iOS 10

Security and Privacy Changes

Alban Diquet - Engineering R&D Lead

DataTheorem

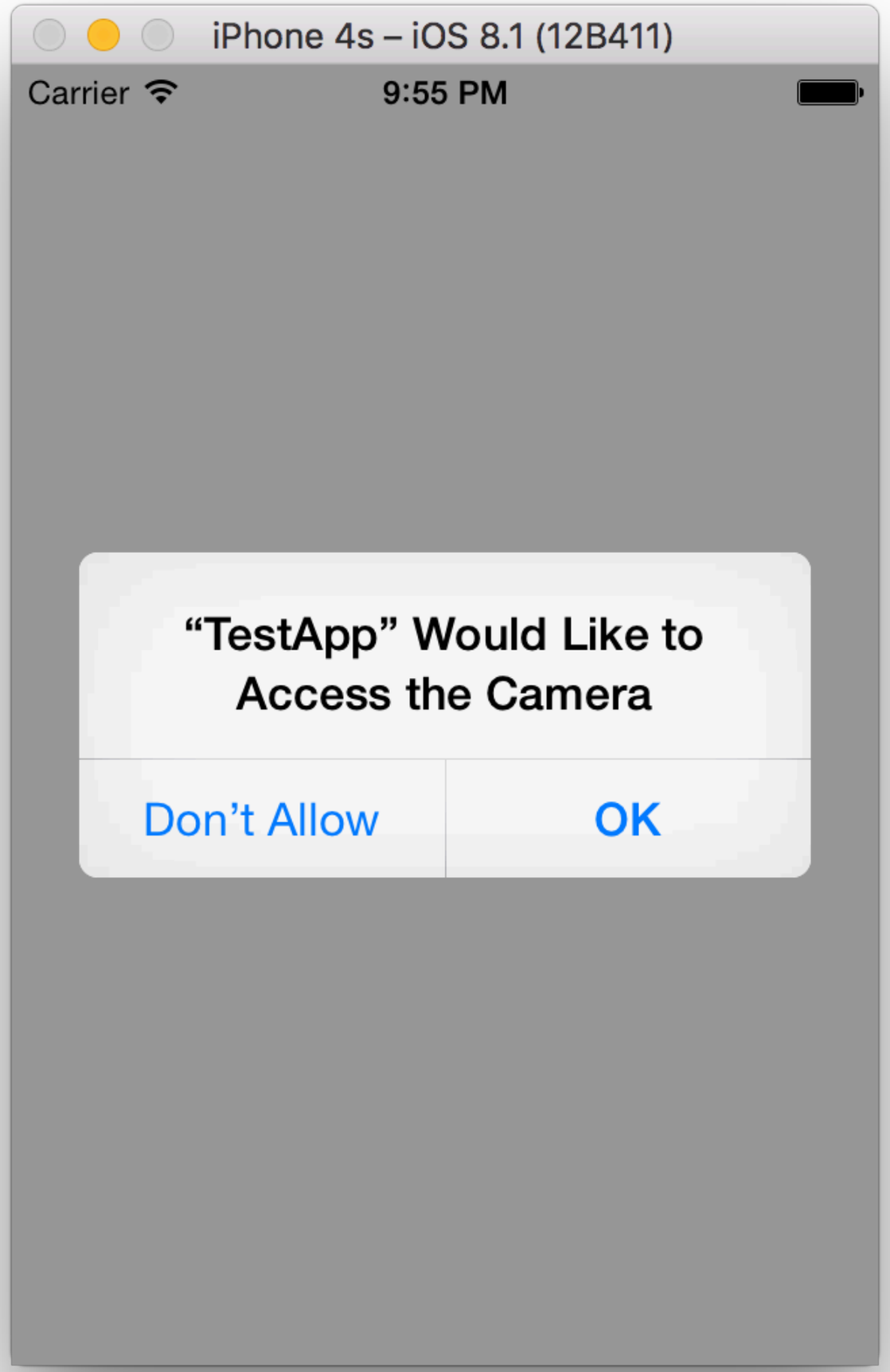
Agenda

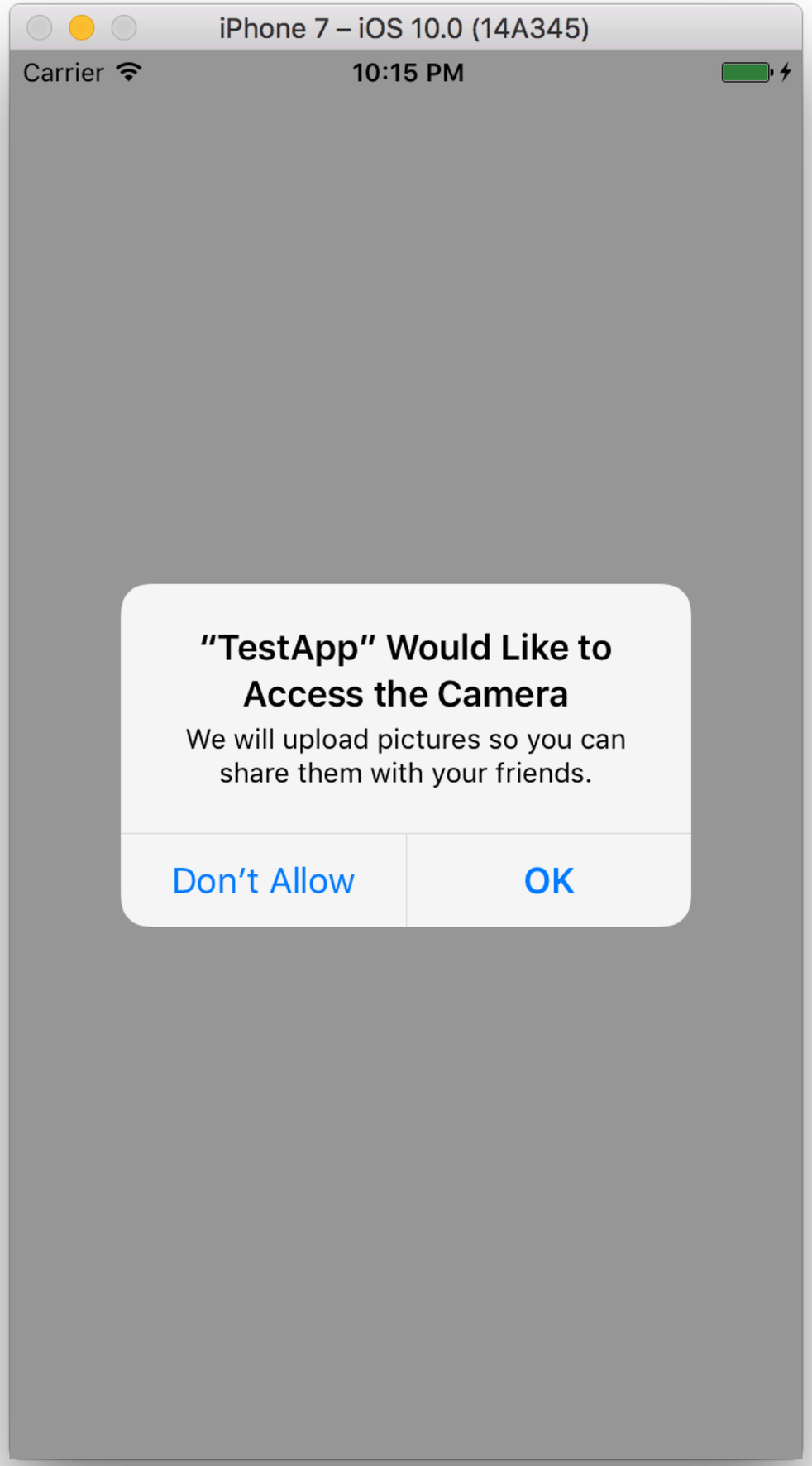
- Privacy Consent
- Pasteboards
- Ad Tracking
- Other Misc. Changes
- App Transport Security

Privacy Consent

Privacy Consent

- Apps can request access to private user data
  - Contacts, Location, Camera, etc…
- The App developer can set a "Privacy Purpose" for each type of user data
  - Quick sentence explaining why the App needs to access the data
  - Introduced in iOS 6


iPhone 4s – iOS 8.1 (12B411)
Carrier
9:55 PM
“TestApp” Would Like to Access the Camera
Don’t Allow
OK


iPhone 7 – iOS 10.0 (14A345)
Carrier
10:15 PM
"TestApp" Would Like to Access the Camera
We will upload pictures so you can share them with your friends.
Don't Allow
OK

```
2016-10-17 21:51:15.831748 TestApp[65786:3674115] subsystem: com.apple.UIKit,
category: HIDEventFiltered, enable_level: 0, persist_level: 0, default_ttl: 0,
info_ttl: 0, debug_ttl: 0, generate_symptoms: 0, enable_oversize: 1,
privacy_setting: 2, enable_private_data: 0
2016-10-17 21:51:15.845162 TestApp[65786:3674115] subsystem: com.apple.UIKit,
category: HIDEventIncoming, enable_level: 0, persist_level: 0, default_ttl: 0,
info_ttl: 0, debug_ttl: 0, generate_symptoms: 0, enable_oversize: 1,
privacy_setting: 2, enable_private_data: 0
2016-10-17 21:51:15.862752 TestApp[65786:3674112] subsystem: com.apple.BaseBoard,
category: MachPort, enable_level: 1, persist_level: 0, default_ttl: 0, info_ttl:
0, debug_ttl: 0, generate_symptoms: 0, enable_oversize: 0, privacy_setting: 0,
enable_private_data: 0
2016-10-17 21:51:15.892179 TestApp[65786:3674022] subsystem: com.apple.UIKit,
category: StatusBar, enable_level: 0, persist_level: 0, default_ttl: 0, info_ttl:
0, debug_ttl: 0, generate_symptoms: 0, enable_oversize: 1, privacy_setting: 2,
enable_private_data: 0
2016-10-17 21:51:15.968353 TestApp[65786:3674114] [access] This app has crashed
because it attempted to access privacy-sensitive data without a usage
description.  The app's Info.plist must contain an NSCameraUsageDescription key
with a string value explaining to the user how the app uses this data.
(lldb)
```

Privacy Consent

- Purpose Strings are mandatory with iOS 10
  - Apps will crash when built with the iOS 10 SDK and running on an iOS 10 device

iTunes Connect: Your app "P1 Alerts" (Apple ID: 1023997459) has one or more issues

Inbox x

iTunes Store <itunesconnect@apple.com> Sep 29

to [redacted]

Dear developer,

We have discovered one or more issues with your recent delivery for "P1 Alerts". To process your delivery, the following issues must be corrected:

This app attempts to access privacy-sensitive data without a usage description. The app's Info.plist must contain an NSPhotoLibraryUsageDescription key with a string value explaining to the user how the app uses this data.

Once these issues have been corrected, you can then redeliver the corrected binary.

Regards,

The App Store team

Privacy Consent

- Purpose Strings are mandatory with iOS 10
  - Apps will crash when built with the iOS 10 SDK and running on an iOS 10 device

Privacy Consent

- Purpose Strings are mandatory with iOS 10
  - Apps will crash when built with the iOS 10 SDK and running on an iOS 10 device
  - Apple will also reject App builds regardless of the Base SDK

Privacy Consent

- Why make the Purpose Strings mandatory?
  - It forces App developers to explain to the user why the App needs access their data
  - It prevents third-party SDKs within the App from requesting access to data the App developers never intended to access in the first place

Purpose String Required

iOS 10

Contacts

Calendar

Reminders

Photos

Bluetooth Sharing

Microphone

Camera

Location

Health—Sharing

Health—Updating

HomeKit

Media Library

Motion and Fitness

CallKit

Speech Recognition

SiriKit

TV Provider

Pasteboards

Pasteboards before iOS 10

- Introduced in iOS 3.0
    - Used by App developers as a hacky inter-App communication mechanism
        - Example: OpenUDID SDK for sharing a unique identifier https://github.com/ylechelle/OpenUDID
- Apple has been locking down Pasteboards on every iOS release
    - iOS 7: Named Pasteboards no longer accessible to Apps from different vendors
    - iOS 9: System and Named Pasteboards no longer accessible from Apps running in the background

Pasteboards on iOS 10

- Persistent Named Pasteboards deprecated in iOS 10
  - The OS will automatically configure Named Pasteboards as non-persistent
    - Allowed sharing data between Apps from the same vendor
    - Should use a shared Container instead
- Remaining use cases for (non-persistent) Named Pasteboards:
  - User requested to copy some data to the System Pasteboard
  - Share some data across different screens within the same App

```objc
// Create a persistent named pasteboard
UIPasteboard *myPasteboard = [UIPasteboard pasteboardWithName:@"MyCustomPasteboard"
                                                       create:YES];
myPasteboard.persistent = YES;
```

⚠ 'setPersistent:' is deprecated: first deprecated in iOS 10.0 - Do not set persistence on pasteboards. This property is set automatically.

Pasteboards on iOS 10

- New controls introduced in iOS 10
  - expirationDate
    - The time and date that you want the system to remove the pasteboard items from the pasteboard
    - Should always be set by your App
  - localOnly
    - The pasteboard items should not be available to other devices through the Handoff feature
      - New Handoff features in iOS 10 and macOS 10.12

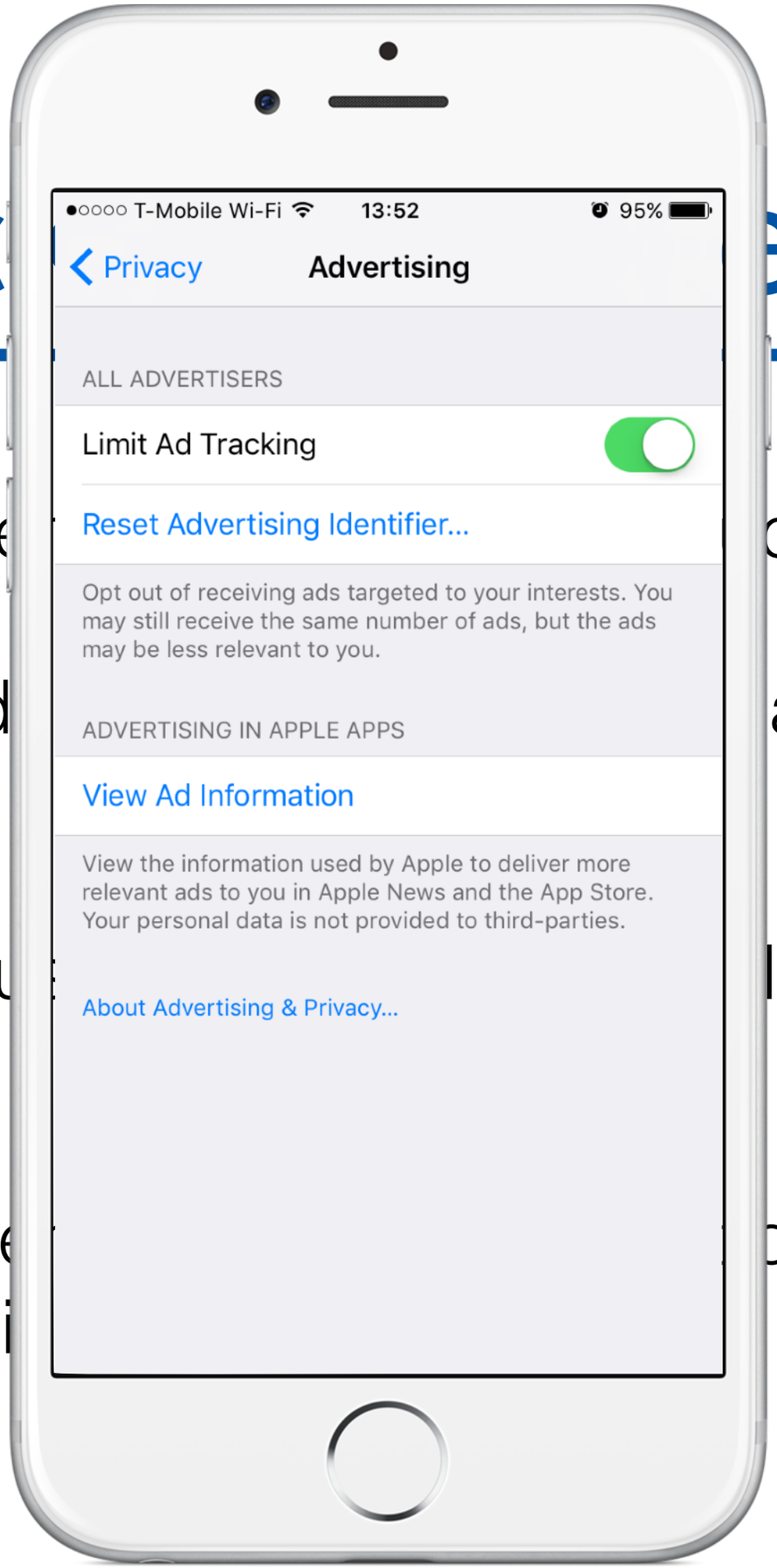
Ad Tracking

Ad Tracking before iOS 10

- Advertiser Identifier (“IDFA”) introduced in iOS 6
  - Uniquely identifies a device, but can be reset by the user
  - Less intrusive than the previously-used device UDID
- Users can enable “Limit Ad Controls” in their device’s privacy settings

Ad Trac e iOS 10

- Advertiser Ide ced in iOS 6
  - Uniquely id an be reset by the user
  - Less intru ly used device UDID
- Users can e ols” in their device’s pri



Ad Tracking before iOS 10

- Advertiser Identifier (“IDFA”) introduced in iOS 6
  - Uniquely identifies a device, but can be reset by the user
    - Less intrusive than the previously used device UDID
- Users can enable “Limit Ad Controls” in their device’s privacy settings

Ad Tracking before iOS 10

- Strict requirements described in the App Store Review guidelines

- App developers are required to:
  - Only use the IDFA for displaying Ads in their Apps
  - Check and honor the user's "Limit Ad Controls" setting
    - Policy-based: no technical means to enforce this
    - IDFA collected by most advertising and analytics SDKs

Limit Ad Tracking setting in iOS

☐ I, Anne Johnson, confirm that this app, and any third party that interfaces with this app, uses the Advertising Identifier checks and honors a user's Limit Ad Tracking setting in iOS and, when it is enabled by a user, this app does not use Advertising Identifier, and any information obtained through the use of the Advertising Identifier, in any way other than for "Limited Advertising Purposes" as defined in the iOS Developer Program License Agreement.

Ad Tracking on iOS 10

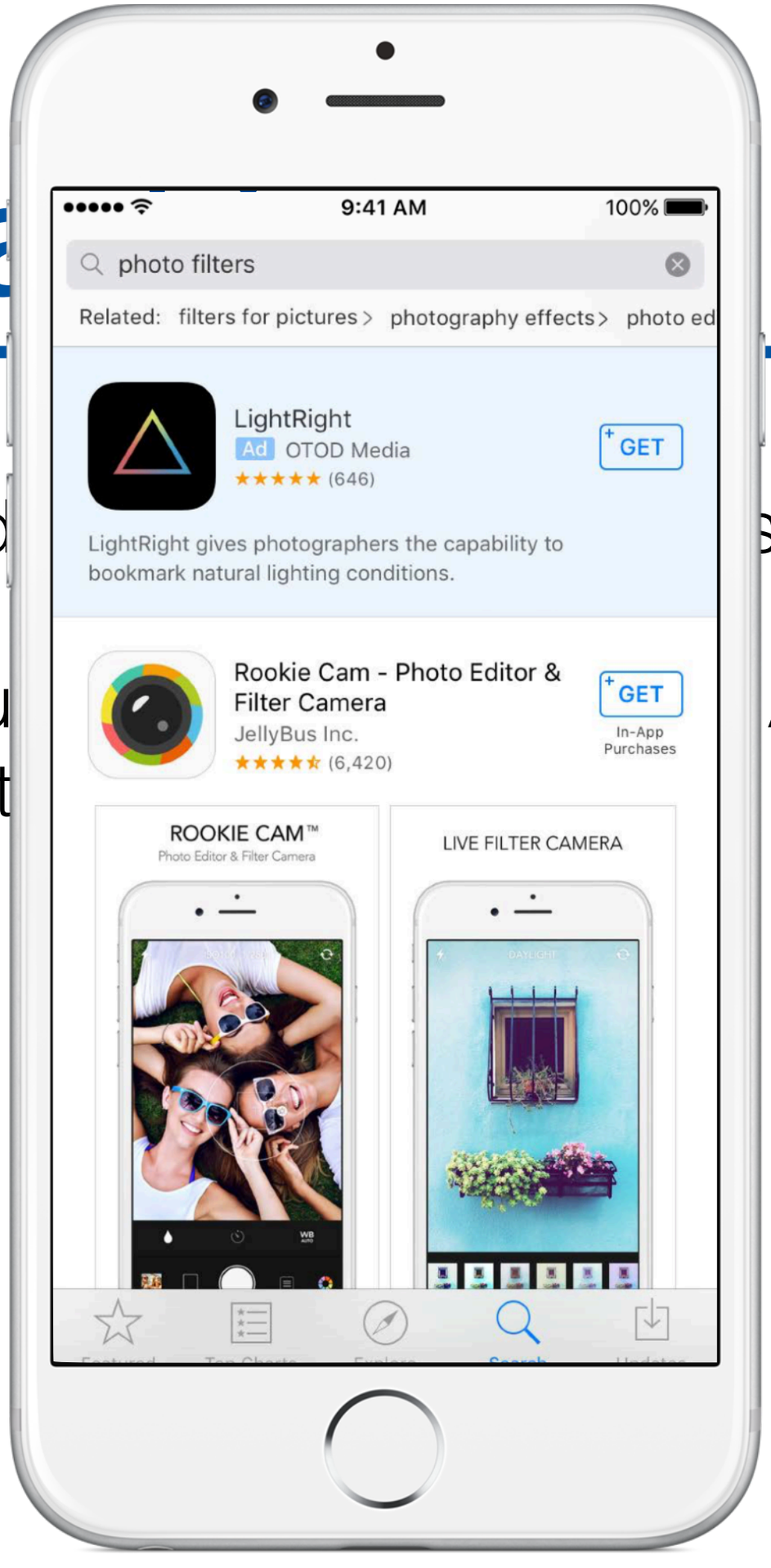
- On iOS 10, devices with “Limit Ad Controls” enabled will return an empty IDFA to Apps that query it:
  - 00000000-0000-0000-0000-000000000000
- The user’s privacy choice is now technically enforced
  - Apps and SDKs can no longer collect the IDFA regardless of the user’s preference
- Less-intrusive vendor identifier can still be used

Ad Tracking on iOS 10

- Additional Ad controls for the end-user
  - New “About this Ad” button in the Apple News and App Store Apps

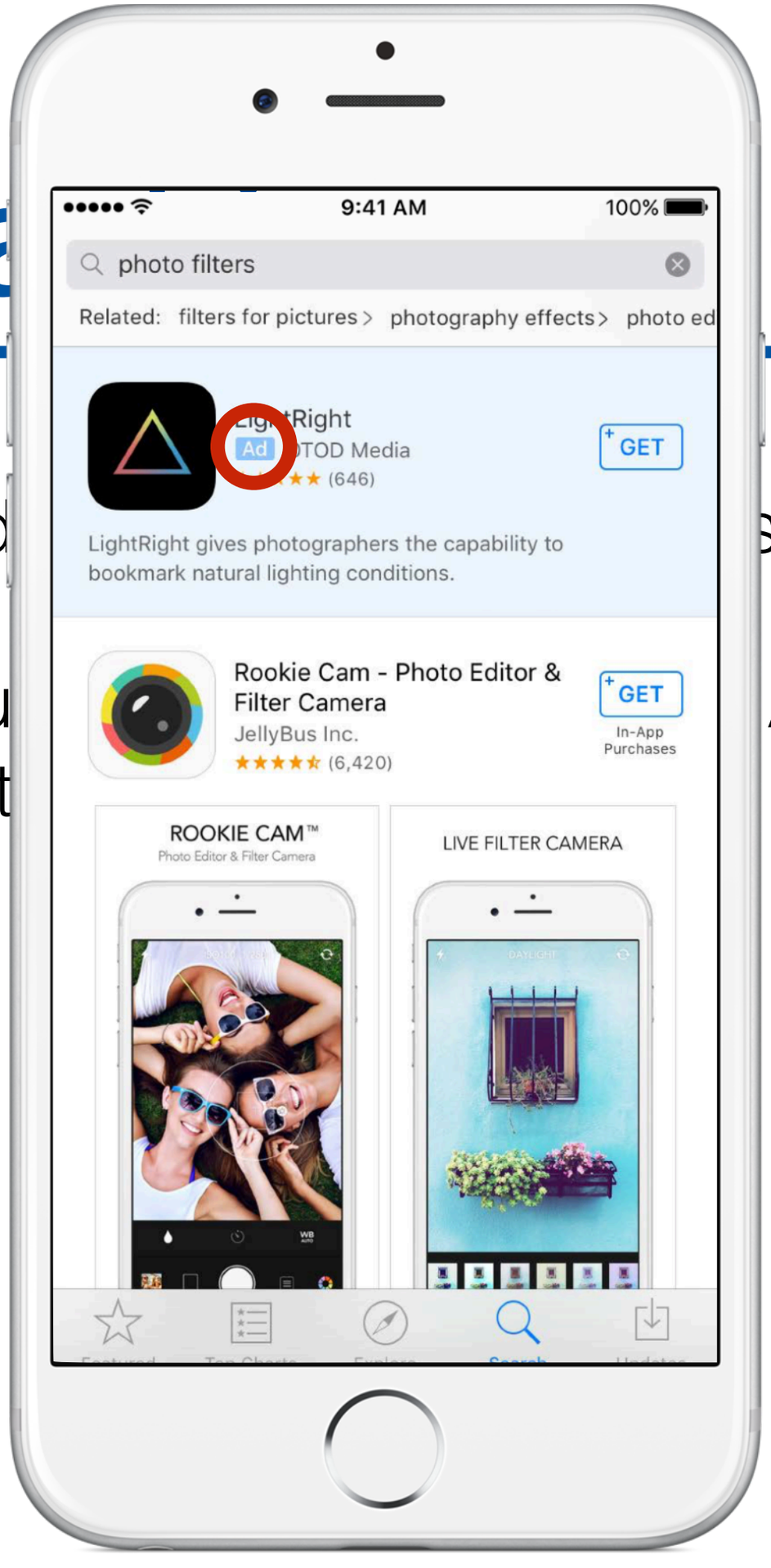
Ad Tra OS 10

- Additional Ad ser
  - New “Abou Apple News and App St



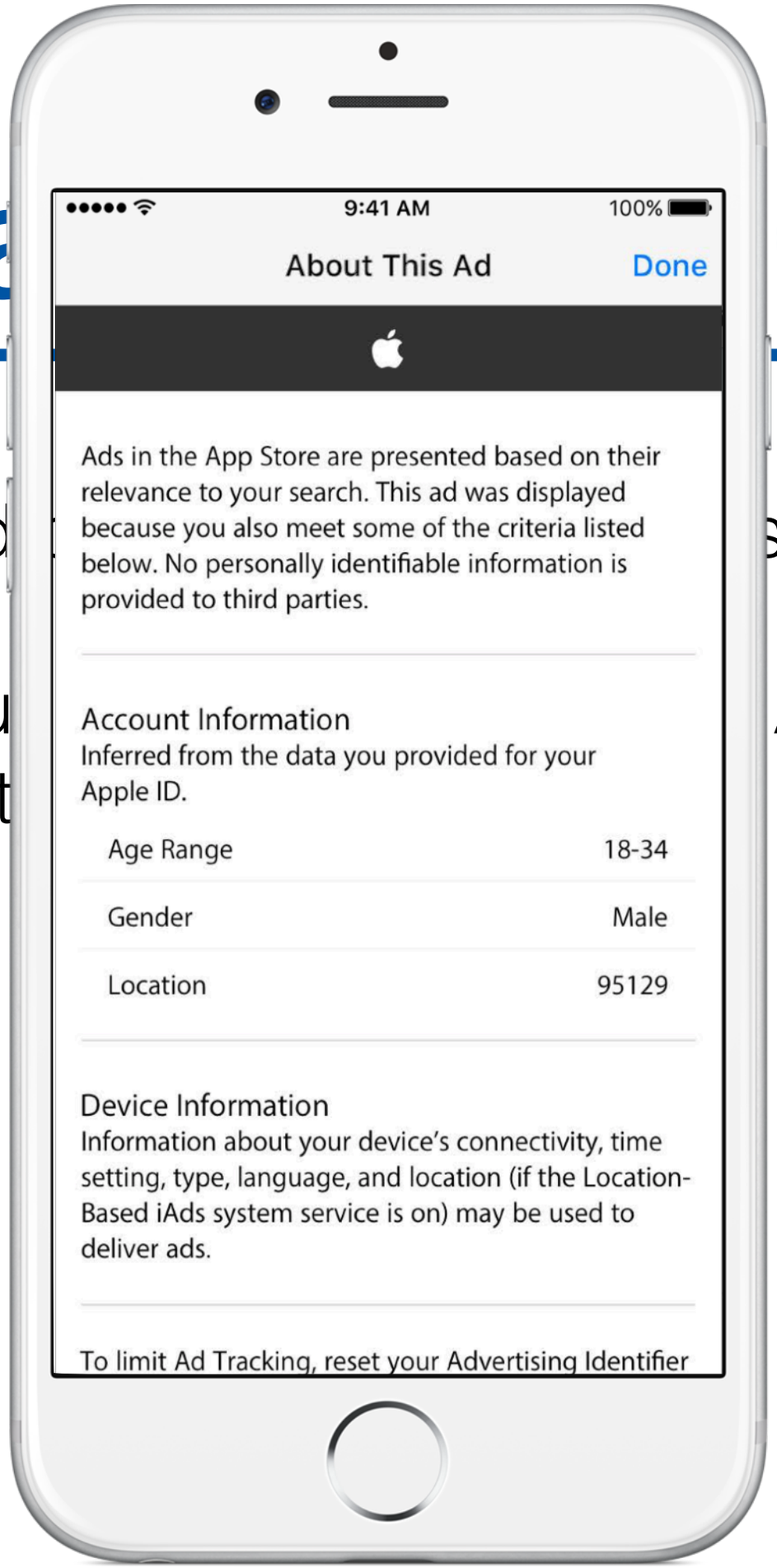
Ad Tra OS 10

- Additional Ad ser
  - New "Abou Apple News and App St



Ad Tra OS 10

- Additional Ad ser
  - New "Abou Apple News and App St



Ad Tracking on iOS 10

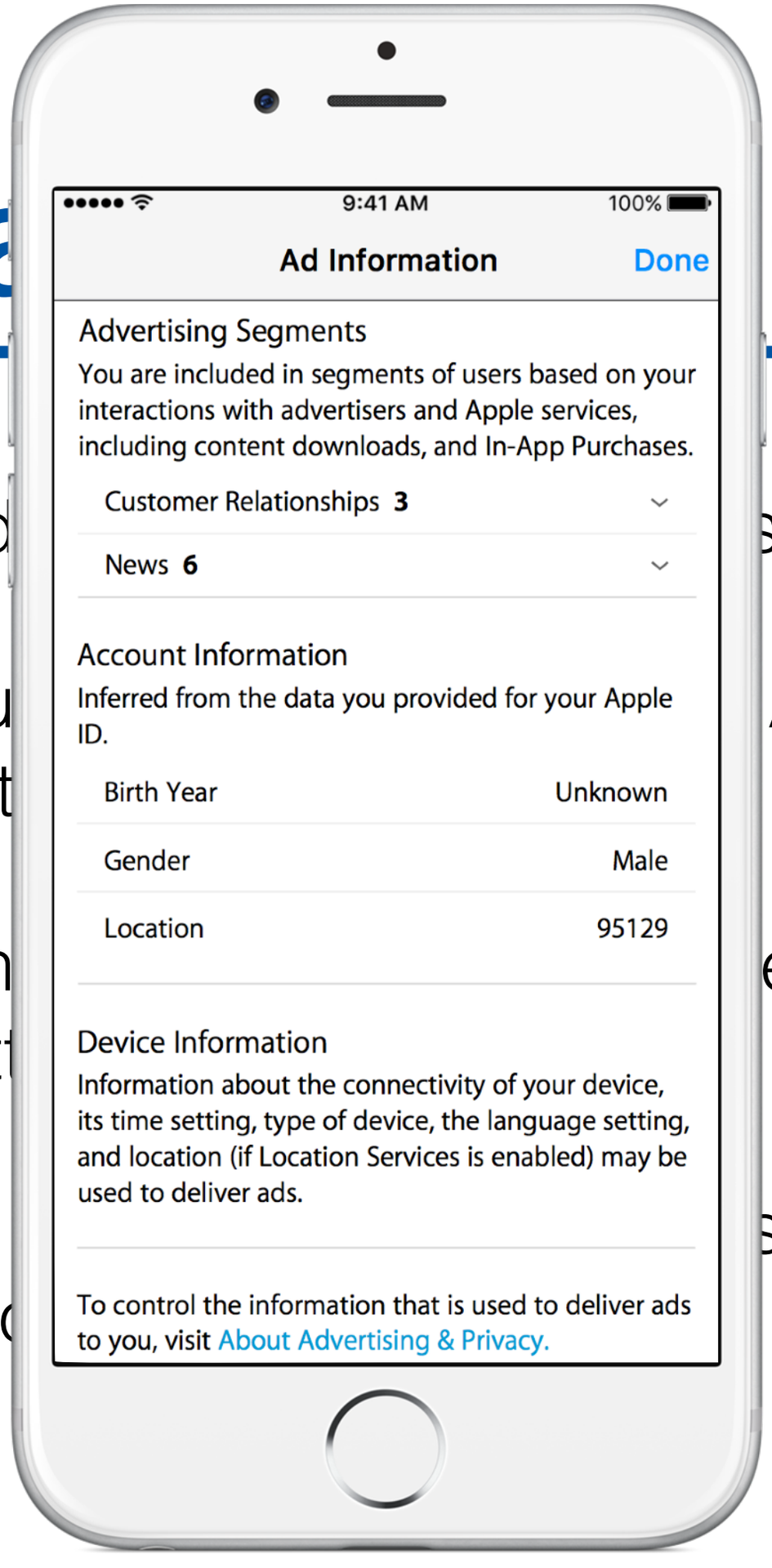
- Additional Ad controls for the end-user
  - New “About this Ad” button in the Apple News and App Store Apps

Ad Tracking on iOS 10

- Additional Ad controls for the end-user
  - New “About this Ad” button in the Apple News and App Store Apps
  - New “Ad Information” screen in the device’s Privacy settings
    - Provides details about the user’s Ad profiling information

Ad Tra OS 10

- Additional Ad ser
  - New “Abou Apple News and App St
- New “Ad In e device’s Privacy sett
  - Provides s Ad profiling informati



Other Misc. Changes

New Extension Points

- SiriKit: process requests originating from Siri or Maps
  - Protected by a user permission
- CallKit: display caller screen
- iMessage: customize messages and content
  - Users identity hidden behind an opaque UUID

Unified Logging APIs

- New logging functions to replace *NSLog()* and such: *os_log()*
  - Supersedes ASL (Apple System Logger) and the Syslog APIs
  - More functionality (logging levels, customization, etc.)
  - Same API on iOS, macOS, tvOS and watchOS
- Also a new security feature: objects and strings are automatically hidden from logs

Unified Logging APIs

```
 NSLog("Started Application.”);

 NSString *idfaString = [[[ASIdentifierManager sharedManager]
advertisingIdentifier] UUIDString];

 NSLog("Retrieved device IDFA %@.", idfaString);
```

```
TestApp[832:199415] Started Application.
TestApp[832:199415] Retrieved device IDFA 49295812-1432-1591-8456-940415481204.
```

Unified Logging APIs

```
 NSLog("Started Application.”);

 NSString *idfaString = [[[ASIdentifierManager sharedManager]
advertisingIdentifier] UUIDString];

 NSLog("Retrieved device IDFA %@.", idfaString);
```

```
TestApp[832:199415] Started Application.
TestApp[832:199415] Retrieved device IDFA 49295812-1432-1591-8456-940415481204.
```

Unified Logging APIs

```
 os_log(OS_LOG_DEFAULT, "Started Application.”);

 NSString *idfaString = [[[ASIdentifierManager sharedManager]
advertisingIdentifier] UUIDString];

 os_log(OS_LOG_DEFAULT, "Retrieved device IDFA %@.", idfaString);
```

```
TestApp[832:199415] Started Application.
TestApp[832:199415] Retrieved device IDFA <private>.
```

Unified Logging APIs

```
 os_log(OS_LOG_DEFAULT, "Started Application.”);

 NSString *idfaString = [[[ASIdentifierManager sharedManager]
advertisingIdentifier] UUIDString];

 os_log(OS_LOG_DEFAULT, "Retrieved device IDFA %{public}@.", idfaString);
```

```
TestApp[832:199415] Started Application.
TestApp[832:199415] Retrieved device IDFA 49295812-1432-1591-8456-940415481204.
```

Unified Logging APIs

```
Sep 25 22:15:23 IPh0n3 securityd[114] <Notice>: cert[0]: NonEmptySubject =(leaf)
[]> 0

Sep 25 22:15:23 IPh0n3 securityd[114] <Notice>: cert[0]: AnchorTrusted =(leaf)
[force]> 0

Sep 25 22:15:24 IPh0n3 locationd[462] <Notice>: message
'kCLConnectionMessageWatchdog' received from client '/System/Library/
PrivateFrameworks/BulletinBoard.framework'

Sep 25 22:15:25 IPh0n3 awdd[94] <Info>: life:#I 3 activities remaining:
<private>

Sep 25 22:15:25 IPh0n3 DuetHeuristic-BM(DuetActivitySchedulerDaemon)[123]
<Notice>: <private>

Sep 25 22:15:25 IPh0n3 UserEventAgent(DuetActivityScheduler)[25] <Notice>:
STARTING: <private>

Sep 25 22:15:25 IPh0n3 UserEventAgent(com.apple.cts)[25] <Info>: DAS told us to
run com.apple.mediaanalysisd.fullanalysis
```

Unified Logging APIs

```
Sep 25 22:15:23 IPh0n3 securityd[114] <Notice>: cert[0]: NonEmptySubject =(leaf)
[]> 0

Sep 25 22:15:23 IPh0n3 securityd[114] <Notice>: cert[0]: AnchorTrusted =(leaf)
[force]> 0

Sep 25 22:15:24 IPh0n3 locationd[462] <Notice>: message
'kCLConnectionMessageWatchdog' received from client '/System/Library/
PrivateFrameworks/BulletinBoard.framework'

Sep 25 22:15:25 IPh0n3 awdd[94] <Info>: life:#I 3 activities remaining:
<private>

Sep 25 22:15:25 IPh0n3 DuetHeuristic-BM(DuetActivitySchedulerDaemon)[123]
<Notice>: <private>

Sep 25 22:15:25 IPh0n3 UserEventAgent(DuetActivityScheduler)[25] <Notice>:
STARTING: <private>

Sep 25 22:15:25 IPh0n3 UserEventAgent(com.apple.cts)[25] <Info>: DAS told us to
run com.apple.mediaanalysisd.fullanalysis
```

Differential Privacy

- Technique to anonymize data collected from each user by adding noise as the data gets collected
    - When aggregating all the data from all users, stats on the global data set are still accurate
    - "Learn from crowd while protecting individual privacy"
- Apple is using differential privacy when collecting popular words, popular emojis, deep links and hints in Notes

Decrypted Kernel Images

- Kernel images no longer encrypted since iOS 10

Decrypted Kernel Images

Decrypted Kernel Images

Apple Opens Up iPhone Code in What Could Be Savvy Strategy or Security Screwup

A preview of Apple's next mobile operating system upgrade revealed some of the system's inner workings for the first time and suggests that the company wants help finding security flaws.

by Tom Simonite June 21, 2016

Decrypted Kernel Images

- Kernel images no longer encrypted since iOS 10

Decrypted Kernel Images

- Kernel images no longer encrypted since iOS 10
  - Obfuscation measure from the iOS 1.0 era
    - Prevented performance optimizations
    - Ultimately not very useful for security
- Will make it easier for security researchers to review the kernel and discover bugs
  - Good timing with Apple's new bug bounty program

App Transport Security

App Transport Security

- ATS introduced last year with iOS 9
    - On by default and blocks “unprotected" connections initiated by Apps
        - HTTP connections
        - HTTPS connections using weak security settings (TLS before 1.2, no perfect-forward secrecy, etc.)
- Developers can disable some or all the requirements in their Apps using ATS exemptions
    - Different exemptions can be applied to all, or specific domains

App Transport Security

```
2016-10-18 22:21:37.869850 TestApp[82472:4120665] subsystem:
com.apple.UIKit, category: StatusBar, enable_level: 0, persist_level: 0,
default_ttl: 0, info_ttl: 0, debug_ttl: 0, generate_symptoms: 0,
enable_oversize: 1, privacy_setting: 2, enable_private_data: 0
2016-10-18 22:21:37.901116 TestApp[82472:4120665] subsystem:
com.apple.libsqlite3, category: logging, enable_level: 0, persist_level: 0,
default_ttl: 0, info_ttl: 0, debug_ttl: 0, generate_symptoms: 0,
enable_oversize: 1, privacy_setting: 2, enable_private_data: 0
2016-10-18 22:21:37.920 TestApp[82472:4120753] App Transport Security has
blocked a cleartext HTTP (http://) resource load since it is insecure.
Temporary exceptions can be configured via your app's Info.plist file.
```

App Transport Security

- Big announcement at WWDC 2016
  - App Review will require “reasonable justification” for most ATS exceptions
  - Starting on January 1st 2017

App Transport Security

- Big announcement at WWDC 2016
  - App Review will require “reasonable justification” for most ATS exceptions
    - Starting on January 1st 2017
- Specifically, three ATS exemptions will be blacklisted:
  - NSAllowsArbitraryLoads - affects all domains
  - NSExceptionAllowsInsecureHTTPLoads - domain-specific
  - NSExceptionMinimumTLSVersion - domain-specific

App Transport Security

- Also new ATS exemptions introduced in iOS 10
- New behavior for NSAllowsArbitraryLoads
  - "In iOS 10 and later, and macOS 10.12 and later, the value of this key is ignored if any of the following keys are present in your app's Info.plist file:"
    - NSAllowsArbitraryLoadsInMedia
    - NSAllowsArbitraryLoadsInWebContent
    - NSAllowsLocalNetworking

App Transport Security

"The goal here is to flush out those folks who, when ATS was first released, simply turned it off globally and moved on.
That will no longer be allowed."

Apple, on the developer forums

App Transport Security

- How to be ready for January 1st 2017?
- Different approach for:
  - Servers you control
  - Third-party servers
  - WebViews displaying arbitrary websites: browser-like functionality

App Transport Security

- How to be ready for January 1st 2017
  - For servers you control:
    - Avoid using any of the blacklisted exemptions
    - All servers must support HTTPS with TLS 1.2

App Transport Security

- How to be ready for January 1st 2017
  - For third-party servers:
    - Any domain-specific ATS exemption can still be used
      - Apple confirmed that not having control of the server was a reasonable justification
  - You will still need to identify the list of third-party servers your App connects to so you can configure the ATS exemptions

App Transport Security

- How to be ready for January 1st 2017
  - For Webviews connecting to any website or server
  - Two possible solutions
    - New *NSAllowsArbitraryLoadsInWebContent* ATS exemption for iOS 10
      - Can be combined with *NSAllowsArbitraryLoads* which will be ignored on iOS 10 but will allow the WebViews to work on iOS 9
    - Replacing the *UIWebView* or *WKWebview* with the *SFSafariViewController*
      - ATS not enforced by design
      - Available since iOS 9

App Transport Security

- How to be ready for January 1st 2017 - **Recap**
  - Easiest to justify to the App Review team:
    - *NSAllowsArbitraryLoads* **disabled** and domain-specific exemptions for **third-party domains**
  - Harder to justify:
    - *NSAllowsArbitraryLoads* **enabled** and domain-specific “un-exemptions" for the **domains you control**
  - Will be blocked:
    - *NSAllowsArbitraryLoads* **enabled** and no additional ATS settings

App Transport Security

- Possible requirements for 2018 or later?
  - Certificate transparency and OCSP stapling
  - SHA-1 certificates rejected
  - 3-DES SSL cipher suites disabled
  - Additional ATS exemptions blacklisted:
    - Perfect forward secrecy exemption
    - Webview exemption

Conclusion

Conclusion

- Many positive security and privacy changes were introduced in iOS 10

- Apple is getting more agressive at making the App Store and its Apps more secure

- The App Review process now also enforces security requirements

  - Only ATS and Privacy Strings for now but most likely more to come

  - How to be pro-active: follow deprecation warnings and try to enable/leverage new security mechanisms and features

Conclusion

- Top priority
  - Add Privacy Purpose Strings for any private data your App is accessing
  - Be ready for the ATS requirements on January 2017

Conclusion

- Top priority
  - Add Privacy Purpose Strings for any private data your App is accessing
  - Be ready for the ATS requirements on January 2017
- Medium priority
  - Set an expiration date for any data your App puts in the Pasteboard
  - Switch to the *SFSafariViewController* if you have a browser-like WebView

Conclusion

- Top priorities as an App developer
  - Add Privacy Purpose Strings for any private data your App is accessing
  - Be ready for the ATS requirements on January 2017
- In the medium term
  - Set an expiration date for any data your App puts in the Pasteboard
  - Switch to the *SFSafariViewController* if you have a browser-like WebView
- In the long term
  - Deploy OCSP Stapling and enable Certificate Transparency on your App servers
  - Switch to the new logging APIs to prevent private data from being logged

Questions?

Alban Diquet - Engineering R&D Lead

DataTheorem

Thank You!

Data Theorem blog: https://datatheorem.github.io/

Alban on Twitter: https://twitter.com/nabla_c0d3

Alban Diquet - Engineering R&D Lead

DataTheorem