Three iOS 0-days revealed by researcher frustrated with Apple’s bug bounty

Public disclosure comes in wake of other grumblings about Apple's bug bounty behavior.

Jim Salter - 9/24/2021, 8:25 PM

Enlarge / Pseudonymous researcher illusionofchaos joins a growing legion of security researchers frustrated with Apple's slow response and inconsistent policy adherence when it comes to security flaws.

Aurich Lawson | Getty Images

Yesterday, a security researcher who goes by `illusionofchaos` dropped public notice of three zero-day vulnerabilities in Apple's iOS mobile operating system. The vulnerability disclosures are mixed in with the researcher's frustration with Apple's Security Bounty program, which `illusionofchaos` says chose to cover up an earlier-reported bug without

giving them credit.

This researcher is by no means the first to publicly express their frustration with Apple over its security bounty program.

Nice bug—now *shhh*

`illusionofchaos` says that they've reported four iOS security vulnerabilities this year—the three zero-days they publicly disclosed yesterday plus an earlier bug that they say Apple fixed in iOS 14.7. It appears that their frustration largely comes from how Apple handled that first, now-fixed bug in `analyticsd`.

This now-fixed vulnerability allowed arbitrary user-installed apps to access iOS's analytics data—the stuff that can be found in `Settings --> Privacy --> Analytics & Improvements --> Analytics Data`—without any permissions granted by the user. `illusionofchaos` found this particularly disturbing, because this data includes medical data harvested by Apple Watch, such as heart rate, irregular heart rhythm, atrial fibrillation detection, and so forth.

Analytics data was available to any application, even if the user disabled the iOS `Share Analytics` setting.

According to `illusionofchaos`, they sent Apple the first detailed report of this bug on April 29. Although Apple responded the next day, it did not respond to `illusionofchaos` again until June 3, when it said it planned to address the issue in iOS 14.7. On July 19, Apple did indeed fix the bug with iOS 14.7, but the security content list for iOS 14.7 acknowledged neither the researcher nor the vulnerability.

Apple told `illusionofchaos` that its failure to disclose the vulnerability and credit them was just a "processing issue" and that proper notice would be given in "an upcoming update." The vulnerability and its resolution still

were not acknowledged as of iOS 14.8 on September 13 or iOS 15.0 on September 20.

Frustration with this failure of Apple to live up to its own promises led `illusionofchaos` to first threaten, then publicly drop this week's three zero-days. In `illusionofchaos`' own words: "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."

We do not have concrete timelines for `illusionofchaos`' disclosure of the three zero-days, or of Apple's response to them—but `illusionofchaos` says the new disclosures still adhere to responsible guidelines: "Google Project Zero discloses vulnerabilities in 90 days after reporting them to vendor, ZDI - in 120. I have waited much longer, up to half a year in one case."

New vulnerabilities: Gamed, nehelper enumerate, nehelper Wi-Fi

The zero-days `illusionofchaos` dropped yesterday can be used by user-installed apps to access data that those apps should not have or have not been granted access to. We've listed them below—along with links to `illusionofchaos`' Github repos with proof-of-concept code—in order of (our opinion of) their severity:

- Gamed zero-day exposes Apple ID email and full name, exploitable Apple ID authentication tokens, and read access to Core Duet and Speed Dial databases
- Nehelper Wi-Fi zero-day exposes Wi-Fi information to apps that have not been granted that access

- Nehelper Enumerate zero-day exposes information about what apps are installed on the iOS device

The Gamed 0-day is obviously the most severe, since it both exposes Personal Identifiable Information (PII) and may be used in some cases to be able to perform actions at `*.apple.com` that would normally need to be either instigated by the iOS operating system itself, or by direct user interactions.

The Gamed zero-day's read access to Core Duet and Speed Dial databases is also particularly troubling, since that access can be used to gain a pretty complete picture of the user's entire set of interactions with others on the iOS device—who is in their contact list, who they've contacted (using both Apple and third-party applications) and when, and in some cases even file attachments to individual messages.

The Wi-Fi zero-day is next on the list, since unauthorized access to the iOS device's Wi-Fi info might be used to track the user—or, possibly, learn the credentials necessary to access the user's Wi-Fi network. The tracking is typically a more serious concern, since physical proximity is generally required to make Wi-Fi credentials themselves useful.

One interesting thing about the Wi-Fi zero-day is the simplicity of both the flaw and the method by which it can be exploited: "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 entitlement` check is skipped." In other words, all you need to do is claim to be using an older software development kit—and if so, your app gets to ignore the check that should disclose whether the user consented to access.

The Nehelper Enumerate zero-day appears to be the least damaging of

the three. It simply allows an app to check whether another app is installed on the device by querying for the other app's `bundleID`. We haven't come up with a particularly scary use of this bug on its own, but a hypothetical malware app might leverage such a bug to determine whether a security or antivirus app is installed and then use that information to dynamically adapt its own behavior to better avoid detection.

Conclusions

Assuming `illusionofchaos`' description of their disclosure timeline is correct—that they've waited for longer than 30 days, and in one case 180 days, to publicly disclose these vulnerabilities—it's hard to fault them for the drop. We do wish they had included full timelines for their interaction with Apple on all four vulnerabilities, rather than only the already-fixed one.

We can confirm that this frustration of researchers with Apple's security bounty policies is by no means limited to this one pseudonymous researcher. Since Ars published a piece earlier this month about Apple's slow and inconsistent response to security bounties, several researchers have contacted us privately to express their own frustration. In some cases, researchers included video clips demonstrating exploits of still-unfixed bugs.

We have reached out to Apple for comment, but we have yet to receive any response as of press time. We will update this story with any response from Apple as it arrives.