From: Daniel Vogel <[redacted]>
Sent: Mon, 27 Aug 2018 12:51:09 +0000 (UTC)
To: Tim Sweeney <[redacted]>; Paul Meegan <[redacted]>; Canon Pence <[redacted]>
Subject: Fwd: Fortnite's Bad Bug, Epic's Complaint, The Downside of Open

---

REDACTED FOR PRIVILEGE

-- Daniel, Epic Games Inc.

---------- Forwarded message ----------
From: **Ben Thompson** <[redacted]>
Date: Mon, Aug 27, 2018 at 8:10 AM
Subject: Fortnite's Bad Bug, Epic's Complaint, The Downside of Open
To: Daniel Vogel <[redacted]>

View in browser

Fortnite's Bad Bug, Epic's Complaint, The Downside of Open

Monday, August 27, 2018
Good morning,

I made a silly mistake in the last update: Android Oreo, with its accompanying changes to the granularity with which users can install apps from "Unknown Sources", came out in August 2017, not August 2018; the current version of Android is Android Pie. However, as of late July only 12.1% of Android devices were running Oreo.

On to the update, which ended up building on that update from last week thanks to some new Fortnite news:

Fortnite's Bad Bug

From Android Central:

> Google has just publicly disclosed that it discovered an extremely serious vulnerability in Epic's first Fortnite installer for Android that allowed any app on your phone to download and install anything in the background, including apps with full permissions granted, without the user's knowledge. Google's security team first disclosed the vulnerability privately to Epic Games on August 15, and has since released the information publicly following confirmation from Epic that the vulnerability was patched.

The bug is pretty straightforward; from Google's issue tracker:

> The Fortnite APK (com.epicgames.fortnite) is downloaded by the Fortnite Installer (com.epicgames.portal) to external storage…Any app with the WRITE_EXTERNAL_STORAGE permission can substitute the APK immediately after the download is completed and the fingerprint is verified. This is easily done using a FileObserver. The Fortnite Installer will proceed to install the substituted (fake) APK…
>
> If the fake APK has a targetSdkVersion of 22 or lower, it will be granted all permissions it requests at install-time. This vulnerability allows an app on the device to hijack the Fortnite Installer to instead install a fake APK with any permissions that would normally require user disclosure.

I swear, this really is straightforward, but let me add some additional context:

- The vast majority of Android phones have external storage, usually a mini-SD card. This storage is fully accessible to all apps that have been granted permission upon first run to access external storage; this is a very common permissions request that the vast majority of users grant without thinking about it. And why not? Shared storage can be quite convenient, as it makes it easy to share data between apps, or even with other devices, like a user's PC.
- Apps also have their own storage on the same disk as the operating system; this storage is private and can only be accessed by the app.
- As noted last week, when users want to install Fortnite they do not go to the Google Play store; rather, they download from Epic's website. However, they don't download the full-game, but rather a lightweight app that is purpose-built to install Fortnite. This is where the connection to Android Oreo comes in: instead of asking users to allow the installation of apps from anywhere (as is necessary on all versions of Android before Oreo), or from the browser (which

is to say, effectively from anywhere), users only need to allow the installation of apps from the Fortnite installer (beyond the installation of the installer itself, that is).

- When the Fortnite installer was run, it requested all necessary permissions from the user, and then downloaded the game itself from Epic's website. However — and this is half of the bug — it downloaded the game to external storage, which remember, all apps have read-and-write access to.
- If the user had a malicious app already on their phone that had been given permission to use external storage (which again, basically all apps have), that app could surreptitiously watch for the Fortnite download, and then substitute its own app package to be installed by the installer, or inject code into Fortnite itself. This is the other half of the bug: Fortnite's installer did not validate the download to make sure it was from Epic and that it hadn't been tampered with.

The end result is that malware could be installed on a user's phone with the same permissions as Fortnite, or Fortnite itself altered to include malware, without the user even knowing.

Epic's Complaint

Epic Games CEO Tim Sweeney was not very happy with Google:

> Epic genuinely appreciated Google's effort to perform an in-depth security audit of Fortnite immediately following our release on Android, and share the results with Epic so we could speedily issue an update to fix the flaw they discovered.
>
> However, it was irresponsible of Google to publicly disclose the technical details of the flaw so quickly, while many installations had not yet been updated and were still vulnerable. An Epic security engineer, at my urging, requested Google delay public disclosure for the typical 90 days to allow time for the update to be more widely installed. Google refused. You can read it all at https://issuetracker.google.com/issues/112630336
>
> Google's security analysis efforts are appreciated and benefit the Android platform, however a company as powerful as Google should practice more responsible disclosure timing than this, and not endanger users in the course of its counter-PR efforts against Epic's distribution of Fortnite outside of Google Play.

Sweeney is wrong on this point: Google's policy is to disclose vulnerabilities *that aren't patched* after 90 days, and "sooner if the vendor releases a fix"; the Google engineer that disclosed the vulnerability to Epic states this explicitly in the initial bug report:

> [NOTE: This bug is subject to a 90-day disclosure deadline. After 90 days elapse or a patch has been made broadly available, the bug report — including any comments and attachments — will become visible to the public.]

Frankly, Google has done quite a bit to accommodate Epic: not only did they find this vulnerability, they actually warn users who search for Fortnite in the Google Play store that it is not there so that they don't install the wrong thing. And, of course, there is the design of Android itself, particularly the ability to install apps from outside the Play Store. Last week I quoted Sweeney praising Google for this openness:

> We're distributing Fortnite to Android users from epicgames.com. You go to our website, click the download button, and go through some prompts to download and install Fortnite. There's no third-party store involved at all. It's just like the PC experience, the way that PC and Mac users install Fortnite. That now works on Android, which is possible because Android is an open platform.

Do you know what else is "like the PC experience"? Having a bunch of disk space that is accessible to every app — that is, external storage, the root cause of this bug. Oh sure, Epic's developers should have used protected internal storage (they were likely concerned about space, as many Android phones skimp here); Google says on the Android developer site:

> Files created on external storage, such as SD cards, are globally readable and writable. Because external storage can be removed by the user and also modified by any application, don't store sensitive information using external storage.
>
> You should perform input validation when handling data from external storage as you would with data from any untrusted source. You should not store executables or class files on external storage prior to dynamic loading. If your app does retrieve executable files from external storage, the files should be signed and cryptographically verified prior to dynamic loading.

The Downside of Open

The problem is that relying on conscientious developers for security is a fool's errand. People make mistakes — note this bit from Check Point, which discovered this class of attacks:

> Among the applications which were tested for this new attack surface were Google Translate, Yandex Translate, Google Voice Typing, Google Text-to-Speech, Xiaomi Browser and various additional applications. After referring to the advice given within Google's guidelines, our team proceeded to compare the advice to what is actually the case.

> In the case of Google Translate, Yandex Translate and Google Voice Typing, we found that the developers failed to validate the integrity of data read from the External Storage. As such, our team was able to compromise certain files required by these apps, resulting in the crash of each of these applications.
>
> Xiaomi Browser was found to be using the External Storage as a staging resource for application updates. As a result, our team was able to carry out an attack by which the application's update code was replaced, resulting in the installation of an alternative, undesired application instead of the legitimate update.

Yep, some of Google's own apps have the same type of vulnerability as Fortnite. Granted, they are not as bad, because they don't have permission to install apps like Fortnite does; many OEM apps do, though, like the Xiaomi browser (also, apparently Xiaomi "chose not to address [the vulnerability] at this time").

This gets at the broader point: there always has been and always will be an inverse correlation between "openness" of the sort preferred by Sweeney and Epic and security. iOS, as the obvious counter-example, doesn't have any shared storage at all. The only way to pass data between apps is to use Apple's extension system; similarly, the only way to move data onto a PC is to use a cloud service or iTunes. To put it another way, there are no vulnerabilities like this one on iOS not because Apple's developers are smarter or better at their jobs; there are no vulnerabilities of this type because the opportunity to make a mistake has been eliminated.

This trade-off goes both ways: to restrict the possibility of bad things is to restrict the possibility of good ones as well. It is hard to imagine PCs having the plethora of use cases that they do had they been closed off like iOS from the beginning (or like Android, which is still much more closed off than a PC); similarly, we will never know what smartphone use cases were never discovered because developers never had full access to the device (or to a full complement of business models).

To go back to Fortnite, Epic chose to pursue the upside of owning the customer relationship and not paying Google 30%; that the company now has to face the downside of having endangered its customers is part of the package, and Sweeney comes off extremely poorly by whining about Google. In fact, Google does deserve blame, not for being, well, open about this vulnerability, but for designing Android in such a way that this vulnerability could exist. And they deserve credit, too.

Thanks for being a supporter, and have a great day!

Discuss on the Stratechery Member Forum

Our mailing address is:
Stratechery LLC
PO Box 9516
Brea, CA 92822

Add us to your address book