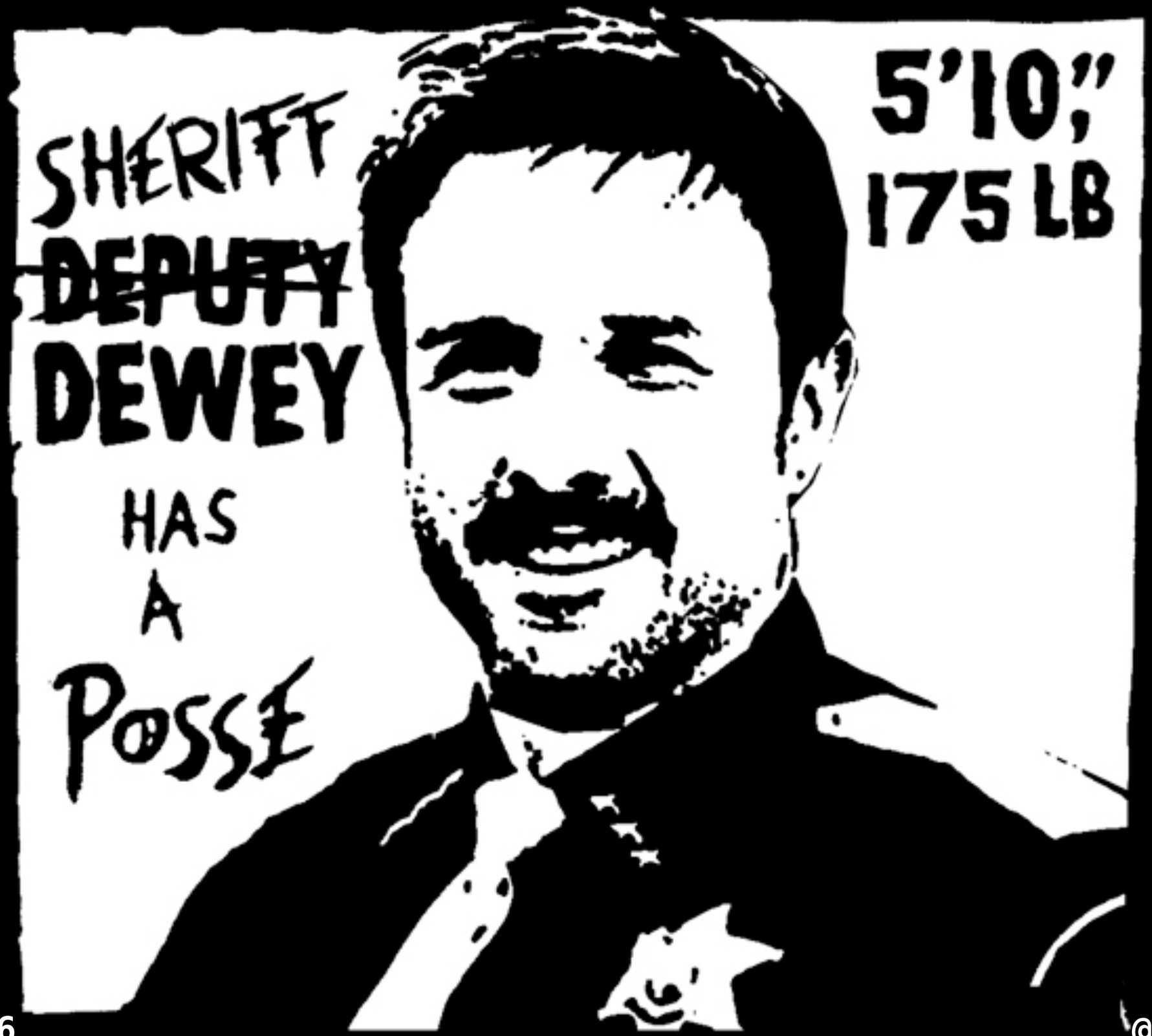
Hiding From the Investigator

Understanding OS X and iOS Code Signing to Hide Data



Why this talk?

If you:

- Do OS X/iOS forensics or incident response
- Use Apple Code Signing as part of your workflow
- Are not really sure how to check code signing properly

Ground Rules

- Post install
- No direct malicious code execution
- See Patrick Wardle’s talk on Sunday

Windows

```
echo “No longer signed” >> signedWindowsPE.exe
```

OS X/iOS

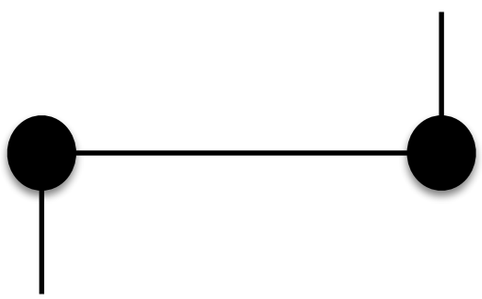
```
echo “Still Signed… Sorta” >> signedOSXbinary
```

‘Bug’ Reporting Timeline

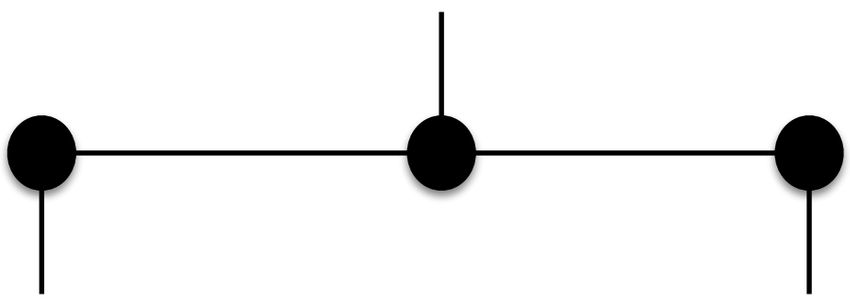
‘Bug’ Reporting Timeline

April 2015 –
Reported to Apple

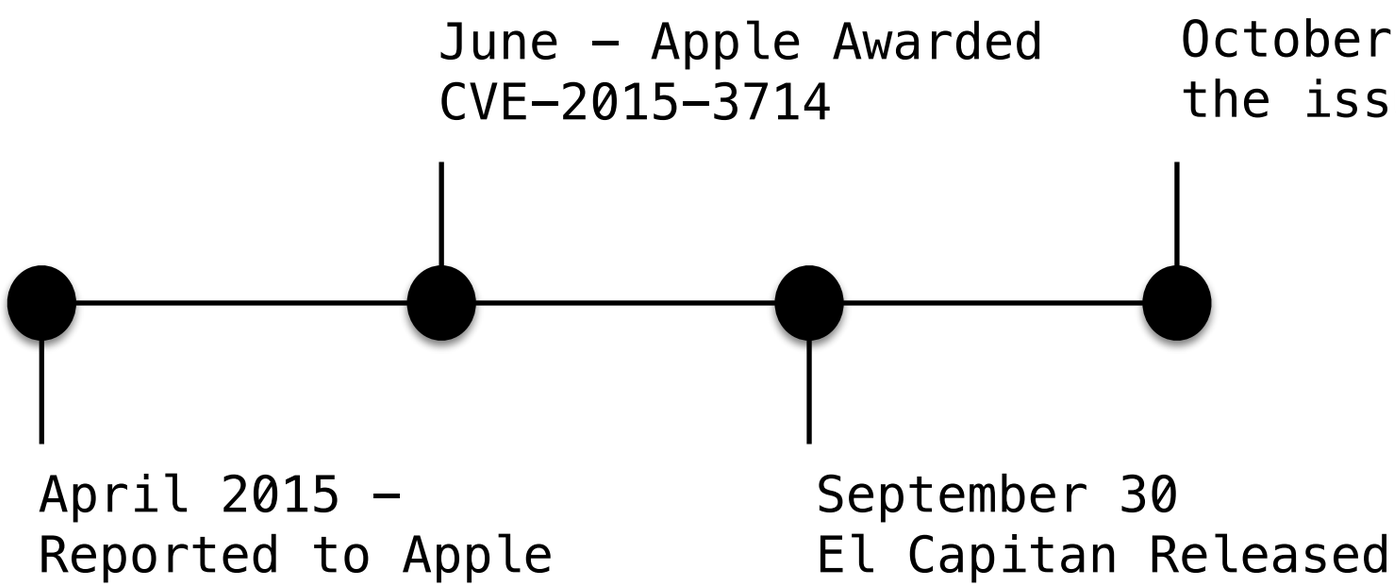
‘Bug’ Reporting Timeline



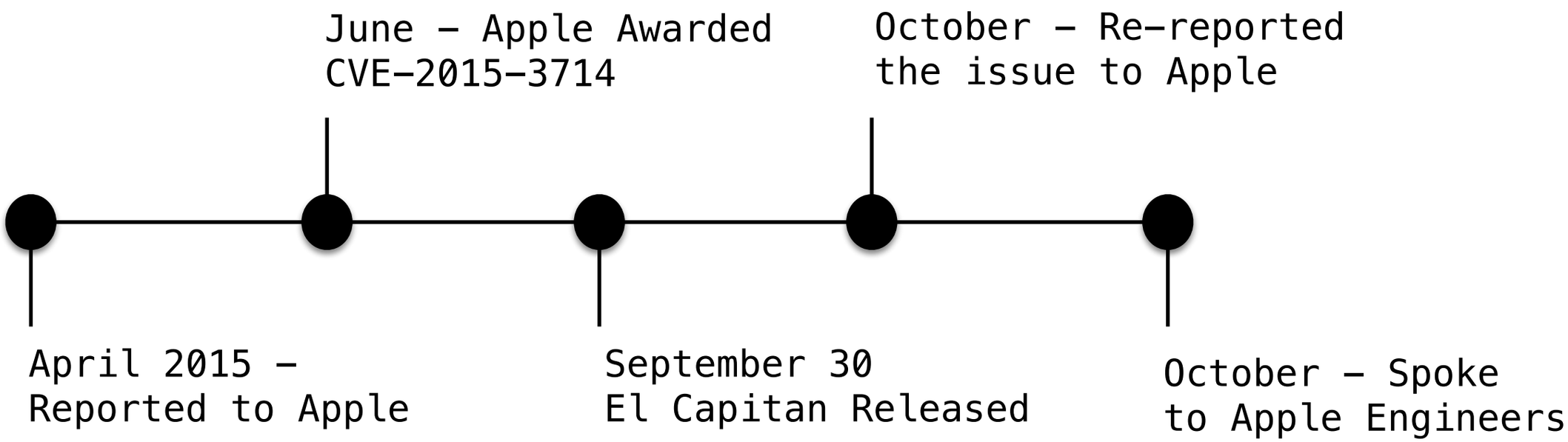
‘Bug’ Reporting Timeline



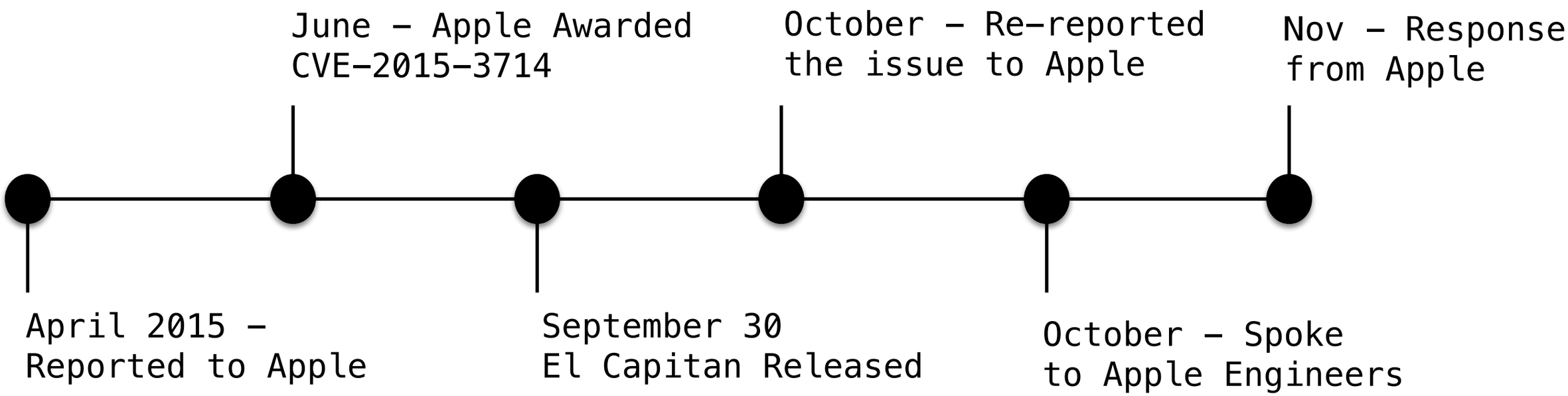
‘Bug’ Reporting Timeline



‘Bug’ Reporting Timeline



‘Bug’ Reporting Timeline

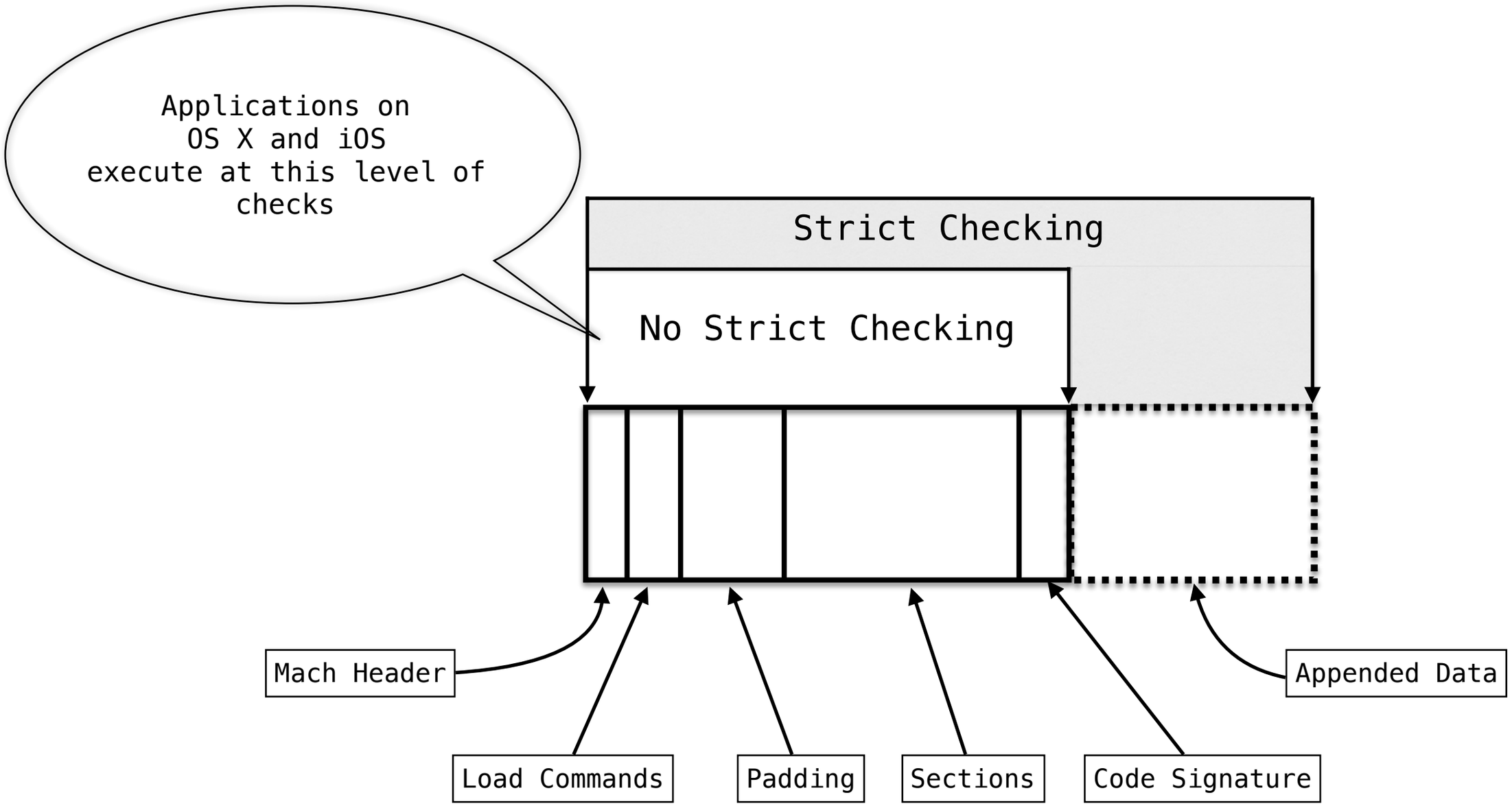


Apple’s Response

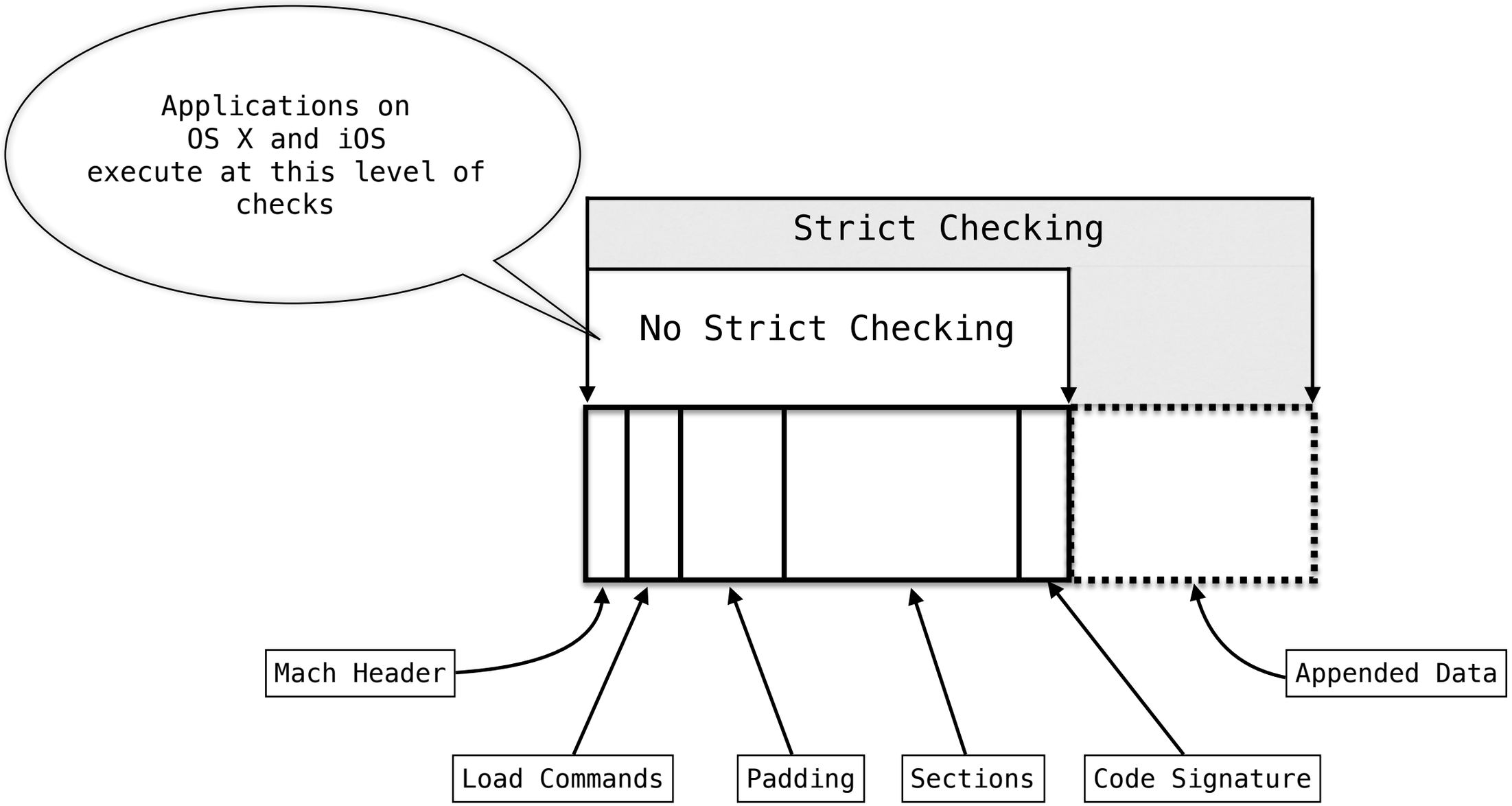
TL;DR Version:
Use only Strict Checking

OSX/iOS CodeSigning
Strict vs No-Strict

Mach-O Binary

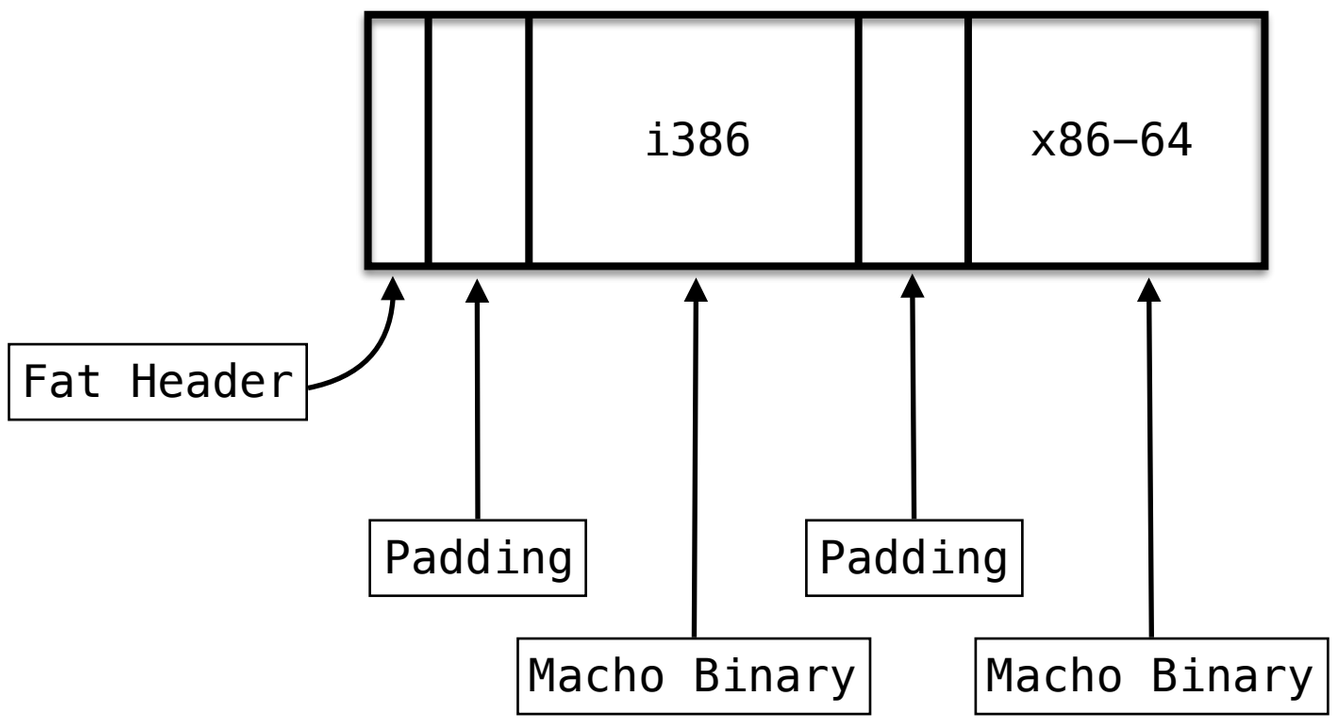


Mach-O Binary

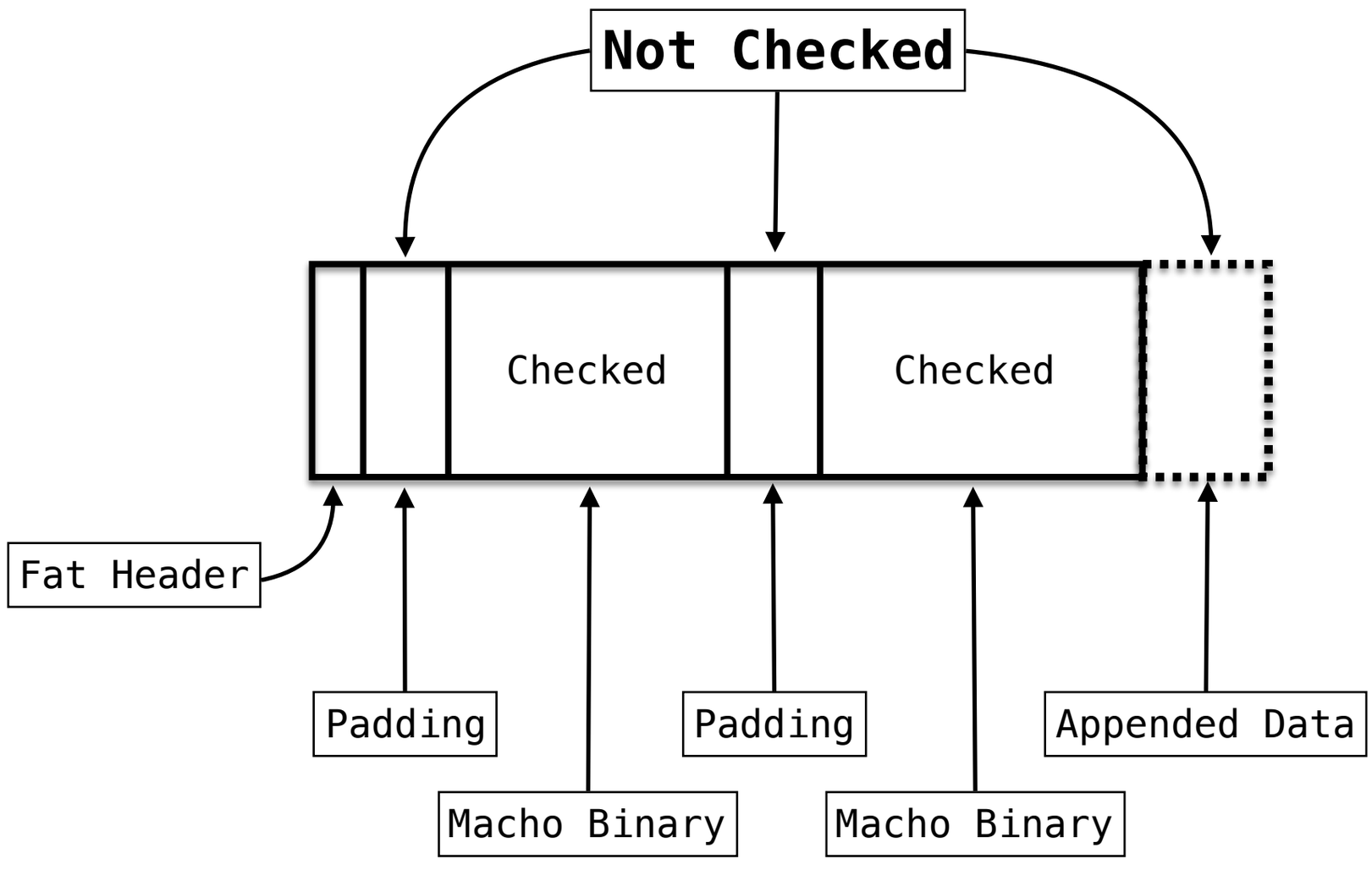


On iOS there is only “No Strict” Checking – Strict checks don’t exist.

Fat/Universal Binary



Fat/Universal Binary



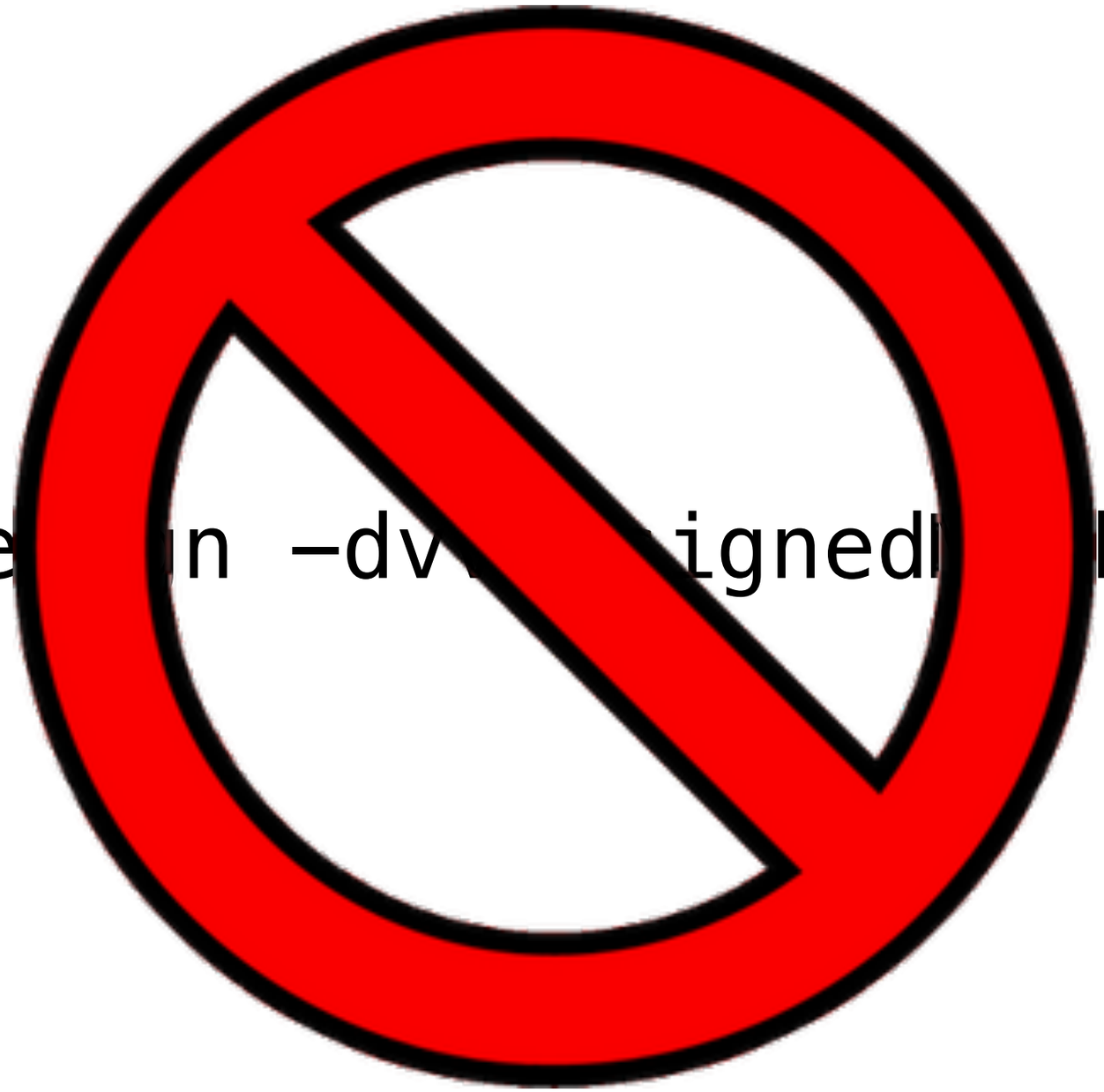
What incident response/ forensics / Security tools use strict checking?

[This slide is blank because no one does it]

Tools that use No-Strict Checking

- KnockKnock
  - OSXCollector
- TaskExplorer
- KextViewr
- Google MacOps-molcodesignchecker (https://github.com/google/macops-molcodesignchecker)
  - Google-Santa
- OSQuery

```
$ codesign -dvvv signedMacho|app
```



```
$ codesign -dvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvvho|app
```

Strict Checking

```
$ codesign -vv signedMacho|app
OR
$ codesign --verify -vvvv signedMacho|app
OR
$ codesign -vv --deep signedMacho|app
OR
$ codesign -vvvv signedMacho|app
OR
$ codesign --verbose=4 signedMacho|app
```

No-Strict Checking

```
$ codesign --verify -vv --no-strict signedMacho|app
```

Invalid Code Envelope

```
➜ /tmp  codesign -dvvv ./ls
Executable=/private/tmp/ls
Identifier=com.apple.ls
Format=Mach-O thin (x86_64)
CodeDirectory v=20100 size=261 flags=0x0(none) hashes=8+2 location=embedded
Hash type=sha1 size=20
CDHash=b583404214ff4e0bee6e0662731bff5555c24621
Signature size=4097
Authority=Software Signing
Authority=Apple Code Signing Certification Authority
Authority=Apple Root CA
Info.plist=not bound
TeamIdentifier=not set
Sealed Resources=none
Internal requirements count=1 size=60
➜ /tmp  codesign -vv ./ls
./ls: invalid signature (code or signature have been modified)
In architecture: x86_64
➜ /tmp  codesign -vv --no-strict ./ls
./ls: invalid signature (code or signature have been modified)
In architecture: x86_64
➜ /tmp
```

Appended Data

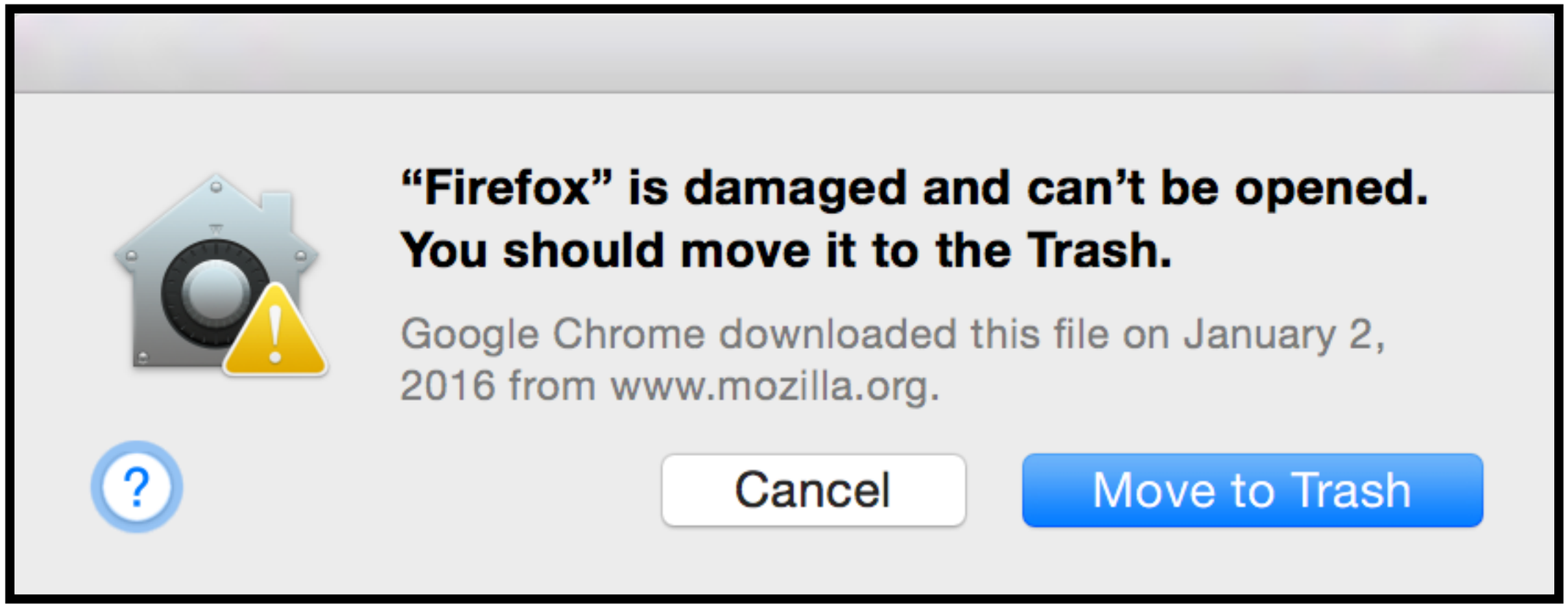
```
➜ kyphosis  codesign -dvvv warmd
Executable=/Users/jp/github/kyphosis/warmd
Identifier=com.apple.warmd
Format=Mach-O thin (x86_64)
CodeDirectory v=20100 size=664 flags=0x0(none) hashes=28+2 location=embedded
Hash type=sha1 size=20
CDHash=518c748f6b5e8534776315f20b88a6185988fd9f
Signature size=4097
Authority=Software Signing
Authority=Apple Code Signing Certification Authority
Authority=Apple Root CA
Info.plist=not bound
TeamIdentifier=not set
Sealed Resources=none
Internal requirements count=1 size=64
➜ kyphosis  codesign -vv warmd
warmd: main executable failed strict validation
➜ kyphosis  codesign -vv --no-strict warmd
warmd: valid on disk
warmd: satisfies its Designated Requirement
```

Codesign Fail

```
bash-3.2$ codesign -vv /Applications/Firefox.app
/Applications/Firefox.app: An internal error has occurred.
```

One change in the Fat File padding section
oops…

Gatekeeper Says No



Introducing LipoCram

- Increases the size of padding between the Fat Header and the first Mach-o binary
- Adds data
- iOS limit to ~ 15K of data between sections
- No Limit on OSX

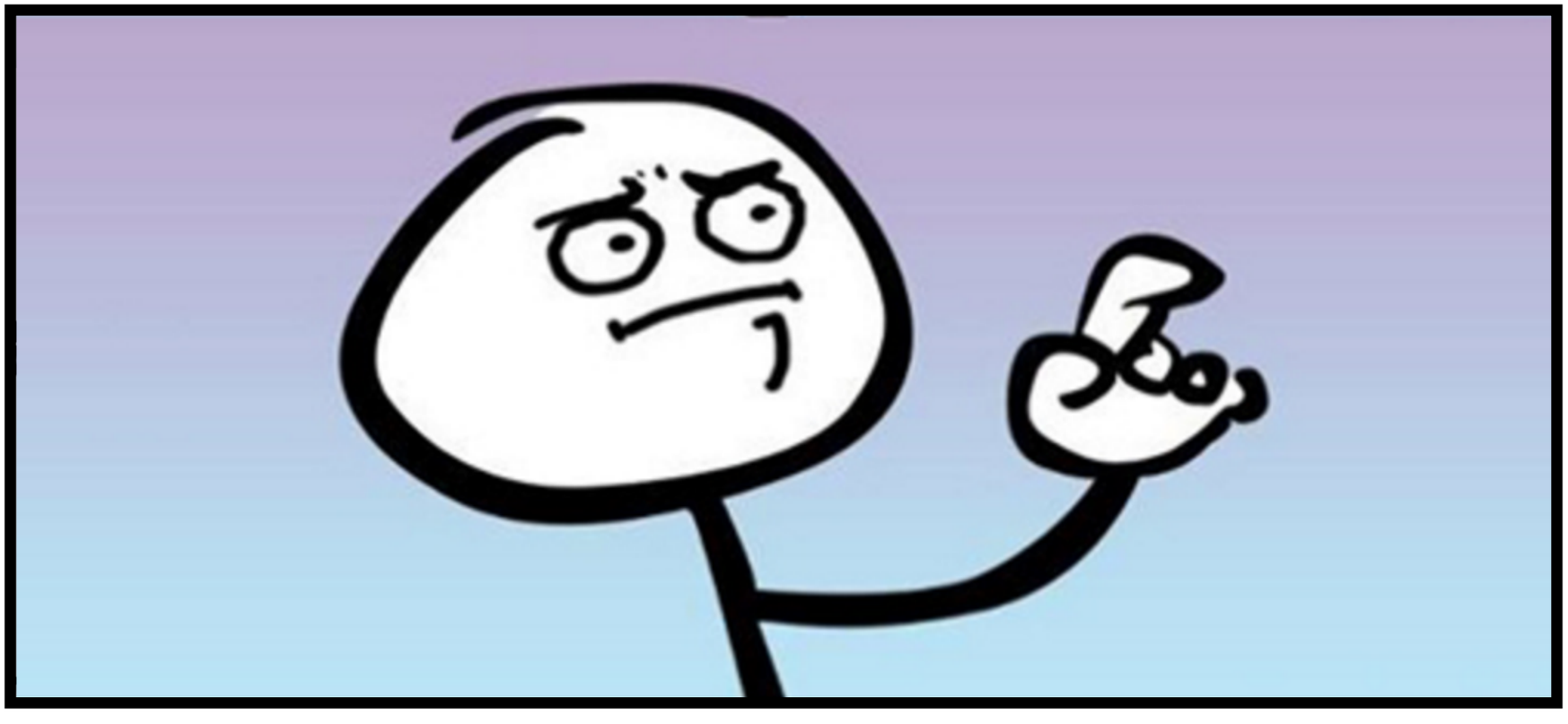
LipoCram Example

```
➜ lipocram  ./lipocram.py
Usage: ./lipocram.py Fatfile DataToCram
➜ lipocram  file firefox
firefox: Mach-O universal binary with 2 architectures
firefox (for architecture x86_64):      Mach-O 64-bit executable x86_64
firefox (for architecture i386):        Mach-O executable i386
➜ lipocram  ./lipocram.py firefox ls
[*] Checking padding size against payload
[*] Finding page aligned new offset for storage area
[*] Size of new padding 0x9000
[*] New data location in Fat File 0x428
[*] Writing ls to FAT file
[*] Fixing up the FAT Headers
```


But Wait..
File Carvers!!

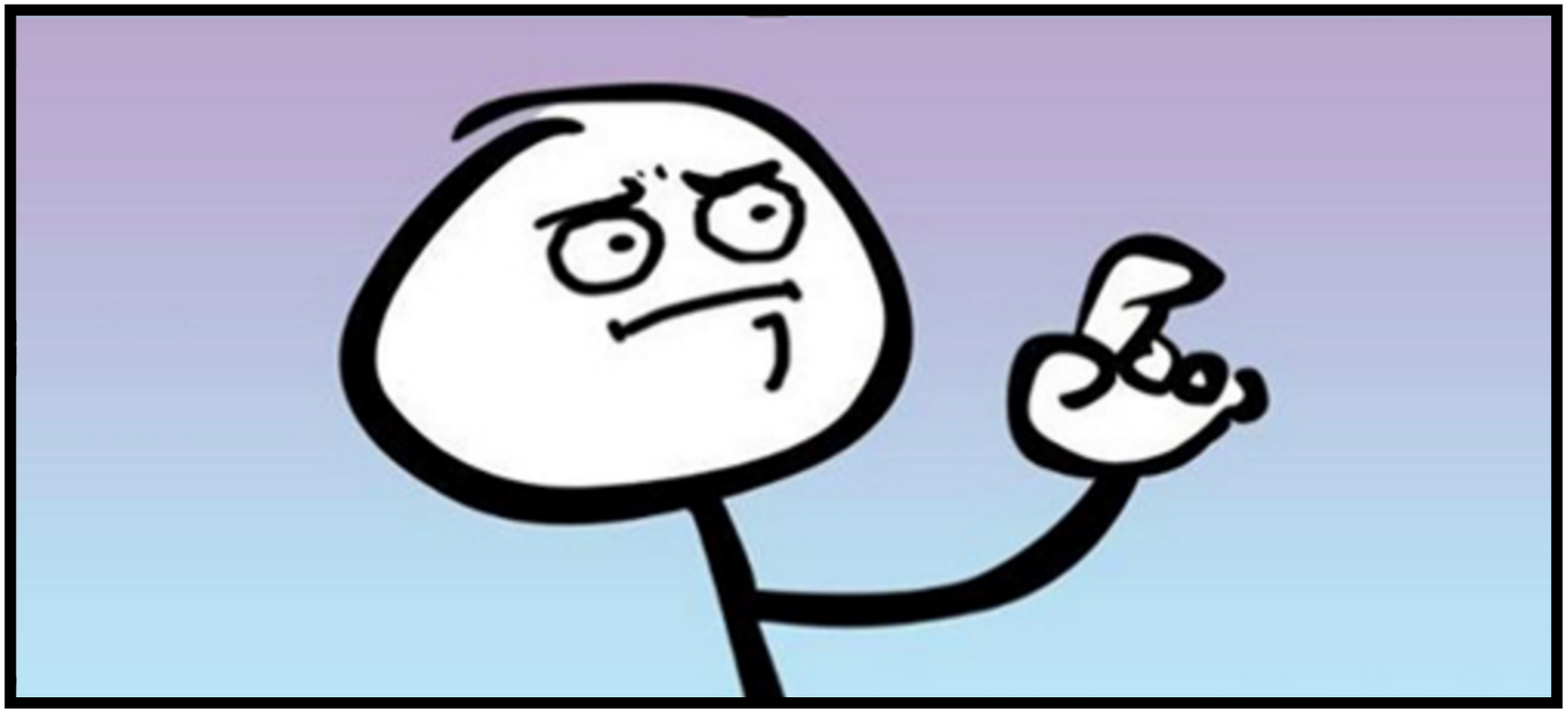
>>

```
openssl enc -e -k moo -aes-256-cbc -in its_happening.gif | tail -c +9 >> firefox
```



>>

```
openssl enc -e -k moo -aes-256-cbc -in its_happening.gif | tail -c +9 >> firefox
```



Side loading a modified iOS Application

```
➜  Mobile Applications  ls
Chromecast 1.12.5715.ipa cc
➜  Mobile Applications  cp Chromecast\ 1.12.5715.ipa cc
➜  Mobile Applications  cd cc
```

```
➜  cc  unzip Chromecast\ 1.12.5715.ipa
Archive:  Chromecast 1.12.5715.ipa
   creating: META-INF/
  inflating: META-INF/com.apple.ZipMetadata.plist
 extracting: META-INF/com.apple.FixedZipMetadata.bin
   creating: Payload/
   creating: Payload/Chromecast.app/
   creating: Payload/Chromecast.app/_CodeSignature/
  inflating: Payload/Chromecast.app/_CodeSignature/CodeResources
  inflating: Payload/Chromecast.app/Info.plist
  inflating: Payload/Chromecast.app/Chromecast
 extracting: Payload/Chromecast.app/AppIcon29x29.png
```

```
➜  cc  cd Payload
➜  Payload  ls
Chromecast.app
➜  Payload  codesign -vv Chromecast.app
Chromecast.app: resource envelope is obsolete (custom omit rules)
➜  Payload  codesign -vv --no-strict Chromecast.app
Chromecast.app: valid on disk
Chromecast.app: satisfies its Designated Requirement
```

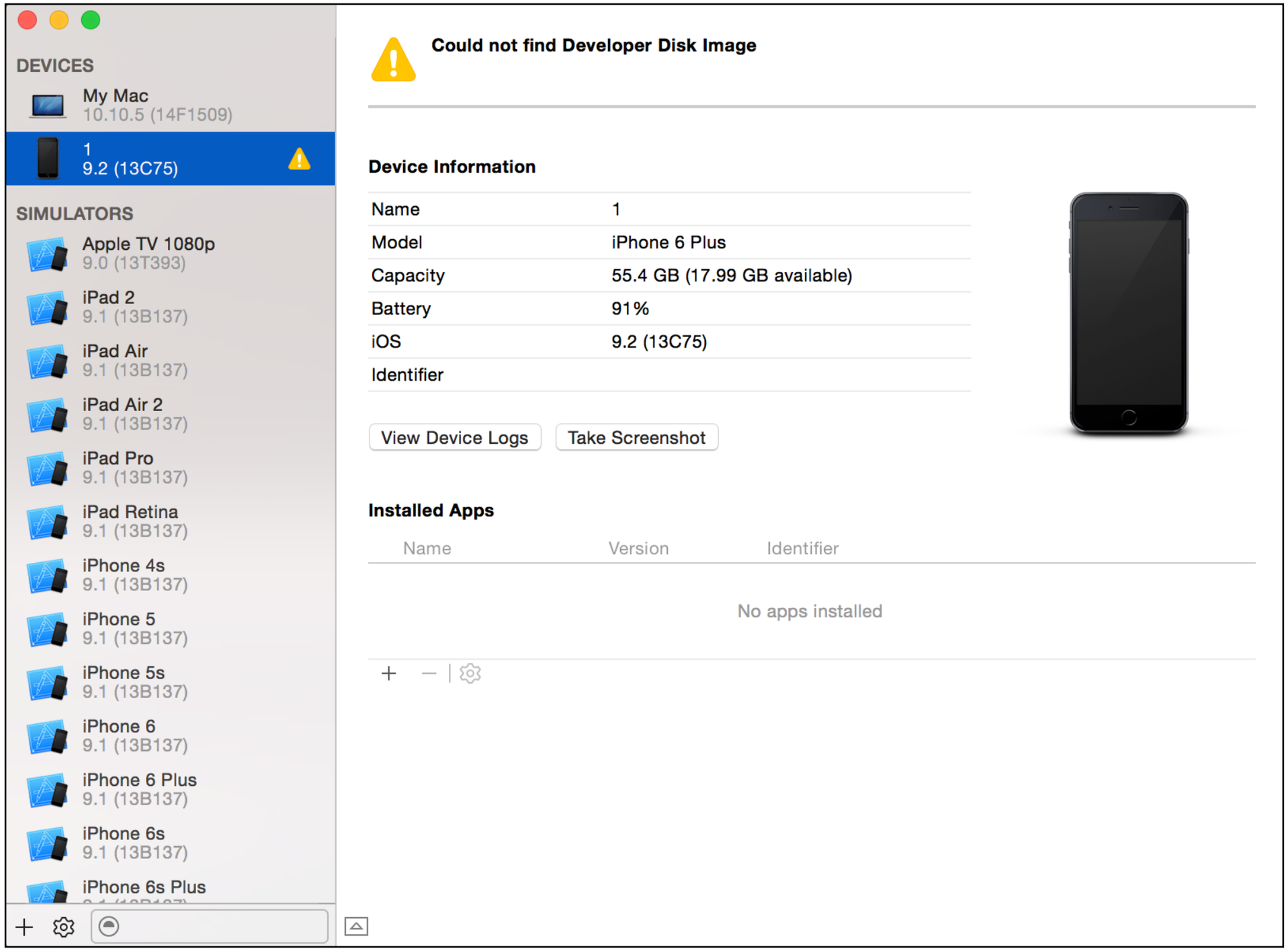
```
➜  Payload  cd Chromecast.app
➜  Chromecast.app  ll Chromecast
-rwxr-xr-x  1 jp  staff    13M Sep 15 17:17 Chromecast
➜  Chromecast.app  cat /dev/random >> Chromecast
^C
➜  Chromecast.app  ll Chromecast
-rwxr-xr-x  1 jp  staff   2.1G Jan  2 18:08 Chromecast
➜  Chromecast.app  cd ..
```

```
➜  Payload  codesign -vv --no-strict Chromecast.app
Chromecast.app: valid on disk
Chromecast.app: satisfies its Designated Requirement
```

```
➜  Payload  cd ..
➜  cc  rm Chromecast\ 1.12.5715.ipa
➜  cc  zip -r Chromecast\ 1.12.5715.ipa ./*
  adding: META-INF/ (stored 0%)
  adding: META-INF/com.apple.FixedZipMetadata.bin (stored 0%)
  adding: META-INF/com.apple.ZipMetadata.plist (deflated 17%)
  adding: Payload/ (stored 0%)
  adding: Payload/Chromecast.app/ (stored 0%)
```

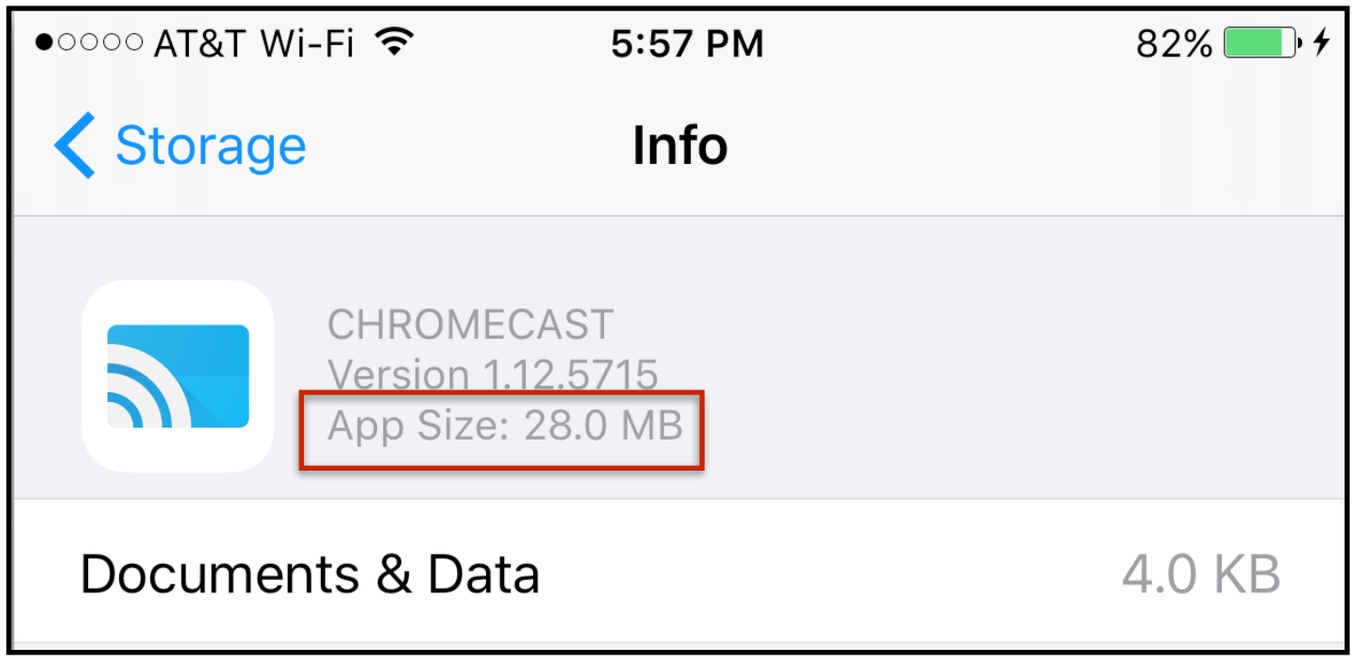
```
➜  cc  ll Chromecast\ 1.12.5715.ipa
-rw-r--r--  1 jp  staff   2.1G Jan  2 18:15 Chromecast 1.12.5715.ipa
```

XCode Side loading…

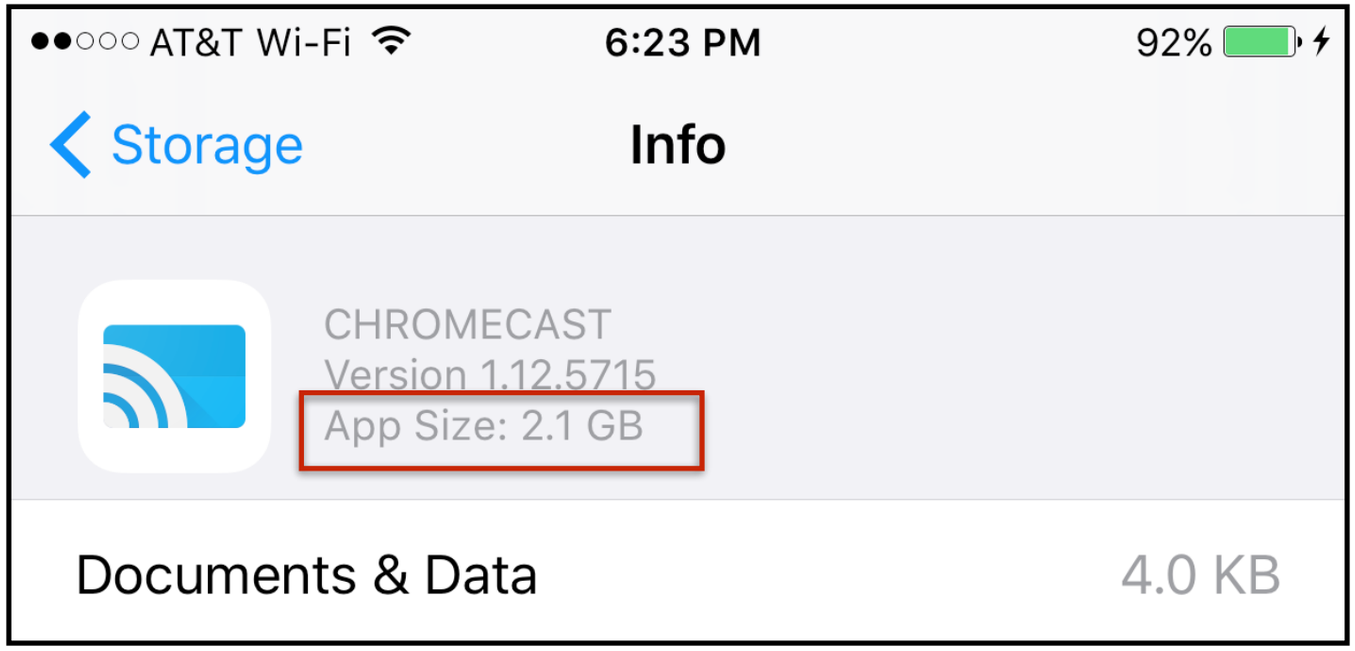


You can also use iTunes!

Before



After



Limitations

- Limited to 1G of appended data or app crashes
- Limited to ~15K of data in padded sections of a FAT file or app crashes
- <= iOS 8 Applications can be transferred back to computer via iTunes
- >= iOS 9 Applications can not be transferred off a non-jailbroken phone

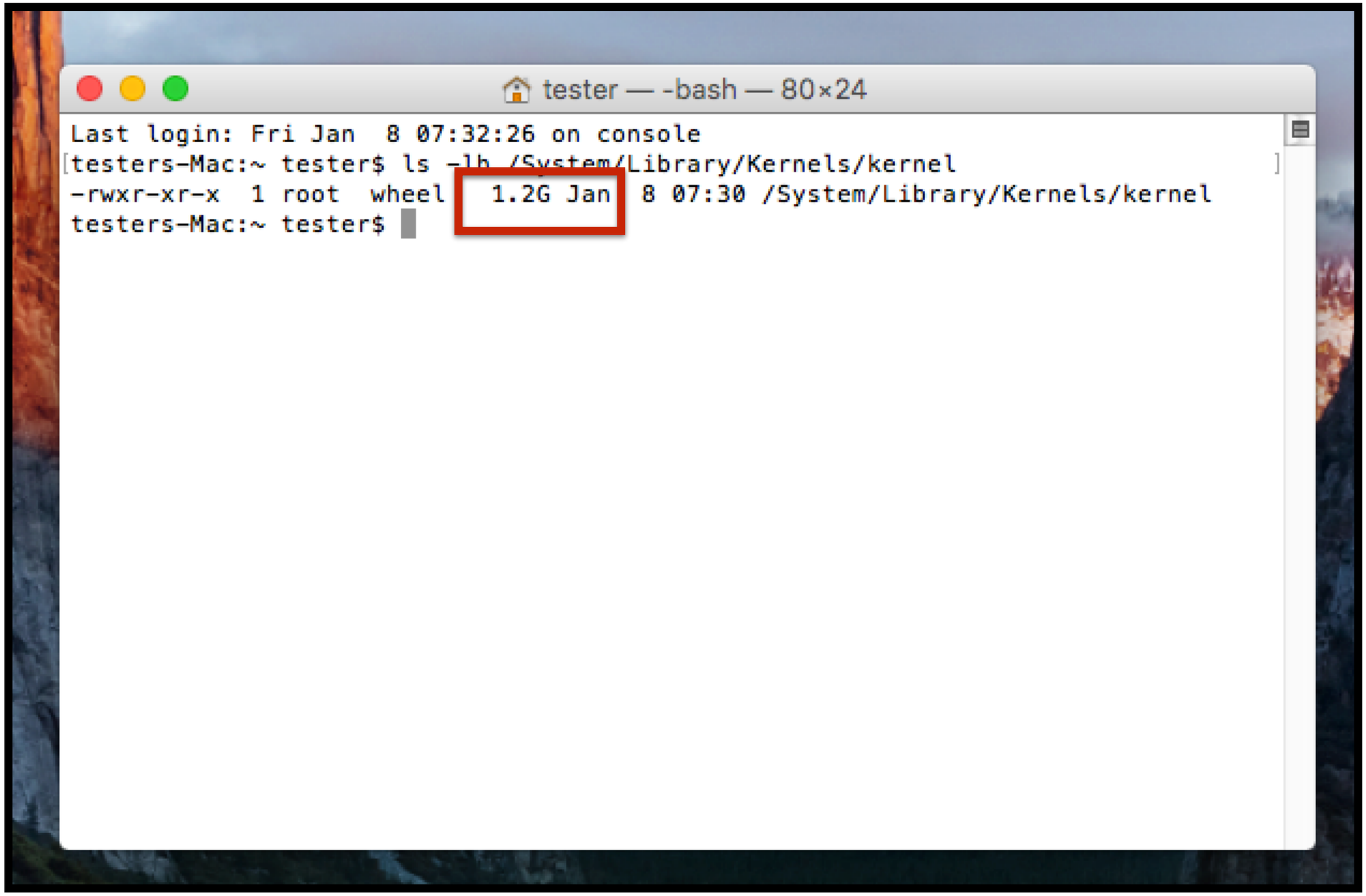
These techniques also work on unsigned binaries such as…

…the OS X Kernel

Append data to the Kernel

```
testers-Mac:~ tester$ cp /System/Library/Kernels/kernel .
testers-Mac:~ tester$ cat /dev/random >> kernel
^C
testers-Mac:~ tester$ ls -lh kernel
-rwxr-xr-x  1 tester  staff   1.2G Jan  8 07:27 kernel
testers-Mac:~ tester$ ls -l /System/Library/Kernels/kernel
-rwxr-xr-x  1 root  wheel  10760928 Dec 27 23:04 /System/Library/Kernels/kernel
testers-Mac:~ tester$ ls -lh /System/Library/Kernels/kernel
-rwxr-xr-x  1 root  wheel    10M Dec 27 23:04 /System/Library/Kernels/kernel
testers-Mac:~ tester$
```

Reboot


tester — -bash — 80×24
Last login: Fri Jan 8 07:32:26 on console
testers-Mac:~ tester$ ls -lh /System/Library/Kernels/kernel
-rwxr-xr-x 1 root wheel   1.2G Jan 8 07:30 /System/Library/Kernels/kernel
testers-Mac:~ tester$

Potential For Abuse

- Malware – See github.com/secretsquirrel/shmoo2016
  - hunchback.c
  - parse.c
- Data Hiding

Solutions

Introducing Kyphosis

- Script to find appended data on Mach-o files and between binaries in the Universal/Fat File
- Python 2.7
- Looks at what is NOT loaded in memory
- Works unsigned and signed binaries

Kyphosis Usage

```
➜ kyphosis  ./kyphosis.py
Usage: ./kyphosis.py macho_binary
*Returns nothing if there is nothing*
➜ kyphosis  ./kyphosis.py firefox
Found extra data in the Fat file slack space for firefox
Writing to firefox.extra_data_section0
➜ kyphosis  hexdump -C firefox.extra_data_section0
00000000  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
00001550  00 00 00 00 00 00 00 00  00 00 00 00 90 00 00 00  |................|
00001560  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  |................|
*
00001fd0
```

IDA Pro Example

```
➜ kyphosis  ./kyphosis.py /Applications/IDA\ Pro\ 6.8/uninstall.app/Contents/MacOS/o
sx-intel
Found extra data at the end of file.. /Applications/IDA Pro 6.8/uninstall.app/Content
s/MacOS/osx-intel
Writing to osx-intel.extra_data_end
```

Tclkit

```
00000000  4A 4C 1A 00  00 2D DC AB  80 21 14 AA  1C D9 CE 0A  JL...-...!......
00000010  00 00 6B 00  81 02 28 B2  21 31 CB 53  19 00 00 F8  ..k...(.!1.S....
00000020  09 D4 1F 28  05 0D 1C EB  01 AD 23 C8  06 54 03 BB  ...(......#..T..
00000030  10 40 05 81  01 6C F1 23  72 9D 1E 0C  C6 09 1D 0C  .@...l.#r.......
00000040  2E 17 AC 0A  23 20 70 0A  B0 11 B3 06  26 0E 79 26  ....# p.....&.y&
00000050  71 25 FF 10  9E 06 4A 03  9D 0B CE 05  77 00 47 0F  q%....J.....w.G.
00000060  D4 0A D6 08  3D 0C F1 0F  D6 08 89 07  32 1F BF 16  ....=.......2...
00000070  1E 06 00 00  C0 0A 1F 1B  9C 6B C0 28  38 AA E8 09  .........k.(8...
00000080  00 00 81 62  BF 30 23 B9  B2 13 00 00  81 01 4F FA  ...b.0#.......O.
00000090  33 00 EF 00  00 6E 01 80  4E 97 37 60  F4 00 00 60  3....n..N.7`...`
000000A0  00 6D 18 80  0B 6D CE 4E  7D 8A 7B 01  C6 00 C0 0D  .m...m.N}.{.....
000000B0  85 0C 96 08  5A 03 38 08  40 06 00 00  5D 07 EE 01  ....Z.8.@...]...
000000C0  88 01 0D 96  5B 77 A5 00  00 9B 07 1D  0D 89 02 00  ....[w..........
000000D0  00 44 06 F2  06 69 03 93  11 0C 1D 2C  18 CE 07 40  .D...i.....,...@
```

Patches Submitted and Accepted

- Synack knockknock
- Yelp OSXCollector
  - Strict check accepted (use -t)
  - Kyphosis pull request pending..
- Facebook OSQUERY

Knockknock Strict / No-Strict

- kSecCSDoNotValidateResources (**No-Strict**): 11.829 secs
- kSecCSStrictValidate (**Strict**): 9.808 secs
- kSecCSStrictValidate | kSecCSCheckAllArchitectures kSecCSCheckNestedCode (**all architectures and Strict and Deep**): 10.810 secs

OSXCollector Strict / No-Strict

- kSecCSDoNotValidateResources (**No-Strict**): 2:25.36 total

- kSecCSStrictValidate (**Strict**): 2:29.361 total

- kSecCSStrictValidate | kSecCSCheckAllArchitectures | kSecCSCheckNestedCode (**all architectures and Strict and Deep**): **2:22.04** total

Wrap Up

- You can stash data on Mach-O and in the Fat file formats
- For OS X you need to do strict checking
- For both, iOS and OS X you need to look at what is not loaded into memory
- PE and ELF files can be abused in the similar ways

Questions?

@midnite_runr
http://github.com/secretsquirrel

Credits

Title Art: http://1974design.com/wp-content/uploads/2011/02/deweyhasaposse.jpg