Vulnerability in Google Play Core Library Remains Unpatched in Google Play Applications

research.checkpoint.com/2020/vulnerability-in-google-play-core-library-remains-unpatched-in-google-play-applications/

December 3, 2020

December 3, 2020
Research by: Aviran Hazum, Jonathan Shimonovich

Overview:

A new vulnerability for the Google Play Core Library was published in late August, which allows Local-Code-Execution (LCE) within the scope of any application that has the vulnerable version of the Google Play Core Library.

In this paper, we analyze the impact and magnitude of this vulnerability from a security perspective.

Background:

What is the Google Play Core Library?

From Google's Android Development Documentation:

The Play Core Library is your app's runtime interface with the Google Play Store. Some of the things you can do with Play Core include the following:

- *Download additional language resources*
- *Manage delivery of feature modules*
- *Manage delivery of asset packs*
- *Trigger in-app updates*
- *Request in-app reviews*

So, basically, the Google Play Core Library is a gateway for interacting with Google Play Services from within the application itself, starting from dynamic code loading (such as downloading additional levels only when needed), to delivering locale-specific resources, to interacting with Google Play's review mechanisms.

Many popular applications utilize this library including:

Facebook and Instagram alone are responsible for 5 billion and 1 billion downloads to date, respectively, from the Google Play Store. Imagine the number of devices that were impacted by this vulnerability.

What is CVE-2020-8913?

OverSecured already covered the technical aspects of this vulnerability. For a more in-depth technical analysis, please refer to their blog.

A brief overview: Inside the sandbox of each application, there are two folders: one for "verified" files received from Google Play, and another for "non-verified" files. Files downloaded from Google Play services go into the verified folder, while files downloaded from other sources are sent to the non-verified folder. When a file is written to the verified folder, it interacts with the Google Play Core library which loads and executes it.

Another feature, an exported intent, allows other sources to push files into the hosting application's sandbox. There are some limitations: the file is pushed into the non-verified folder, and it is not automatically handled by the library.

The vulnerability lies within the combination of the two features mentioned above, and also utilizes file traversal, a concept as old as the internet itself. When a 3rd party source pushes a file into another application, it needs to supply a path for the file to be written to. If an attacker uses file traversal (../verified_splits/my_evil_payload.apk), the payload is written to the verified folder, and is automatically loaded into the vulnerable application and executed within its scope.

Google patched this vulnerability on April 6, 2020.

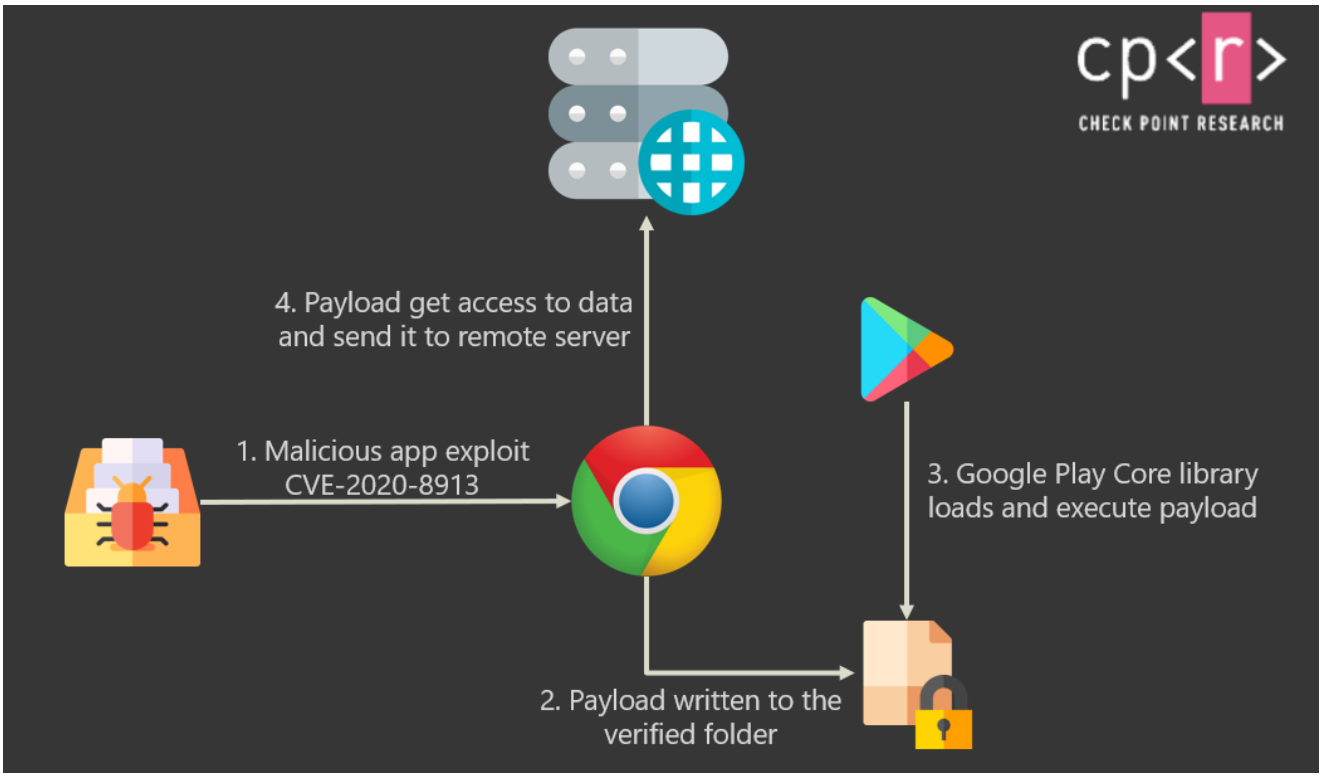


Figure 1 – Infographic showing the attack chain.

Impact and Magnitude:

When we combine popular applications that utilize the Google Play Core library, and the Local-Code-Execution vulnerability, we can clearly see the risks. If a malicious application exploits this vulnerability, it can gain code execution inside popular applications and have the same access as the vulnerable application.

The possibilities are limited only by our creativity. Here are just a few examples:

- Inject code into banking applications to grab credentials, and at the same time have SMS permissions to steal the Two-Factor Authentication (2FA) codes.
- Inject code into Enterprise applications to gain access to corporate resources.
- Inject code into social media applications to spy on the victim, and use location access to track the device.
- Inject code into IM apps to grab all messages, and possibly send messages on the victim's behalf.

Since the vulnerability was patched in April, why is there cause for concern now? The answer is because the patch needs to be pushed by the developers into the application. Unlike server-side vulnerabilities, where the vulnerability is patched completely once the patch is applied to the server, for client-side vulnerabilities, each developer needs to grab the latest version of the library and insert it into the application.

As the human factor is one of the most difficult to overcome when it comes to security, we decided to see which applications patched the vulnerability, and which are still vulnerable to get an overall better understanding of the vulnerability's magnitude.

Since the publication of this vulnerability, we started monitoring vulnerable applications. During the month of September 2020, 13% of Google Play applications analyzed by SandBlast Mobile used this library, and 8% of those apps had a vulnerable version.

We also compared the September versions to the current versions on Google Play so we could see which applications are still affected. To our surprise, we discovered applications from a large variety of genres:

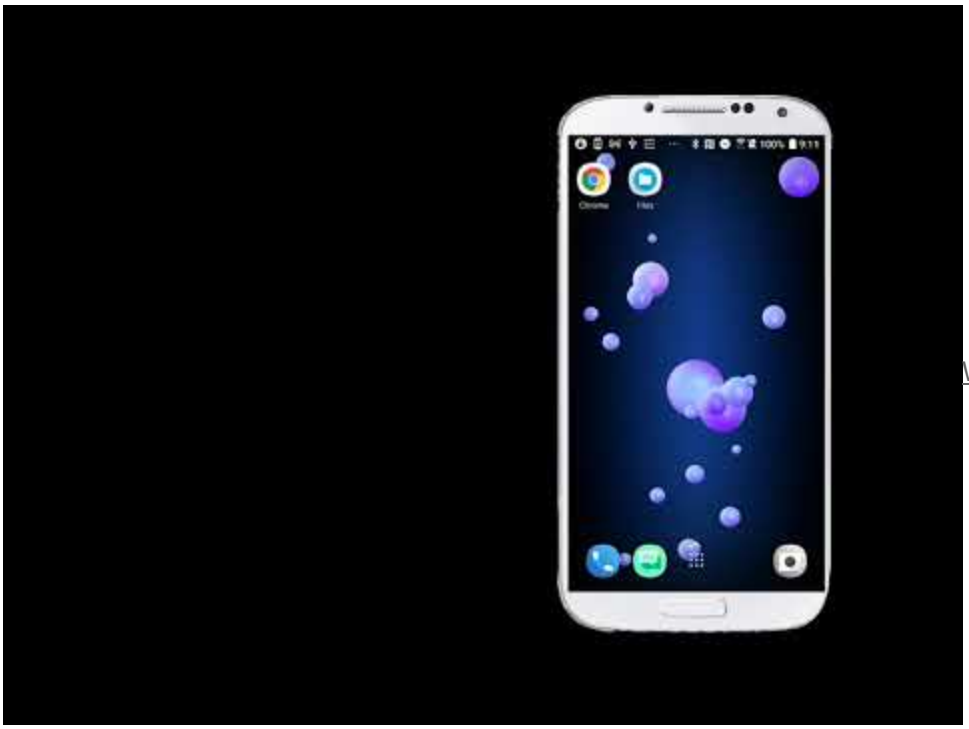
- Social – *Viber
- Travel – *Booking
- Business – ***Cisco Teams
- Maps and Navigation – Yango Pro (Taximeter), **Moovit
- Dating – **Grindr, OKCupid
- Browsers – Edge
- Utilities – Xrecorder, PowerDirector

*Prior to this publication, we have notified the Apps about the vulnerability and the need to update the version of the library , in order not to be affected. Viber & Booking updated to the patched versions after our notification.
** 19:00 December 3rd 2020 – Both Grindr & Moovit have updated their versions to the patched version and are no longer vulnerable

*** 19:25 December 3rd 2020 – Cisco teams updated to the latest version and the app is no longer vulnerable

Demo:

As our demo video shows, this vulnerability is easy to exploit. All you need to do is to create a "hello world" application that calls the exported intent in the vulnerable app to push a file into the verified files folder with the file-traversal path.
Then sit back and watch the magic happen. To demonstrate targeting a specific application, we took a vulnerable version of the Google Chrome application and created a dedicated payload to grab its bookmarks.



Watch Video At:

https://youtu.be/Dfa8JEvnteY

SandBlast Mobile can detect this vulnerability in both legitimate vulnerable applications, and malicious applications seeking to exploit it.

How to protect yourself:

Check Point SandBlast Mobile is the market-leading Mobile Threat Defense (MTD) solution, providing the widest range of capabilities to help you secure your mobile workforce. SandBlast Mobile provides protection for all mobile vectors of attack, including the download of malicious applications and applications with malware embedded in them.

Appendix 1 – Vulnerable applications in Google Play

Package Name	Name	Version	Download Count
com.viber.voip	*Viber	*14.1.0.16	500,000,000
com.booking	*Booking.com	*24.8.2	100,000,000
com.aloha.browser	Aloha	2.23.0	1,000,000
com.walla.wallasports	Walla! Sports	1.8.3.1	100,000
videoeditor.videorecorder.screenrecorder	XRecorder	1.4.0.3	100,000,000
com.tranzmate	Moovit	5.56.0.459	50,000,000

com.walla.wallahamal	Hamal	2.2.2.1	1,000,000
com.indiamart.m	IndiaMART	12.7.4	10,000,000
com.microsoft.emmx	Edge	45.09.4.5083	10,000,000
com.grindrapp.android	Grindr	6.32.0	10,000,000
ru.yandex.taximeter	Yango Pro (Taximeter)	9.56	5,000,000
com.cyberlink.powerdirector	PowerDirector	7.5.0	50,000,000
com.okcupid.okcupid	OkCupid	47.0.0	10,000,000
com.cisco.wx2.android	Teams	40.10.1.274	1,000,000