

Code signing flaw in macOS

```
> whoami
```

```
Thomas Reed  
Director of  
Mac & Mobile  
@ Malwarebytes
```

```
@thomasareed
```



Old-school malware

> Viruses!?

> "A virus operates by inserting or attaching itself to a legitimate program or document [...] in order to execute its code."¹



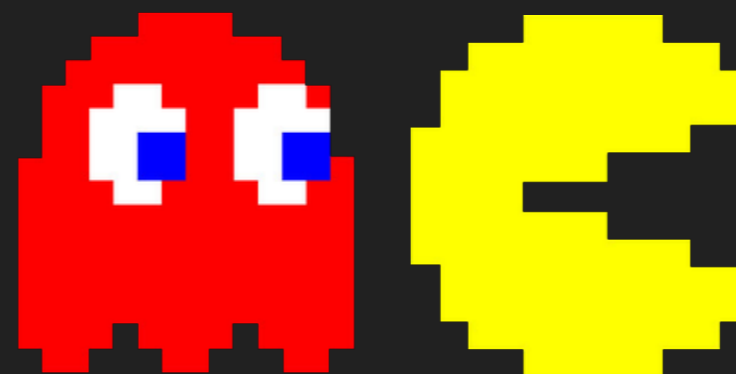
> None currently active on Macs

> Cases where malware was added to an existing app were done manually, not automatically

1 - <https://us.norton.com/internetsecurity-malware-what-is-a-computer-virus.html>

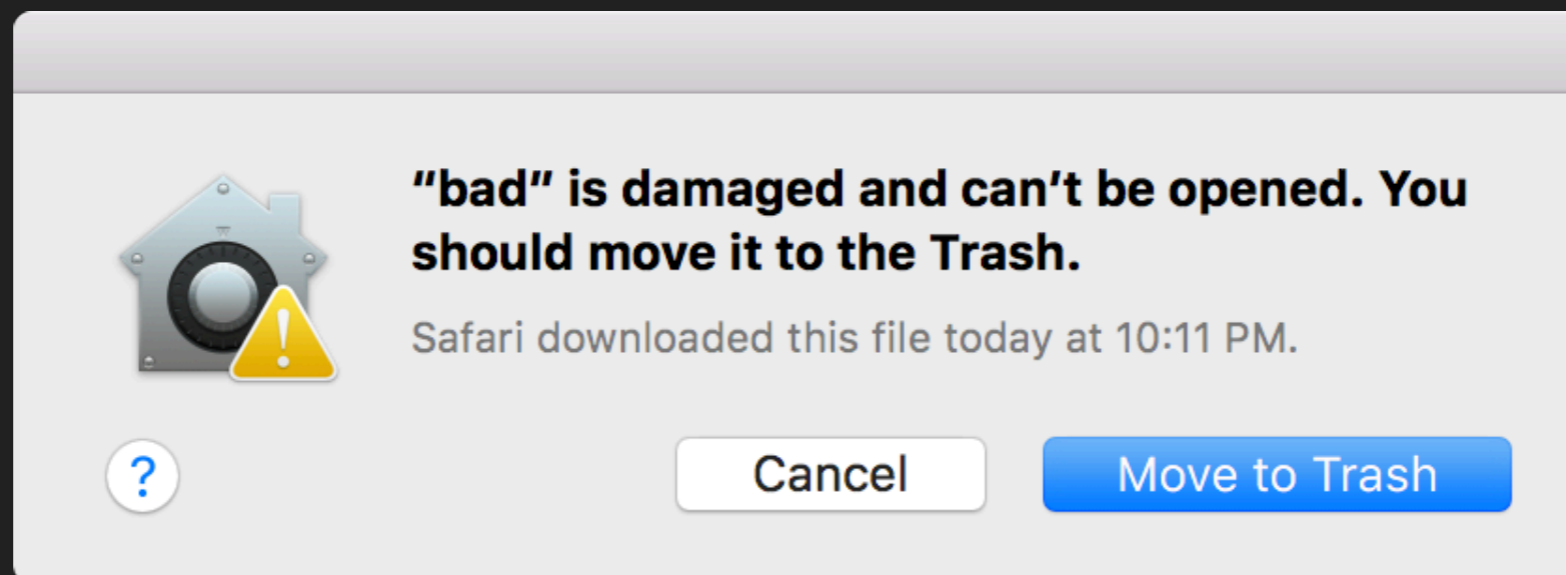
Mac viruses?

- > Are viruses impossible on modern macOS, due to code signing?
- > Unfortunately, no.
- > Why not? Let's look at how code signing works.

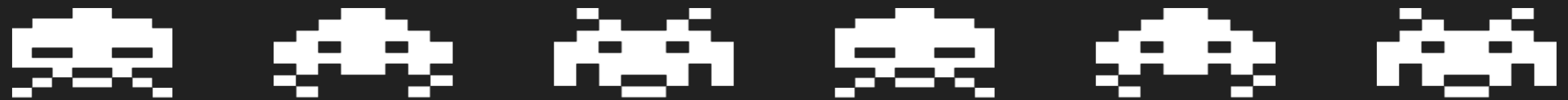


Code signing on Mac

- > Most apps code signed today
- > Unsigned apps not allowed by default
- > macOS verifies code signature before running downloaded apps



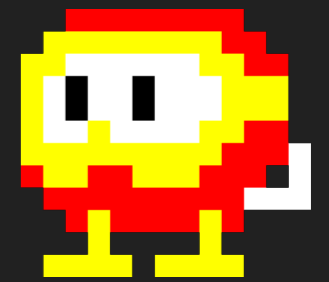
Code signing on Mac



> So, code signed apps are safe, right?

WRONG!

Code signing on Mac



- > Apps are "quarantined" when downloaded
- > Gatekeeper only checks code signature for quarantined apps
- > After opening, quarantine flag is removed
- > Code signature is never checked again!



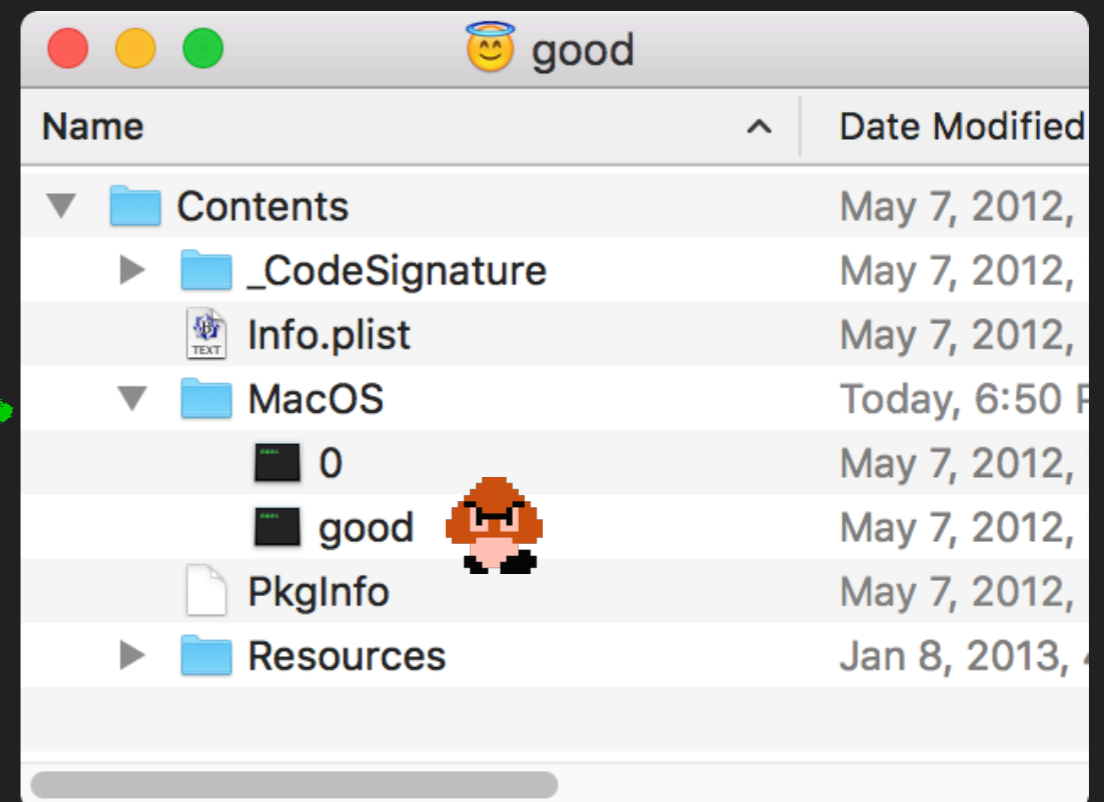
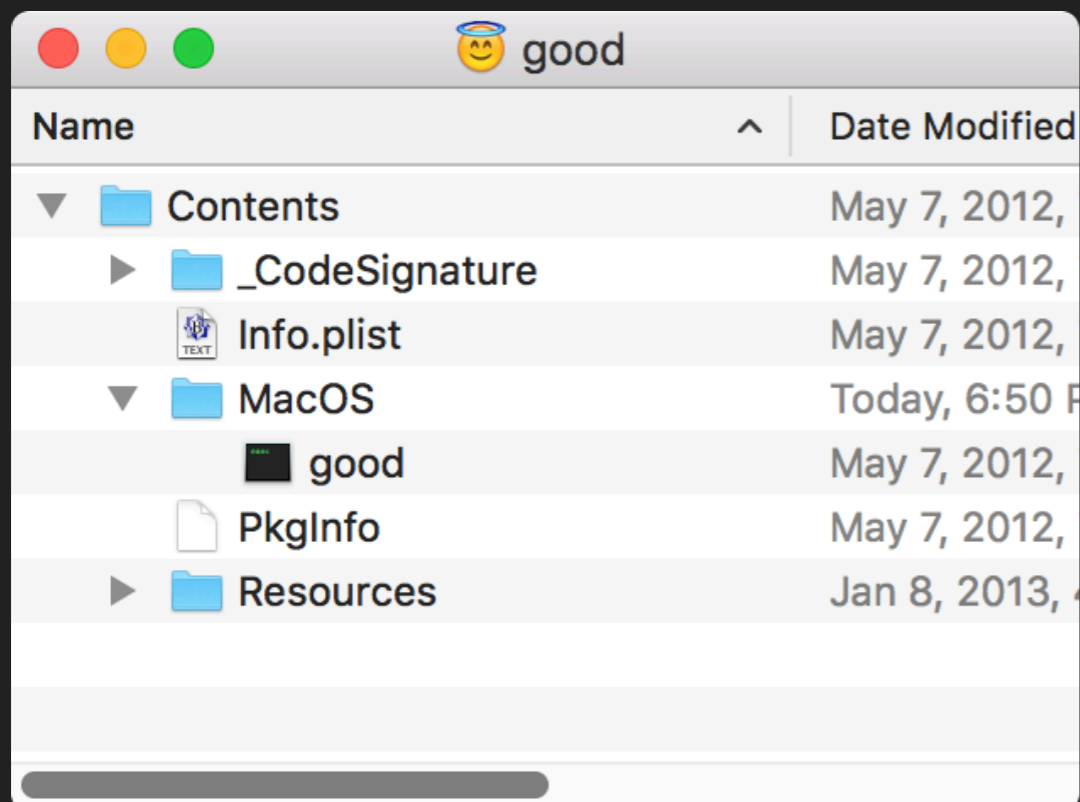
Time for mischief!

- > Malware almost always wants persistence
- > Malware almost always wants to be hidden
- > Achieve both by infecting apps that are no longer quarantined!
- > Malicious code will run every time an infected app is opened



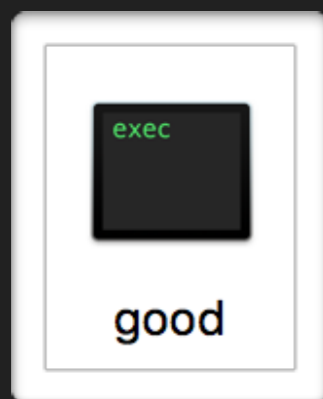
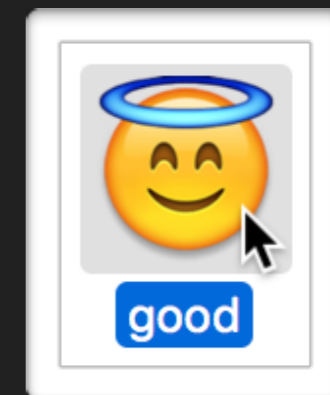
Infecting an app

- > Rename "good" to something else - like "0"
- > Add malicious executable named "good"
- > "good" executable loads "0" to make the app seem normal



Infecting an app

User double-clicks "good" app



System opens "good" executable file (which is actually malicious)

"good" executable opens original, renamed executable, to avoid suspicion



Infecting an app

The screenshot shows the macOS Activity Monitor window titled "Activity Monitor (All Processes)". The "CPU" tab is selected. A search bar at the top right contains the text "goo". The process list shows two entries, both named "good" with a "good" emoji icon. A green arrow points to the first "good" process. Below the arrow, the text "Malicious process" is written in green. At the bottom of the window, a summary box displays system statistics.

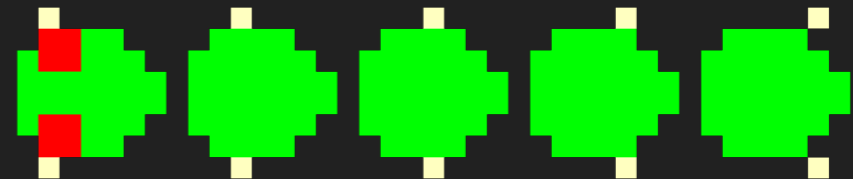
Process Name	% CPU	CPU Time	Threads	Idle Wake Ups	PID	User
good	0.3	0.11	4	13	50089	thomas
good	0.0	0.23	4	1	50091	thomas

System:	4.74%	CPU LOAD	Threads	1535
User:	2.00%		Processes:	401
Idle:	93.27%			



You have dysentery.

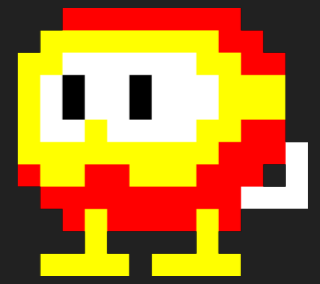
How hard is this?



- > Not very!
- > 22 lines of Swift code - malicious executable
- > 18 lines of AppleScript - dropper part 1
- > 16 lines of shell script - dropper part 2



Exceptions



- > Apple's apps can't be modified
- > If you try it, they crash
- > Malicious code still runs!



Chess quit unexpectedly.

Click Reopen to open the application again. Click Report to see more detailed information and send a report to Apple.



Ignore

Report...

Reopen

Exceptions



- > Some third-party apps have self-protection
- > If you change them, they'll let the user know
- > Malicious code still runs!



Something has modified Pacifist's application bundle. The application could be damaged, or could be infected by a virus. Please download an unaltered copy of Pacifist.

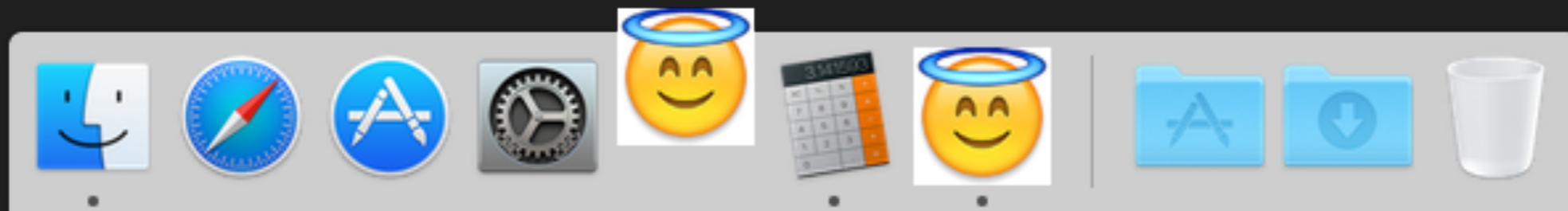
OK



Potential giveaways

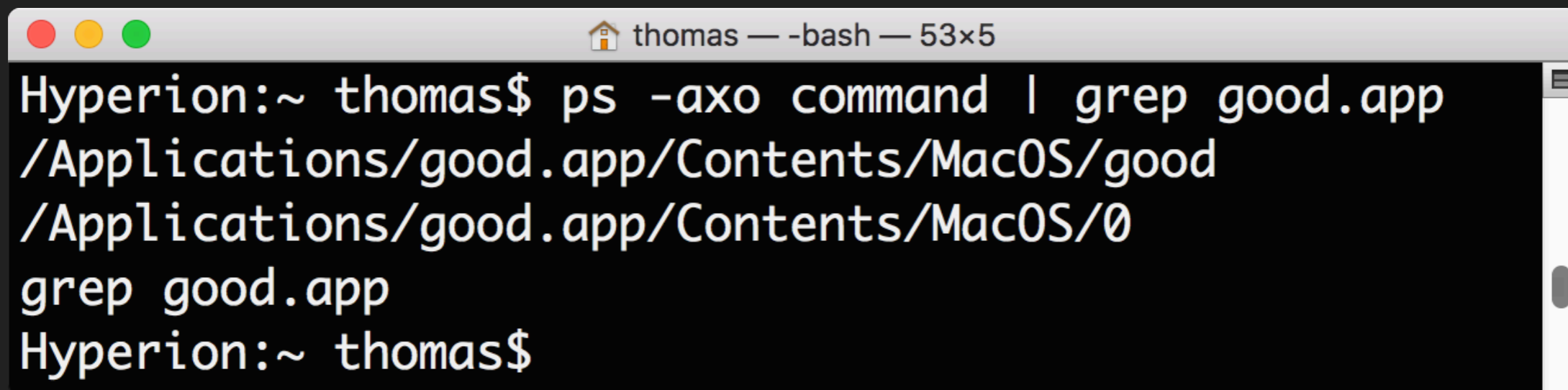
- > Doubled Dock icons
- > Malicious process shows as bouncing icon
- > Original process appears normally
- > Can be prevented

GG



Potential giveaways

- > Two processes in Activity Monitor
- > Two processes in ps output
- > Could make this less suspicious fairly easily



```
Hyperion:~ thomas$ ps -axo command | grep good.app
/Applications/good.app/Contents/MacOS/good
/Applications/good.app/Contents/MacOS/0
grep good.app
Hyperion:~ thomas$
```

Demo time...

How to detect

> Use `spctl` to verify signature

Good signature:



```
thomas — -bash — 80x5
Hyperion:~ thomas$ spctl --assess --verbose=4 /Applications/good.app
/Applications/good.app: accepted
source=Developer ID
Hyperion:~ thomas$
```

Bad signature:



```
thomas — -bash — 80x5
Hyperion:~ thomas$ spctl --assess --verbose=4 /Applications/good.app
/Applications/good.app: code has no resources but signature indicates they must
be present
Hyperion:~ thomas$
```

How to detect

> Use osquery to check signature



```
thomas — -bash — 80x10
Hyperion:~ thomas$ osqueryi --line 'select * from signature where path="/Applications/good.app"'
  path = /Applications/good.app
  signed = 1
  identifier = com.thesafemac.good
  cdhash = 7bdebb6406c5148f5b055971af11864da4633d40
  team_identifier = DKYMKWTFCU
  authority = Developer ID Application: Thomas Reed (DKYMKWTFCU)
Hyperion:~ thomas$
```

How to detect

> Use osquery to check signature



```
thomas — -bash — 80x10
Hyperion:~ thomas$ osqueryi --line 'select * from signature where path="/Applications/good.app"'
  path = /Applications/good.app
  signed = 0
  identifier = MalTest-55554944f1486d7a59f535fabef31b690dbbc92b
  cdhash = 93c51b7dc46ca82eba78e00b93685ccb914757b6
  team_identifier =
  authority =
Hyperion:~ thomas$
```

Problem...

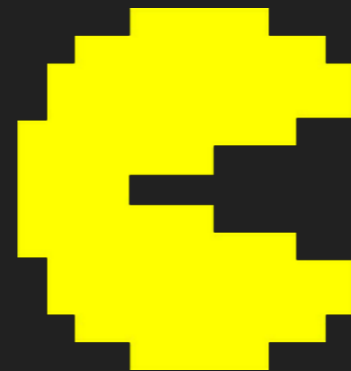
> What if the dropper re-signs the app with a different certificate?



```
thomas — -bash — 80x9
Hyperion:~ thomas$ osqueryi --line 'select * from signature where path="/Applications/good.app"'
  path = /Applications/good.app
  signed = 1
  identifier = com.goomba.good
  cdhash = 93c51b7dc46ca82eba78e00b93685ccb914757b6
  team_identifier = ABCDEFGHIJ
  authority = Developer ID Application: Joe Smith (ABCDEFGHIJ)
Hyperion:~ thomas$
```

Naughty or nice?

- > Possible solution: Santa
<https://github.com/google/santa>
- > Use in lockdown mode to allow only whitelisted apps to run
- > Modified apps will be blocked



Naughty or nice?

Santa

The following application has been blocked from executing because its trustworthiness cannot be determined.

Application	good
Filename	good
Path	/Applications/good.app/Contents/MacOS/good
Publisher	Not code-signed
Identifier	41bf94e3896dacc15fc00f09b2a3eafe fcc28bfe43c3e58f73480a8b5ddf2f65
Parent	launchd (1)
User	test

Prevent future notifications for this application for a day

Ignore



Naughty or nice?

> Pros:

> Difficult to bypass



> Cons:

> Whitelisting will keep you jumping with user requests!

> Unrealistic for certain users (eg, developers)



We're all set, then?

> Not so fast...

> Some legit apps have no or broken signatures

> Some apps even break their own signatures!





Not signed

```
$ codesign --display --verbose /Applications/  
OpenVPN/OpenVPN\ Connect.app
```

```
/Applications/OpenVPN/OpenVPN Connect.app: code  
object is not signed at all
```



Obsolete

```
$ codesign --verify --verbose /Applications/  
Google\ Chrome.app
```

```
/Applications/Google Chrome.app: resource  
envelope is obsolete (custom omit rules)
```



Broken

```
$ codesign --display --verbose /Applications/  
YubiKey\ PIV\ Manager.app
```

```
Executable=/Applications/YubiKey PIV Manager.app/  
Contents/MacOS/piuma  
Identifier=YubiKey PIV Manager  
...  
TeamIdentifier=LQA3CS5MM7
```

```
$ codesign --verify --verbose /Applications/  
YubiKey\ PIV\ Manager.app
```

```
/Applications/YubiKey PIV Manager.app: code object  
is not signed at all  
In subcomponent: /Applications/YubiKey PIV  
Manager.app/Contents/MacOS/piuman.pkg
```



Broken

```
$ codesign --verify --verbose /Applications/  
zoom.us.app
```

```
/Applications/zoom.us.app: a sealed resource is  
missing or invalid  
file added: /Applications/zoom.us.app/Contents/  
Frameworks/zMacResRetina.bundle  
file missing: /Applications/zoom.us.app/Contents/  
Plugins/ZoomAudioDevice.kext/Contents/Info.plist  
file missing: /Applications/zoom.us.app/Contents/  
Plugins/ZoomAudioDevice.kext/Contents/Resources/  
en.lproj/InfoPlist.strings  
file missing: /Applications/zoom.us.app/Contents/  
Plugins/ZoomAudioDevice.kext/Contents/MacOS/  
ZoomAudioDevice
```

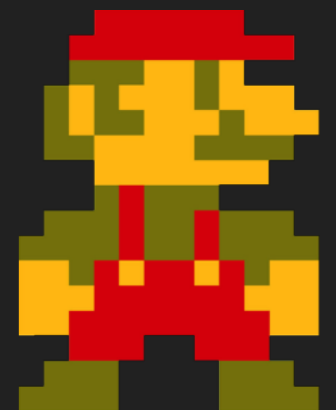
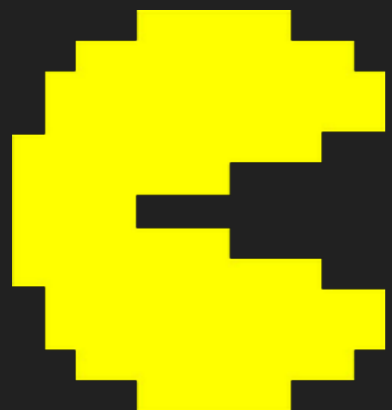
What to do?

- > Look at the signature for every app you use
- > If broken/missing/etc, report it to the vendor
- > If you are a developer, get it right!
- > Spread the word


Thanks!




<https://preview.tinyurl.com/ydgf32mk>




Bonus points

Space Invaders.....10 points

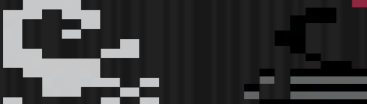
Blinky (PAC-MAN).....15 points

Centipede.....20 points

Pooka & Fygar (Dig-Dug).....50 points

Goomba (Super Mario).....75 points

Spider (Centipede).....100 points

snake & scorpion (Pitfall).....200 points

Coily (Q*bert).....500 points