SECURITY IS AN EXPENSIVE HASSLE —

Samsung's Android app-signing key has leaked, is being used to sign malware

The cryptographic key proves an update is legit, assuming your OEM doesn't lose it.

Dsimic

RON AMADEO - 12/2/2022, 3:13 PM

A developer's cryptographic signing key is one of the major linchpins of Android security. Any time Android updates an app, the signing key of the old app on your phone needs to match the key of the update you're installing. The matching keys ensure the update actually comes from the company that originally made your app and isn't some malicious hijacking plot. If a developer's signing key got leaked, anyone could distribute malicious app updates and Android would happily install them, thinking they are legit.

On Android, the app-updating process isn't just for apps downloaded from an app store, you can also update bundled-in system apps made by Google, your device manufacturer, and any other bundled apps. While downloaded apps have a strict set of permissions and controls, bundled-in Android system apps have access to much more powerful and invasive permissions and aren't subject to the usual Play Store limitations (this is why Facebook always pays to be a bundled app). If a third-party developer ever lost their signing key, it would be bad. If an *Android OEM* ever lost their system app signing key, it would be really, really bad.

Guess what has happened! Łukasz Siewierski, a member of Google's Android Security Team, has a post on the Android Partner Vulnerability Initiative (AVPI) issue tracker detailing leaked platform certificate keys that are actively being used to sign malware. The post is just a list of the keys, but running each one through APKMirror or Google's VirusTotal site will put names to some of the compromised keys: Samsung, LG, and Mediatek are the heavy hitters on the list of leaked keys, along with some smaller OEMs like Revoview and Szroco, which makes Walmart's Onn tablets.

These companies somehow had their signing keys leaked to outsiders, and now you can't trust that apps that claim to be from these companies are really from them. To make matters worse, the "platform certificate keys" that they lost have some serious permissions. To quote the AVPI post:

> A platform certificate is the application signing certificate used to sign the "android" application on the system image. The "android" application runs with a highly privileged user id—android.uid.system—and holds system permissions, including permissions to access

> user data. Any other application signed with the same certificate can declare that it wants to run with the same user id, giving it the same level of access to the Android operating system.

Esper Senior Technical Editor Mishaal Rahman, as always, has been posting great info about this on Twitter. As he explains, having an app grab the same UID as the Android system isn't quite root access, but it's close and allows an app to break out of whatever limited sandboxing exists for system apps. These apps can directly communicate with (or, in the case of malware, spy on) other apps across your phone. Imagine a more evil version of Google Play Services, and you get the idea.

Samsung: Actually, our key was compromised years ago

Samsung isn't just the biggest Android OEM to have a signing key leak, it's also the biggest user of a leaked key. That earlier APKMirror link shows just how bad it is. Samsung's compromised key is used for everything: Samsung Pay, Bixby, Samsung Account, the phone app, and a million other things you can find on the 101 pages of results for that key. It would be possible to craft a malicious update for any one of these apps, and Android would be happy to install it overtop of the real app. Some of the updates are from *today*, indicating Samsung has still not changed the key.

The story gets even weirder, though. As APKMirror founder Artem Russakovskii points out, some of the samples of officially signed malware on VirusTotal are from 2016! So has this problem been going on for six years? Samsung gave the following statement to XDA Developers' Adam Conway:

> Samsung takes the security of Galaxy devices seriously. We have issued security patches since 2016 upon being made aware of the issue, and there have been no known security incidents regarding this potential vulnerability. We always recommend that users keep their devices up-to-date with the latest software updates.

Frankly, that statement doesn't make any sense. If Samsung has known about this for years, why is it still using a compromised key? There might be some logistical issues in updating an already shipped phone that Samsung might not want to deal with, but Samsung has made a lot of new devices between 2016 and now. Making a new build of the OS with new keys, on a new phone, seems like something it should have done years ago.

There's also a statement from the Android Security Team in the comments of the AVPI post:

> OEM partners promptly implemented mitigation measures as soon as we reported the key compromise. End users will be protected by user mitigations implemented by OEM partners. Google has implemented broad detections for the malware in Build Test Suite, which scans system images. Google Play Protect also detects the malware. There is no indication that this

malware is or was on the Google Play Store. As always, we advise users to ensure they are running the latest version of Android.

That statement doesn't give a lot of detail about what users can do to protect themselves, but bringing up the Play Store is a good point. Sideloading a random app update from the Internet is particularly dangerous right now, since nothing can guarantee that a Samsung/LG/Mediatek app is actually from that company, and the main security mechanism for sideloading updates no longer works. The Play Store at least does some amount of virus scanning, and the existing package names that would trigger an update notification are all claimed under an OEM's Google account. Google's "Play Protect" system is included in all Android phones and has the ability to remotely uninstall identified malware apps without any user input, even if you sideloaded it.

What OEMs really need to do is stop using the compromised keys to secure their apps. It's not clear why Samsung continues to use the key. Android's APK Signature Scheme V3 allows developers to change app keys with just an update—you authenticate an app with the new and old key and indicate that only the new key is supported for updates. This is a requirement for Play Store apps, but again, system apps from OEMs are not subject to any of the Play Store rules, so some OEMs are still using the old v2 signature scheme.

Thankfully, these leaked keys are only for apps and not the keys used to sign OS updates. So even if the v3 signature scheme is not in use, theoretically the affected companies could ship a still-secure OTA update that includes new system apps with new keys, and they could make new corresponding Play Store updates that are compatible with those new keys. That sounds like a lot of work, though.

Consumers are now left in the dark about how this happened and how it's being handled. We're going to be very generous and hope it's just because this is a newly developing situation right now. We'll update this post if Samsung or Google answers any of our myriad questions.

READER COMMENTS 137 SHARE THIS STORY

RON AMADEO
Ron is the Reviews Editor at Ars Technica, where he specializes in Android OS and Google products. He is always on the hunt for a new gadget and loves to rip things apart to see how they work.

EMAIL ron@arstechnica.com // **TWITTER** @RonAmadeo

SITREP: F-16 replacement