




FixedBugP2+ Add Hotlist

 STATUS UPDATE No update yet. Edit

 DESCRIPTION

ch...@gmail.com created issue [#1](#)

Jun 4, 2018 06:56PM



Experiencing issue when using App Bundles and Realm native libs.

See attached APK analyzer screenshot for bundled realm native libs in the APK

Shipping a bundle caused following issues:

...

Caused by com.getkeepsafe.relinker.MissingLibraryException
lib/armeabi/librealm-jni.so
com.getkeepsafe.relinker.ApkLibraryInstaller.a (Unknown Source)
com.getkeepsafe.relinker.ReLinkerInstance.b (Unknown Source)
com.getkeepsafe.relinker.ReLinkerInstance.a (Unknown Source)
com.getkeepsafe.relinker.ReLinker.a (Unknown Source)
io.realm.internal.RealmCore.a (Unknown Source)
io.realm.Realm.a (Unknown Source)
...

(Batmobile Defy Mini - Android 4.3.1)


and

...

Caused by com.getkeepsafe.relinker.MissingLibraryException
lib/x86/librealm-jni.so
com.getkeepsafe.relinker.ApkLibraryInstaller.a (Unknown Source)
com.getkeepsafe.relinker.ReLinkerInstance.b (Unknown Source)
com.getkeepsafe.relinker.ReLinkerInstance.a (Unknown Source)
com.getkeepsafe.relinker.ReLinker.a (Unknown Source)
io.realm.internal.RealmCore.a (Unknown Source)
io.realm.Realm.a (Unknown Source)
...

(Nexus 5X - Android 6.0.1)

As far as I can see those two abis are included in the APK but cause trouble in the Bundle.

 **nativelibs.png**
72 KB [View](#) [Download](#)

Reporter

Type

Priority

Severity

Status

Access

Assignee

Verifier

Collaborators

CC

AOSP ID

Blocking Release

Release Status

Found In

Targeted To

Verified In


In Prod

Show 1 addition

✓ Mentioned issues (1)


✓ Links (4)

Hide all

 Mentioned issues (1)

P2 By using android new Android App Bundle i have found Resource Not Found ["https://issuetracker.google.com/111233819"](https://issuetracker.google.com/111233819)

le...@ [#25](#)

 Links (4)

"<https://github.com/KeepSafe/ReLinker>"

an...@ [#9](#)

"<https://medium.com/keepsafe-engineering/the-perils-of-loading-native-libraries-on-a-...>"

an...@ [#9](#)

"<https://github.com/KeepSafe/ReLinker/blob/master/relinker/src/main/java/com/getkeepsafe/relinker/A...>"

le...@ [#11](#)


"The bug is tracked here on ReLinker's GitHub repo: <https://github.com/KeepSafe/ReLinker/issues/44>"

le...@ [#15](#)


COMMENTS

All comments


↓ Oldest first

 **cm...@google.com** <cm...@google.com> [#2](#)


Jun 4, 2018 10:59PM




Is it possible / likely that these were copied from another device rather than being installed by Play?

 **ch...@gmail.com** <ch...@gmail.com> [#3](#)


Jun 5, 2018 03:08PM



That may be the case. Cannot confirm though. Both devices are rooted according to crashlytics.
But would be a bit strange as it was a staged rollout and I guess there were around 1500 installs overall, so app was not "publicly" available

 **h4...@gmail.com** <h4...@gmail.com> [#4](#)

Jun 5, 2018 06:15PM



I have the same issue for 76 users. 27% of them are rooted, which is unusual high. It really could be that the users sideloaded a wrong apk. Is there a way for us to find out?

st...@gmail.com <st...@gmail.com> [#5](#)


Jun 15, 2018 06:56AM ⋮

I'm experiencing the same issue with Realm and Android App Bundle:

```
Caused by com.getkeepsafe.relinker.MissingLibraryException: lib/armeabi/librealm-jni.so
    at com.getkeepsafe.relinker.a.a(ApkLibraryInstaller.java:85)
    at com.getkeepsafe.relinker.c.c(ReLinkerInstance.java:180)
    at com.getkeepsafe.relinker.c.a(ReLinkerInstance.java:136)
    at com.getkeepsafe.relinker.b.a(ReLinker.java:70)
    at com.getkeepsafe.relinker.b.a(ReLinker.java:57)
    at io.realm.internal.l.a(RealmCore.java:60)
    at io.realm.v.a(Realm.java:258)
```

Strangely only on devices which use MediaTek chipsets like:

Quantum MUV PRO - MediaTek MT6753
Lenovo PHAB 2 Plus - MediaTek MT8783
Xperia XA - MediaTek MT6755 Helio P10
Sony Xperia XA1 Ultra Dual - MediaTek MT6757 Helio P20

 **Bildschirmfoto 2018-06-14 um 22.50.51.png**
293 KB [View](#) [Download](#)

dm...@remitly.com <dm...@remitly.com> [#6](#)

Jun 22, 2018 07:01AM ⋮

I have also just encountered a similar issue (with the React Native native lib in our case) while doing a staged rollout of our first ever app bundle-based release. If you to the "Android vitals" section of your Play console, and look at the Crashes tab, you can filter results by Play store installs vs apps installed from elsewhere. So far I am only seeing the native library crash for under the "Installed from elsewhere" category, which confirms the theory that these crashes come from APKs that were not installed through the official play store channel.

to...@blueapron.com <to...@blueapron.com> [#7](#)

Jun 27, 2018 12:00AM ⋮

FWIW, I looked at the same filter. On my app, I've seen 175 instances of this crash in the last seven days when looking over all installs, but 71 when filtering for "installed from Play". So while this is definitely more prevalent if the build is sideloaded, it still seems to happen on Play Store installs.

This seems to occur mostly on Android 8.0 - 97%+ of these crashes are from 8.0/8.1 devices.

dm...@remitly.com <dm...@remitly.com> [#8](#)

Jun 27, 2018 06:13AM ⋮

Your observation is correct - after letting our app roll out to a larger percentage of users, I have started seeing these crashes coming from Play installs - about 25% of total users I would say. Overall this is a little troubling. Maybe modded/non-standard Android OSes like Cyanogenmod are encountering this? I will try to reach out to Play's support team personally to get their opinion on this.

an...@google.com <an...@google.com> [#9](#)

Jun 27, 2018 07:37PM ⋮

Note: Realm uses ReLinker
<https://github.com/KeepSafe/ReLinker>
<https://medium.com/keepsafe-engineering/the-perils-of-loading-native-libraries-on-android-befa49dce2db>

christophwies@, Can you attach a sample?

ch...@gmail.com <ch...@gmail.com> [#10](#)

Jun 28, 2018 03:51PM ⋮

what do you mean with sample?
Stacktrace is already there. You mean the Bundle?

le...@google.com <le...@google.com> [#11](#)

Jul 4, 2018 03:28AM ⋮

I had a quick look at the ReLinker source code and I think I understand the issue now:

ReLinker loads the native library directly from the APK file, without using the OS' System.load() or System.loadLibrary() methods. The issue is that it searches for the native libraries in the base APK, whereas with the App Bundle, the native libraries are in config splits (i.e. ReLinker searches in base.apk whereas the native libraries are actually in base-x86.apk, base-armeabi.apk, etc.).

Source:
<https://github.com/KeepSafe/ReLinker/blob/master/relinker/src/main/java/com/getkeepsafe/relinker/ApkLibraryInstaller.java#L55>
Instead of searching in "sourceDir", ReLinker should be searching over the different directories under "splitSourceDirs".

ma...@gmail.com <ma...@gmail.com> [#12](#)


Jul 10, 2018 03:10AM ⋮

Got the same issue and here's what I've spotted
Such crashes are happening only on devices that in some cases have a x86 arch, and sometimes armeabi (armeabi-v7a)

So, I got crashes on
ASUS:
ZenFone 2(ZE551ML)
ZenPad C 7.0 (Z170C)

Samsung:
Galaxy S4
Galaxy Tab A 7.0
SM-G955F

All of these devices have look like this in Google Play Device Catalog

 **Screen Shot 2018-07-09 at 20.09.04.png**
45 KB [View](#) [Download](#)

ma...@gmail.com <ma...@gmail.com> [#13](#)


Jul 10, 2018 03:18AM ⋮

Well
When you come to an Artifact library, and try to download an APK for a specific device, here's what you got
ZE551ML - downloads APK with x86 ABI split - crashes only on armeabi
Z170C - same
dream2lte (Samsung Galaxy S8+) - downloads arm64 - crashes on armeabi
So you got an idea
I'm not sure how do the Google Play choose which APK to install on a particular user device, but it really does it wrong

ma...@gmail.com <ma...@gmail.com> [#14](#)

Jul 10, 2018 03:20AM ⋮

(Proof of concept)

 **Screen Shot 2018-07-09 at 20.19.06.png**
103 KB [View](#) [Download](#)

le...@google.com <le...@google.com> [#15](#)

Jul 10, 2018 06:38AM ⋮

The bug is tracked here on ReLinker's GitHub repo: <https://github.com/KeepSafe/ReLinker/issues/44>

ch...@gmail.com <ch...@gmail.com> [#16](#)

Nov 8, 2018 07:15PM ⋮

Relinker have fixed this issue and released a fix in Relinker 1.3.0, so I think this issue can be closed.

kb...@yelp.com <kb...@yelp.com> [#17](#)

Nov 14, 2018 10:10AM ⋮

This issue does not appear to be fixed by Relinker 1.3.0. We're still seeing this crash from Play Store installed app bundles when using Realm 5.7 (Relinker 1.3.0 was added in 5.6). According to the Play console it's about ~1000 crashes from Play Store installed app bundles, and Bugsnag is reporting about ~4000 crashes overall (not sure how many are side loads vs from the Play Store).

Sadly, our stacktrace doesn't mention what path it is trying to load:
Caused by: com.getkeepsafe.relinker.MissingLibraryException:
at com.getkeepsafe.relinker.ApkLibraryInstaller.installLibrary\$757ca6fb (ApkLibraryInstaller.java:123)
at com.getkeepsafe.relinker.ReLinkerInstance.loadLibraryInternal (ReLinkerInstance.java:180)
at com.getkeepsafe.relinker.ReLinkerInstance.loadLibrary\$11a583b2 (ReLinkerInstance.java:136)
at com.getkeepsafe.relinker.ReLinker.loadLibrary (ReLinker.java:70)
at com.getkeepsafe.relinker.ReLinker.loadLibrary (ReLinker.java:57)
at io.realm.internal.RealmCore.loadLibrary (RealmCore.java:60)
at io.realm.Realm.init (Realm.java:269)

Both Bugsnag and the Play console show a majority of affected devices are newer Samsung devices (S9, S9+, Note9, S8). Bugsnag also reports that ~90% of the crashes report an abi triplet of [arm64-v8a, armeabi-v7a, armeabi]. We have been unable to reproduce the issue on a Samsung S9 by downloading the bundle through the Play Store or using bundletool, but are planning on adding more logging to verify that the apk downloaded contains at least one of the abi types the device supports.

le...@google.com <le...@google.com> [#18](#)

Nov 14, 2018 10:15AM ⋮

Have you tried sideloading only the base-master.apk to see if the same stacktrace can be reproduced?
You can extract the base-master.apk from the APK Set generated by bundletool (using the "bundletool extract-apks" command for example).

kb...@yelp.com <kb...@yelp.com> [#19](#)

Nov 14, 2018 01:29PM ⋮

Just sideloading the base-master.apk on the S9 replicates the crash and stacktrace. Though this is expected is it not since the split native libraries would be missing?

le...@google.com <le...@google.com> [#20](#)

Nov 14, 2018 10:26PM ⋮

Yes, it's expected, but if it wasn't the same stacktrace, it may have indicated that the issue was not a split APK missing. Now, I'm more confident that it's a split APK not being installed and not an Android platform issue. This all indicates that those would be sideloaded APKs. We are working on ways to mitigate this issue,

but this is still work in progress.

kb...@yelp.com <kb...@yelp.com> [#21](#)

Nov 15, 2018 07:49AM ⋮

I think I'm a bit confused. Are you saying all the crashes are because of sideloads? What about the crashes the Play Console reports are from apps installed through the Play Store?

le...@google.com <le...@google.com> [#22](#)

Nov 15, 2018 09:49PM ⋮

Assigned to le...@google.com.

That's a good point, let me have a closer look.

le...@google.com <le...@google.com> [#23](#)

Nov 20, 2018 01:30AM ⋮

Marked as fixed.

The source of the install can unfortunately be faked quite easily with "adb" and it looks like a bunch of users have learnt to do that (for reasons that I'm not entirely sure of yet). We haven't received any end-user report either that an install from Play was failing for them. A few other developers are facing the same issue and none of them have been able to reproduce the issue either.

All of this seems to confirm that the issue is indeed sideloading (with for some users faking the source of the install) so I will close this bug for now.

If you have any end-user reporting that the install is failing when installing from Play store, feel free to re-open this bug providing the details.

Note that we are also actively working on a solution to avoid the crashes on the users having installed only a subset of the required APKs. This should help remove those crashes from your reports altogether. Stay tuned.

at...@bandlab.com <at...@bandlab.com> [#24](#)

Feb 21, 2019 04:26PM ⋮

We started seeing a similar issue in our app since we switched to app bundle: we have ~2k google play console crash reports with UnsatisfiedLinkError when the app tries to load our custom native library (we are not using Relinker).

80% of the crashes are from google play installs, though we still suspect it has to do with sideloading of our app.

Is there anything we can do, or any open issue to follow regarding this?

le...@google.com <le...@google.com> [#25](#)

Feb 21, 2019 09:15PM ⋮

You can track <https://issuetracker.google.com/issues/111233819>
We should have a solution ready soon.

vi...@gmail.com <vi...@gmail.com> [#26](#)

Feb 3, 2020 09:48PM ⋮

Similar issue com.getkeepsafe.relinker.MissingLibraryException: librealm-jni.so

le...@google.com <le...@google.com> [#27](#)

Feb 3, 2020 10:41PM ⋮

Re comment #26, unfortunately your comment does not provide enough information to help you.

he...@globant.com <he...@globant.com> [#28](#)

Sep 28, 2023 07:25AM ⋮

Hi I recently started getting missing link error on arm v7a architectures:

```
Exception java.lang.UnsatisfiedLinkError: dlopen failed: "/data/data/com.my.app/app_lib/librealm-jni.so.10.9.0" has bad ELF magic
at java.lang.Runtime.load0 (Runtime.java:928)
at java.lang.System.load (System.java:1633)
at com.getkeepsafe.relinker.SystemLibraryLoader.loadPath
at com.getkeepsafe.relinker.ReLinkerInstance.loadLibraryInternal
at com.getkeepsafe.relinker.ReLinkerInstance.loadLibrary
at com.getkeepsafe.relinker.ReLinker.loadLibrary
at com.getkeepsafe.relinker.ReLinker.loadLibrary
at io.realm.internal.RealmCore.loadLibrary
at io.realm.Realm.initializeRealm
at io.realm.Realm.init
```

I use android bundles and have been doing so for the past year with no problem and suddenly it started to crash on devices for my last publication. Do you know if this is a known issue or something related to Google SDK 33 Policies?