PROMON

PROMON SECURITY RESEARCH

StrandHogg 2.0

www.promon.co
info@promon.no
+47 22 02 11 30

Table of contents

PROMON

Brief description

Promon has discovered a new Android vulnerability that allows hackers to gain access to almost all apps. Classified «critical severity» by Google, the vulnerability has been named StrandHogg 2.0 by Promon due to its similarities with the infamous StrandHogg (1.0) vulnerability discovered by the company in 2019.

StrandHogg 2.0, like its predecessor, is a vulnerability in the task management system on Android that allows an attacker to hijack tasks. This can be exploited such that when the end user starts a legitimate app, instead of actually getting that app, the user is presented with a malicious look-a-like instead.

The key difference between StrandHogg (1.0), and StrandHogg 2.0 is that the former uses an attribute called taskAffinity to achieve the aforementioned task hijacking. For the attacker, the disadvantage of taskAffinity is that it has to be compiled into AndroidManifest.xml of the malicious app, in plaintext. While taskAffinity has many legitimate uses, it still means that this serves as a tip-off to Google Play Protect to detect malicious apps exploiting StrandHogg (1.0).

This is not the case with StrandHogg 2.0, which uses a different method for task hijacking - one which leaves no markers, and can be combined with reflection and obfuscation to hide the malicious parts extremely well.

After disclosing the findings to Google, they acknowledged the vulnerability and assigned it a Critical Severity Rating. Google asked for time to implement a patch to solve the issue. Since we had not seen any active malware abusing the vulnerabilities, we decided to comply with Google to extend the commonly accepted three-month timeline for reasonable disclosure, rather than to go public with a zero-day vulnerability. The vulnerabilities detailed in this document – CVE-2020-0096 – do not impact devices running Android 10.

Details

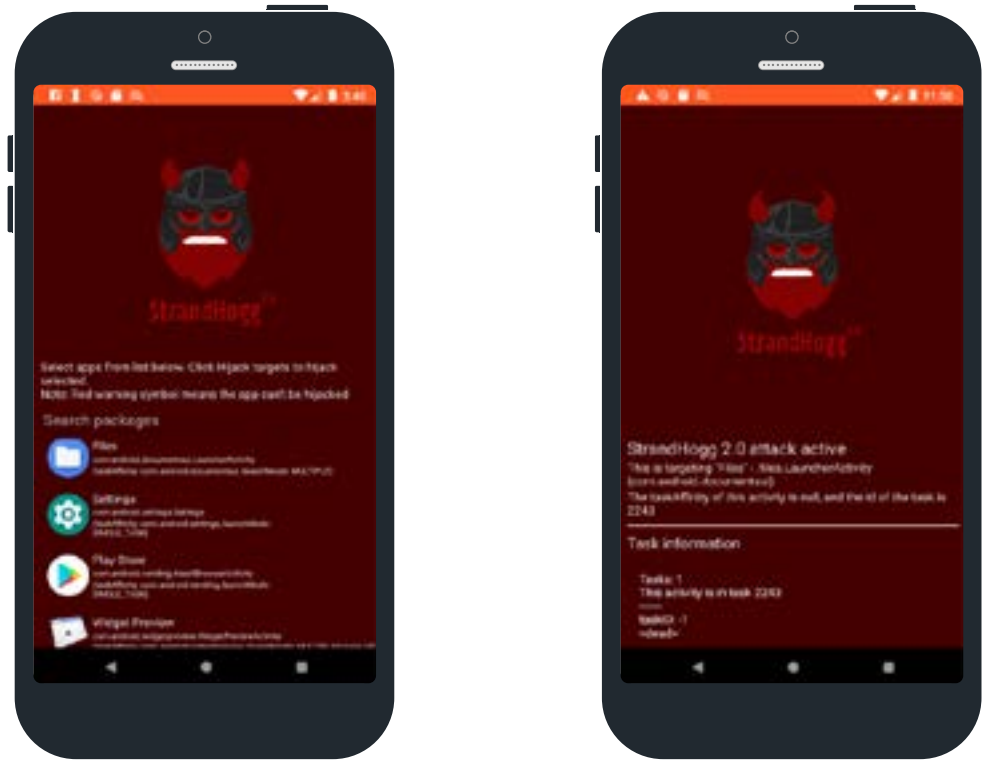
The attack is centered around android.app.Activity#startActivities(android.content.Intent[]) which has a peculiarity that shares some state on the passed Intents. As a result, this can be abused to create any arbitrary task stack.

In order to hijack a task correctly so that the launcher/Home will launch the modified task with the StrandHogg 2.0 attack activity on top (visible), the base Intent of the task has to identically appear as though the launcher/Home started the app.

startActivities(Intent[]) allows for the first creation of the launch intent in an identical manner to how the Launcher would launch it, and then adds in one or more intents that will launch into the same task. Only by setting FLAG_ACTIVITY_NEW_TASK is the implicit relationship to the
previous intent's task cleared. This is then abused in order to chain several attacks into one call by setting this flag on each baseIntent.

The last activity passed into startActivities(Intent[]) Intent array is the one (and only) activity which will actually be visible and shown to the device user.

A malicious app can fairly easily hijack all - or a subset - of the installed packages. The task hijacking effects are the same as with StrandHogg (1.0) (taskAffinity hijacking). By selecting target apps through the ‘recents’ screen, or by clicking the target app icon in the Launcher/Home, the hijacked task will show the malicious app’s activity, rather than the activity of the the originally targeted app. View our POC video on our information page.



PROMON

This can be used for various types of phishing attack, such as displaying a fake login screen, gathering different types of sensitive information, denial of service, and/or collecting permissions under the guise of the target app (such as SMS, GPS positioning and more).

- The vulnerability does not require any special markings, attributes or modifications to the malicious app (e.g. AndroidManifest.xml)
- The vulnerability does not require any modifications to the target app(s)
- The vulnerability does not require any special privileges, configurations, permissions or any other kind of modifications to the device
- The vulnerability does require that the malicious app must have target app(s) installed on the device
- The vulnerability appears to be effective on all Android versions below Android 10. Early versions (<4.0.1) have not yet been tested

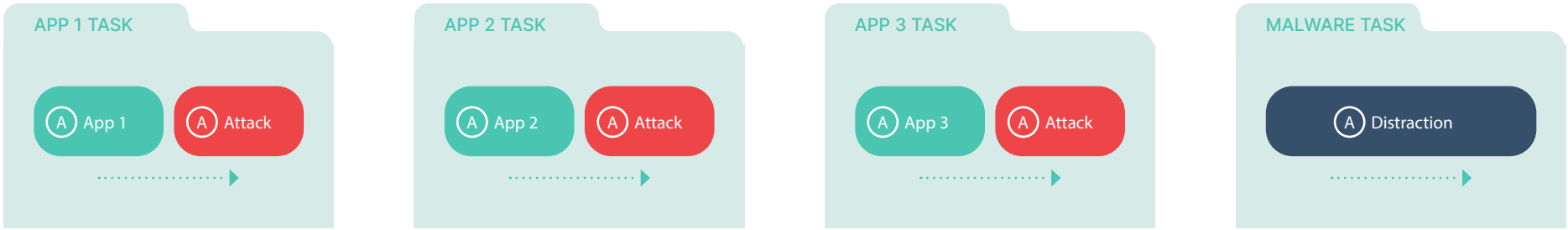
On Android 10, the attack is entirely ineffective, with the activities split into different tasks, and into separate task stacks according to analysis of the «adb shell dumpsys activity» activities.

The vulnerability explained (POC)

By implementing code, an attacker can replace the end-user view of almost any Android app by hijacking the tasks. This can be used to gain various platform permissions (by hijacking apps such as the SMS app, mail app, camera app, maps etc.), and stealing login credentials. The malicious app can hide its intention by obfuscation and reflection, making static analysis of the malicious app challenging.

The StrandHogg 2.0 vulnerability allows malware to exploit a peculiarity in startActivities(Intent[]). This call takes an Intent Array of activities to start. By ensuring that this array has a specific order, and certain elements are tagged with NEW_TASK, the malicious app gets free reign in dictating what these tasks should contain.

PROMON



NEW_TASK

What user will see when starting app

What user sees when malware launches attack

In the schematic above, we start three new tasks: App 1, 2 and 3. For each, we first inject the original launcher activity of the app we are targeting, then we put in our own attack activity.

The vulnerability allows us to place this attack activity into each of these tasks. With these tasks, the first activity gives the individual tasks all of its information and authority.

In the first task example for instance, this particular task will appear to be the original task belonging to the app. However, the second activity (the attack activity) which we placed into this task, is what the user will actually see when this task is activated.

As a result, the next time the app is invoked, for instance, by a user clicking its app icon, the Android OS will evaluate the existing tasks and find the task we created. Because it looks genuine to the app, it will bring the task we created to the foreground and with it our attack will now be activated.

Finally, in our illustration, we have mentioned a Distraction activity. The startActivities(Intent[]) that the malware needs to setup its attack is a call that will change the active activity (screen).

The last intent sent to startActivities is the one that the user will end up seeing, and all the other activities will be placed into the background, awaiting the next launch of their corresponding tasks. By placing a benign innocent-looking activity at the end, we can ensure that the end-user does not see anything suspicious and thus remains oblivious to all of the task-hijacking that just took place.

Mitigation recommendation

1. Android end-users should update their devices to the latest firmware as soon as possible in order to protect themselves against attacks utilizing StrandHogg 2.0.

2. To mitigate against StrandHogg 2.0 and other malware as an app developer, ensure you protect your app with appropriate security, such as In-App Protection by Promon SHIELD™ provides active protection mechanisms against StrandHogg (1.0) and StrandHogg 2.0. Promon SHIELD™ will monitor tasks in the process as they enter/are created and determine if they are malicious or not. Offending tasks will be blocked, and the app will be notified about the findings.

3. Customize your own: In order to mitigate the StrandHogg 2.0 attack, all public activities of the app, and especially the launch activity (the activity that contains action main, category launcher) must all be set to launchMode=”singleTask” OR launchMode=”singleInstance” in AndroidManifest.xml.

   Note, however, that these attributes are still vulnerable to StrandHogg (1.0) through taskAffinity combined with allowTaskReparenting. This will also have some side-effects on the app flow that

may be undesirable, as each app-click will now bring you back to the original launcher activity instead of the screen that you were on. To counteract this effect the tasks must be handled manually. Promon SHIELD™ on the other hand will handle this for you.

launchMode="<default>"

launchMode="singleInstance"

Prevents any other activity from entering this task

Timeline

Dec 4, 2019	–	Reported issue to Google
Dec 4, 2019	–	Shared a PoC «malicious app» and video with Google
Dec 4, 2019	–	Google confirmed receiving the report
Dec 9, 2019	–	Google set the severity of the finding as «Critical»
Dec 9, 2019	–	Google confirms that they are able to reproduce the issue
Feb 14, 2020	–	We inform Google the 90-day disclosure is nearing in the beginning of March, and ask for status on their side
Feb 14, 2020	–	Google responds that April is the soonest they can roll out a fix
Feb 14, 2020	–	We inform Google we are working on mitigations
Feb 14, 2020	–	Google responds. They are working on remediations, and ask if we can share what mitigations we are recommending
Feb 17, 2020	–	We inform Google that we can hold back the disclosure until April. We request the CVE number
Feb 17, 2020	–	We share our mitigation strategies, as well as how we envisage a platform mitigation
Mar 23, 2020	–	Google responds with the CVE ID (CVE-2020-0096)
Mar 23, 2020	–	Google responds that general availability of the fix for Android will be available in May
Mar 23, 2020	–	Google asks if we will consider delaying disclosure to May
Mar 27, 2020	–	We respond that we will delay disclosure until May
Apr 22, 2020	–	Google informs us that the May Security Bulletin is scheduled to contain a patch for the vulnerability