Subject: Re: Seemingly benign "Jekyll" app passes Apple review, then becomes "evil" | Ars Technica

From: Eddy Cue [redacted]

Received(Date): Tue, 27 Aug 2013 07:54:14 -0700

Cc: Phil Schiller [redacted], "C.K. Haun" [redacted],Ron Okamoto [redacted], Emily Blumsack [redacted],Kevin Saul [redacted], Henri Lamiraux [redacted],Greg Joswiak [redacted], Matt Fischer [redacted],Craig Federighi [redacted]

To: Phillip Shoemaker [redacted]

Date: Tue, 27 Aug 2013 07:54:14 -0700

Of the 3 things, this seems like the most important to do. We should speak to Craig to get this prioritized. At the same time, we should speak to Niall about the proxy. Both of these would help us not just with malware but to keep honest developers honest.

On Aug 27, 2013, at 7:49 AM, Phillip Shoemaker [redacted] wrote:

The dev tools team.

On Aug 27, 2013, at 7:23 AM, Eddy Cue [redacted] wrote:

> Who does the work on the analyzer?
>
> On Aug 27, 2013, at 7:16 AM, Phillip Shoemaker [redacted] wrote:
>
> Privileged and Confidential
>
> Phil,
>
> We didn't have time to discuss at ERB this past Friday, but I want you to be armed with the information on what we need in order to prevent this from happening in the future.
>
> There were three key points from this document, and they are:
> 1. The use of private APIs can get through the process when these APIs are obfuscated. This is because we use a static analyzer.
> 2. Once these private APIs are used, apps can access content that they're not allowed to, and send these off device to a back end server.
> 3. That developers can have a backdoor switch to have the app act one way during review, and an entirely different way once the app is live on the store.
>
> **Static Analyzer**
> Because Objective-C has calls like dlopen and dlsym, developers are able to dynamically open libraries on the fly, using APIs that are undetected using a static analyzer. This static analyzer looks at API names rather than true APIs being called, so there's often the issue of false

positives. The Static Analyzer enables us to catch direct accessing of Private APIs, but it completely misses apps using indirect methods of accessing these Private APIs. This is what the authors used in their Jekyll apps.

A dynamic analyzer has been on the roadmap from DevTools for close to four years now, but doesn't appear to be getting any closer to being done. This is an important component in our ability to catch these types of apps, but more importantly, without this, I can guarantee that we will continue to allow apps accessing Private APIs to get onto the store, because we don't have the tools to catch these types of shenanigans.

A dynamic analyzer also needs the smoketest from the iTunes team, but we have no ETA on it's availability.

Privacy Proxy
Once an app has accessed any content on the device, the developer has the ability to send it to a back end server. While the team looks for apps appropriately accessing data, and informing the user, we have no real knowledge what they are doing with this data. To solve this, we designed a "Privacy Proxy" that is simply a Wifi Access Point, that you simply log into with your review device, and it watches all of the data that is being transferred through it. When it detects that a UDID, or Mac address, or contacts are being sent to a back end server, it messages that reviewer's Mac, and the reviewer is then armed with the knowledge that this data is being transferred. It is now up to that reviewer to make the determination if this is allowed or not (if the contacts are being backed up to the cloud and this is a key feature of the app, then this should be ok).

We have worked with a variety of teams on this including iOS, DTS, IS&T and InfoSec, but it appears to be in no-mans land right now. IS&T wants to support this, but we just don't have the pressure to get this onto their roadmap any time soon.

Backdoor Switch
This is one that App Review have lived with since the beginning, and we are nowhere near closer to figuring out how to handle it. One idea is to capture stack shots during review, and compare them to real user's stack shots in the wild (for users that have data collection turned on), and see if there are differences. If so, this should automatically trigger a re-review. Any other ideas would be greatly appreciated.

We need some help in convincing other teams to implement this functionality for us. Until then, it is more brute force, and somewhat ineffective.

On Aug 20, 2013, at 12:24 PM, Phillip Shoemaker [redacted] wrote:

Privileged and Confidential

I will be prepared to discuss this issue from the App Review side at ERB on Friday.

On Aug 20, 2013, at 12:20 PM, Philip Schiller [redacted] wrote:

Privileged and Confidential

I'd like to hear in ERB on Friday what our plan is to handle this issue going forward

http://arstechnica.com/security/2013/08/seemingly-benign-jekyll-app-passes-apple-review-then-becomes-evil/

Seemingly benign "Jekyll" app passes Apple review, then becomes "evil"

Computer scientists say they found a way to sneak malicious programs into Apple's exclusive app store without being detected by the mandatory review process that's supposed to automatically flag such apps.

The researchers from the Georgia Institute of Technology used the technique to create what appeared to be a harmless app that Apple reviewers accepted into the iOS app store. They were later able to update the app to carry out a variety of malicious actions without triggering any security alarms. The app, which the researchers titled "Jekyll," worked by taking the binary code that had already been digitally signed by Apple and rearranging it in a way that gave it new and malicious behaviors.

"Our method allows attackers to reliably hide malicious behavior that would otherwise get their app rejected by the Apple review process," the researchers wrote in a paper titled *Jekyll on iOS: When Benign Apps Become Evil*. "Once the app passes the review and is installed on an end user's device, it can be instructed to carry out the intended attacks. The key idea is to make the apps remotely exploitable and subsequently introduce malicious control flows by rearranging signed code. Since the new control flows do not exist during the app review process, such apps, namely Jekyll apps, can stay undetected when reviewed and easily obtain Apple's approval."

Apple representatives didn't immediately respond to a request for comment, but company spokesman Tom Neumayr told MIT Technology Review that developers have made changes to the iOS operating system in response to issues identified in the paper. It remains unclear if the vulnerabilities have been completely fixed.

The Jekyll app was only active for a few minutes following it's launch in March and during that time it wasn't installed by anyone not involved in the experiment, the researchers said. The app had the ability to carry out a variety of malicious attacks, including stealthily sending tweets, e-mails, and text messages; stealing device ID numbers; taking photos; and attacking other apps. The app could also cause Apple's Safari browser to

load booby-trapped websites.

"Such a seemingly benign app can pass the app review because it neither violates any rules imposed by Apple nor contains functional malice," the researchers wrote. "However, when a victim downloads and runs the app, attackers can remotely exploit the planted vulnerabilities and in turn assemble the gadgets to accomplish various malicious tasks."

The attack described by the researchers in some ways resembles an exploit that allows hackers to inject malicious code into legitimate Android apps without invalidating their digital signature. The so-called "master key vulnerability" was also discovered by white-hat researchers, but it's now being exploited in some third-party marketplaces.