Introduction

iOS (formerly **iPhone OS**) is a mobile operating system created and developed by Apple Inc. exclusively for its hardware. It is the operating system that powers many of the company's mobile devices, including the iPhone and iPod Touch; the term also included the versions running on iPads until the name *iPadOS* was introduced with version 13 in 2019. It is the world's second-most widely installed mobile operating system, after Android. It is the basis for three other operating systems made by Apple: iPadOS, tvOS, and watchOS. It is proprietary software, although some parts of it are open source under the Apple Public Source License and other licenses.

Unveiled in 2007 for the first-generation iPhone, iOS has since been extended to support other Apple devices such as the iPod Touch (September 2007) and the iPad (introduced: January 2010; availability: April 2010.) As of March 2018, Apple's App Store contains more than 2.1 million iOS applications, 1 million of which are native for iPads. These mobile apps have collectively been downloaded more than 130 billion times.

Major versions of iOS are released annually. The current stable version, iOS 15, was released to the public on September 20, 2021.

History

In 2005, when Steve Jobs began planning the iPhone, he had a choice to either "shrink the Mac, which would be an epic feat of engineering, or enlarge the iPod". Jobs favored the former approach but pitted the Macintosh and iPod teams, led by Scott Forstall and Tony Fadell, respectively, against each other in an internal competition, with Forstall winning by creating the iPhone OS. The decision enabled the success of the iPhone as a platform for third-party developers: using a well-known desktop operating system as its basis allowed the many third-party Mac developers to write software for the iPhone with minimal retraining. Forstall was also responsible for creating a software development kit for programmers to build iPhone apps, as well as an App Store within iTunes.

The operating system was unveiled with the iPhone at the Macworld Conference & Expo on January 9, 2007, and released in June of that year. At the time of its unveiling in January, Steve Jobs claimed: "iPhone runs OS X" and runs "desktop class applications", but at the time of the iPhone's release, the operating system was renamed "iPhone OS". Initially, third-party native applications were not supported. Jobs' reasoning was that developers could build web applications through the Safari web browser that "would behave like native apps on the iPhone". In October 2007, Apple announced that a native Software Development Kit (SDK) was under development and that they planned to put it "in developers' hands in February". On March 6, 2008, Apple held a press event, announcing the iPhone SDK.

The iOS App Store was opened on July 10, 2008, with an initial 500 applications available. This quickly grew to 3,000 in September 2008, 15,000 in January 2009, 50,000 in June 2009, 100,000 in November 2009, 250,000 in August 2010, 650,000 in July 2012, 1 million in October 2013, 2 million in June 2016, and 2.2 million in January 2017. As of March 2016, 1 million apps are natively compatible with the iPad tablet computer. These apps have collectively been downloaded more than 130 billion times. App intelligence firm Sensor Tower estimated that the App Store would reach 5 million apps by 2020.

In September 2007, Apple announced the iPod Touch, a redesigned iPod based on the iPhone form factor. On January 27, 2010, Apple introduced their much-anticipated media tablet, the iPad, featuring a larger screen than the iPhone and iPod Touch, and designed for web browsing, media consumption, and reading, and offering multi-touch interaction with multimedia formats including newspapers, e-books, photos, videos, music, word processing documents, video games, and most existing iPhone apps using a 9.7-inch screen. It also includes a mobile version of Safari for web browsing, as well as access to the App Store, iTunes Library, iBookstore, Contacts, and Notes. Content is downloadable via Wi-Fi and optional 3G service or synced through the user's computer. AT&T was initially the sole U.S. provider of 3G wireless access for the iPad.

In June 2010, Apple rebranded iPhone OS as "iOS". The trademark "IOS" had been used by Cisco for over a decade for its operating system, IOS, used on its routers. To avoid any potential lawsuit, Apple licensed the "IOS" trademark from Cisco.

The Apple Watch smartwatch was announced by Tim Cook on September 9, 2014, being introduced as a product with health and fitness-tracking. It was released on April 24, 2015. It uses watchOS as operative system, which is based on IOS.

On November 22, 2016, a five-second video file originally named "IMG_0942.MP4" started crashing iOS on an increasing count of devices, forcing users to reboot. It gained massive popularity through social media channels and messaging services.

In October 2016, Apple opened its first iOS Developer Academy in Naples inside University of Naples Federico II's new campus. The course is completely free, aimed at acquiring specific technical skills on the creation and management of applications for the Apple ecosystem platforms. At the academy there are also issues of business administration (business planning and business management with a focus on digital opportunities) and there is a path dedicated to the design of graphical interfaces. Students have the opportunity to participate in the "Enterprise Track", an in-depth training experience on the entire life cycle of an app, from design to implementation, to security, troubleshooting, data storage and cloud usage. As of 2020, the academy graduated almost a thousand students from all over the world, who have worked on 400 app ideas and have already published about 50 apps on the iOS App Store. In the 2018–2019 academic year, students from more than 30 countries arrived. 35 of these have been selected to attend the Worldwide Developer Conference, the annual Apple Developer Conference held annually in California in early June.

On June 3, 2019, iPadOS, the branded version of iOS for iPad, was announced at the 2019 WWDC; it was launched on September 25, 2019.

Components

Most iOS apps are built using components from UIKit, a programming framework that defines common interface elements. This framework lets apps achieve a consistent appearance across the system, while at the same time offering a high level of customization. UIKit elements are flexible and familiar. They’re adaptable, enabling you to design a single app that looks great on any iOS device, and they automatically update when the system introduces appearance changes. The interface elements provided by UIKit fit into three main categories:

Bars. Tell people where they are in your app, provide navigation, and may contain buttons or other elements for initiating actions and communicating information.

Views. Contain the primary content people see in your app, such as text, graphics, animations, and interactive elements. Views can enable behaviors such as scrolling, insertion, deletion, and arrangement.

Controls. Initiate actions and convey information. Buttons, switches, text fields, and progress indicators are examples of controls.

In addition to defining the interface of iOS, UIKit defines functionality your app can adopt. Through this framework, for example, your app can respond to gestures on the touchscreen and enable features such as drawing, accessibility, and printing.

iOS tightly integrates with other programming frameworks and technologies too, such as Apple Pay, HealthKit, and ResearchKit, enabling you to design amazingly powerful apps.

Applications

iOS devices come with preinstalled apps developed by Apple including Mail, Maps, TV, Music, FaceTime, Wallet, Health, and many more.

Applications ("apps") are the most general form of application software that can be installed on iOS. They are downloaded from the official catalog of the App Store digital store, where apps are subjected to security checks before being made available to users. In June 2017, Apple updated its guidelines to specify that app developers will no longer have the ability to use custom prompts for encouraging users to leave reviews for their apps. IOS applications can also be installed directly from an IPA file provided by the software distributor, via unofficial ways. They are written using iOS Software Development Kit (SDK) and, often, combined with Xcode, using officially supported programming languages, including Swift and Objective-C. Other companies have also created tools that allow for the development of native iOS apps using their respective programming languages.

Applications for iOS are mostly built using components of UIKit, a programming framework. It allows applications to have a consistent look and feel with the OS, nevertheless offering customization.

Elements automatically update along with iOS updates, automatically including new interface rules. UIKit elements are very adaptable, this allows developers to design a single app that looks the same on any iOS device. In addition to defining the iOS interface, UIKit defines the functionality of the application.

At first, Apple did not intend to release an SDK to developers, because they did not want third-party apps to be developed for iOS, building web apps instead. However, this technology never entered into common use, this led Apple to change its opinion, so in October 2007 the SDK for developers was announced, finally released on March 6, 2008.

The SDK includes an inclusive set of development tools, including an audio mixer and an iPhone simulator. It is a free download for Mac users. It is not available for Microsoft Windows PCs. To test the application, get technical support, and distribute applications through App Store, developers are required to subscribe to the Apple Developer Program.

Over the years, the Apple Store apps surpassed multiple major milestones, including 50,000, 100,000, 250,000, 500,000, 1 million, and 2 million apps. The billionth application was installed on April 24, 2009.

Home screen

The home screen, rendered by SpringBoard, displays application icons and a dock at the bottom where users can pin their most frequently used apps. The home screen appears whenever the user unlocks the device or presses the physical "Home" button while in another app. Before iOS 4 on the iPhone 3GS (or later), the screen's background could be customized only through jailbreaking, but can now be changed out-of-the-box. The screen has a status bar across the top to display data, such as time, battery level, and signal strength. The rest of the screen is devoted to the current application. When a passcode is set and a user switches on the device, the passcode must be entered at the Lock Screen before access to the Home screen is granted.

In iPhone OS 3, Spotlight was introduced, allowing users to search media, apps, emails, contacts, messages, reminders, calendar events, and similar content. In iOS 7 and later, Spotlight is accessed by pulling down anywhere on the home screen (except for the top and bottom edges that open Notification Center and Control Center). In iOS 9, there are two ways to access Spotlight. As with iOS 7 and 8, pulling down on any homescreen will show Spotlight. However, it can also be accessed as it was in iOS 3 – 6. This endows Spotlight with Siri suggestions, which include app suggestions, contact suggestions and news. In iOS 10, Spotlight is at the top of the now-dedicated "Today" panel.

Since iOS 3.2, users are able to set a background image for the Home Screen. This feature is only available on third-generation devices—iPhone 3GS, third-generation iPod Touch (iOS 4.0 or newer), and all iPad models (since iOS 3.2)—or newer.

iOS 7 introduced a parallax effect on the Home Screen, which shifts the device's wallpaper and icons in response to the movement of the device, creating a 3D effect and an illusion of floating icons. This effect is also visible in the tab view of Mail and Safari.

Researchers found that users organize icons on their homescreens based on usage frequency and relatedness of the applications, as well as for reasons of usability and aesthetics.

System font
iOS originally used Helvetica as the system font. Apple switched to Helvetica Neue exclusively for the iPhone 4 and its Retina Display, and retained Helvetica as the system font for older iPhone devices on iOS 4. With iOS 7, Apple announced that they would change the system font to Helvetica Neue Light, a decision that sparked criticism for inappropriate usage of a light, thin typeface for low-resolution mobile screens. Apple eventually chose Helvetica Neue instead. The release of iOS 7 also introduced the ability to scale text or apply other forms of text accessibility changes through Settings. With iOS 9, Apple changed the font to San Francisco, an Apple-designed font aimed at maximum legibility and font consistency across its product lineup.

Folders
iOS 4 introduced folders, which can be created by dragging an application on top of another, and from then on, more items can be added to the folder using the same procedure. A title for the folder is automatically selected by the category of applications inside, but the name can also be edited by the user. When apps inside folders receive notification badges, the individual numbers of notifications are added up and the total number is displayed as a notification badge on the folder itself. Originally, folders on an iPhone could include up to 12 apps, while folders on iPad could include 20. With increasing display sizes on newer iPhone hardware, iOS 7 updated the folders with pages similar to the home screen layout, allowing for a significant expansion of folder functionality. Each page of a folder can contain up to nine apps, and there can be 15 pages in total, allowing for a total of 135 apps in a single folder. In iOS 9, Apple

updated folder sizes for iPad hardware, allowing for 16 apps per page, still at 15 pages maximum, increasing the total to 240 apps.

Notification Center

Main article: Notification Center

Before iOS 5, notifications were delivered in a modal window and couldn't be viewed after being dismissed. In iOS 5, Apple introduced Notification Center, which allows users to view a history of notifications. The user can tap a notification to open its corresponding app, or clear it. Notifications are now delivered in banners that appear briefly at the top of the screen. If a user taps a received notification, the application that sent the notification will be opened. Users can also choose to view notifications in modal alert windows by adjusting the application's notification settings. Introduced with iOS 8, widgets are now accessible through the Notification Center, defined by 3rd parties.

When an app sends a notification while closed, a red badge appears on its icon. This badge tells the user, at a glance, how many notifications that app has sent. Opening the app clears the badge.

Accessibility

iOS offers various accessibility features to help users with vision and hearing disabilities. One major feature, VoiceOver, provides a voice reading information on the screen, including contextual buttons, icons, links and other user interface elements, and allows the user to navigate the operating system through gestures. Any apps with default controls and developed with a UIKit framework gets VoiceOver functionality built in. One example includes holding up the iPhone to take a photo, with VoiceOver describing the photo scenery. As part of a "Made for iPhone" program, introduced with the release of iOS 7 in 2013, Apple has developed technology to use Bluetooth and a special technology protocol to let compatible third-party equipment connect with iPhones and iPads for streaming audio directly to a user's ears. Additional customization available for Made for iPhone products include battery tracking and adjustable sound settings for different environments. Apple made further efforts for accessibility for the release of iOS 10 in 2016, adding a new pronunciation editor to VoiceOver, adding a Magnifier setting to enlarge objects through the device's camera, software TTY support for deaf people to make phone calls from the iPhone, and giving tutorials and guidelines for third-party developers to incorporate proper accessibility functions into their apps.

In 2012, Liat Kornowski from *The Atlantic* wrote that "the iPhone has turned out to be one of the most revolutionary developments since the invention of Braille", and in 2016, Steven Aquino of *TechCrunch* described Apple as "leading the way in assistive technology", with Sarah Herrlinger, Senior Manager for Global Accessibility Policy and Initiatives at Apple, stating that "We see accessibility as a basic human right. Building into the core of our products supports a vision of an inclusive world where opportunity and access to information are barrier-free, empowering individuals with disabilities to achieve their goals".

Criticism has been aimed at iOS depending on both internet connection (either WiFi or through iTunes) and a working SIM card upon first activation. This restriction has been loosened in iOS 12, which no longer requires the latter.

Multitasking

Multitasking for iOS was first released in June 2010 along with the release of iOS 4. Only certain devices—iPhone 4, iPhone 3GS, and iPod Touch 3rd generation—were able to multitask. The iPad did not get multitasking until iOS 4.2.1 in that November.

The implementation of multitasking in iOS has been criticized for its approach, which limits the work that applications in the background can perform to a limited function set and requires application developers to add explicit support for it.

Before iOS 4, multitasking was limited to a selection of the applications Apple included on the device. Users could however "jailbreak" their device in order to unofficially multitask. Starting with iOS 4, on third-generation and newer iOS devices, multitasking is supported through seven background APIs:

1. Background audio – application continues to run in the background as long as it is playing audio or video content
2. Voice over IP – application is suspended when a phone call is not in progress
3. Background location – application is notified of location changes
4. Push notifications
5. Local notifications – application schedules local notifications to be delivered at a predetermined time
6. Task completion – application asks the system for extra time to complete a given task
7. Fast app switching – application does not execute any code and may be removed from memory at any time

In iOS 5, three new background APIs were introduced:

8. Newsstand – application can download content in the background to be ready for the user
9. External Accessory – application communicates with an external accessory and shares data at regular intervals
10. Bluetooth Accessory – application communicates with a bluetooth accessory and shares data at regular intervals

In iOS 7, Apple introduced a new multitasking feature, providing all apps with the ability to perform background updates. This feature prefers to update the user's most frequently used apps and prefers to use Wi-Fi networks over a cellular network, without markedly reducing the device's battery life.

Switching applications

In iOS 4.0 to iOS 6.x, double-clicking the home button activates the application switcher. A scrollable dock-style interface appears from the bottom, moving the contents of the screen up. Choosing an icon switch to an application. To the far left are icons which function as music controls, a rotation lock, and on iOS 4.2 and above, a volume controller.

With the introduction of iOS 7, double-clicking the home button also activates the application switcher. However, unlike previous versions it displays screenshots of open applications on top of the icon and horizontal scrolling allows for browsing through previous apps, and it is possible to close applications by dragging them up, similar to how WebOS handled multiple cards.

With the introduction of iOS 9, the application switcher received a significant visual change; while still retaining the card metaphor introduced in iOS 7, the application icon is smaller, and appears above the screenshot (which is now larger, due to the removal of "Recent and Favorite Contacts"), and each application "card" overlaps the other, forming a rolodex effect as the user scrolls. Now, instead of the home screen appearing at the leftmost of the application switcher, it appears rightmost. In iOS 11, the application switcher receives a major redesign. In the iPad, the Control Center and app switcher are combined. The app switcher in the iPad can also be accessed by swiping up from the bottom. In the iPhone, the app switcher cannot be accessed if there are no apps in the RAM.

Ending tasks

In iOS 4.0 to iOS 6.x, briefly holding the icons in the application switcher makes them "jiggle" (similarly to the homescreen) and allows the user to *force* quit the applications by tapping the red minus circle that appears at the corner of the app's icon. Clearing applications from multitasking stayed the same from iOS 4.0 through 6.1.6, the last version of iOS 6.

As of iOS 7, the process has become faster and easier. In iOS 7, instead of holding the icons to close them, they are closed by simply swiping them upwards off the screen. Up to three apps can be cleared at a time compared to one in versions up to iOS 6.1.6.

Task completion

Task completion allows apps to continue a certain task after the app has been suspended. As of iOS 4.0, apps can request up to ten minutes to complete a task in the background. This doesn't extend to background uploads and downloads though (e.g. if a user starts a download in one application, it won't finish if they switch away from the application).

Siri

Main article: Siri

Siri () is an intelligent personal assistant integrated into iOS. The assistant uses voice queries and a natural language user interface to answer questions, make recommendations, and perform actions by delegating requests to a set of Internet services. The software adapts to users' individual language usages, searches, and preferences, with continuing use. Returned results are individualized.

Originally released as an app for iOS in February 2010, it was acquired by Apple two months later, and then integrated into iPhone 4S at its release in October 2011. At that time, the separate app was also removed from the iOS App Store.

Siri supports a wide range of user commands, including performing phone actions, checking basic information, scheduling events and reminders, handling device settings, searching the Internet, navigating areas, finding information on entertainment, and is able to engage with iOS-integrated apps. With the release of iOS 10 in 2016, Apple opened up limited third-party access to Siri, including third-party messaging apps, as well as payments, ride-sharing, and Internet calling apps. With the release of iOS 11, Apple updated Siri's voices for more clear, human voices, it now supports follow-up questions and language translation, and additional third-party actions.

Game Center

Main article: Game Center

Game Center is an online multiplayer "social gaming network" released by Apple. It allows users to "invite friends to play a game, start a multiplayer game through matchmaking, track their achievements, and compare their high scores on a leaderboard." iOS 5 and above adds support for profile photos.

Game Center was announced during an iOS 4 preview event hosted by Apple on April 8, 2010. A preview was released to registered Apple developers in August. It was released on September 8, 2010, with iOS 4.1 on iPhone 4, iPhone 3GS, and iPod Touch 2nd generation through 4th generation. Game Center made its public debut on the iPad with iOS 4.2.1. There is no support for the iPhone 3G, original iPhone and the first-generation iPod Touch (the latter two devices did not have Game Center because they did not get iOS 4). However, Game Center is unofficially available on the iPhone 3G via a hack.

Development

Main article: iOS SDK

The iOS SDK (Software Development Kit) allows for the development of mobile apps on iOS.

While originally developing iPhone prior to its unveiling in 2007, Apple's then-CEO Steve Jobs did not intend to let third-party developers build native apps for iOS, instead directing them to make web applications for the Safari web browser. However, backlash from developers prompted the company to reconsider, with Jobs announcing in October 2007 that Apple would have a software development kit available for developers by February 2008. The SDK was released on March 6, 2008.

The SDK is a free download for users of Mac personal computers. It is not available for Microsoft Windows PCs. The SDK contains sets giving developers access to various functions and services of iOS devices, such as hardware and software attributes. It also contains an iPhone simulator to mimic the look and feel of the device on the computer while developing. New versions of the SDK accompany new versions of iOS. In order to test applications, get technical support, and distribute apps through App Store, developers are required to subscribe to the Apple Developer Program.

Combined with Xcode, the iOS SDK helps developers write iOS apps using officially supported programming languages, including Swift and Objective-C. Other companies have also created tools that allow for the development of native iOS apps using their respective programming languages.

Update schedule

Main article: iOS version history

Apple provides major updates to the iOS operating system annually via iTunes and since iOS 5, also over-the-air. The device checks an XML-based PLIST file on mesu.apple.com for updates. The updates are delivered in plain unencrypted ZIP files. On all recent iOS devices, iOS regularly checks on the availability of an update, and if one is available, will prompt the user to permit its automatic installation.

The latest stable version is iOS 15, released on September 20, 2021. It is available for iPhone 6S and later, and the seventh-generation iPod Touch. In addition to the release of iOS 15, iPadOS 15 was released. Apple debuted iOS 15 and iPadOS 15 at its annual WWDC keynote on June 22, 2020. iPadOS 15 is available on the same devices as iOS 14. Devices supported are iPad Air 2 and later, iPad fifth-generation and later, iPad mini 4 and later and all versions of the iPad Pro. The update introduced new features such as redesigned notifications, a more informative Weather app, Focus Mode, SharePlay, Live Text, and more.

Originally, iPod Touch users had to pay for system software updates. This was due to accounting rules that designated it not a "subscription device" like iPhone or Apple TV, and improvements to the device required payments. The requirement to pay to upgrade caused iPod Touch owners to stay away from updates. However, in September 2009, a change in accounting rules won tentative approval, affecting Apple's earnings and stock price, and allowing iPod Touch updates to be delivered for free.

Apple has significantly extended the cycle of updates for iOS supported devices over the years. The iPhone (1st generation) and iPhone 3G only received two iOS updates, while later models had support for five, six, and seven years.

XNU kernel

The iOS kernel is the XNU kernel of Darwin. The original iPhone OS (1.0) up to iPhone OS 3.1.3 used Darwin 9.0.0d1. iOS 4 was based on Darwin 10. iOS 5 was based on Darwin 11. iOS 6 was based on Darwin 13. iOS 7 and iOS 8 are based on Darwin 14. iOS 9 is based on Darwin 15. iOS 10 is based on Darwin 16. iOS 11 is based on Darwin 17. iOS 12 is based on Darwin 18. iOS 13 is based on Darwin 19.

In iOS 6 the kernel is subject to ASLR, similar to that of OS X Mountain Lion. This makes exploit possibilities more complex since it is not possible to know the location of kernel code.

Since XNU is based on the BSD kernel, it is open source. The source is under a 3-clause BSD license for the original BSD parts, with parts added by Apple under the Apple Public Source License. The versions contained in iOS are not available; only the versions used in macOS are available.

iOS does not have kernel extensions (kexts) in the file system, even if they are actually present. The kernel cache can be decompressed to show the correct kernel, along with the kexts (all packed in the __PRELINK_TEXT section) and their plists (in the __PRELINK_INFO section).

The kernel cache can also be directly decompressed (if decrypted) using third-party tools. With the advent of iOS 10 betas and default plain text kernelcaches, these tools can only be used after unpacking and applying lzssdec to unpack the kernel cache to its full size.

The kextstat provided by the Cydia alternative software does not work on iOS because the kextstat is based on `kmod_get_info(...)`, which is a deprecated API in iOS 4 and Mac OS X Snow Leopard. There are other alternative software that can also dump raw XML data.

On developing devices, the kernel is always stored as a statically linked cache stored in /System/Library/Caches/com.apple.kernelcaches/kernelcache which is unpacked and executed at boot.

In the beginning, iOS had a kernel version usually higher than the corresponding version of macOS. Over time, the kernels of iOS and macOS have gotten closer. This is not surprising, considering that iOS introduced new features (such as the ASLR Kernel, the default freezer, and various security-strengthening features) that were first incorporated and subsequently arrived on macOS. It appears Apple is gradually merging the iOS and macOS kernels over time. The build date for each version varies slightly between processors. This is due to the fact that the builds are sequential.

The latest version of the Darwin Kernel updated to iOS 13.6 is 19.6.0, dated July 27, 2020, while for iOS 14 beta 4 it is 20.0.0, dated July 27, 2020.

Kernel Image

The kernel image base is randomized by the boot loader (iBoot). This is done by creating random data, doing a SHA-1 hash of it and then using a byte from the SHA-1 hash for the kernel slide. The slide is calculated with this formula:

```
base=0x01000000+(slide_byte*0x00200000)
```

If the slide is 0, the static offset of 0x21000000 is used instead.

The adjusted base is passed to the kernel in the boot arguments structure at offset `0x04`, which is equivalent to gBootArgs->virtBase.

Kernel Map

The kernel map is used for kernel allocations of all types (`kalloc()`, `kernel_memory_allocate()`, etc.) and spans all of kernel space (`0x80000000-0xFFFEFFFF`). The kernel based maps are submaps of the `kernel_map`, for example `zone_map`, `ipc_kernel_map`, etc.

The strategy is to randomize the base of the `kernel_map`. A random 9-bit value is generated right after `kmem_init()` which establishes `kernel_map`, is multiplied by the page size. The resulting value is used as the size for the initial `kernel_map` allocation. Future `kernel_map` (and submap) allocations are pushed forward by a random amount. The allocation is silently removed after the first garbage collection and reused. This behaviour can be overridden with the "`kmapoff`" boot parameter.

Attacks

`Kext_request()` allows applications to request information about kernel modules, divided into active and passive operations. Active operations (load, unload, start, stop, etc.) require root access. iOS removes the ability to load kernel extensions. Passive operations were originally (before iOS 6) unrestricted and allowed unprivileged users to query kernel module base addresses. iOS6 inadvertently removed some limitations; only the load address requests are disallowed. So attackers can use `kKextRequestPredicateGetLoaded` to get load addresses and mach-o header dumps. The load address and mach-o segment headers are obscured to hide the ASLR slide, but mach-o section headers are not. This reveals the virtual addresses of loaded kernel sections.

This information leak has been closed with iOS 6.0.1.

Versions codenames

Main article: List of Apple codenames § iOS

Internally, iOS identifies each version by a **codename**, often used internally only, normally to maintain secrecy of the project. For example, the codename for iOS 14 is *Azul.*

Jailbreaking

Main article: iOS jailbreaking

Since its initial release, iOS has been subject to a variety of different hacks centered around adding functionality not allowed by Apple. Prior to the 2008 debut of Apple's native iOS App Store, the primary motive for jailbreaking was to bypass Apple's purchase mechanism for installing the App Store's native applications. Apple claimed that it would not release iOS software updates designed specifically to break these tools (other than applications that perform SIM unlocking); however, with each subsequent iOS update, previously un-patched jailbreak exploits are usually patched.

When a device is booting, it loads Apple's own kernel initially, so a jailbroken device must be exploited and have the kernel patched each time it is booted up.

There are different types of jailbreak. An *untethered* jailbreak uses exploits that are powerful enough to allow the user to turn their device off and back on at will, with the device starting up completely, and the kernel will be patched without the help of a computer – in other words, it will be jailbroken even after each reboot.

However, some jailbreaks are tethered. A tethered jailbreak is only able to temporarily jailbreak the device during a single boot. If the user turns the device off and then boots it back up without the help of a jailbreak tool, the device will no longer be running a patched kernel, and it may get stuck in a partially started state, such as Recovery Mode. In order for the device to start completely and with a patched kernel, it must be "re-jailbroken" with a computer (using the "boot tethered" feature of a tool) each time it is turned on. All changes to the files on the device (such as installed package files or edited system files) will persist between reboots, including changes that can only function if the device is jailbroken (such as installed package files).

In more recent years, two other solutions have been created – *semi-tethered* and *semi-untethered*.

A semi-tethered solution is one where the device is able to start up on its own, but it will no longer have a patched kernel, and therefore will not be able to run modified code. It will, however, still be usable for normal functions, just like stock iOS. To start with a patched kernel, the user must start the device with the help of the jailbreak tool.

A semi-untethered jailbreak gives the ability to start the device on its own. On first boot, the device will not be running a patched kernel. However, rather than having to run a tool from a computer to apply the kernel patches, the user is able to re-jailbreak their device with the help of an app (usually sideloaded using Cydia Impactor) running on their device. In the case of the iOS 9.2-9.3.3 jailbreak, a Safari-based exploit was available, thereby meaning a website could be used to rejailbreak.

In more detail: Each iOS device has a bootchain that tries to make sure only trusted/signed code is loaded. A device with a tethered jailbreak is able to boot up with the help of a jailbreaking tool because the tool executes exploits via USB that bypass parts of that "chain of trust", bootstrapping to a pwned (no signature check) iBSS, iBEC, or iBoot to finish the boot process.

Since the arrival of Apple's native iOS App Store, and—along with it—third-party applications, the general motives for jailbreaking have changed. People jailbreak for many different reasons, including gaining filesystem access, installing custom device themes, and modifying SpringBoard. An additional motivation is that it may enable the installation of pirated apps. On some devices, jailbreaking also makes it possible to install alternative operating systems, such as Android and the Linux kernel. Primarily, users jailbreak their devices because of the limitations of iOS. Depending on the method used, the effects of jailbreaking may be permanent or temporary.

In 2010, the Electronic Frontier Foundation (EFF) successfully convinced the U.S. Copyright Office to allow an exemption to the general prohibition on circumvention of copyright protection systems under the Digital Millennium Copyright Act (DMCA). The exemption allows jailbreaking of iPhones for the sole purpose of allowing legally obtained applications to be added to the iPhone. The exemption does not affect the contractual relations between Apple and an iPhone owner, for example, jailbreaking voiding the iPhone warranty; however, it is solely based on Apple's discretion on whether they will fix jailbroken devices in the event that they need to be repaired. At the same time, the Copyright Office exempted unlocking an iPhone from DMCA's anticircumvention prohibitions. Unlocking an iPhone allows the iPhone to be used with any wireless carrier using the same GSM or CDMA technology for which the particular phone model was designed to operate.

Unlocking

Main article: SIM lock

Initially most wireless carriers in the US did not allow iPhone owners to unlock it for use with other carriers. However AT&T allowed iPhone owners who have satisfied contract requirements to unlock their iPhone. Instructions to unlock the device are available from Apple, but it is ultimately the sole discretion of the carrier to authorize the device to be unlocked. This allows the use of a carrier-sourced iPhone on other networks. Modern versions of iOS and the iPhone fully support LTE across multiple carriers despite where the phone was originally purchased from. There are programs to remove SIM lock restrictions, but are not supported by Apple and most often not a permanent unlock – a soft-unlock.

A software unlock is the process by which the iPhone is modified such that the baseband will accept the SIM card of any GSM carrier. This is entirely different from a jailbreak; jailbreaking one's iPhone does not unlock it. A jailbreak is, however, required for all currently public, unofficial software unlocks.

The legality of software unlocking varies in each country; for example, in the US, there is a DMCA exemption for unofficial software unlocking, but the exemption is limited to devices purchased before January 26, 2013 (so software unlocks for newer devices are in a legal grey area).

Digital rights management

The closed and proprietary nature of iOS has garnered criticism, particularly by digital rights advocates such as the Electronic Frontier Foundation, computer engineer and activist Brewster Kahle, Internet-law specialist Jonathan Zittrain, and the Free Software Foundation who protested the iPad's introductory event and have targeted the iPad with their "Defective by Design" campaign. Competitor Microsoft, via a PR spokesman, criticized Apple's control over its platform.

At issue are restrictions imposed by the design of iOS, namely digital rights management (DRM) intended to lock purchased media to Apple's platform, the development model (requiring a yearly subscription to distribute apps developed for the iOS), the centralized approval process for apps, as well as Apple's general control and lockdown of the platform itself. Particularly at issue is the ability for Apple to remotely disable or delete apps at will.

Some in the tech community have expressed concern that the locked-down iOS represents a growing trend in Apple's approach to computing, particularly Apple's shift away from machines that hobbyists can "tinker with" and note the potential for such restrictions to stifle software innovation. Former Facebook developer Joe Hewitt protested against Apple's control over its hardware as a "horrible precedent" but praised iOS's sandboxing of apps.

Security and privacy

See also: Mobile security and WARRIOR PRIDE

iOS utilizes many security features in both hardware and software. Below are summaries of the most prominent features.

Secure Boot

Before fully booting into iOS, there is low-level code that runs from the Boot ROM. Its task is to verify that the Low-Level Bootloader is signed by the Apple Root CA public key before running it. This process is to ensure that no malicious or otherwise unauthorized software can be run on an iOS device. After the Low-Level Bootloader finishes its tasks, it runs the higher level bootloader, known as iBoot. If all goes well, iBoot will then proceed to load the iOS kernel as well as the rest of the operating system.

Secure Enclave

The Secure Enclave is a coprocessor found in iOS devices part of the A7 and newer chips used for data protection, Touch ID and Face ID. The purpose of the Secure Enclave is to handle keys and other info such as biometrics that is sensitive enough to not be handled by the Application Processor (AP). It is isolated with a hardware filter so the AP cannot access it. It shares RAM with the AP, but its portion of the RAM (known as TZ0) is encrypted. The secure enclave itself is a flashable 4 MB AKF processor core called the secure enclave processor (SEP) as documented in Apple Patent Application 20130308838. The technology used is similar to ARM's TrustZone/SecurCore but contains proprietary code for Apple KF cores in general and SEP specifically. It is also responsible for generating the UID key on A9 or newer chips that protects user data at rest.

It has its own secure boot process to ensure that it is completely secure. A hardware random number generator is also included as a part of this coprocessor. Each device's Secure Enclave has a unique ID that is given to it when it is made and cannot be changed. This identifier is used to create a temporary key that encrypts the memory in this portion of the system. The Secure Enclave also contains an anti-replay counter to prevent brute force attacks.

The SEP is located in the devicetree under IODeviceTree:/arm-io/sep and managed by the AppleSEPManager driver.

In 2020, security flaws in the SEP were discovered, causing concerns about Apple devices such as iPhones.

Face ID

Main article: Face ID

Face ID is a face scanner that is embedded in the notch on iPhone models X, XS, XS Max, XR, 11, 11 Pro, 11 Pro Max, 12, 12 Mini, 12 Pro, and 12 Pro Max, and 13, 13 Mini, 13 Pro, and 13 Pro Max. It can be used to unlock the device, make purchases, and log into applications among other functions. When used, Face ID only temporarily stores the face data in encrypted memory in the Secure Enclave, as described below. There is no way for the device's main processor or any other part of the system to access the raw data that is obtained from the Face ID sensor.

Passcode

iOS devices can have a passcode that is used to unlock the device, make changes to system settings, and encrypt the device's contents. Until recently, these were typically four numerical digits long. However, since unlocking the devices with a fingerprint by using Touch ID has become more widespread, six-digit passcodes are now the default on iOS with the option to switch back to four or use an alphanumeric passcode.

Touch ID

Main article: Touch ID

Touch ID is a fingerprint scanner that is embedded in the home button and can be used to unlock the device, make purchases, and log into applications among other functions. When used, Touch ID only temporarily stores the fingerprint data in encrypted memory in the Secure Enclave, as described above. There is no way for the device's main processor or any other part of the system to access the raw fingerprint data that is obtained from the Touch ID sensor.

Address Space Layout Randomization

Main article: Address Space Layout Randomization

Address Space Layout Randomization (ASLR) is a low-level technique of preventing memory corruption attacks such as buffer overflows. It involves placing data in randomly selected locations in memory in order to make it more difficult to predict ways to corrupt the system and create exploits. ASLR makes app bugs more likely to crash the app than to silently overwrite memory, regardless of whether the behavior is accidental or malicious.

Non-Executable Memory

iOS utilizes the ARM architecture's Execute Never (XN) feature. This allows some portions of the memory to be marked as non-executable, working alongside ASLR to prevent buffer overflow attacks including return-to-libc attacks.

Encryption

As mentioned above, one use of encryption in iOS is in the memory of the Secure Enclave. When a passcode is utilized on an iOS device, the contents of the device are encrypted. This is done by using a hardware AES 256 implementation that is very efficient because it is placed directly between the flash storage and RAM.

iOS, in combination with its specific hardware, uses crypto-shredding when erasing all content and settings by obliterating all the keys in 'effaceable storage'. This renders all user data on the device cryptographically inaccessible.

Keychain

The iOS keychain is a database of login information that can be shared across apps written by the same person or organization. This service is often used for storing passwords for web applications.

App Security

Third-party applications such as those distributed through the App Store must be code signed with an Apple-issued certificate. In principle, this continues the chain of trust all the way from the Secure Boot process as mentioned above to the actions of the applications installed on the device by users. Applications are also sandboxed, meaning that they can only modify the data within their individual home directory unless explicitly given permission to do otherwise. For example, they cannot access data owned by other user-installed applications on the device. There is a very extensive set of privacy controls contained within iOS with options to control apps' ability to access a wide variety of permissions such as the camera, contacts, background app refresh, cellular data, and access to other data and services. Most of the code in iOS, including third-party applications, runs as the "mobile" user which does not have root privileges. This ensures that system files and other iOS system resources remain hidden and inaccessible to user-installed applications.

App Store bypasses

Companies can apply to Apple for enterprise developer certificates. These can be used to sign apps such that iOS will install them directly (sometimes called "sideloading"), without the app needing to be distributed via the App Store. The terms under which they are granted make clear that they are only to be used for companies who wish to distribute apps directly to their employees.

Circa January–February 2019, it emerged that a number of software developers were misusing enterprise developer certificates to distribute software directly to non-employees, thereby bypassing the App Store. Facebook was found to be abusing an Apple enterprise developer certificate to distribute an application to underage users that would give Facebook access to all private data on their devices. Google was abusing an Apple enterprise developer certificate to distribute an app to adults to collect data from their devices, including unencrypted data belonging to third parties. TutuApp, Panda Helper, AppValley, and TweakBox have all been abusing enterprise developer certificates to distribute apps that offer pirated software.

Network Security

iOS supports TLS with both low- and high-level APIs for developers. By default, the App Transport Security framework requires that servers use at least TLS 1.2. However, developers are free to override this framework and utilize their own methods of communicating over networks. When Wi-Fi is enabled, iOS uses a randomized MAC address so that devices cannot be tracked by anyone sniffing wireless traffic.

Two-Factor Authentication

Main article: Multi-factor authentication

Two-factor authentication is an option in iOS to ensure that even if an unauthorized person knows an Apple ID and password combination, they cannot gain access to the account. It works by requiring not only the Apple ID and password, but also a verification code that is sent to an iDevice or mobile phone number that is already known to be trusted. If an unauthorized user attempts to sign in using another user's Apple ID, the owner of the Apple ID receives a notification that allows them to deny access to the unrecognized device.

References:

Clover, Juli (March 31, 2022). "Apple Releases iOS 15.4.1 With Fix for Battery Drain Issue". MacRumors. Retrieved 2022.

"iOS 15.4.1 (19E258) - Releases - Apple Developer". Apple Developer. Apple Inc. March 31, 2022. Retrieved 2022.

Clover, Juli (April 5, 2022). "Apple Seeds First Betas of iOS 15.5 and iPadOS 15.5 to Developers". MacRumors. Retrieved 2022.

"iOS 15.5 beta (19F5047e) - Releases - Apple Developer". Apple Developer. Apple Inc. April 5, 2022. Retrieved 2022.

“Apple – iPad Pro – Specs". Apple. Archived from the original on January 4, 2019. Retrieved 2019.

"Apple – iPad mini 4 – Specs". Apple. Archived from the original on October 24, 2015. Retrieved 2015.

"Apple – iPad Air 2 – Technical Specifications". Apple. Archived from the original on October 26, 2015. Retrieved 2015.

"Apple – iPhone XS – Technical Specifications". Apple. Archived from the original on January 4, 2019. Retrieved 2019.

Tim Brookes (October 17, 2019). "Where Are iTunes Features in macOS Catalina?". How-To Geek.

"Apple Open Source". Retrieved 2020.

"Charting The Explosive Growth of the App Store". Lifewire. Retrieved 2018.

"iOS 15 Preview". Apple. Retrieved 2021.

Satariano, Adam; Burrows, Peter; Stone, Brad (October 14, 2011). "Scott Forstall, the Sorcerer's Apprentice at Apple". Bloomberg Businessweek. Bloomberg L.P. Archived from the original on April 7, 2017. Retrieved 2017.

Mike Wuerthele (October 5, 2016). "Apple's first European iOS Developer Academy opening on Thursday in Naples, Italy". AppleInsider. Archived from the original on December 21, 2016. Retrieved 2016.

"Chi è entrato, chi è scappato e cosa c'è dentro alla iOS Developer Academy di Napoli". Wired (in Italian). October 7, 2016. Retrieved 2020.

"Dopo Apple in arrivo a Napoli altri big dell'hi-tech". Il Sole 24 ORE (in Italian). October 18, 2019. Retrieved 2020.

"iOS Developer Academy aprirà a Napoli | In Ateneo". University of Naples Federico II. Retrieved 2020.

"Developer Academy | Università Federico II". University of Naples Federico II. Retrieved 2020.

"Apple Developer Academy di Napoli, al via le nuove iscrizioni". lastampa.it (in Italian). May 15, 2019. Retrieved 2020.

"Apple unveils iPadOS, adding features specifically to iPad". AppleInsider. Retrieved 2020.

"Interface Essentials – iOS – Human Interface Guidelines – Apple Developer". developer.apple.com. Retrieved 2020.

"Adaptivity and Layout – Visual Design – iOS – Human Interface Guidelines – Apple Developer". developer.apple.com. Retrieved 2020.

"Widgets – System Capabilities – iOS – Human Interface Guidelines – Apple Developer". developer.apple.com. Retrieved 2020.