A very deep dive into iOS Exploit chains found in the wild

Archived: 2026-04-05 16:00:18 UTC

Posted by Ian Beer, Project Zero

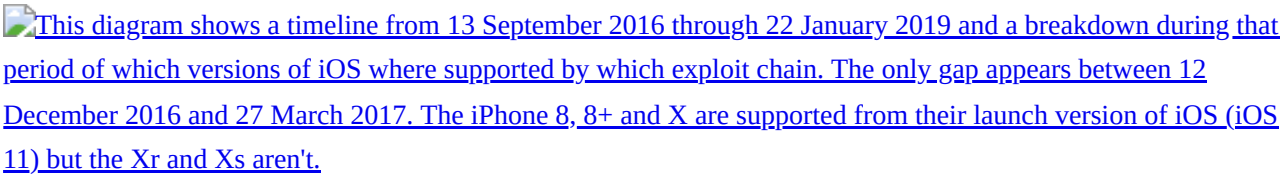
Project Zero's mission is to make 0-day hard. We often work with other companies to find and report security vulnerabilities, with the ultimate goal of advocating for structural security improvements in popular systems to help protect people everywhere.

Earlier this year Google's Threat Analysis Group (TAG) discovered a small collection of hacked websites. The hacked sites were being used in indiscriminate watering hole attacks against their visitors, using iPhone 0-day.

There was no target discrimination; simply visiting the hacked site was enough for the exploit server to attack your device, and if it was successful, install a monitoring implant. We estimate that these sites receive thousands of visitors per week.

TAG was able to collect five separate, complete and unique iPhone exploit chains, covering almost every version from iOS 10 through to the latest version of iOS 12. This indicated a group making a sustained effort to hack the users of iPhones in certain communities over a period of at least two years.

I'll investigate what I assess to be the root causes of the vulnerabilities and discuss some insights we can gain into Apple's software development lifecycle. The root causes I highlight here are not novel and are often overlooked: we'll see cases of code which seems to have never worked, code that likely skipped QA or likely had little testing or review before being shipped to users.


This diagram shows a timeline from 13 September 2016 through 22 January 2019 and a breakdown during that period of which versions of iOS where supported by which exploit chain. The only gap appears between 12 December 2016 and 27 March 2017. The iPhone 8, 8+ and X are supported from their launch version of iOS (iOS 11) but the Xr and Xs aren't.

Working with TAG, we discovered exploits for a total of fourteen vulnerabilities across the five exploit chains: seven for the iPhone's web browser, five for the kernel and two separate sandbox escapes. Initial analysis indicated that at least one of the privilege escalation chains was still 0-day and unpatched at the time of discovery (CVE-2019-7287 & CVE-2019-7286). We reported these issues to Apple with a 7-day deadline on 1 Feb 2019, which resulted in the out-of-band release of iOS 12.1.4 on 7 Feb 2019. We also shared the complete details with Apple, which were disclosed publicly on 7 Feb 2019.

Now, after several months of careful analysis of almost every byte of every one of the exploit chains, I'm ready to share these insights into the real-world workings of a campaign exploiting iPhones en masse.

This post will include:

- detailed write-ups of all five privilege escalation exploit chains;
- a teardown of the implant used, including a demo of the implant running on my own devices, talking to a reverse-engineered command and control server and demonstrating the capabilities of the implant to steal private data like iMessages, photos and GPS location in real-time, and
- analysis by fellow team member Samuel Groß on the browser exploits used as initial entry points.

Let's also keep in mind that this was a failure case for the attacker: for this one campaign that we've seen, there are almost certainly others that are yet to be seen.

Real users make risk decisions based on the public perception of the security of these devices. The reality remains that security protections will never eliminate the risk of attack if you're being targeted. To be targeted might mean simply being born in a certain geographic region or being part of a certain ethnic group. All that users can do is be conscious of the fact that mass exploitation still exists and behave accordingly; treating their mobile devices as both integral to their modern lives, yet also as devices which when compromised, can upload their every action into a database to potentially be used against them.

I hope to guide the general discussion around exploitation away from a focus on the the million dollar dissident and towards discussion of the marginal cost for monitoring the n+1'th potential future dissident. I shan't get into a discussion of whether these exploits cost $1 million, $2 million, or $20 million. I will instead suggest that all of those price tags seem low for the capability to target and monitor the private activities of entire populations in real time.

I recommend that these posts are read in the following order:

1.
2.
3.
4.
5.
6.
7.
