not only the most advanced, but also the most successful OS X malware we've ever seen" -fSecure

OSX.XSLCMD (2014)

chinese apt backdoor, containing a neat 0day

remote shell

screen capture

keylogger

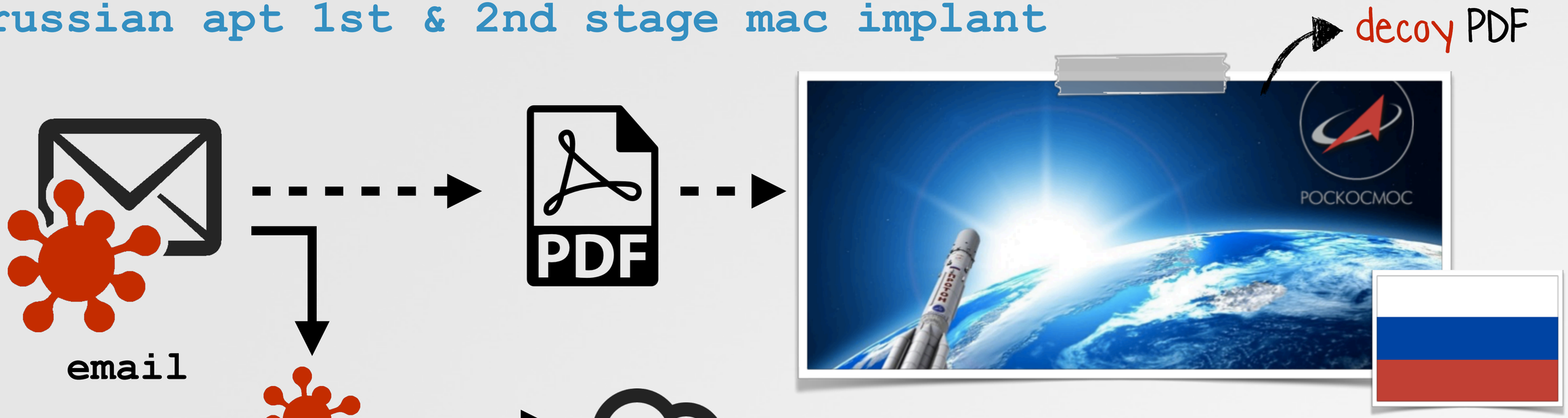
exploiting 'rootpipe' as an 0day!

```
void sub_10000c007()
r12 = [Authenticator sharedAuthenticator];
rax = [SFAuthorization authorization];
rbx = rax;
rax = [rax obtainWithRight:"system.preferences" flags:0x3 error:0x0];
if (rax != 0x0) {
   [r12 authenticateUsingAuthorizationSync:rbx];
   rax = [r12 isAuthenticated];
   if (rax != 0x0) {
      rbx = [NSDictionary dictionaryWithObject:@(0x124) forKey:*_NSFilePosixPermissions];
      rax = [NSData dataWithBytes:"a" length:0x1];
      rax = [UserUtilities createFileWithContents:rax path:@"/var/db/.AccessibilityAPIEnabled" attributes:rbx];
```

exploit
(gains root privileges)

OSX.Komplex / OSX.XAgent (2016)

russian apt 1st & 2nd stage mac implant



OS/XAgent install vector

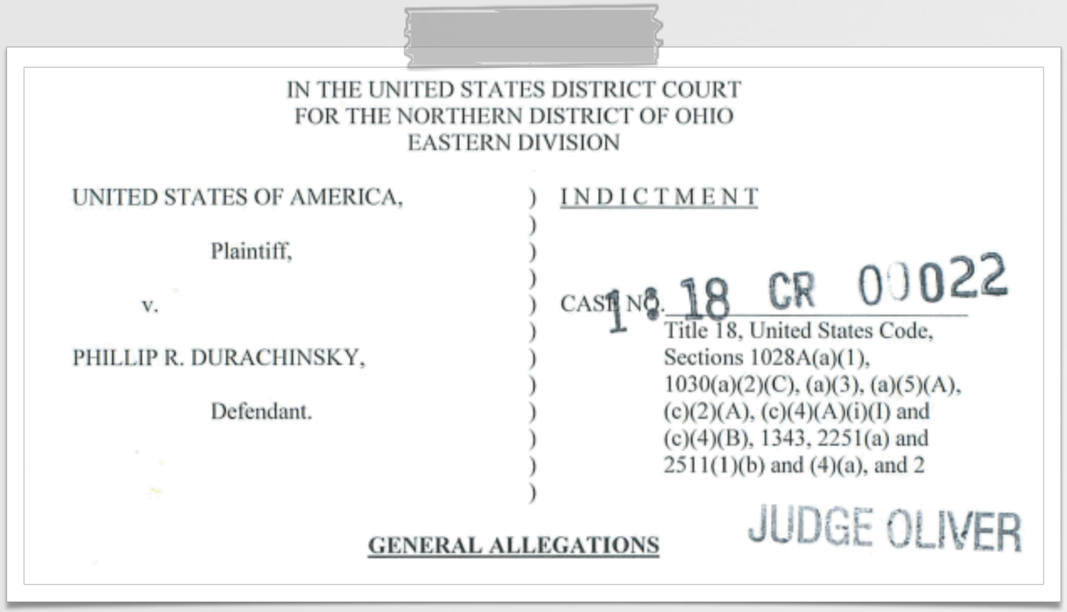
- OSX/Komplex (via email)
  - checks in w/ C&C
  - installs OSX/XAgent

OSX.FruitFly (2017)

spying on mac users (& children) for over a decade

```
$ cat fpsaud
#!/usr/bin/perl use strict;use warnings;use IO::Socket;use
IPC::Open2;my$1;sub G{die if!defined syswrite$1,$_[0]}sub J{my($U,
$A)=('','');while($_[0]>length$U){die if!sysread$I,$A,$_[0]-......
```

backdoor (obfuscated perl)

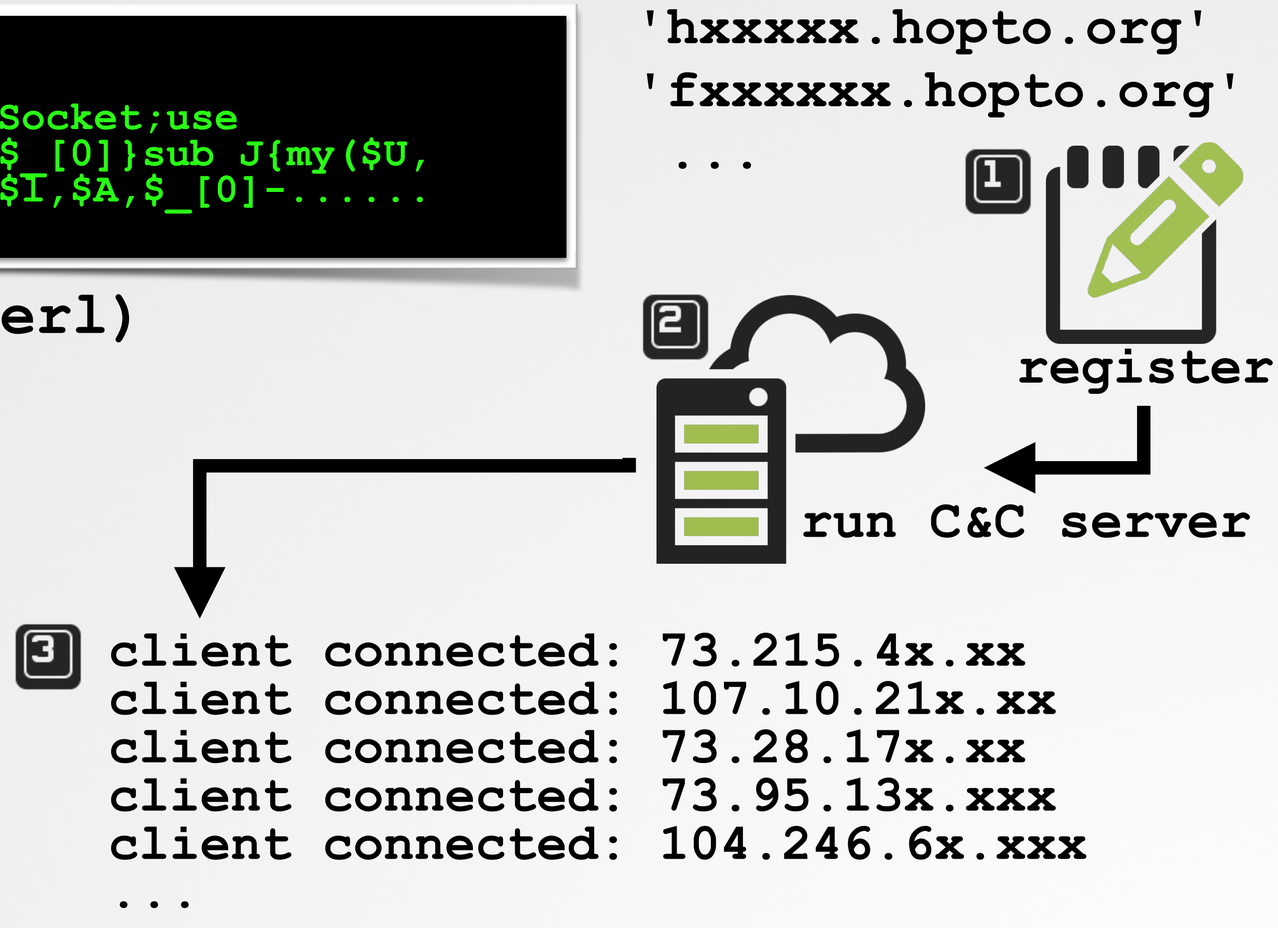


IN THE UNITED STATES DISTRICT COURT
FOR THE NORTHERN DISTRICT OF OHIO
EASTERN DIVISION

UNITED STATES OF AMERICA,) INDICTMENT
)
Plaintiff,)
)
v.) CASE NO. 1:18 CR 00022
)
PHILLIP R. DURACHINSKY,) Title 18, United States Code,
) Sections 1028A(a)(1),
Defendant.) 1030(a)(2)(C), (a)(3), (a)(5)(A),
) (c)(2)(A), (c)(4)(A)(i)(I) and
) (c)(4)(B), 1343, 2251(a) and
) 2511(1)(b) and (4)(a), and 2

JUDGE OLIVER

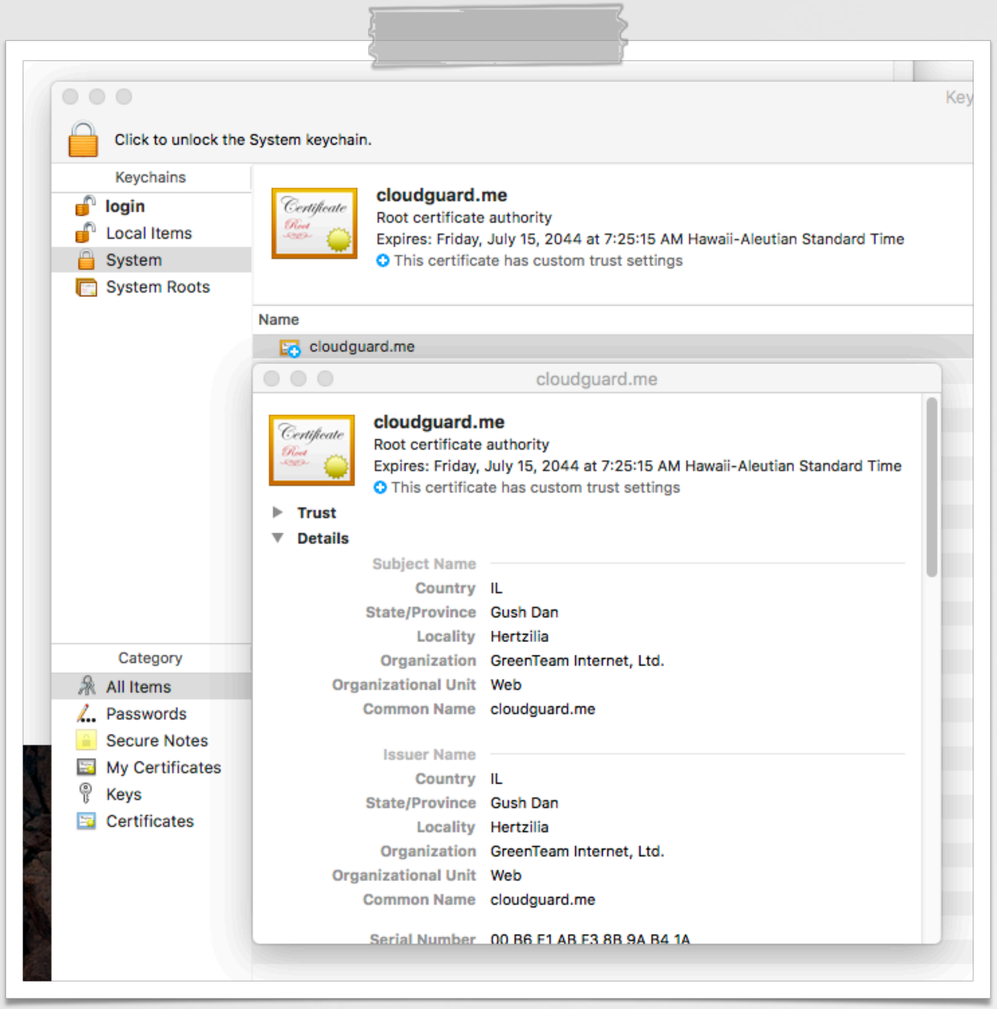
GENERAL ALLEGATIONS



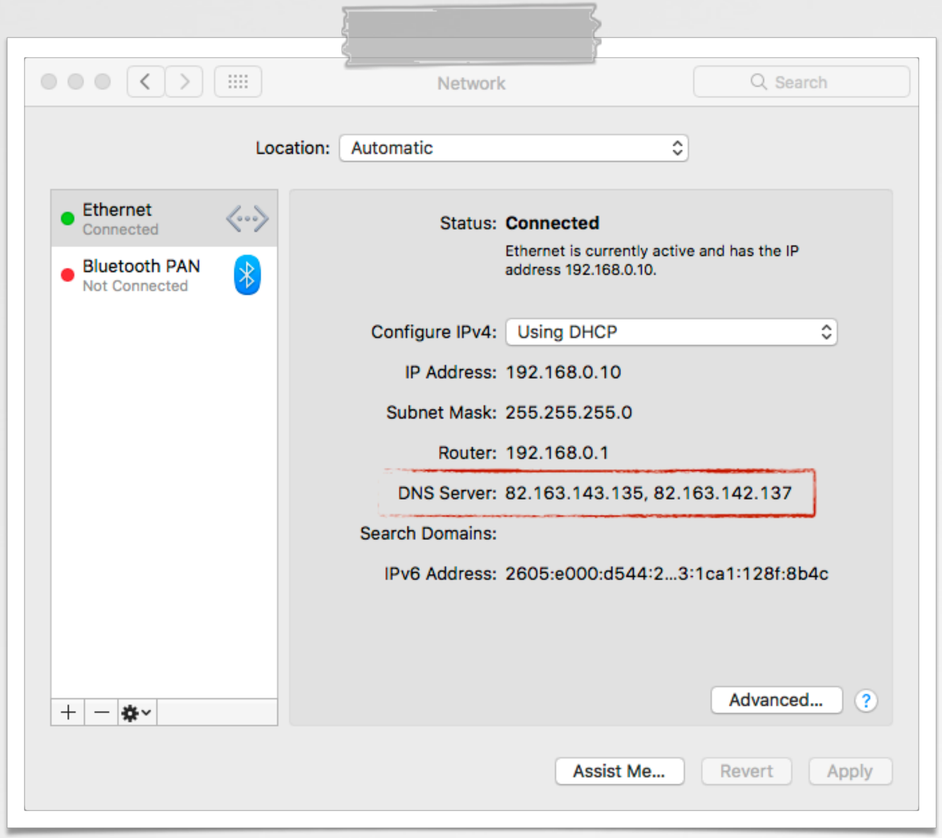
"For more than 13 years, Phillip Durachinsky allegedly infected with malware the computers of thousands of Americans and stole their most personal data and communications" -Assistant Attorney General Cronan.

OSX.MaMi (2018)

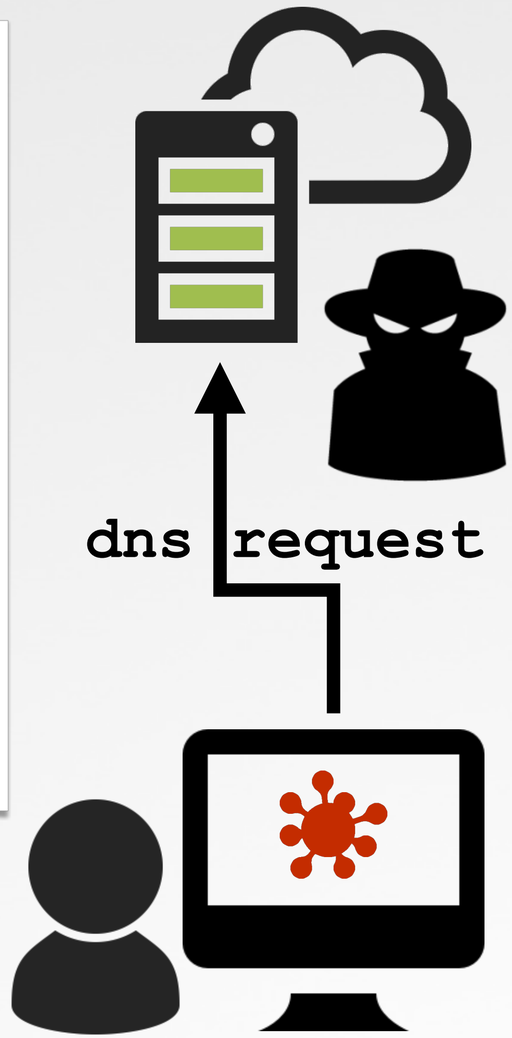
a dns hijacker



1 install certificate

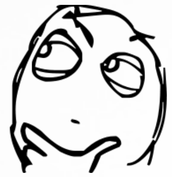


2 hijack DNS servers



a Mac port of Window's 'DNSUnlocker'?

- 'same' malicious DNS servers
- certificate thumbprint



Win/OSX/Linux CrossRAT (2018)

apt surveillance operation backdoor

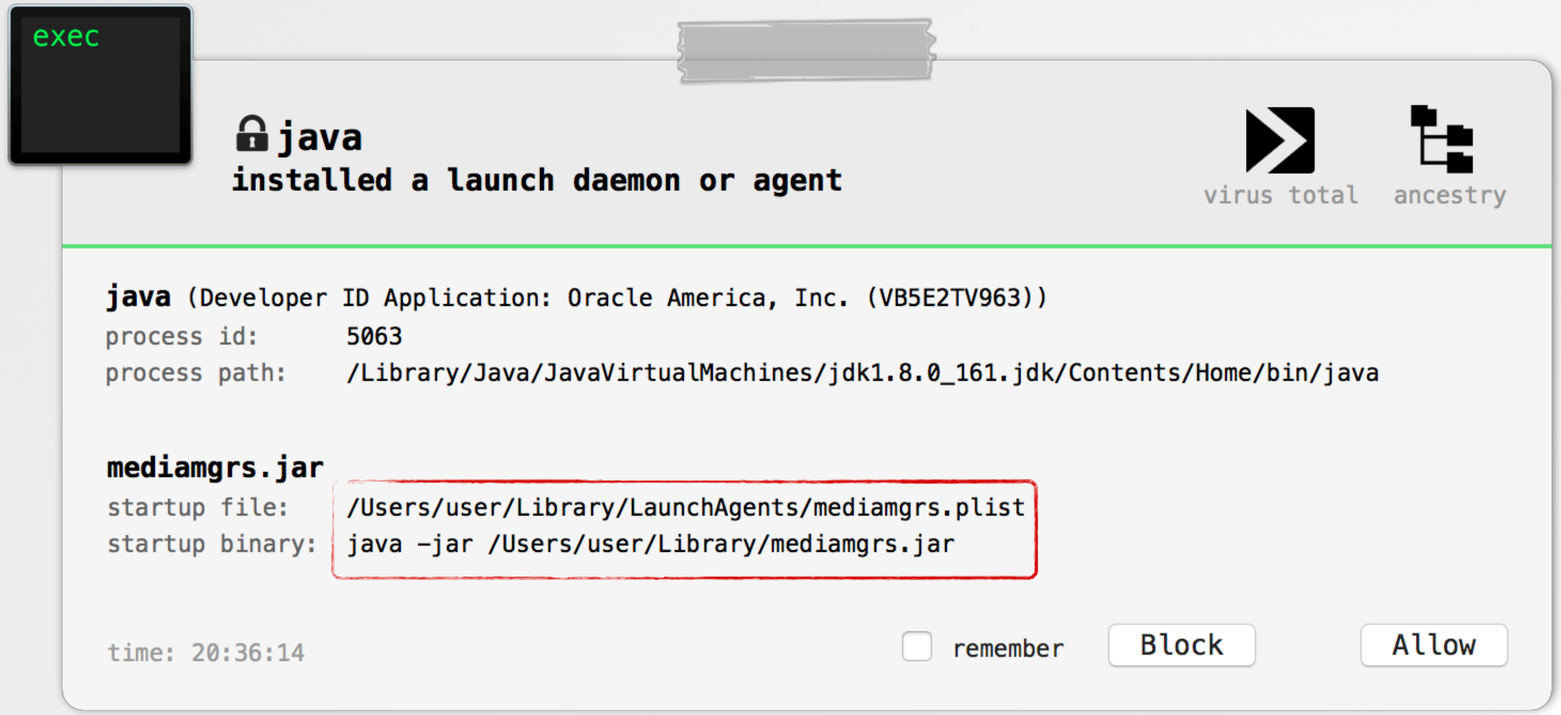
> *"desktop 'surveillanceware' tool...which is able to target Windows, OSX, and Linux."* -EFF/Lookout

java
cross-platform

files

commands

screen capture

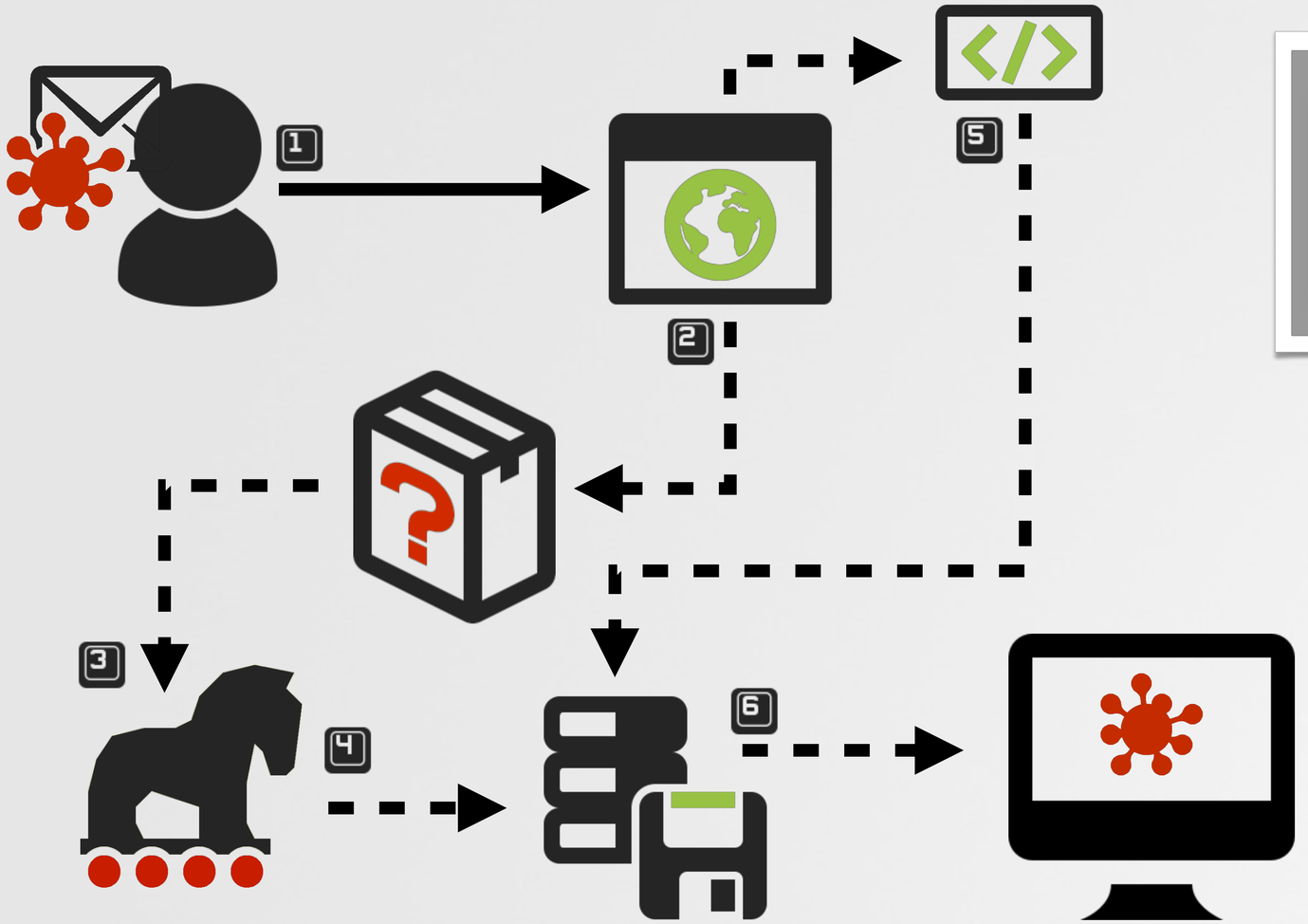


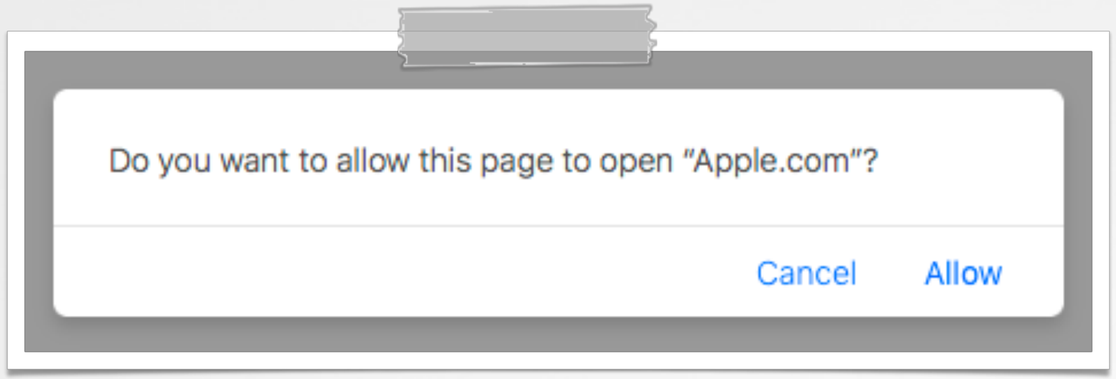
Mac persistence
(launch agent)

OSX.WINDSHIFT (2018)

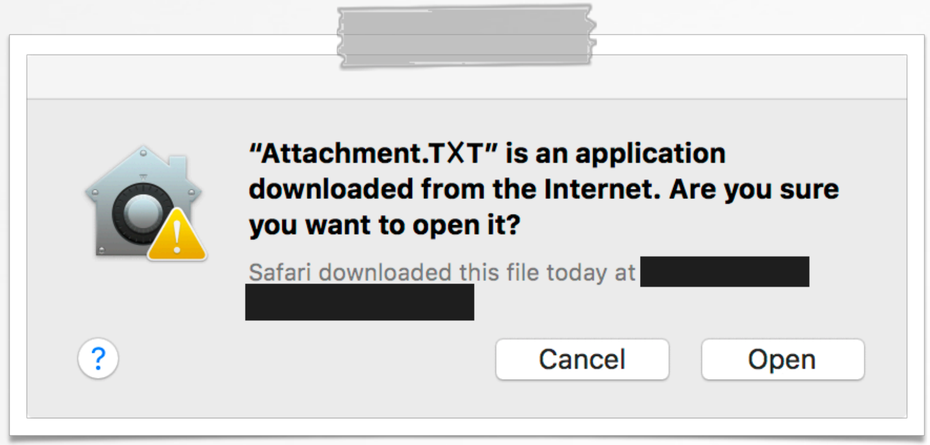
apt backdoor, with a unique infection vector

> *"[WINDSHIFT APT] sole focus [is] on specific individuals for espionage and surveillance purposes"* -DarkMatter





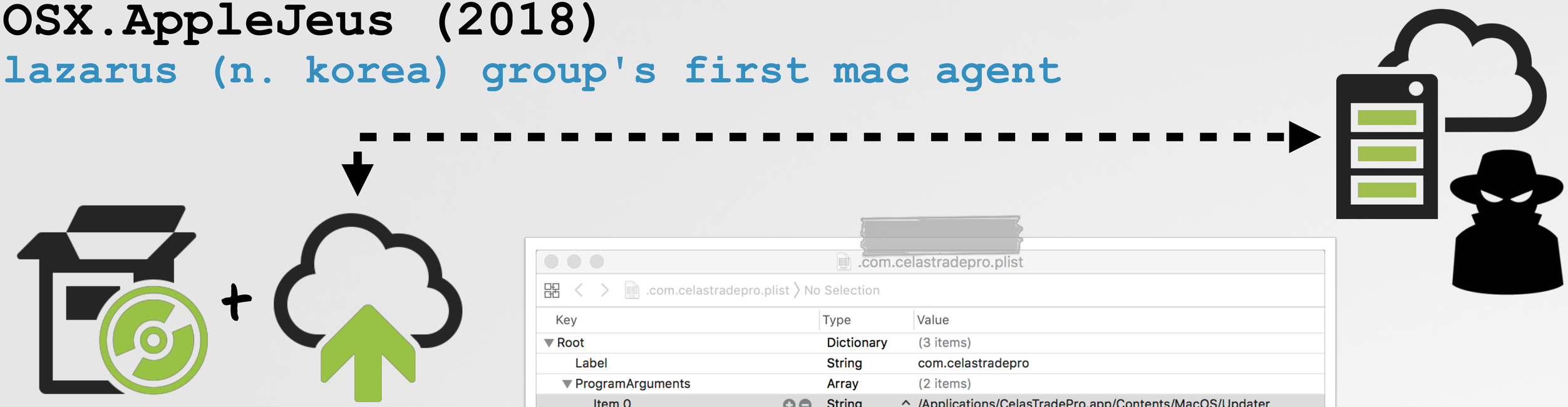
alert: custom URL scheme



alert: application launch

OSX.AppleJeus (2018)

lazarus (n. korea) group's first mac agent



Celas Trade Pro,
from "Celas Limited"

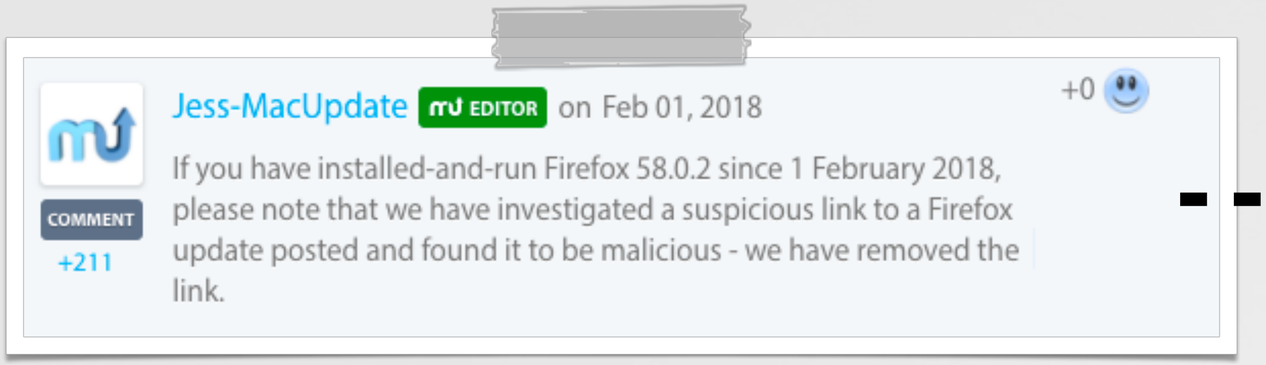
fake company!

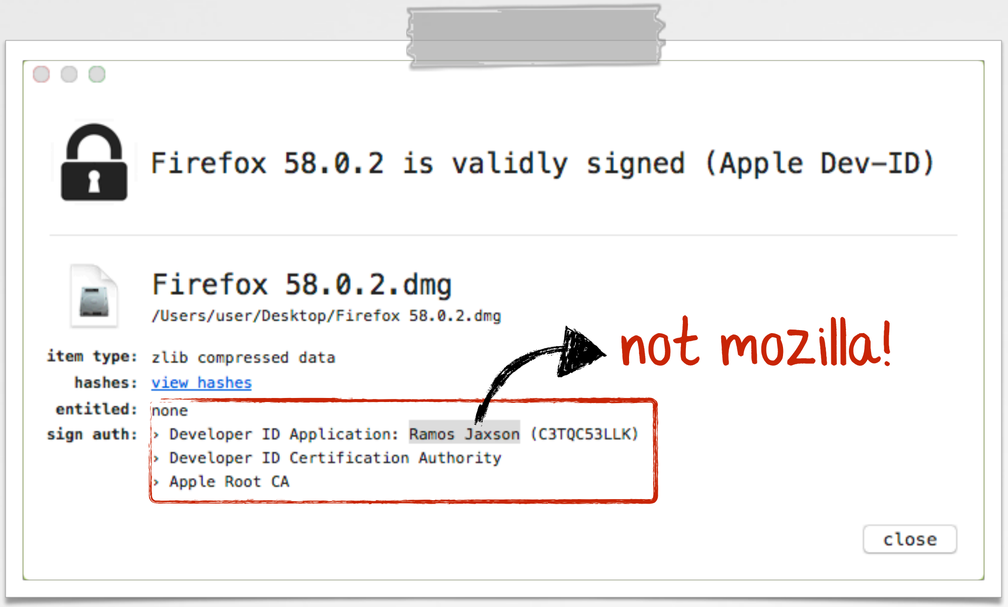
malicious updater's persistence
(plist)

"Lazarus hits cryptocurrency exchange with fake installer and macOS malware" -Kaspersky

OSX.CreativeUpdater (2018)

cryptominer distributed via 'macupdate.com'

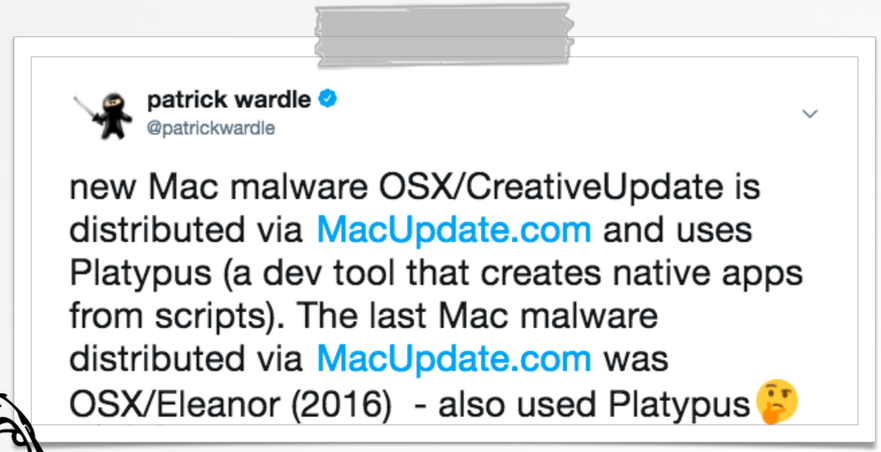


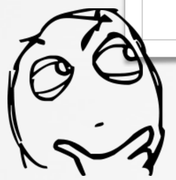


signed

```
$ cat ~/Library/LaunchAgents/MacOS.plist

<key>ProgramArguments</key>
<array>
    <string>sh</string>
    <string>-c</string>
    <string>
      ~/Library/mdworker/mdworker
      -user walker18@protonmail.ch -xmr
    </string>
</array>
...
```

persistent cryptominer (xmr)





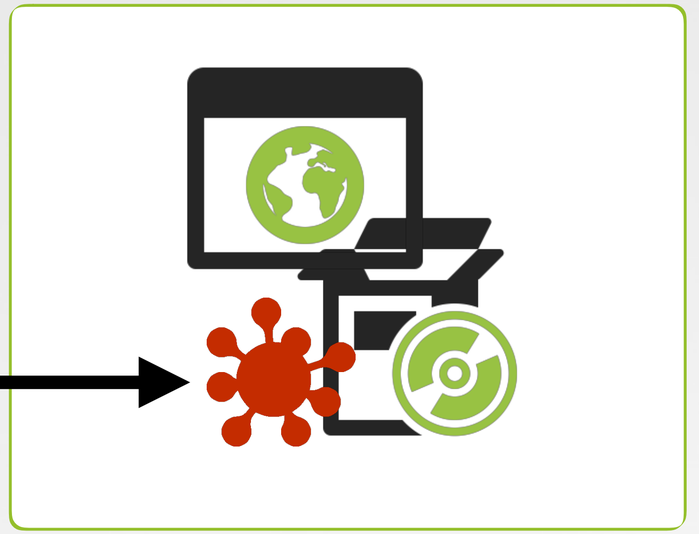
interesting connection

Mac Malware

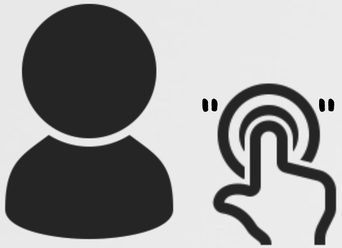
trends in infection vectors

"supply chain" attacks

attachments webpages installers



somewhere.com



user interaction

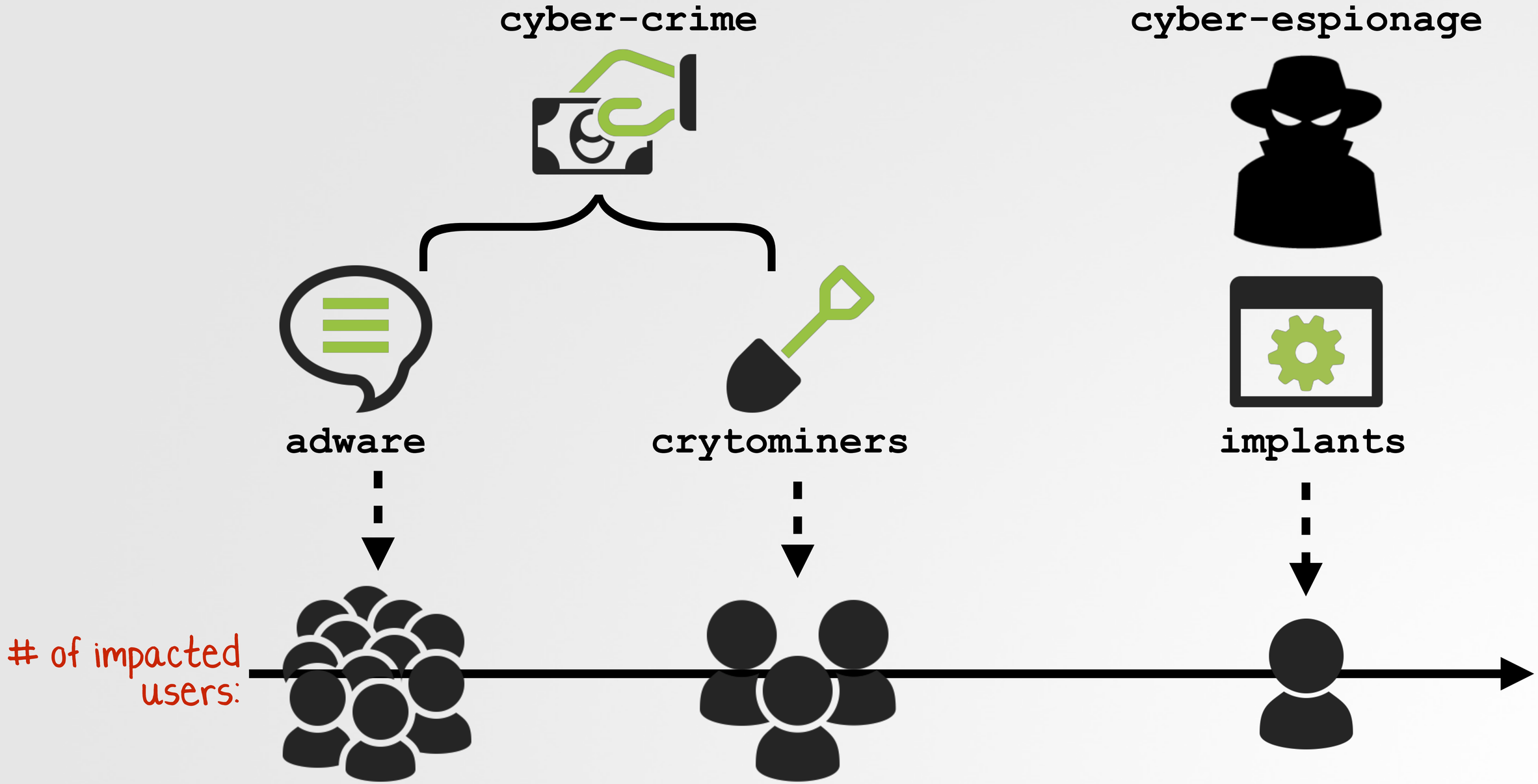
Technical Details

The attack vector included the scanning and identification of externally facing Mac services to include the Apple Filing Protocol (AFP, port 548), RDP, VNC, SSH (port 22), and Back to My Mac (BTMM), which would be targeted with weak passwords or passwords derived from 3rd party data breaches.

'pure' remote (OSX.FruitFly)

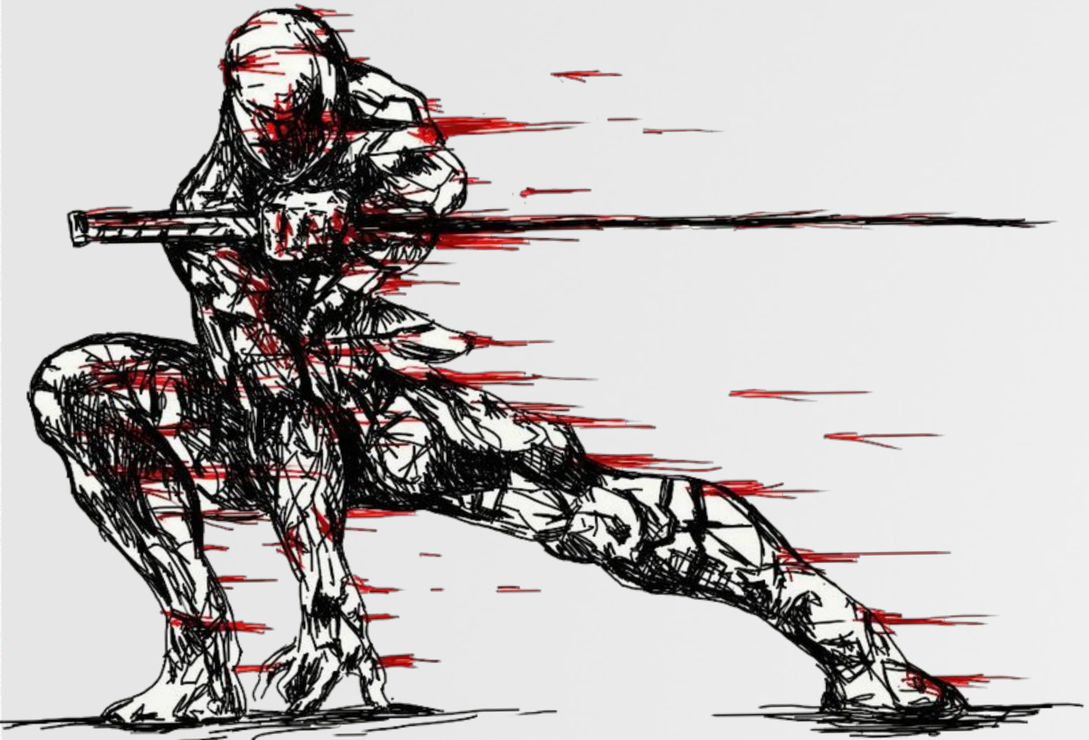
Mac Malware

trends in payload / objective(s)



Mac Vulnerabilities

...the ones that make us cry



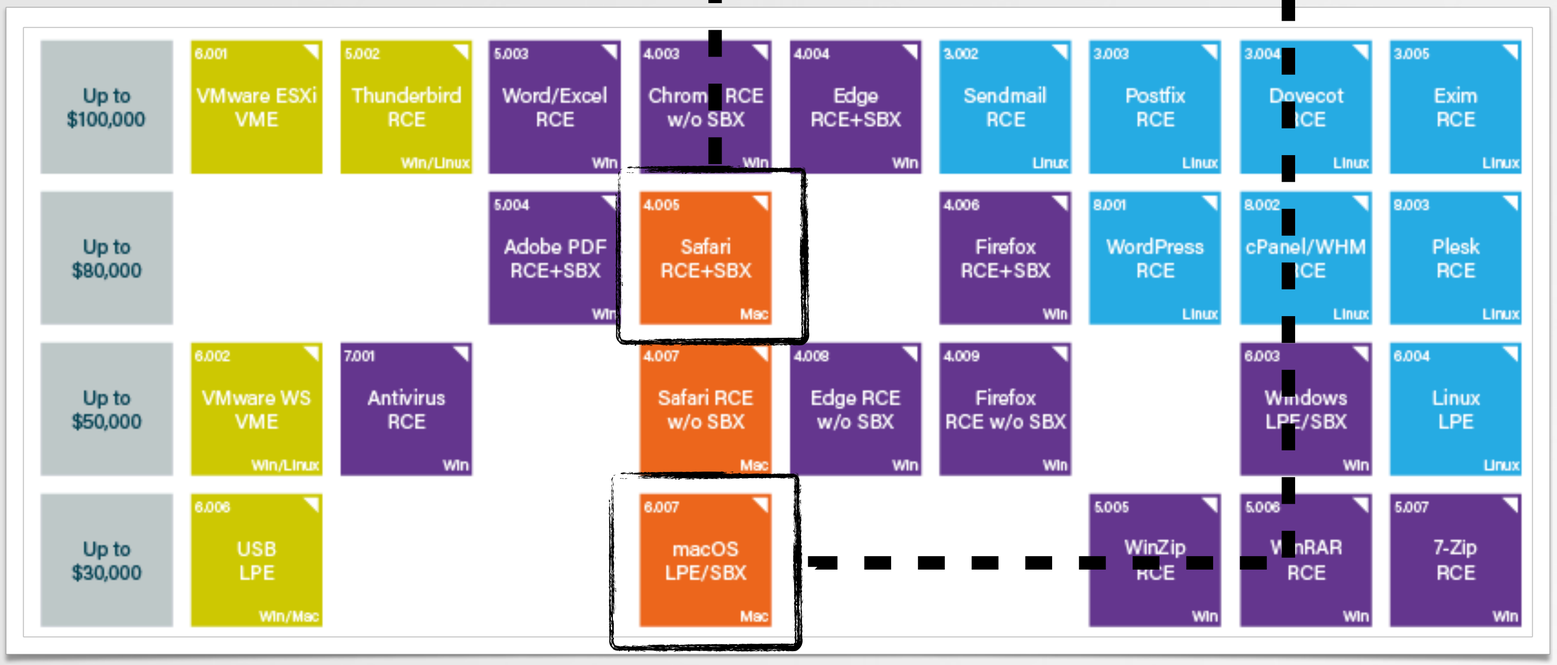
Digita Security

The Reality

Macs are just as susceptible to vulnerabilities as their (modern) Windows counterparts...if not more so!

Safari < Chrome, Edge

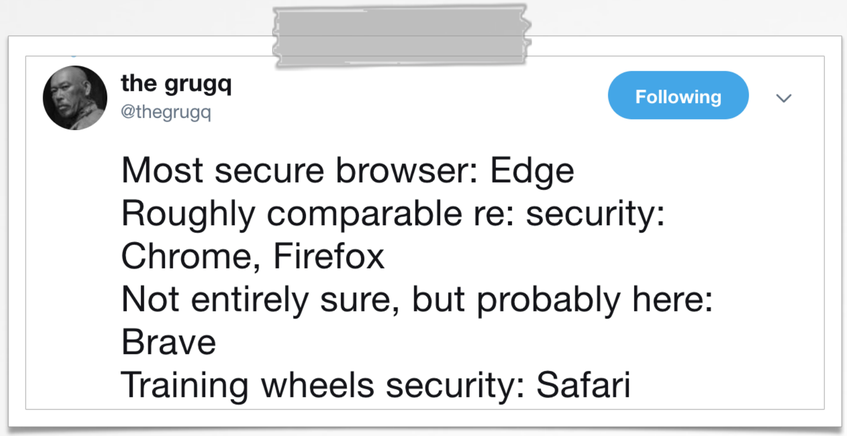
macOS < Window,Linux



market rates for 0days

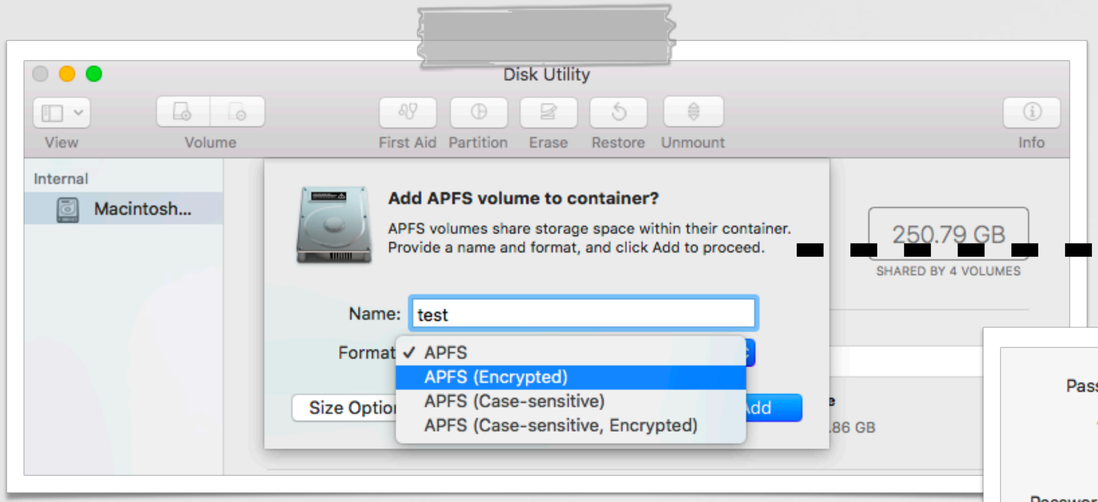
CVE Ranking	Product
#5	macOS
#6	Windows 10

CVE count (2017)

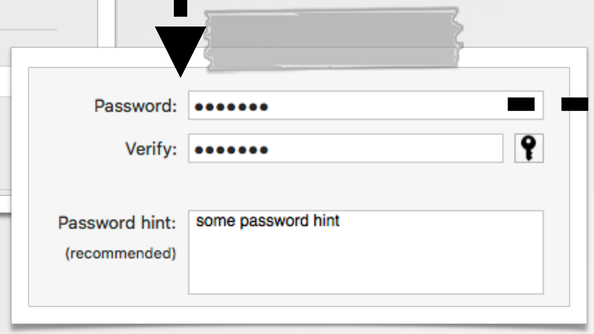


CVE-2017-7149: Password Exposure

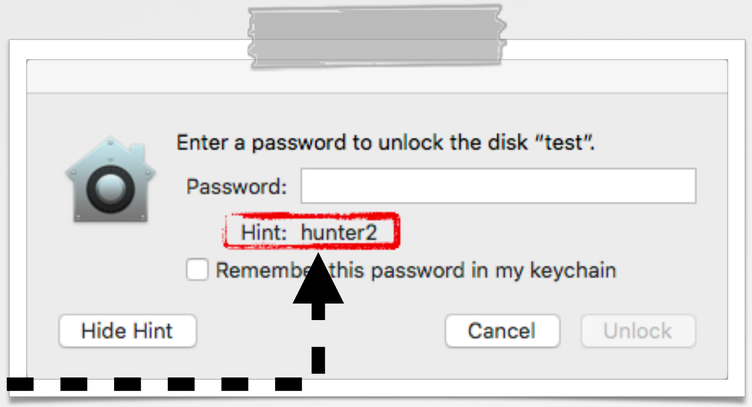
in encrypted volumes, hint == password!? (@martiano_)



1 create encrypted AFPS



2 set password hint



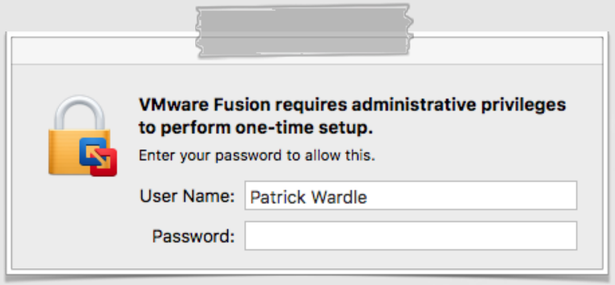
3 later: 'Show Hint' ...reveals password!

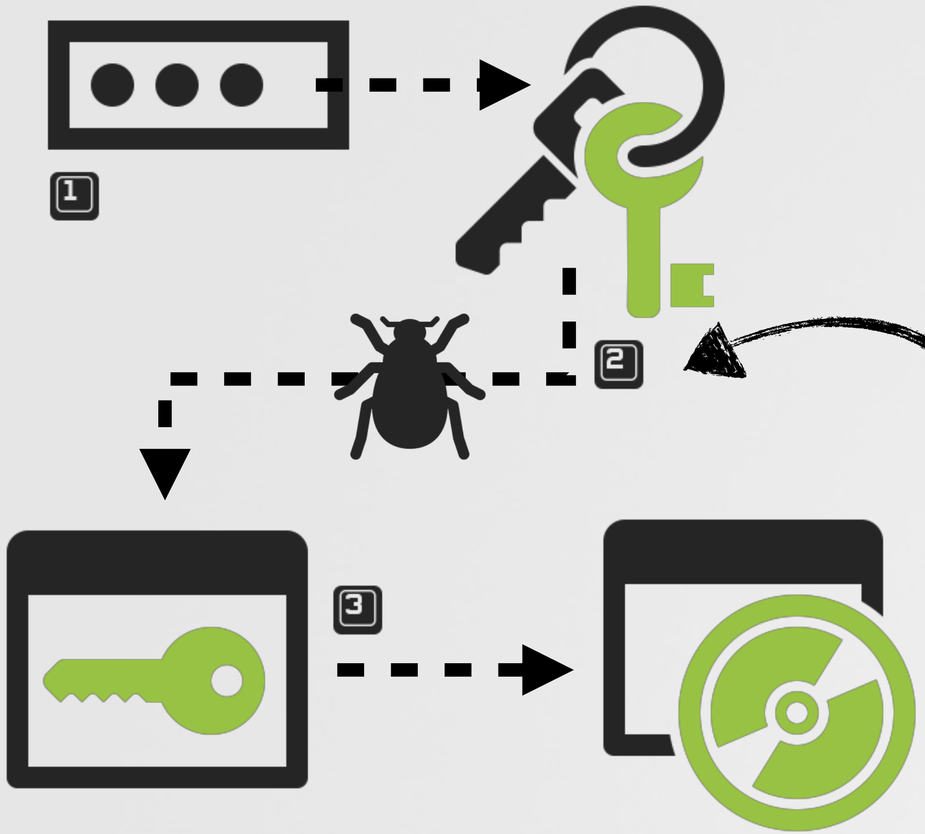
```
[SKHelperClient addChildVolumeToAPFSContainer: ...password: passwordHint: ...]

 //save password
 if(password != nil)
   [infoDictionary setObject:password forKey:@"kSKAPFSDiskPasswordOption"];

 //save password hint
 if(passwordHint != nil)
   [infoDictionary setObject:password forKey:@"kSKAPFSDiskPasswordHintOption"];
```

password value stored in 'hint' key

CVE-2017-7170: Privilege Escalation

sniffing 'authentication' tokens (p. wardle)





authentication

Authorization.h

```
/*
SECURITY NOTE:
Applications should take care to not disclose the AuthorizationExternalForm
to potential attackers since it would authorize rights to them.
*/
```

WARNING!

trampolineClient.cpp

```
#define TRAMPOLINE /usr/libexec/security_authtrampoline

FILE *mbox = tmpfile();
if (fwrite(&extForm, sizeof(extForm), 1, mbox) != 1)
...

char mboxFdText[20];
snprintf(mboxFdText, sizeof(mboxFdText), "auth %d", fileno(mbox));

fork()
...
execv(trampoline, (char *const*)argv);
```

auth token saved to temp file!?
(since OSX Tiger!)

CVE-2017-7170: Privilege Escalation

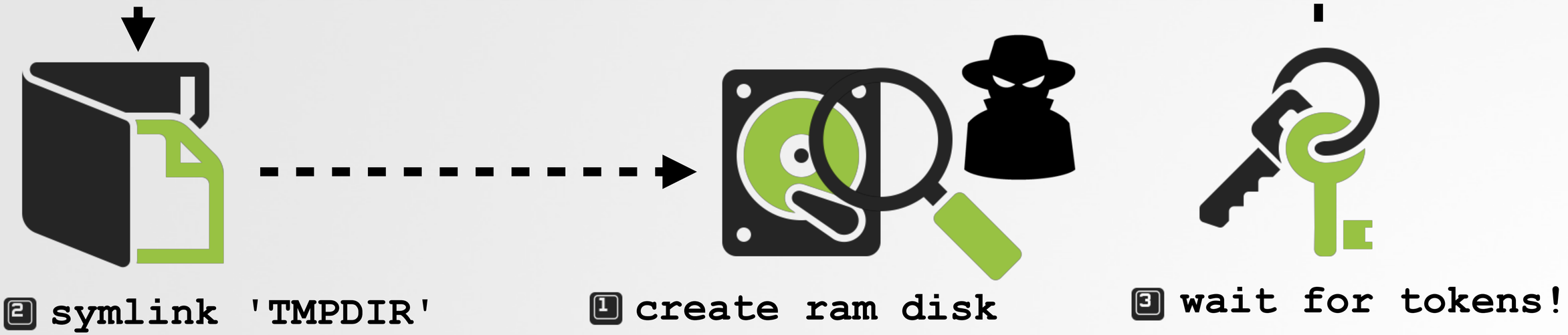
sniffing 'authentication' tokens (p. wardle)

```
FILE *mbox = tmpfile();
fwrite(&extForm, sizeof(extForm), 1, mbox) ...
```

tmpfile:
randomly named :(
immediately unlinked :(

```
$ man tmpfile()

The tmpfile() function returns a pointer to a stream associated with a file descriptor
returned by the routine mkstemp(3).  The created file is unlinked before tmpfile() returns...
```

tmpfile API



2 symlink 'TMPDIR'

1 create ram disk

3 wait for tokens!

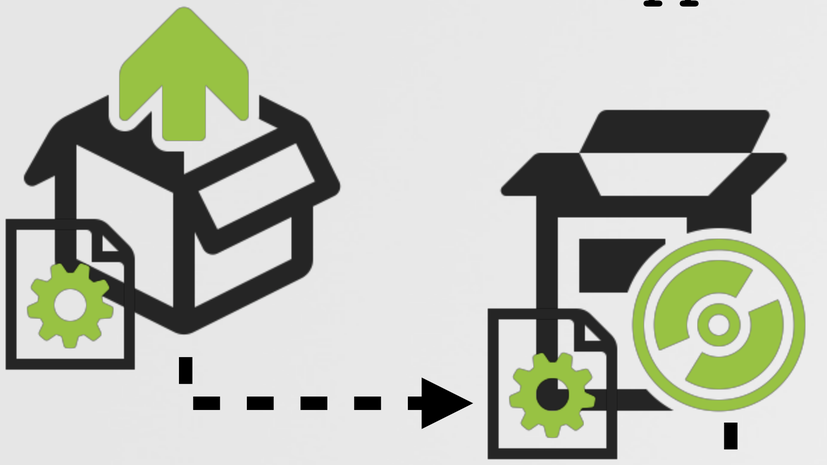
CVE-2017-13837: Stealing FileVault Key

theft via a malicious application (p. wardle)

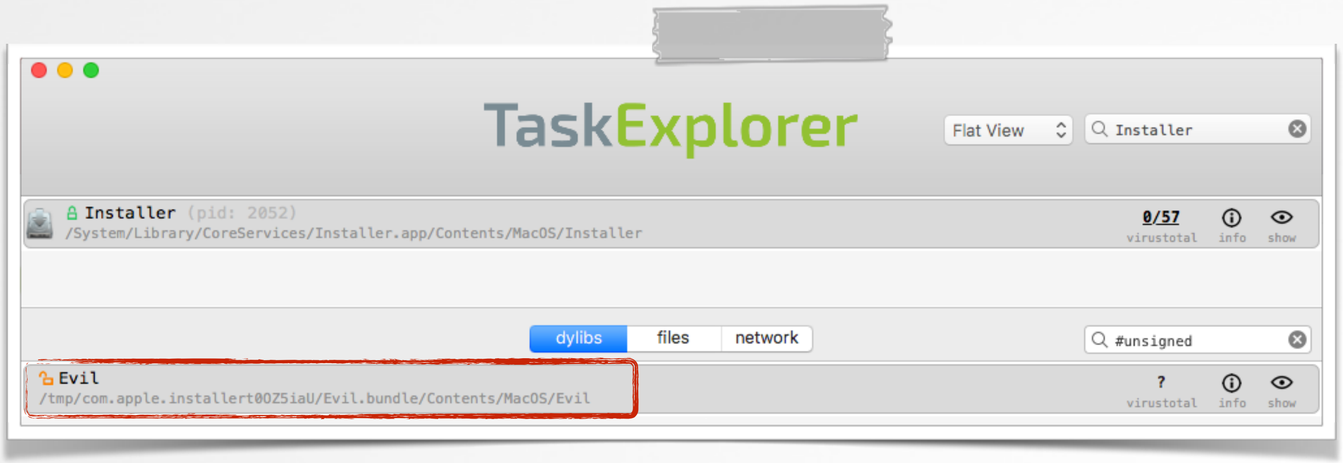
```
$ codesign -d --entitlements - Installer.app

<?xml version="1.0" encoding="UTF-8"?>
<plist version="1.0">
<dict>
  <key>com.apple.private.securityd.stash</key>
  <true/>
</dict>
</plist>
```

Installer.app's entitlements

access:
FileVault Unlock Key



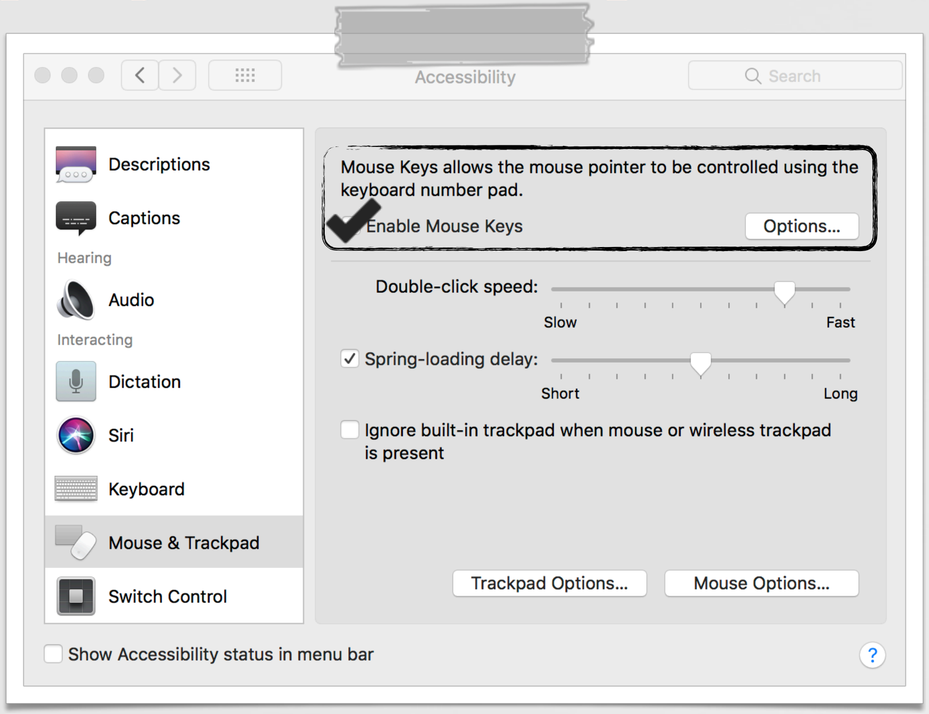
Installer.app loads
.pkg plugins!



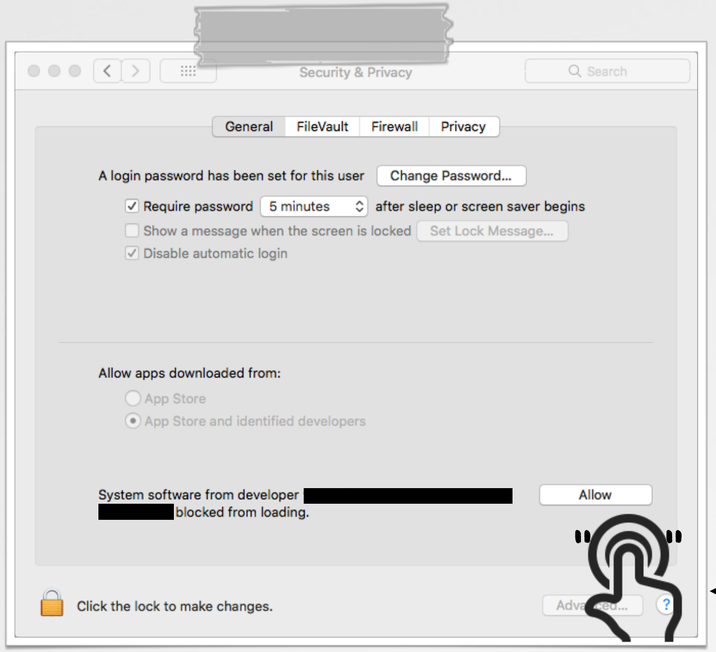
malicious plugin, in Installer.app

CVE-2017-7150: Synthetic Interactions

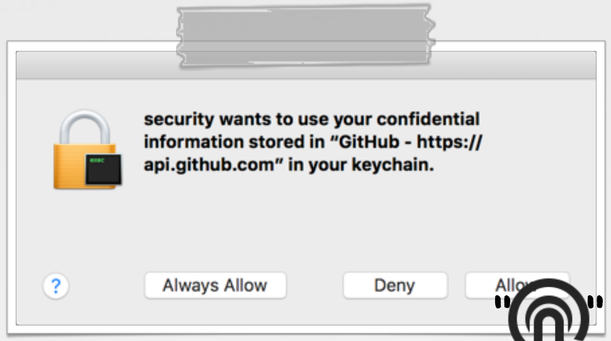
programmatically click all thingz (p. wardle)



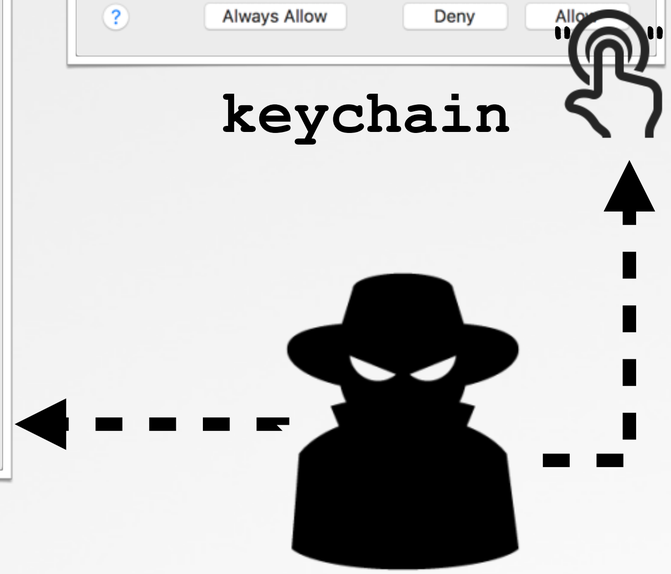
mouse keys



kext loading



keychain



```
//click via mouse keys
void clickAllow(float X, float Y)
{
    //move mouse
    CGEventPost(kCGHIDEventTap, CGEventCreateMouseEvent(nil, kCGEventMouseMoved, CGPointMake(X, Y), kCGMouseButtonLeft));

    //apple script
    [[[NSAppleScript alloc] initWithSource:
       @"tell application \"System Events\" to key code 87\n"] executeAndReturnError:nil];
}
```

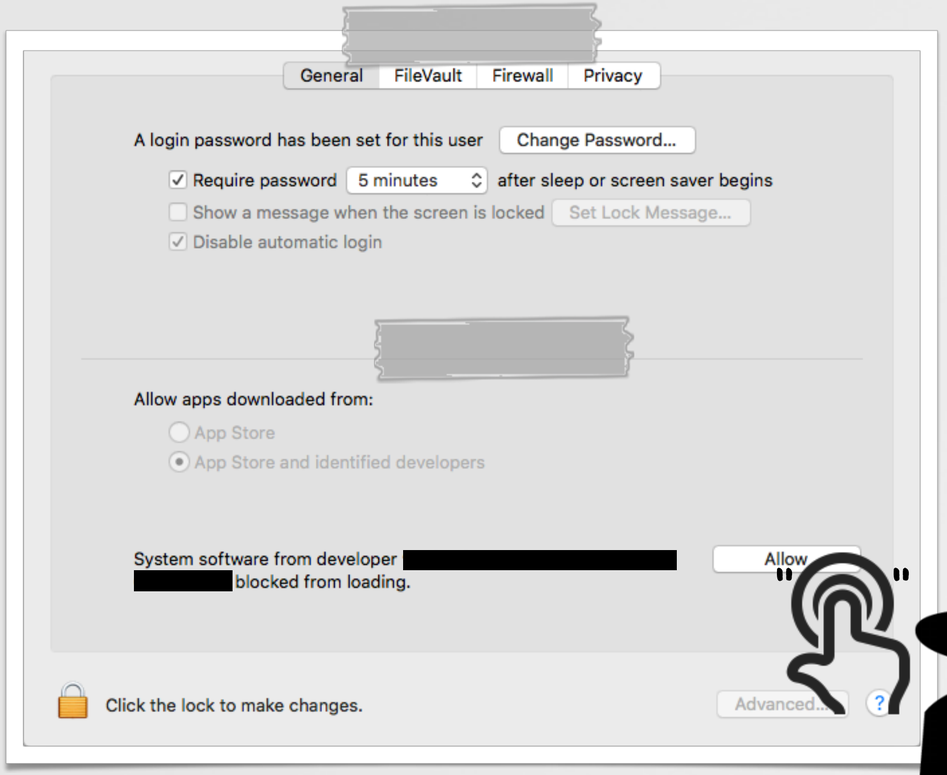
0day (macOS < 10.14): Synthetic Interactions

still, programmatically click all thingz (p. wardle)

```
//given some point {x, y}
// generate synthetic event...
CGPostMouseEvent(point, true, 1, true);
CGPostMouseEvent(point, true, 1, true);
```

mouse down

mouse down
...AGAIN!!

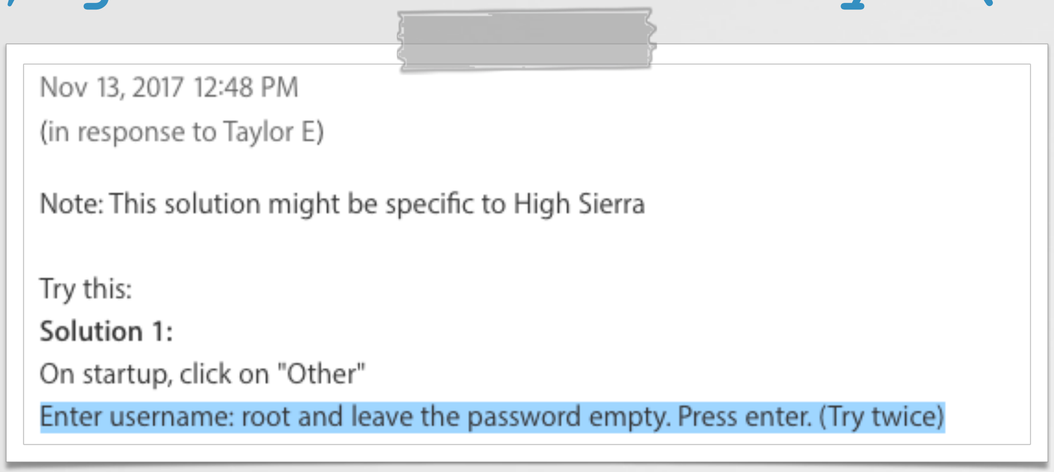


kext loading bypass

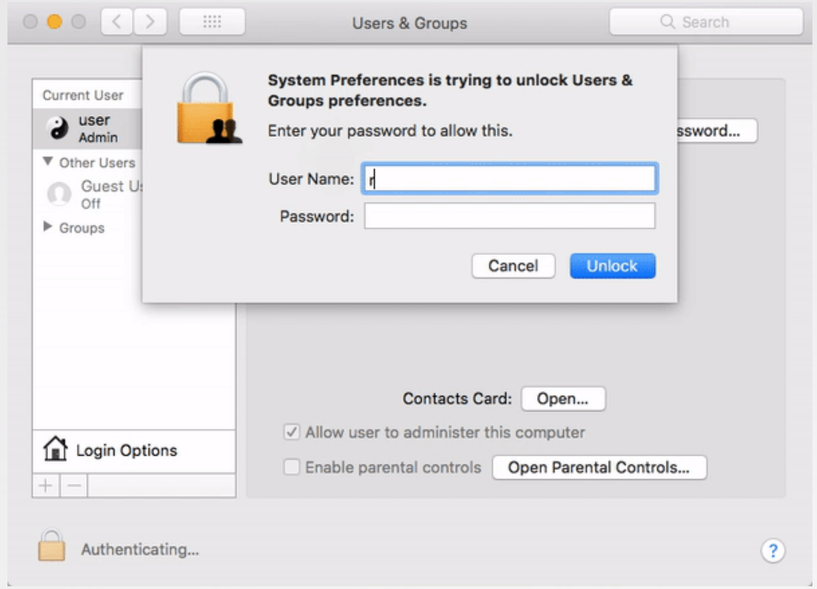
Before Mojave, Apple's
"*User Assisted Kext Loading*"
...was ***NEVER*** implemented securely

CVE-2017-13872: #iamroot

click 2x, get r00t...no really :(



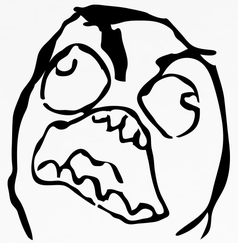
0day on Apple's forums



proof of concept

```
//verify
int match = kODErrorCredentialsInvalid;
if(kODErrorSuccess != od_verify_crypt_password(accountHash, providedPassword, &match, ...);
{
    //bail, w/ with error
    goto bail;
}

//ok happy! allow authentication/login
```

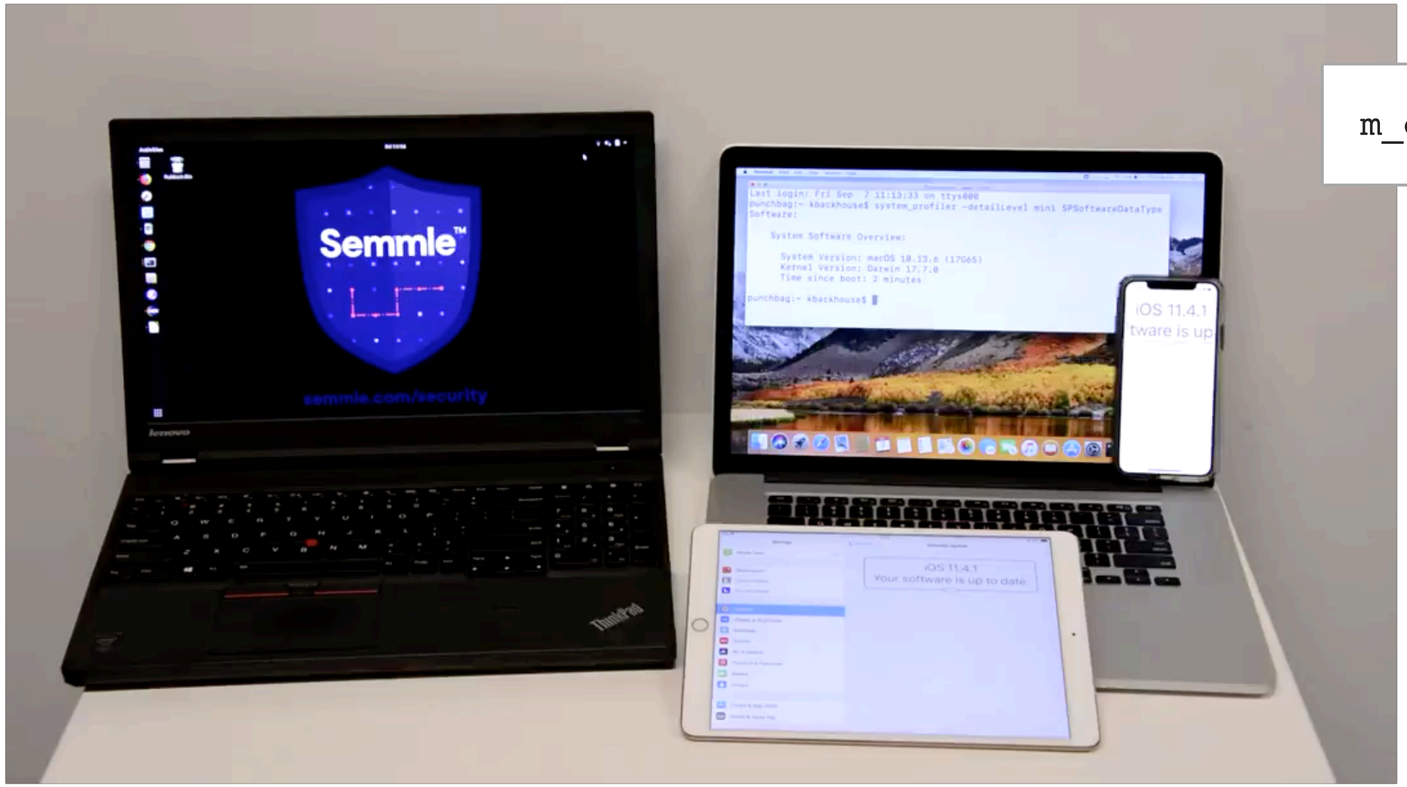
never checked



read: "Why <blank> Gets You Root"
objective-see.com/blog/blog_0x24.html

CVE-2018-4407: pure remote ring-0!?

heap-overflow in (kernel) icmp packet handling (@kevin_backhouse)



ip_icmp.c

```
m_copydata(n, 0, icmplen, (caddr_t)&icp->icmp_ip);
```

- remote
- ring-0
- no user interaction

detailed write-up

"Kernel RCE caused by buffer overflow in Apple's ICMP packet-handling code (CVE-2018-4407)" -https://lgtm.com/blog/

Mac Patches!?

...rarely fix the vulnerability?

MACOS UPDATE ACCIDENTALLY UNDOES APPLE'S "ROOT" BUG PATCH

oops

Apple's OS X 'Rootpipe' patch flops, fails to fix flaw

Researcher finds 'trivial way' to exploit privilege escalation vulnerability after Apple tries to plug Yosemite hole

oops

Apple Fails to Properly Fix Gatekeeper Security Hole, Claims Researcher

oops

Google warns Apple: Missing bugs in your security bulletins are 'disincentive to patch'

oops

want CVE's or 0days

1. reverse apple's patches
2. find bypasses

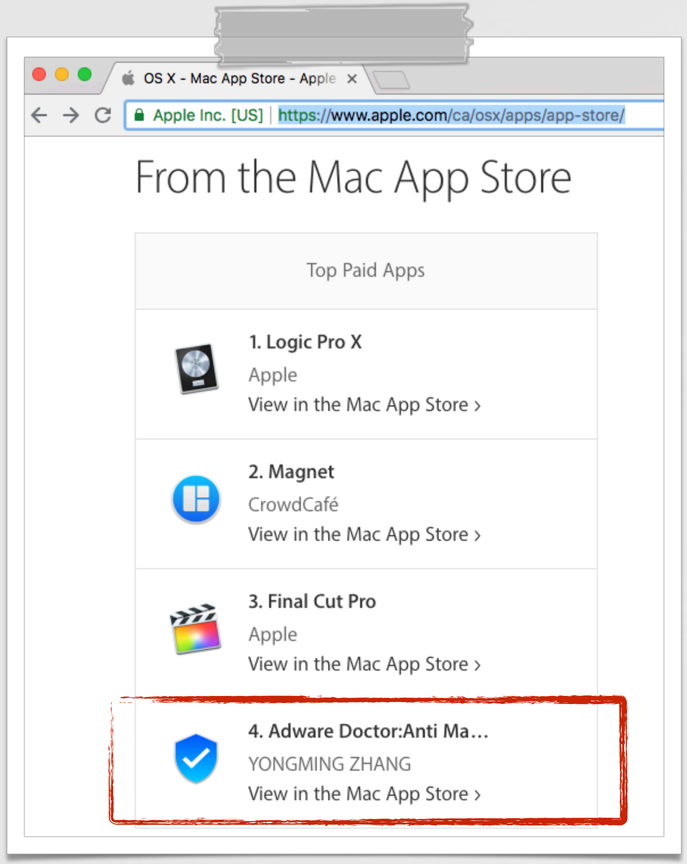
Patrick's CVE collection++

- Gatekeeper bypasses (2x!)
- "Rootpipe" bypass
- Synthetic click bypass

The Mac App Store

...unfortunately, not safe either!

> *"The safest place to download apps for your Mac is the Mac App Store. Apple reviews each app before it's accepted"* -apple.com

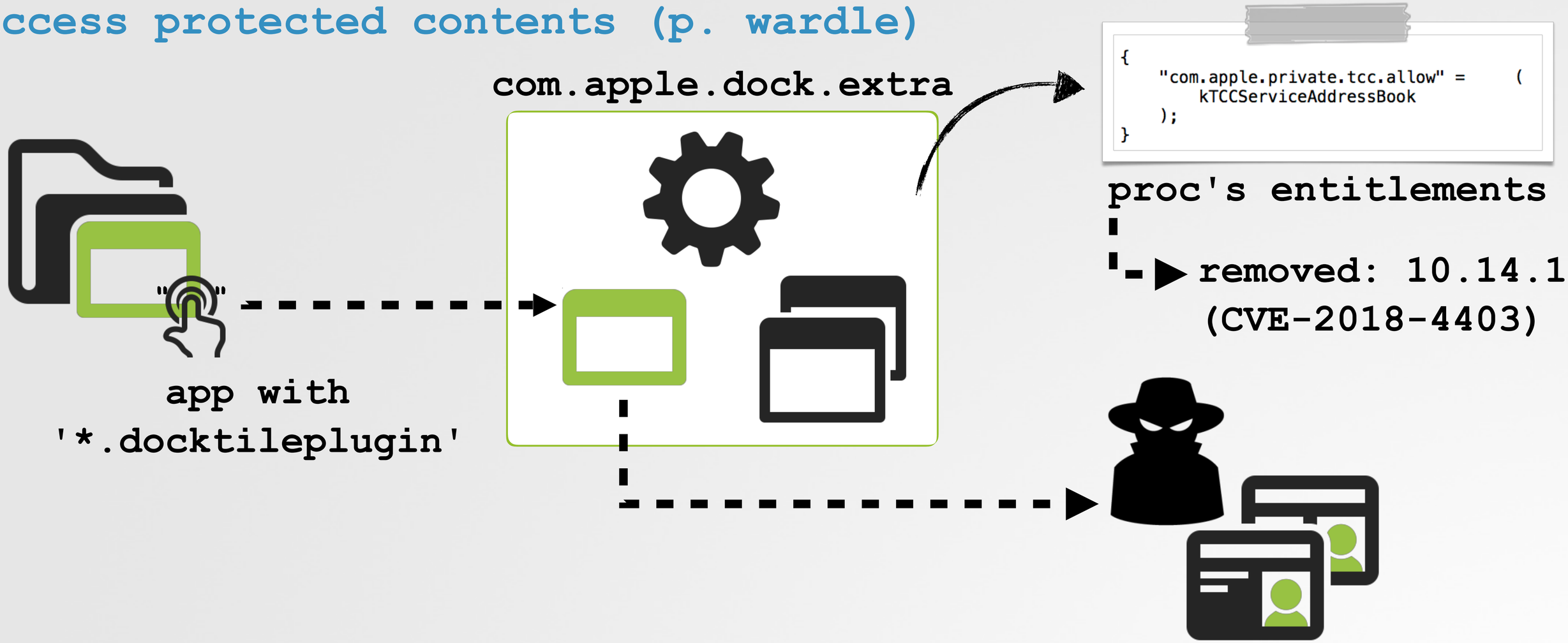


"Adware Doctor"
#4 (#1 utility)

stole user's browser history

Previous 0day (Mojave): Privacy Bypass

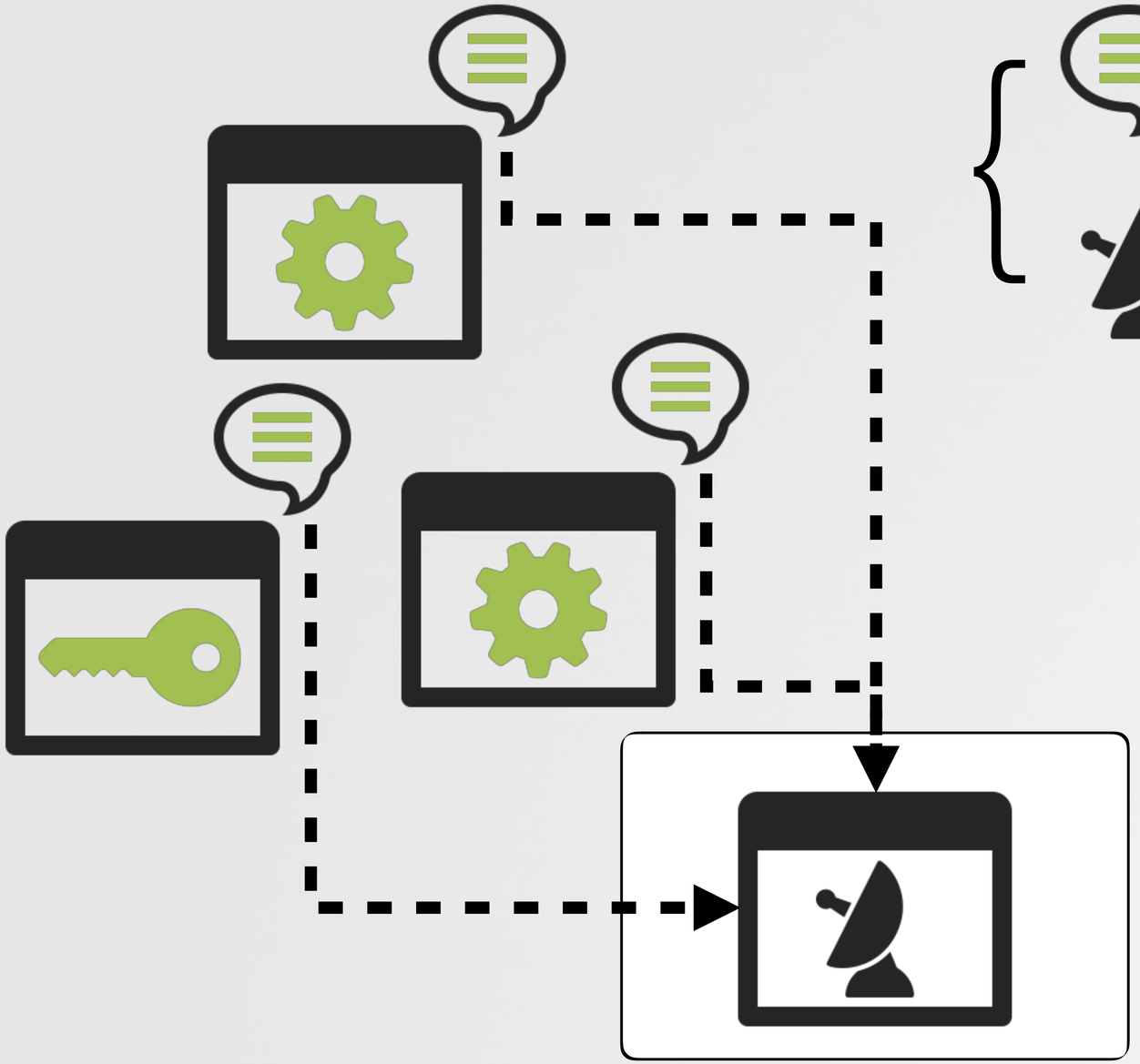
access protected contents (p. wardle)



"The plugin is loaded in a system process at login time or when the application tile is added to the Dock"

0day (Mojave): 'Leaky' Sandbox

all your notifications are belong to us? (p.wardle)



the os, or apps broadcast 'distributed' notifications

...that others can listen for

(malicious) listener

```
$ sniffNotifications

*** attempt to register for all distributed
notifications thwarted by sandboxing.

OS Version:     18A391
Application:    sniffNotifications

Backtrace:
0   __CFGenerateReport + 197
1   __CFXNotificationRegisterObserver + 1035
2   CFNotificationCenterAddObserver + 204
3   sniffNotifications
```

macOS prevents "sandboxed listeners"

0day (Mojave): 'Leaky' Sandbox

all your notifications are belong to us? (p.wardle)

yes!!

In the sandbox, can't register globally for "all" notifications...but can we register for individual notifications?

```
NSString* name = @"<any notification>";
CFNotificationCenterAddObserver(CFNotificationCenterGetDistributedCenter(), nil, callback,
                                (__bridge CFStringRef)name, nil, CFNotificationSuspensionBehaviorDeliverImmediately);

void callback(CFNotificationCenterRef center, void *observer, CFStringRef name_cf, const void *object, CFDictionaryRef userInfo) {
    NSLog(@"event/object: %@/%@", name, object);
}
```

notification listener (sandboxed)

"com.apple.LaunchServices.applicationRegistered"

"com.apple.DownloadFileFinished"

"com.apple.unmountassistant.process.start"

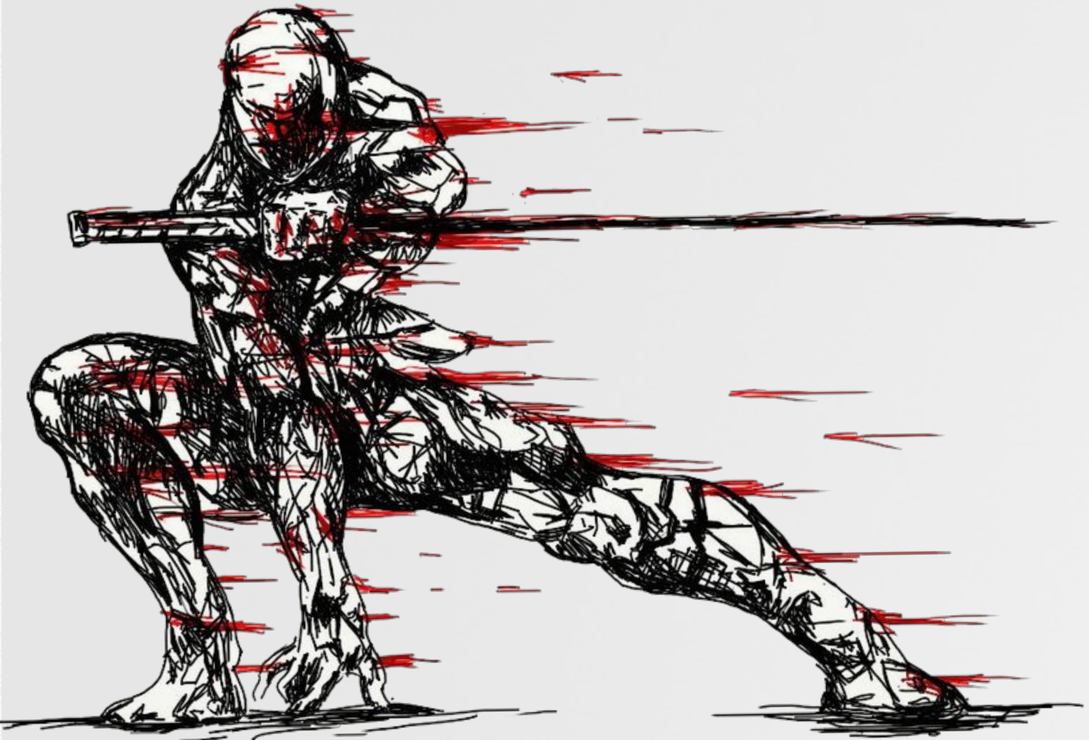
...and any others!!

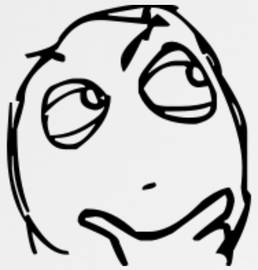
} works in the sandbox

```
$ sniffNotifications
  event: com.apple.DownloadFileFinished
  object: /Users/user/Downloads/omgSecrets.zip
```

(Mac) Musings

so, what's going on in cupertino!?





Security Experts/Researchers

what do they think of Apple's security?

The Secure Parser in C Guru @osxreverser — Following

As long security issues don't generate a cash flow problem to Apple, Apple will not give a flying fuck about the security of its users.

qwertyoruiop @qwertyoruiopz — Following

objective-see.com/blog/blog_0x21… <
Apple is doing this on the iPhone side too by using kernel patch protection.
Pseudosecurity designed to harm users

> Unfortunately when such 'security' features are introduced - even if done so with the noblest of intentions - t
> just complicate the lives of 3rd-party developers and users without affecting the bad guys (who don't have to
> the rules'). High Sierra's SKEL's flawed implementation is a perfect example of this.

Ian Beer @i41nbeer — Following

Hi @tim_cook, I've been working for years to help make iOS more secure. Here's a list of all the bugs I reported which qualified for your bug bounty since its launch, could you invite me to the program so we can donate this money to @amnesty?

Stefan Esser @i0n1c — Following

When it comes to Apple's „iOS Bug Bounty Program" always keep in mind that top iOS kernel bug killer Ian Beer has seen exactly 0 cents from Apple for disclosing tons and tons of iOS kernel bugs.

Charlie Miller @0xcharlie — Following

Can you believe Apple still hasn't reinstated me in the iOS developers program? Nice relationship they have with researchers.

qwertyoruiop @qwertyoruiopz — Following

Apple removing VPN apps in China is a great example of why the walled garden approach is fundamentally anti-freedom.

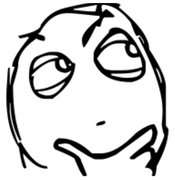
11:40 PM - 15 Sep 2017

Stefan Esser @i0n1c — Following

Replying to @howardnoakley @patrickwardle

Apple does nothing until publicly shamed.

1:22 AM - 11 Sep 2018

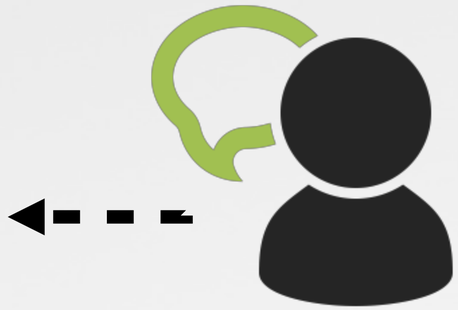


"Even if these researchers are all wrong (they are not) it's interesting to observe the relationship...or lack thereof!

So What Is Apple's (Main) Goal?

user privacy? security? to make $$$?

> *"privacy to us, is a human right"*
> -apple ceo



apple's words

hrmmmm

FINANCIAL TIMES

Chinese business & finance

Apple drops hundreds of VPN apps at Beijing's request

apple's actions

WIRED SUBSCRIBE

ANDY GREENBERG SECURITY 07.10.18 03:22 PM

APPLE'S CHINA-FRIENDLY CENSORSHIP CAUSED AN IPHONE-CRASHING BUG

Confused?

...don't be!

Apple is a publicly traded corporation... First & foremost, it is legally beholden to its shareholders. Never forget that.

Case: eBay Domestic Holdings Inc. v. Newmark

Corporate directors are bound by "fiduciary duties and standards" which include "acting to promote the value of the corporation for the benefit of its stockholders."

Research: "A Duty to Shareholder Value"

"[the] law requires corporate directors and managers to pursue long-term, sustainable shareholder wealth maximization in preference to the interests of other stakeholders or society at large"

...tl;dr shareholders (not users) are #1

Microsoft's Approach

modern, transparent, and emotionally mature

if your priority is security, then maybe!

is it time to switch to Windows (or *nix)!?

Microsoft, has embraced the fact that any software will have bugs

blue hat (security conf)

bounty program

transparent security posture
(e.g. "Global Security Intelligence Report")

emotionally mature relationships w/ researchers
& proactive engagement with the security community

bug sophistication++

exploit prices++

Cupertino

...on the other hand

Mac (in)security

lack of transparency

lack of comms with media & researchers

faulty patches

no bug bounty (macOS)

trivial bugs

more $$ on marketing than security (IMHO)

easy targets for hackers, and job security for Patrick ;)

And why is this a problem?

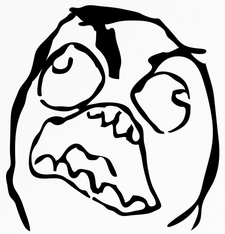
"Many people also consider [mac] to be a more secure computing platform, which may lead to a dangerous sense of complacency in both IT departments and with users."

case in point:
OSX.FruitFly

12+ years

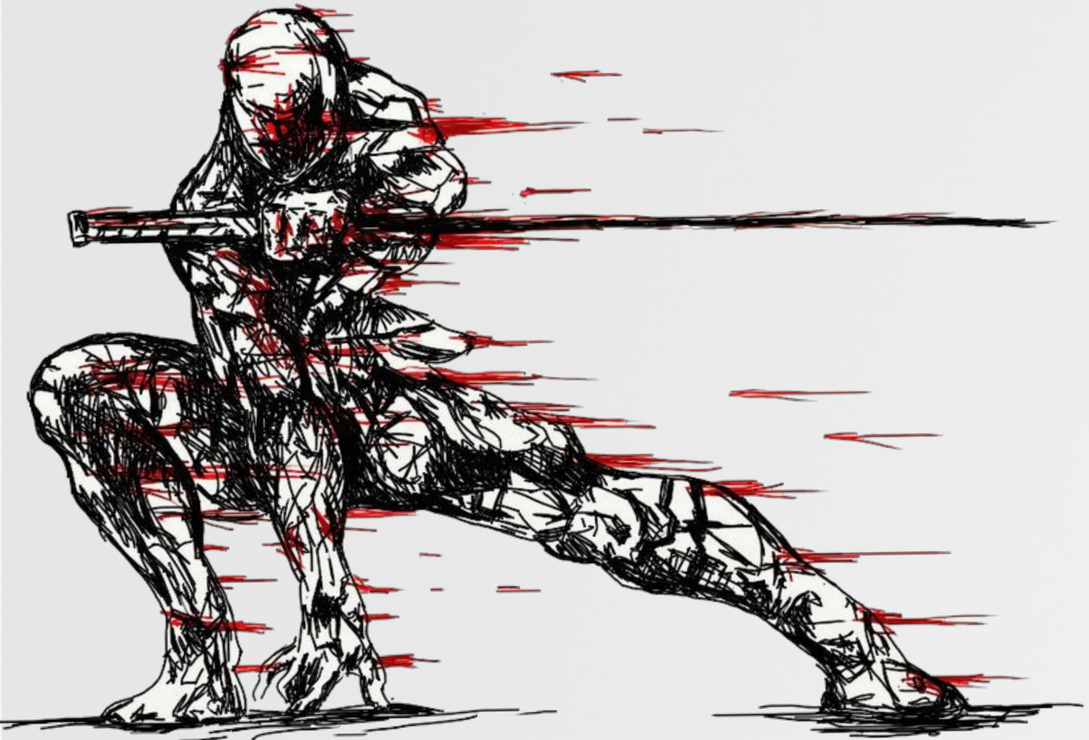
spying on children

Apple's response:
[silence]



Mac Protections

mojave to the rescue?



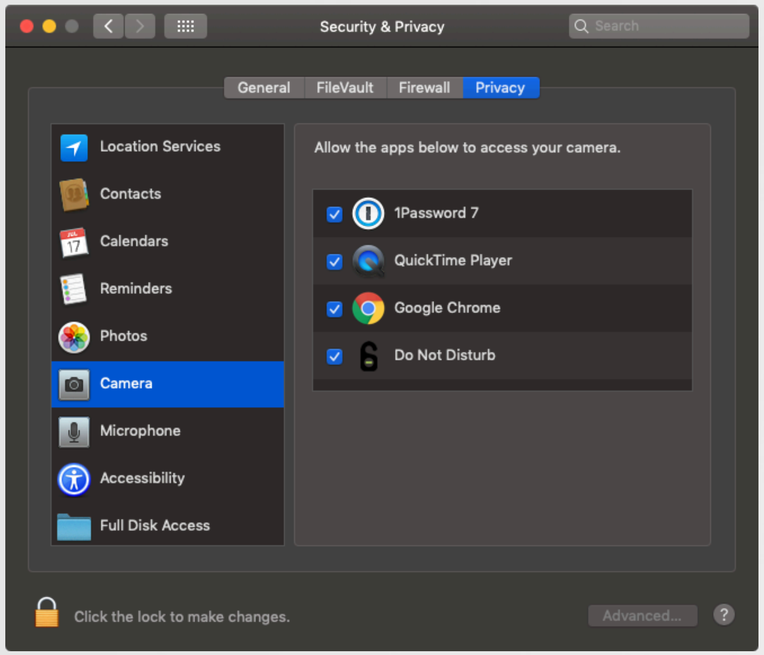
Digita Security

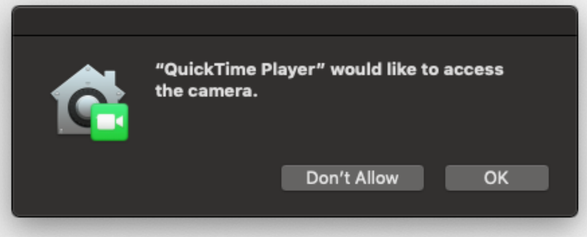
Improved Privacy Protections

camera, mic, & private user files

i spy malware:

- OSX.FruitFly
- OSX.Eleanor
- OSX.Crisis
- OSX.Mokes

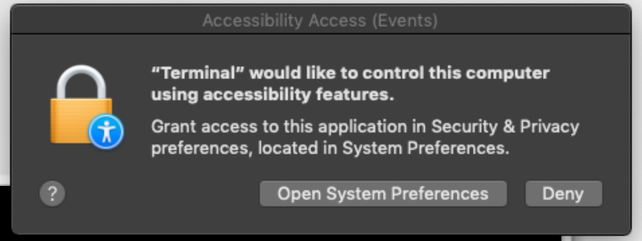




mojave (privacy) alerts

Improved Privacy Protections

camera, mic, and private user files

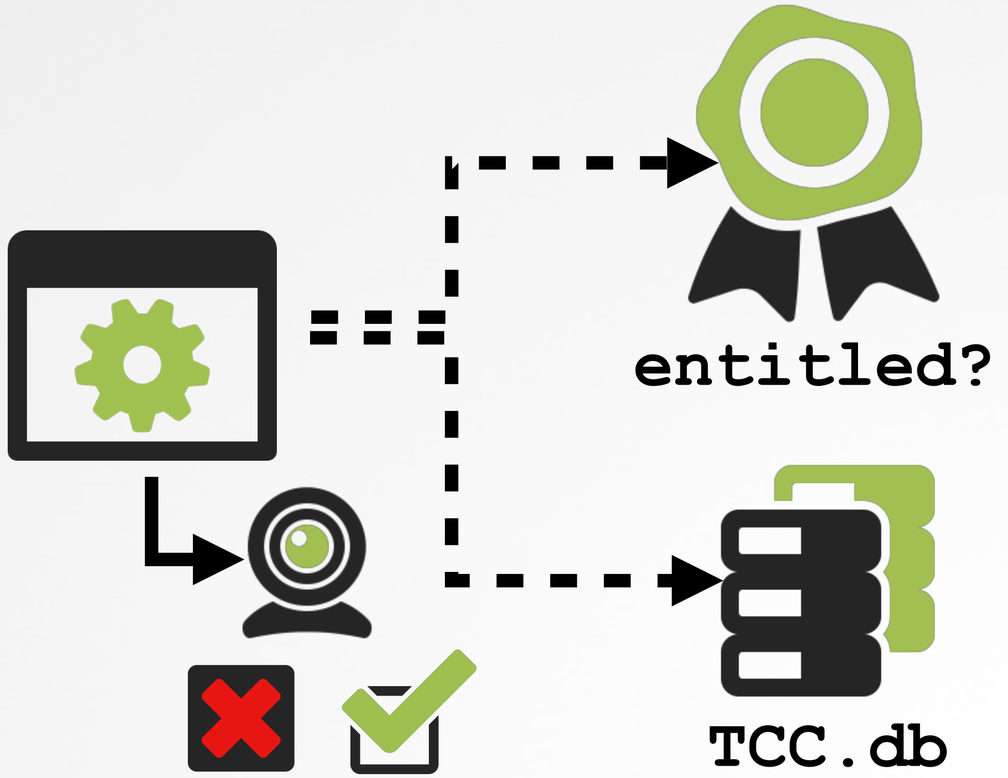



```
# fs_usage -w -f filesystem
  stat64     ~/Library/Application Support/com.apple.TCC/TCC.db-journal        tccd.2978
  stat64     ~/Library/Application Support/com.apple.TCC/TCC.db-wal            tccd.2978
  RdData[A] ~/Library/Application Support/com.apple.TCC/TCC.db                 tccd.2978
  guarded_open_np  ~/Library/Application Support/com.apple.TCC/TCC.db-journal tccd.2978
```

checking 'TCC.db'

```
$ codesign -d --entitlements - /Applications/FaceTime.app

  <?xml version="1.0" encoding="UTF-8"?>

    <plist version="1.0">
    <dict>

      <key>com.apple.private.tcc.allow</key>
      <array>
        <string>kTCCServiceAddressBook</string>
        <string>kTCCServiceReminders</string>
        <string>kTCCServiceMicrophone</string>
        <string>kTCCServiceCamera</string>
      </array>
      ...
```

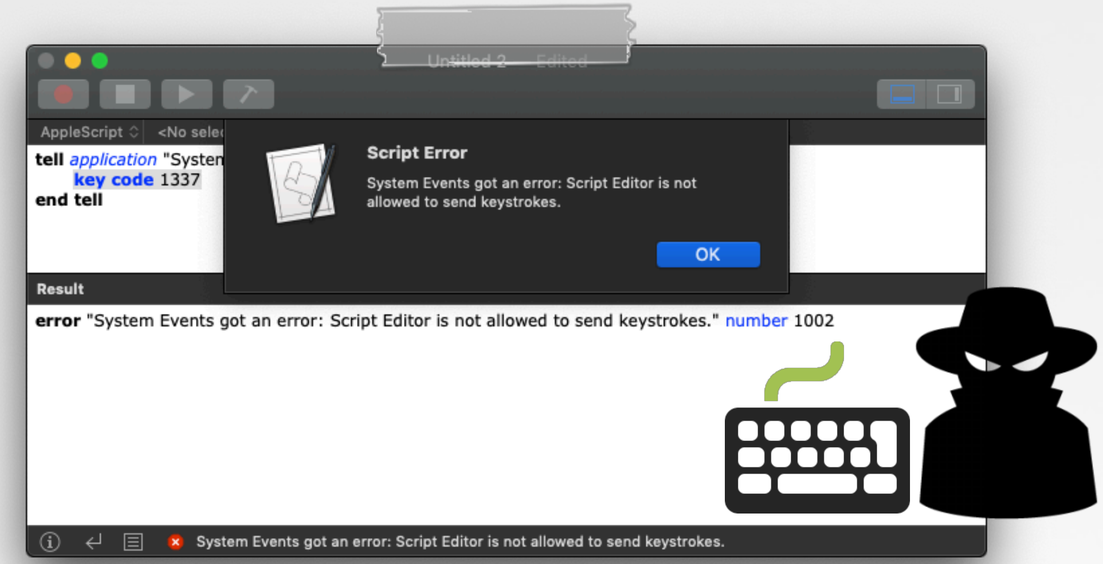
'com.apple.private.tcc.allow' entitlements



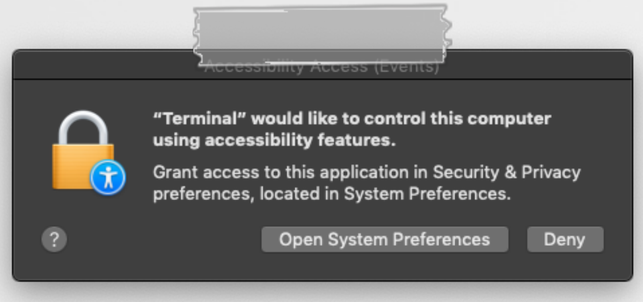
Blocking Synthetic Events

mouse clicks/keystrokes now *generically* blocked

```
//create point
CGPoint point = CGPointMake(atoi(argv[1]), atoi(argv[2]));

//mouse down
CGPostMouseEvent(point, true, 1, true);

//mouse up
CGPostMouseEvent(point, true, 1, false);
```



synthetic keypress: blocked



synthetic mouse click: blocked

synthetic attacks/exploits (pre-mojave)

"*The Mouse is Mightier than the Sword*" (p. wardle)

Apple Event Sandboxing

authorization now required to send Apple Events

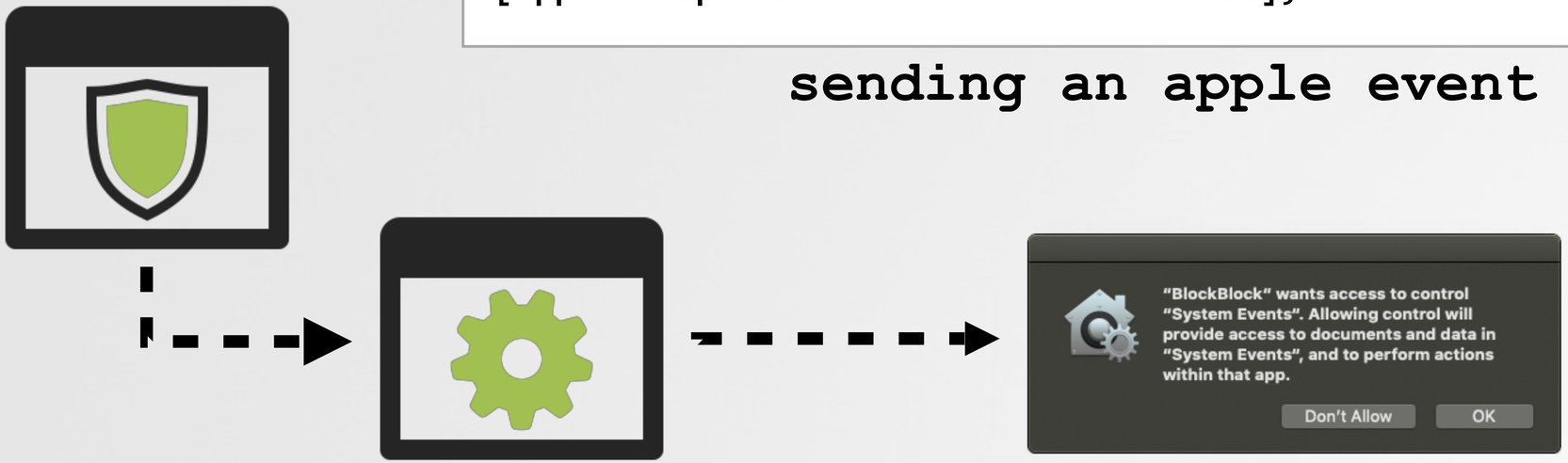
@felix_schwarz

"Apple Event sandboxing in macOS Mojave"

→ "*apps can no longer send Apple Events to other apps without user authorization*"

```
appleScriptCmd = [NSString stringWithFormat:@"tell application \"System Events\"
to delete login item \"%@\"", name];

appleScript = [[NSAppleScript alloc] initWithSource:appleScriptCmd];
[appleScript executeAndReturnError:nil];
```

sending an apple event (blockblock)

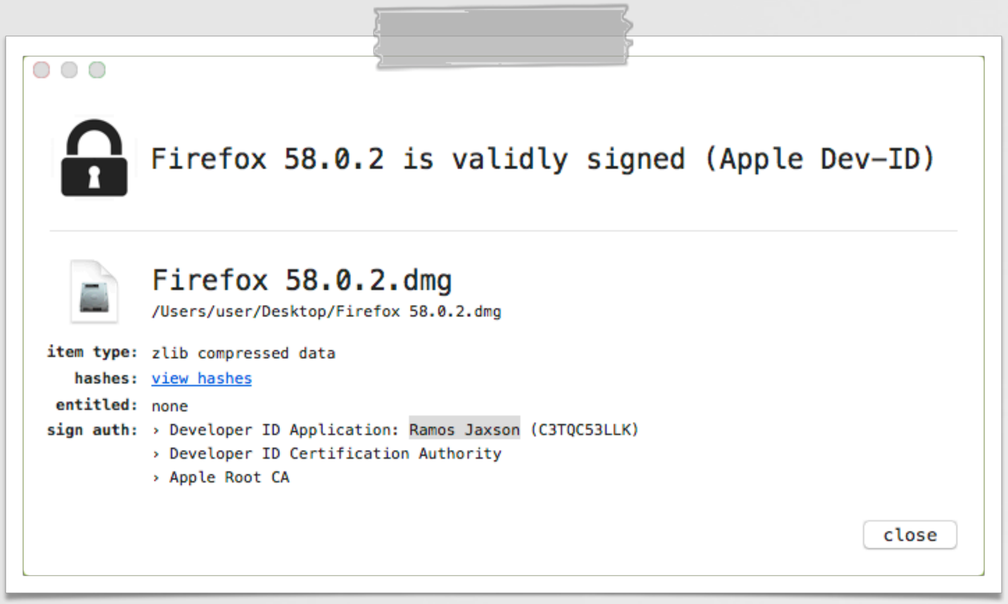


(ab)using applescript / events to bypass (pre-mojave)

"*Making and Breaking Mac Firewalls*" (p. wardle)

Application Notarization

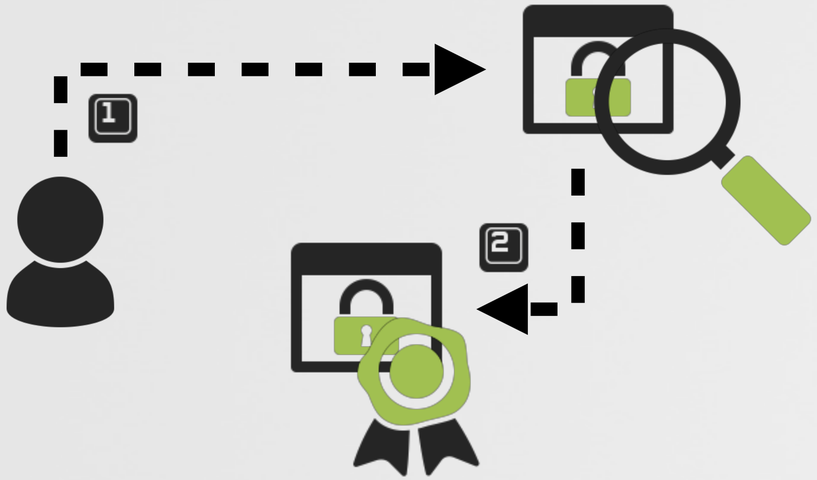
validating 'external' applications



OSX.CreativeUpdater

fraudulent/stolen DevID

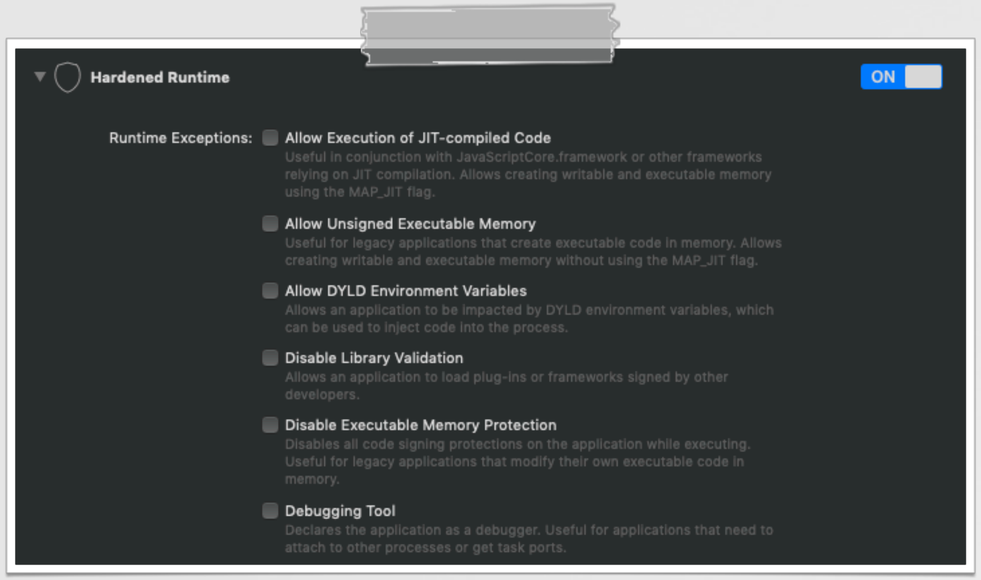
Gatekeeper allows signed apps



1. signed app submitted to Apple
2. after scanning, Apple "notarizes"
   - → *both* DevID & app are verified

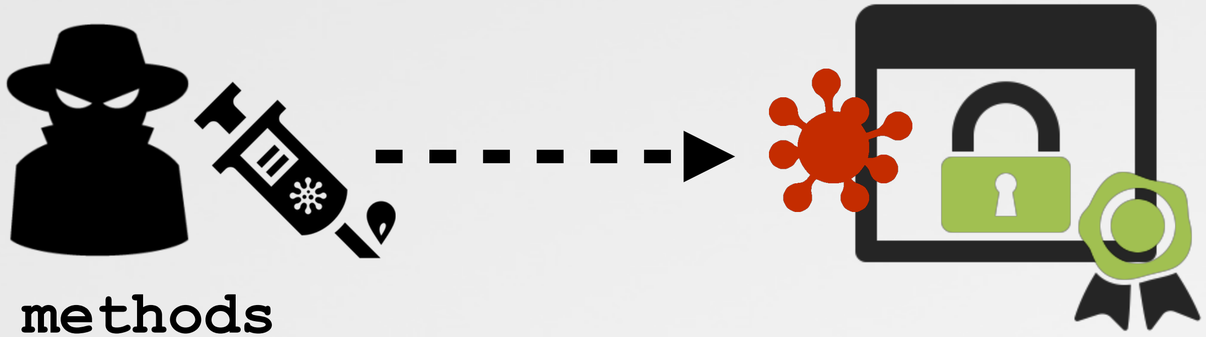
Application Notarization

requires the 'hardened runtime'



hardened runtime
(specify at built-time)

thwarts these:



methods
of injection:

write to remote memory

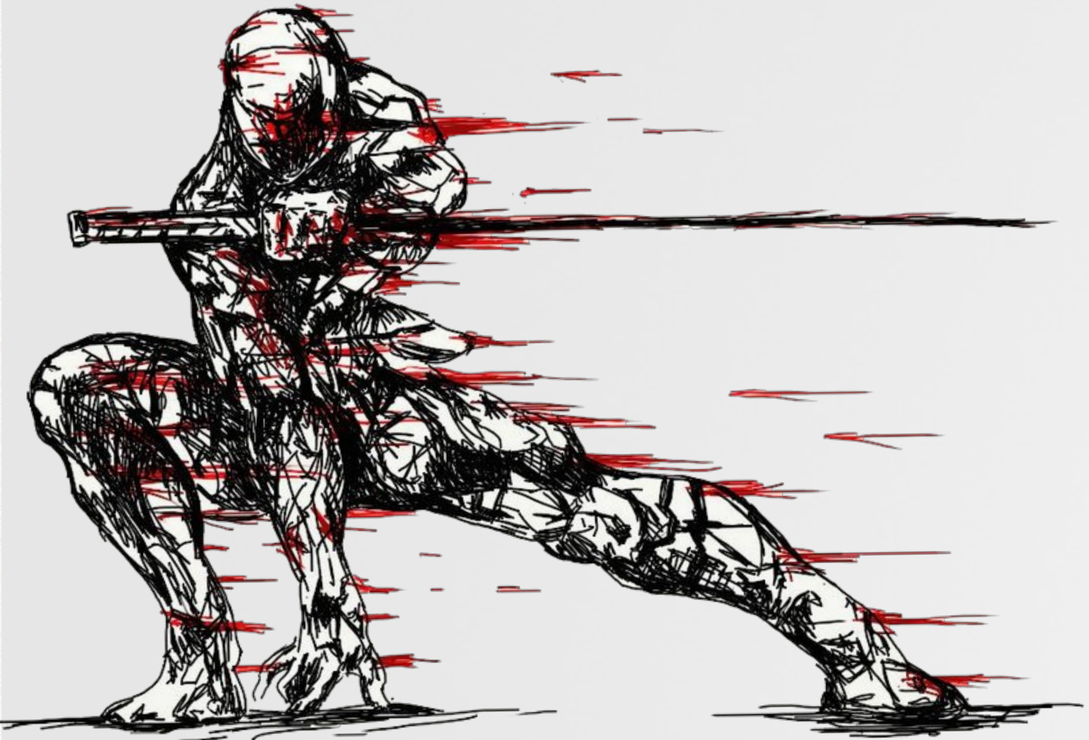
malicious plugins

environment variables

dylib proxying

3rd-Party Protections

generic detection of behaviors



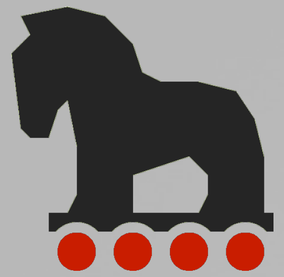
Runtime Detection of Malicious Events

the best method of defense?

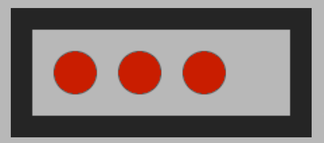
detect these actions!

infection vectors

emails



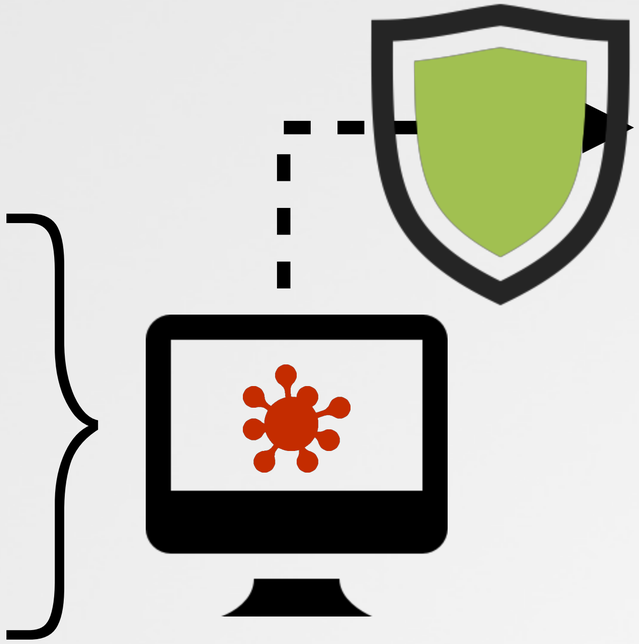
trojans



hacked logins

web popups

0days



will get infected

persist

spy

ransom

exfiltrate

Man charged with spying on thousands of Mac users for 13 years

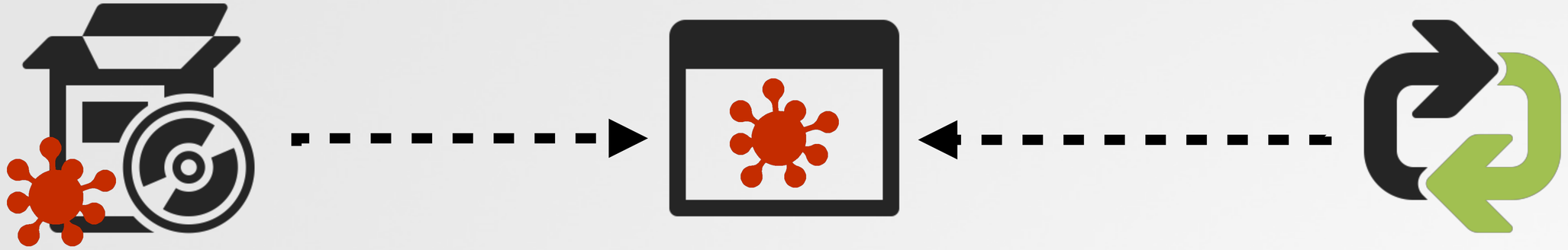
traditional AV isn't going to save you

Persistence

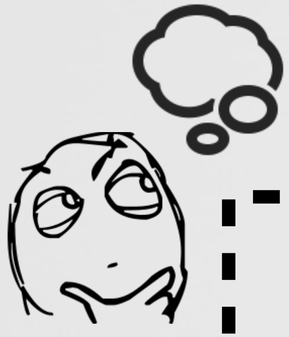
malware persists; let's detect that!

persistence, defined

malware installs such that it will be *automatically* restarted by the OS on (any) reboot



malware's installer | persistent component | system reboot

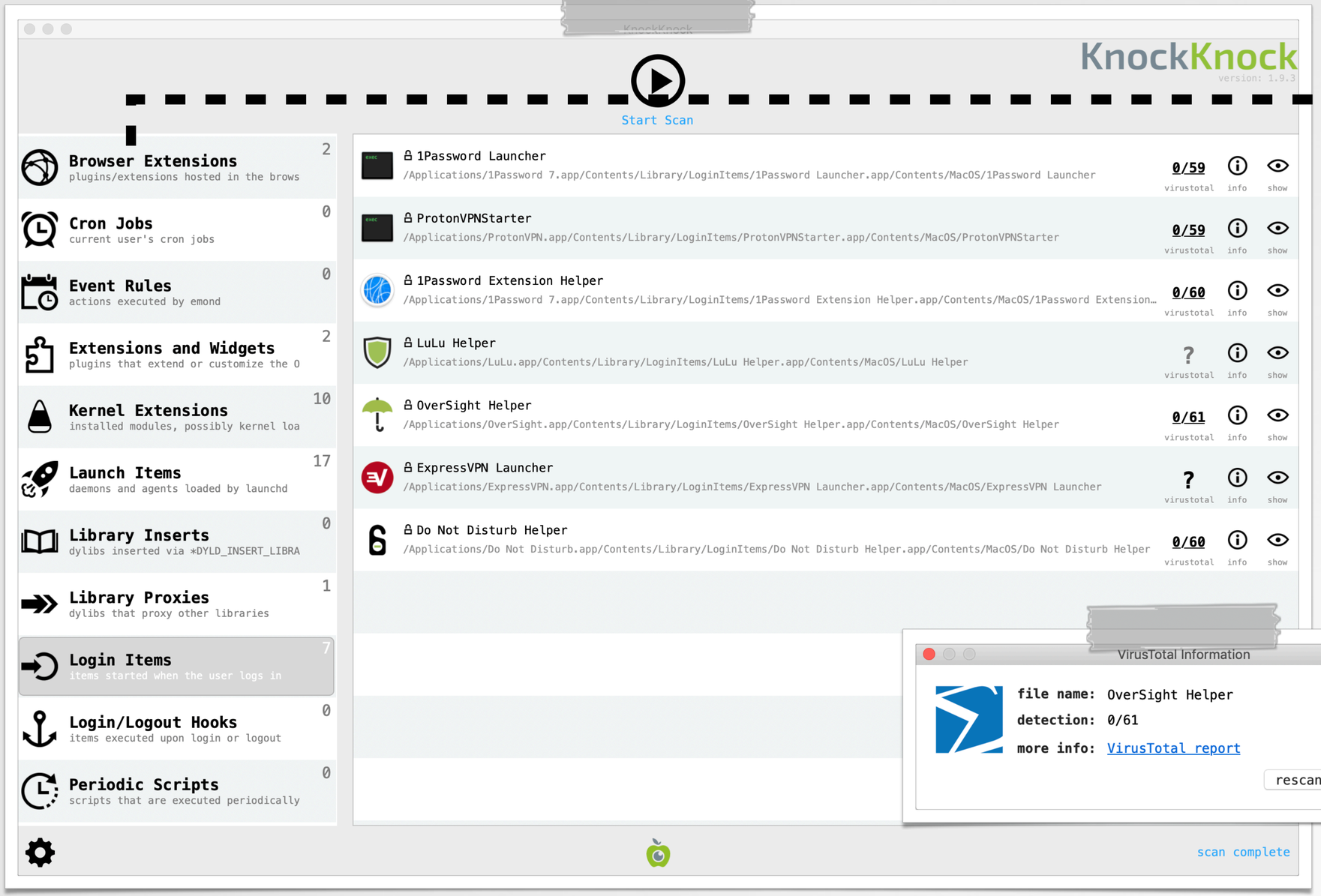


1. malware persists
2. finite methods of persistence
3. ...generically detect malware via persistence

PDF "Methods of Malware Persistence on Mac OS X" (p. wardle)

Persistence

detect persistent software/malware with KnockKnock



KnockKnock

categories

persistent items

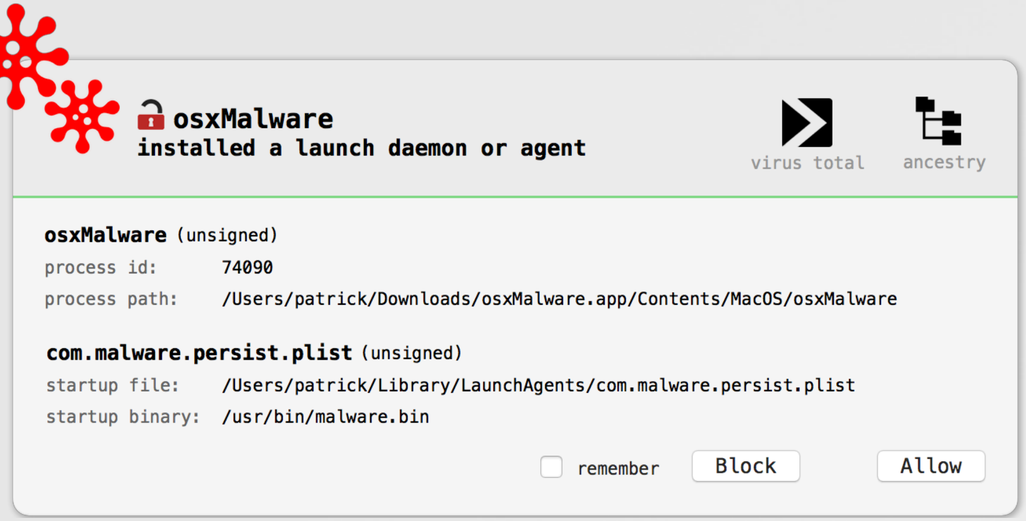
path

signing info

virustotal integration

Persistence

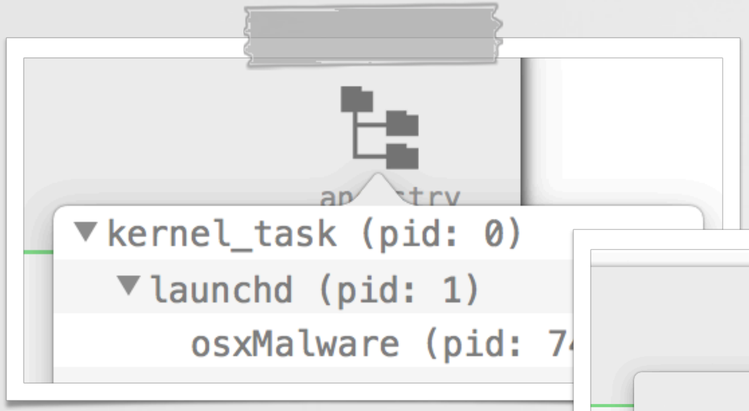
prevent persistent software/malware with BlockBlock



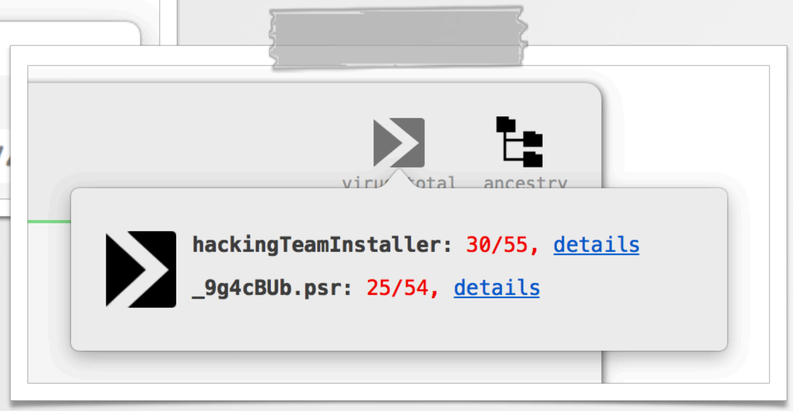
BlockBlock alert

how does BlockBlock work?

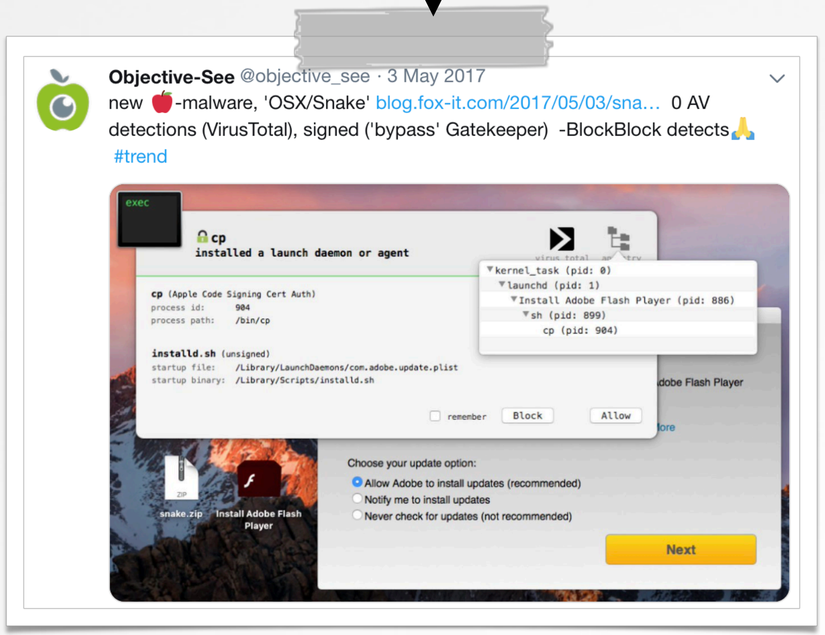
1. monitor persistence locations
2. alert user



process hierarchy



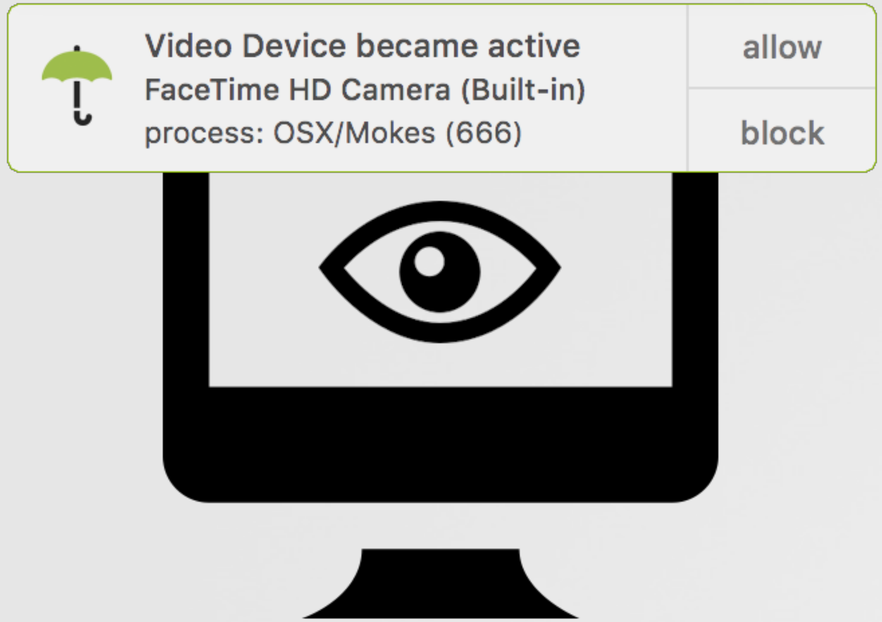
virustotal integration



generic detection(s)!

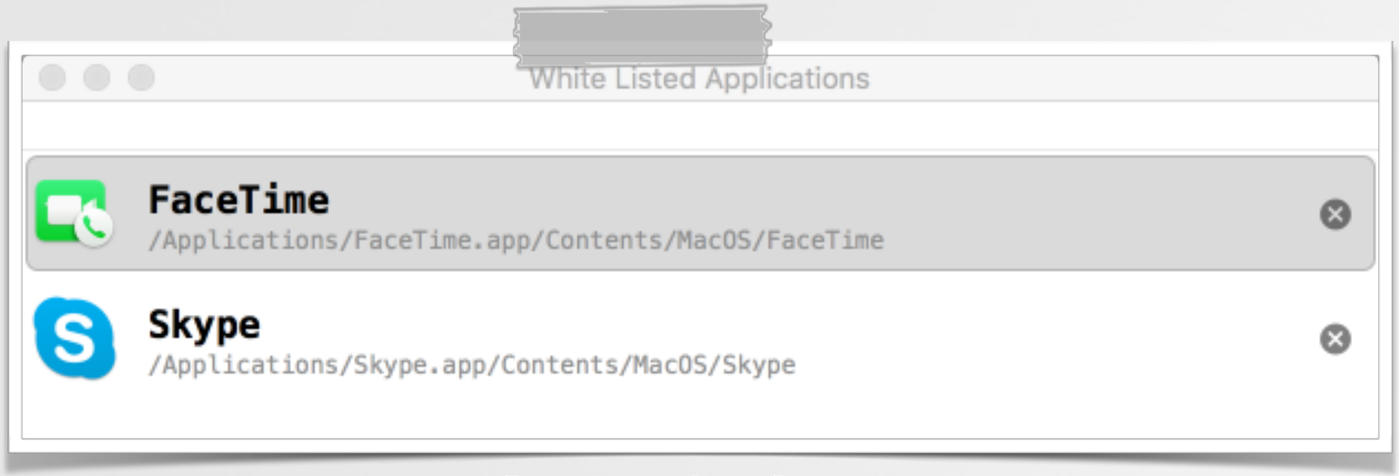
Detecting Webcam/Mic Access

tool: OverSight



how does OverSight work?

1. detects audio/video use
2. identify process
3. alert user



rules window

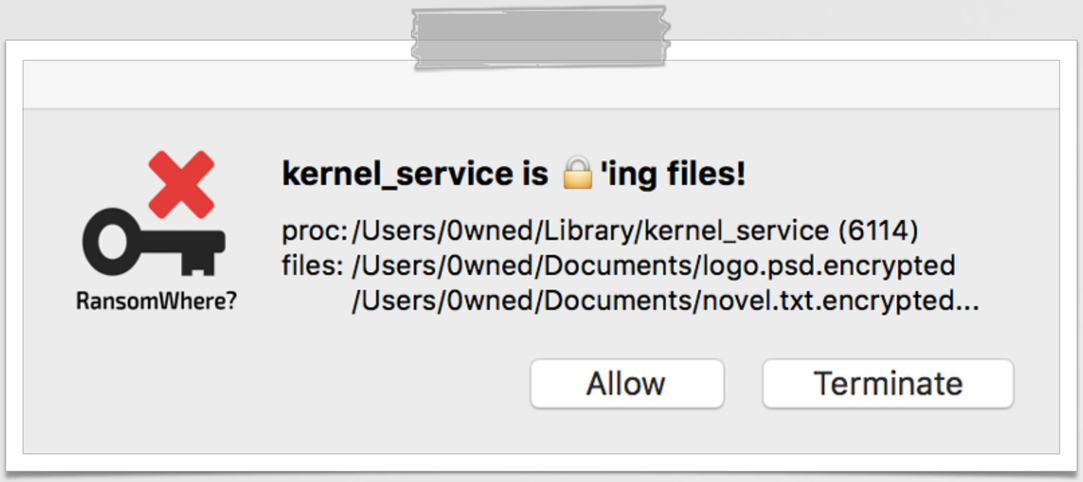
generic detection(s)!

Detecting Ransomware

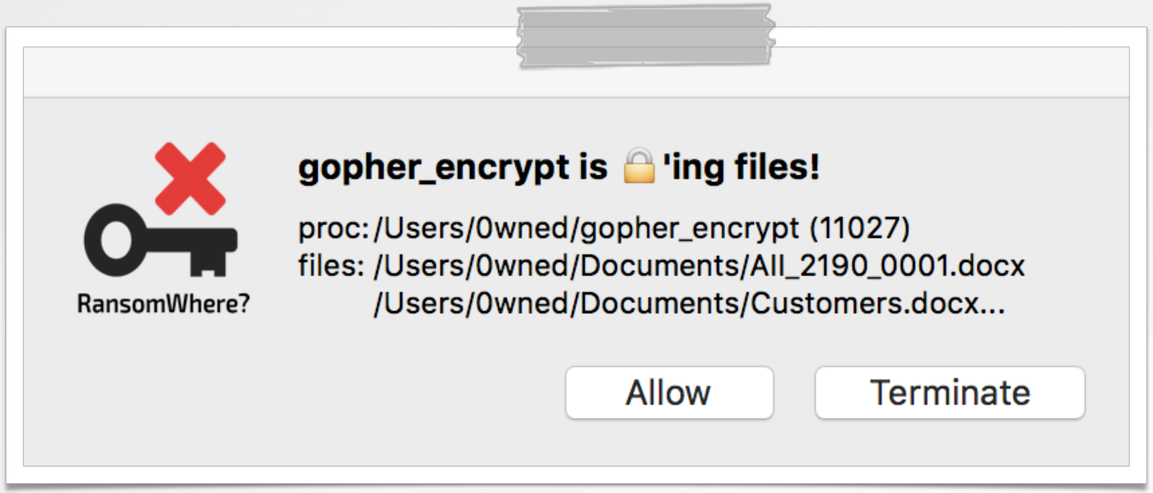
tool: RansomWhere?

research/technical details

"Towards Generic Ransomware Detection"



OSX.KeyRanger



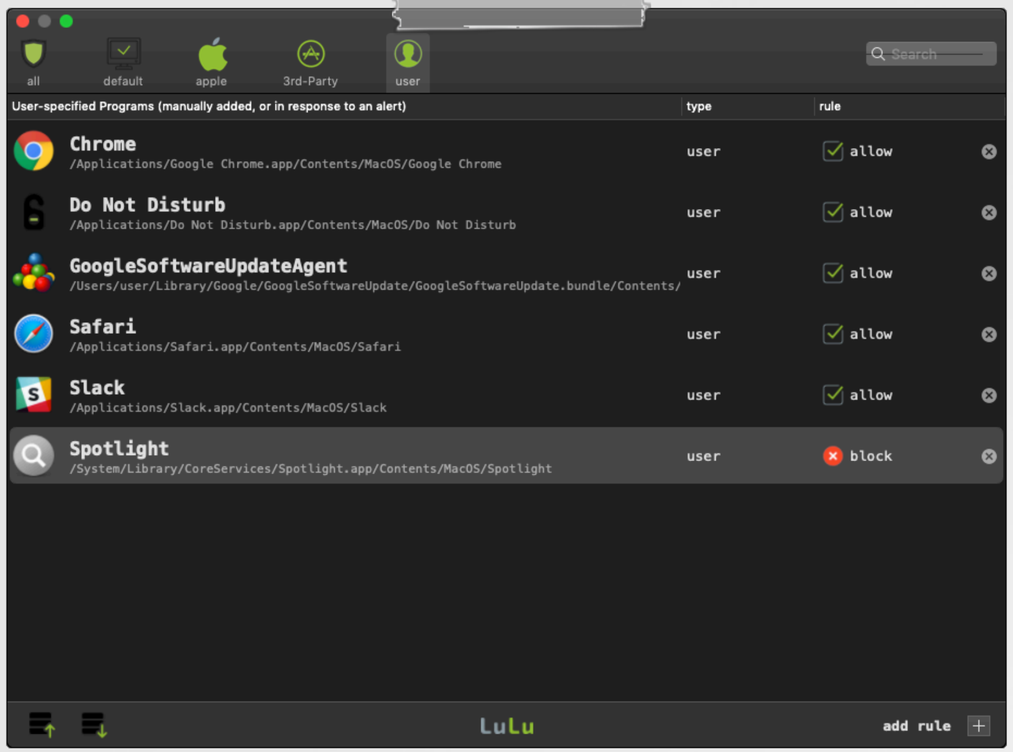
OSX.Gopher

RansomWhere?

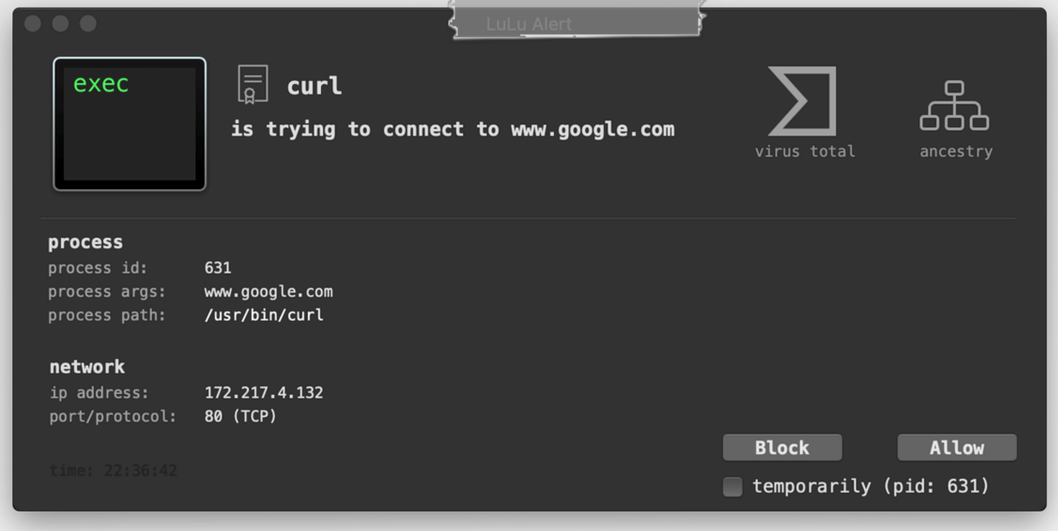
- creating encrypted files
- rapidly / high number
- by an untrusted process

Detecting Exfiltration

tool: Lulu (firewall)



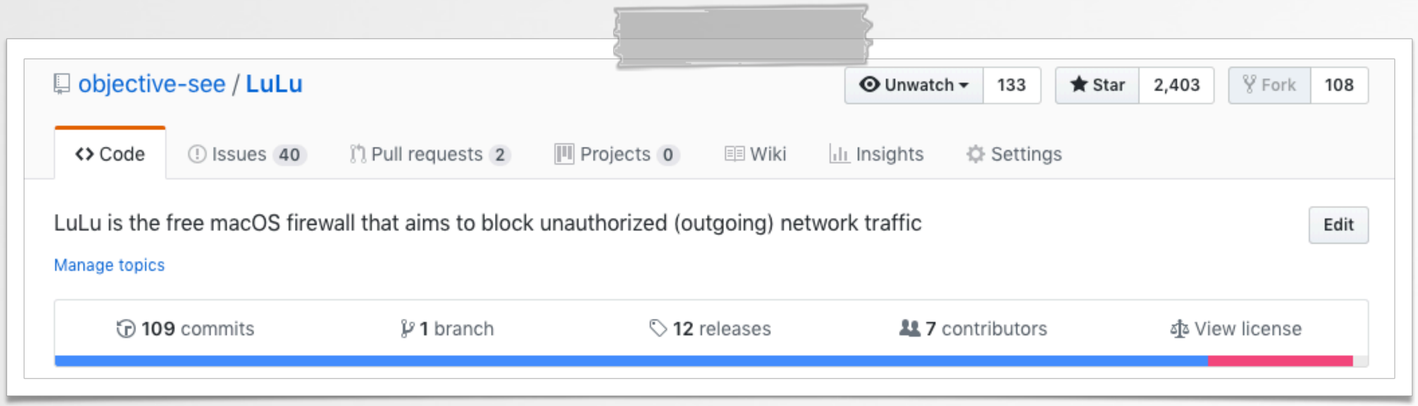
rules



alert window

Lulu's full source

github.com/objective-see/LuLu



All in One

MonitorKit + GamePlan (Digita)

MonitorKit

all events

macOS gaming engine

```
($event.file.path LIKE[cd] "/System/Library/Extensions/*" OR
$event.file.path LIKE[cd] "/Library/Extensions/*") AND
$event.isNewDirectory == 1
```

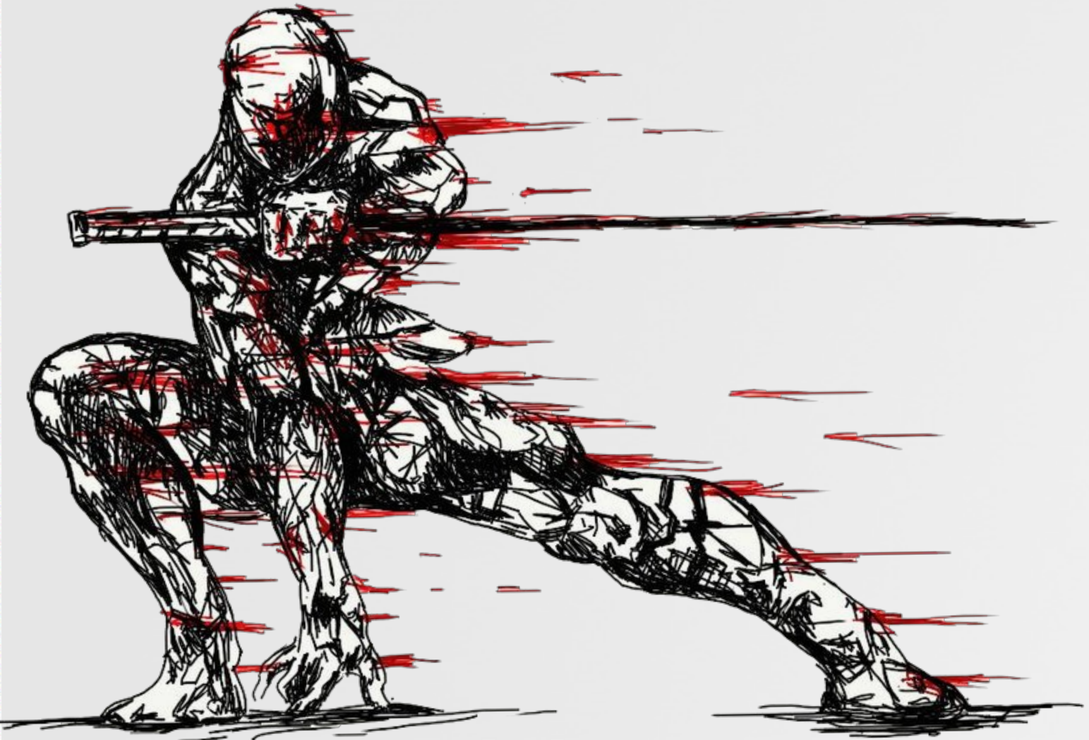
predicate

fact: "new kext"

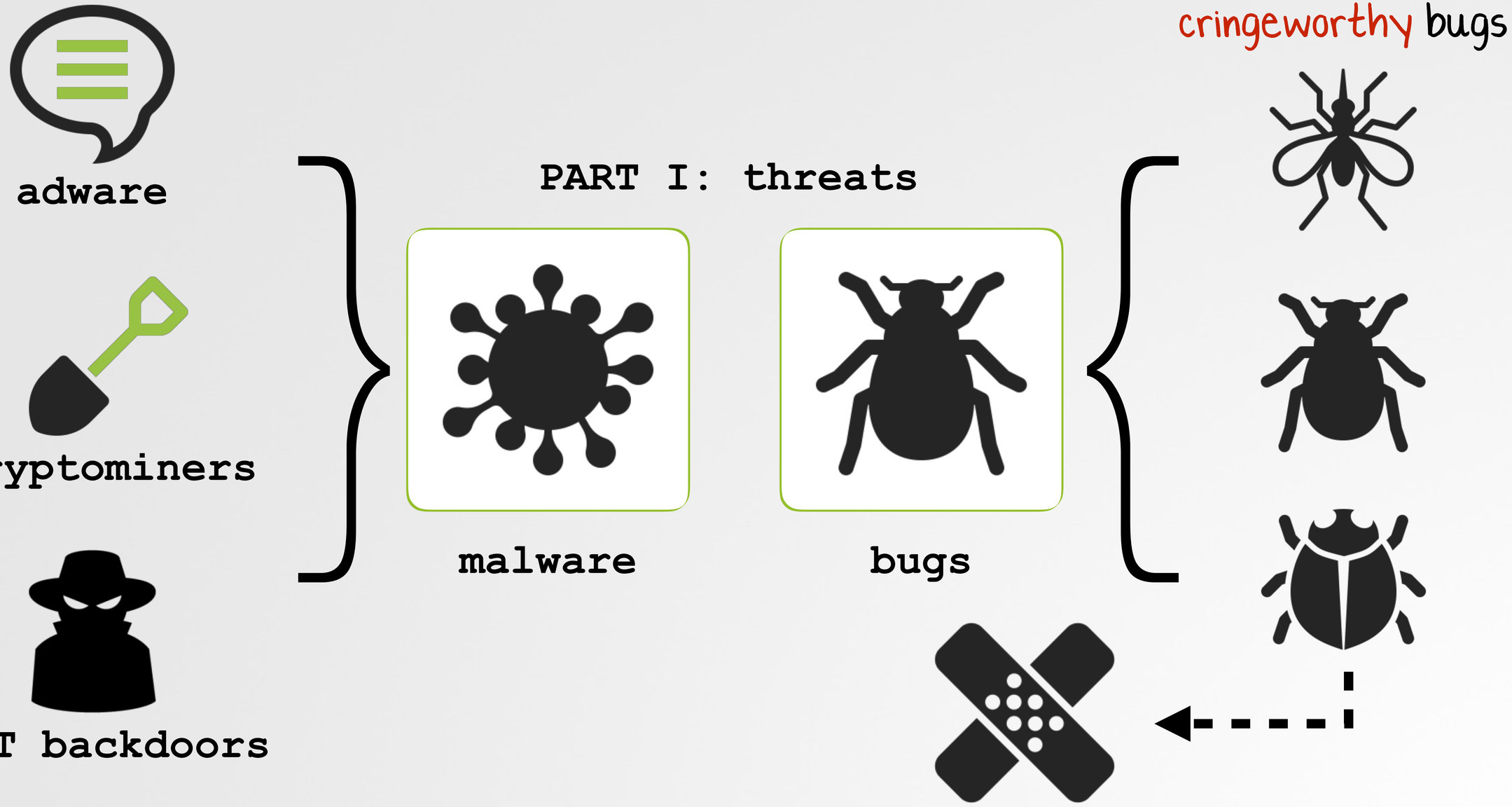
action: alert & collect

Conclusion

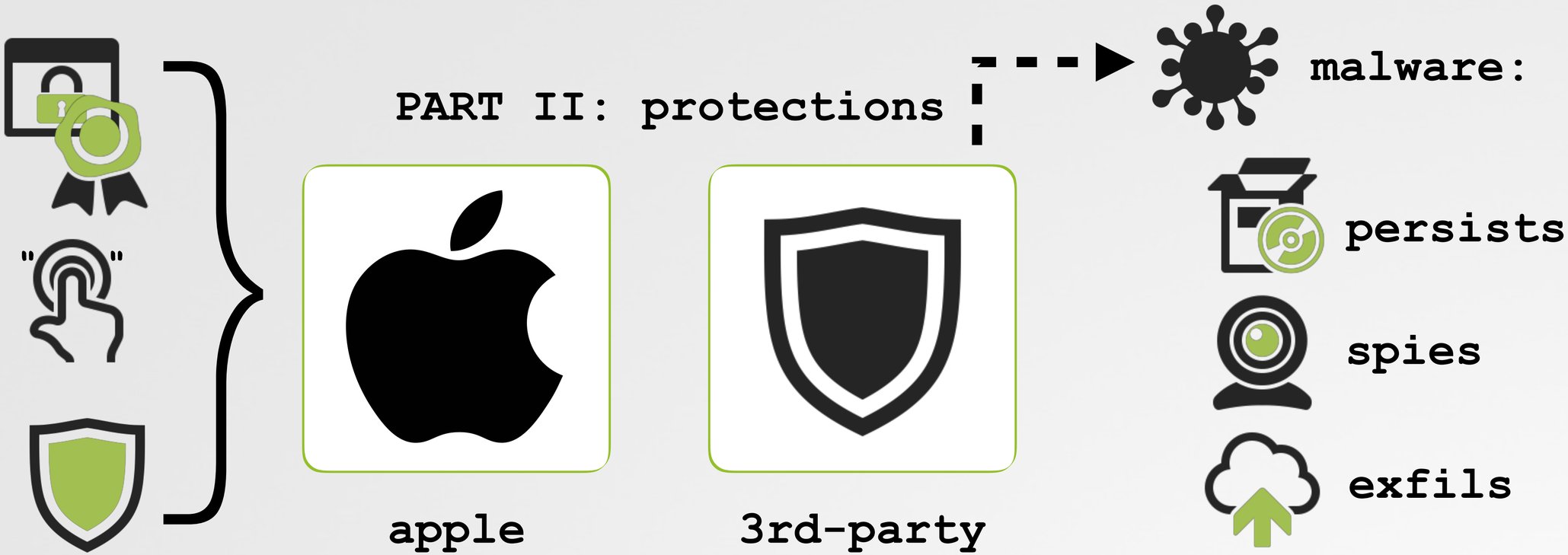
wrapping this up

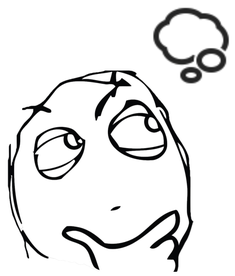


Conclusions



Conclusions





macOS's built-in mitigations aren't enough

3rd-party security tools can address shortcomings and ensure user's remain secure!

Finale

@patrickwardle

friends of Objective-See

Digita Security

SOPHOS

Malwarebytes

SmugMug

SecureMac

Guardian Mobile Firewall

digita security

cybersecurity solutions for the mac enterprise

Credits

images

- iconexperience.com
- wirdou.com/2012/02/04/is-that-bad-doctor
- http://pre04.deviantart.net/2aa3/th/pre/f/2010/206/4/4/441488bcc359b59be409ca02f863e843.jpg

resources

- opensource.apple.com
- newosxbook.com (*OS Internals)
- github.com/objective-see/LuLu
- securelist.com/operation-applejeus/87553/