Crash in OatHeader::GetCompilerFilter

+1 25 Hotlists (8) Mark as Duplicate

Comments (38) Dependencies Duplicates (1) Blocking (0) Resources (8)

Fixed Bug P3 + [AOSP] assigned NeedsInfo

STATUS UPDATE No update yet. Edit

DESCRIPTION sb...@opera.com created issue #1

We're seeing a lot of crashes in the wild on Android 10 devices with the following backtrace:

```
backtrace:
  #00  pc 000000000005f2b6  /apex/com.android.runtime/lib/bionic/libc.so (abort+166)
  #01  pc 0000000000376049  /apex/com.android.runtime/lib/libart.so (art::Runtime::Abort(char const*)+1676)
  #02  pc 000000000000855f  /system/lib/libbase.so (android::base::LogMessage::~LogMessage()+406)
  #03  pc 00000000003240ff  /apex/com.android.runtime/lib/libart.so (art::OatHeader::GetCompilerFilter() const+170)
  #04  pc 0000000000329a99  /apex/com.android.runtime/lib/libart.so (art::OatFile::GetCompilerFilter() const+20)
  #05  pc 0000000000331e4b  /apex/com.android.runtime/lib/libart.so (art::OatFileManager::DumpForSigQuit(std::__1::basic_ostream<char, std::__1::char_traits<char>>&)+242)
  #06  pc 000000000037fe95  /apex/com.android.runtime/lib/libart.so (art::Runtime::DumpForSigQuit(std::__1::basic_ostream<char, std::__1::char_traits<char>>&)+68)
  #07  pc 000000000038e845  /apex/com.android.runtime/lib/libart.so (art::SignalCatcher::HandleSigQuit()+1036)
  #08  pc 000000000038dcaf  /apex/com.android.runtime/lib/libart.so (art::SignalCatcher::Run(void*)+242)
  #09  pc 00000000000a6093  /apex/com.android.runtime/lib/bionic/libc.so (__pthread_start(void*)+20)
  #10  pc 0000000000060763  /apex/com.android.runtime/lib/bionic/libc.so (__start_thread+30)
```

The report above is from Google Pixel (sailfish), 3840MB RAM, Android 10 device (according to Play Console).

We have not been able to reproduce this locally but the abort message coming from this is "Abort message: 'Check failed: key_value != nullptr compiler-filter not found in oat header'". Is there any this to manifest (in which case perhaps we can work around it somehow) or is this purely a system bug?

✓ Mentioned issues (2) ✓ Links (6)

Mentioned issues (2)

--	--	"cannot access b/143155012 , Is this CL for this issue? seems was."
P3	Missing answers to questions asked in https://issuetracker.google.com/issues/150633385	"https://issuetracker.google.com/181142263"

Links (6)

"For steps to capture a bug report, please refer: https://developer.android.com/studio/debug/bug-report#bugreportdevice"
"https://android.googlesource.com/platform/art/+/dee09f90d1b445c3a133c392a0afc03c…"
"Upload the full bug report file after the issue has occurred. Steps followed here: https://developer.android.com/studio/debug/bug-report"
"http://sample.package.name"
"https://android-review.googlesource.com/c/platform/art/…"
"http://com.google.android.gms.ads.nonagon.slot.reward… "

COMMENTS

ra...@google.com <ra...@google.com> #2

Assigned to ra...@google.com.

Thank you for reporting this issue. For us to further investigate this issue, please provide the following additional information:

Steps to reproduce
What steps are needed to reproduce this issue?

Android bug report (to be captured after reproducing the issue)
For steps to capture a bug report, please refer: https://developer.android.com/studio/debug/bug-report#bugreportdevice

Alternate method
Navigate to "Developer options", ensure "USB debugging" is enabled, then enable "Bug report shortcut". Capture bug report by holding the power button and selecting the "Take bug report" o

Note: Please place the files in a folder, upload it to google drive and share the folder to android-bugreport@google.com, then share the link here.

sb...@opera.com <sb...@opera.com> #3

As mentioned in the initial description we have not been able to reproduce this ourselves, the reports are coming in through crash reporting system, so unfortunately we can not provide the r hoping the call stack in itself might give you some information into why this could happen. Especially, in which situations would this call stack appear in the first place, and is there something

This is one of our top crashes so if you've not been getting similar reports from other directions then any hint into why it could be happening to a specific app would be very helpful.

sh...@amlogic.corp-partner.google.com <sh...@amlogic.corp-partner.google.com> #4

sh...@amlogic.corp-partner.google.com <sh...@amlogic.corp-partner.google.com> #4

Currently we run Netflix on a Q TV device, easy to repro this same issue(normal use the device, after several hours, can found tombstone file exactly as this).

li...@gmail.com <li...@gmail.com> #5

we found the same issue on our phone, also Android Q, please check the log

deleted
0 B ?

deleted
0 B ?

li...@gmail.com <li...@gmail.com> #6

It seems a bug of ART, because it happens on a lot of apps.

sh...@amlogic.corp-partner.google.com <sh...@amlogic.corp-partner.google.com> #7

cannot access b/143155012, Is this CL for this issue? seems was.
https://android.googlesource.com/platform/art/+/dee09f90d1b445c3a133c392a0afc03c62d75ef1%5E%21/#F0

sb...@opera.com <sb...@opera.com> #8

So this could potentially be triggered by using InMemoryDexClassLoader in the app?

ad...@google.com <ad...@google.com> #9

Marked as fixed.

The issue has been fixed and it will be available in a future build.

sb...@opera.com <sb...@opera.com> #10

With a future build I assume you mean a new Android version so would be nice to know if it truly is related to InMemoryDexClassLoader or something different since as an app developer I'd li
existing Android version(s) that have this problem. If it is specifically related to mentioned class usage then perhaps I can avoid using it on those Android versions.

sb...@opera.com <sb...@opera.com> #11

Anyone willing to answer the above question? As I mentioned there it is very valuable if you can share any information on how to work around this issue on Android 10 or at least provide con

vi...@gmail.com <vi...@gmail.com> #12

How to resolve it?

da...@gmail.com <da...@gmail.com> #13

Bump, we are still seeing these errors only on Android 10 devices

la...@phonepe.com <la...@phonepe.com> #14

Bump, any update here?

wa...@gmail.com <wa...@gmail.com> #15

1 #00 pc 00054156 /apex/com.android.runtime/lib/bionic/libc.so (abort+165) [armeabi-v7a]
2 #01 pc 0005413f /apex/com.android.runtime/lib/bionic/libc.so (abort+142) [armeabi-v7a]
3 java:
4 [Failed to get java stack]
we are still seeing these errors only on Android 10 devices ,How to resolve it?

ka...@gmail.com <ka...@gmail.com> #16

Still observing it on Android 10 and some of Android 9 devices. Is there any workaround for this issue? Over 32K occurrences of this crash in the last 30 days so it would be nice to fix it or at

ch...@hennge.com <ch...@hennge.com> #17

Seeing this crash in production as well. Is there a work around?

sb...@opera.com <sb...@opera.com> #18

Decided to open a new issue in an attempt to get some attention from the ART team since this report seems to be a dead end: https://issuetracker.google.com/issues/181142263

ja...@google.com <ja...@google.com> #19

Thank you comment #18 - to keep things streamlined we will update our internal product and engineering teams, under this issue, with your new inquiry. That way all the relevant history is ea

We've shared this with our product and engineering teams and will continue to provide updates as more information becomes available.

ja...@google.com <ja...@google.com> #20

In the mean time, would you please kindly provide the following to help us further investigate the issue with more accuracy? It looks like the above bug report info has since been removed.

- Upload the full bug report file after the issue has occurred. Steps followed here: https://developer.android.com/studio/debug/bug-report
- Steps to reproduce issue specific to the provided sample project: [Be as specific as possible]
  - 1
  - 2
  - 3
- Expected Results:
- Observed Results:

Note: Please upload to google drive and share the folder to android-bugreport@google.com, then share the link here.

The more information we have, the more accurately our product and engineering team can solve the issue. Thank you so much for your time and for your cooperation.

sb...@opera.com <sb...@opera.com> #21

All necessary information from our end is already available in this report, only thing missing is some answers from the ART team around any possible workarounds or root cause information

ag...@gmail.com <ag...@gmail.com> #22

```
*** *** *** *** *** *** *** *** *** *** *** *** *** *** *** ***
pid: 0, tid: 0 >>> sample.package.name <<<

backtrace:
  #00  pc 0000000000039228  /system/lib64/libandroidfw.so (android::Theme::~Theme()+40)
  #00  pc 0000000000149078  /system/lib64/libandroid_runtime.so (android::NativeThemeDestroy(_JNIEnv*, _jclass*, long)+24)
  #00  pc 00000000009957b8  /system/framework/arm64/boot-framework.oat (art_jni_trampoline+152)
  #00  pc 0000000000e4cb6c  /system/framework/arm64/boot-framework.oat (android.content.res.AssetManager.releaseTheme+76)
  #00  pc 0000000000a42990  /system/framework/arm64/boot-framework.oat (android.content.res.ResourcesImpl$ThemeImpl.finalize+64)
  #00  pc 00000000004f4e54  /system/framework/arm64/boot-core-libart.oat (java.lang.Daemons$FinalizerDaemon.doFinalize+100)
  #00  pc 00000000004f50cc  /system/framework/arm64/boot-core-libart.oat (java.lang.Daemons$FinalizerDaemon.runInternal+492)
  #00  pc 000000000033be10  /system/framework/arm64/boot-core-libart.oat (java.lang.Daemons$Daemon.run+160)
  #00  pc 00000000004afde8  /system/framework/arm64/boot.oat (java.lang.Thread.run+72)
  #00  pc 0000000000137334  /apex/com.android.runtime/lib64/libart.so (art_quick_invoke_stub+548)
  #00  pc 00000000001460ac  /apex/com.android.runtime/lib64/libart.so (art::ArtMethod::Invoke(art::Thread*, unsigned int*, unsigned int, art::JValue*, char const*)+244)
  #00  pc 00000000004b2768  /apex/com.android.runtime/lib64/libart.so (art::(anonymous namespace)::InvokeWithArgArray(art::ScopedObjectAccessAlreadyRunnable const&, art::ArtMetho
art::JValue*, char const*)+104)
  #00  pc 00000000004b3818  /apex/com.android.runtime/lib64/libart.so (art::InvokeVirtualOrInterfaceWithJValues(art::ScopedObjectAccessAlreadyRunnable const&, _jobject*, _jmethodID*
  #00  pc 00000000004f3d68  /apex/com.android.runtime/lib64/libart.so (art::Thread::CreateCallback(void*)+1176)
  #00  pc 00000000000e24a4  /apex/com.android.runtime/lib64/bionic/libc.so (__pthread_start(void*)+36)
  #00  pc 0000000000083d98  /apex/com.android.runtime/lib64/bionic/libc.so (__start_thread+64)
```

I am getting this on Android 10 ..... Any solution for this???

ja...@google.com <ja...@google.com> #23

Thanks comment #10

- Can you please provide the sample project and entire full bug report please?

ng...@google.com <ng...@google.com> #24

Note to Jaime: #22 is unrelated to this bug. agarwalanchal31@ can you file another bug for this?

Hi Simon, sorry it took so long for this bug to reach us and to answer your questions. I can confirm the issue is due to the use of ImMemoryDexClassLoader, and it happens mostly when an a issued an ANR.

The issue is fixed in Android 10 for Pixel, we are reaching to other partners to see if they can incorporate this change in their devices as well.

The best workaround I see for now is whether you could guard the use of InMemoryDexClassLoader to only be used in Android 11?

sb...@opera.com <sb...@opera.com> #25

Hi, thank you very much for confirming the root cause of this! We have tried to remove known usages of InMemoryDexClassLoader but are unfortunately still seeing these crashes. We're con
is being loaded dynamically (such as webview or play services) which in turn might use this class. But in the meantime I'd like to ask:
1. Is there any other known trigger (other code usage or similar) that might cause this issue?
2. Is there a chance an issue might persist across application updates somehow? That is, could there be some lingering optimized dex or similar still causing the problem even after releasing
usages removed?

ng...@google.com <ng...@google.com> #26

1. Is there any other known trigger (other code usage or similar) that might cause this issue?

ANR and bugreports are the only trigger I know of.

2. Is there a chance an issue might persist across application updates somehow? That is, could there be some lingering optimized dex or similar still causing the problem even after releasing
usages removed?

No, once the app updates, nothing from the previous version persists.

sb...@opera.com <sb...@opera.com> #27

Ok, so no other variants of dex class loading etc (other than InMemoryDexClassLoader) should be able to trigger this issue? (trying to figure out if we need to be looking for any other classes
InMemoryDexClassLoader)

ng...@google.com <ng...@google.com> #28

Correct, only InMemoryDexClassLoader will hit the issue.

ja...@guardsquare.com <ja...@guardsquare.com> #29

Hi,

Are there any other workarounds for this, other than not using InMemoryDexClassLoader?

Is this only a problem on Android 10? 8, 9, 11 are OK?

Is there a way to reproduce the issue easily?

Can you describe what the cause of the problem was and point to the fixes in the code?

Thanks

ng...@google.com <ng...@google.com> #30

> Are there any other workarounds for this, other than not using InMemoryDexClassLoader?

Unfortunately no

> Is this only a problem on Android 10? 8, 9, 11 are OK?

Yes, I believe only 10 has the issue.

> Is there a way to reproduce the issue easily?

You use the app for some time, restart it a few times. And then take a bugreport.

> Can you describe what the cause of the problem was and point to the fixes in the code?

The fix is here:
https://android-review.googlesource.com/c/platform/art/+/1149193

It's an ART internal problem.

ja...@guardsquare.com <ja...@guardsquare.com> #31

Is there a way to detect whether the device has this fix, rather than avoiding the use of InMemoryDexClassLoader on all Android 10 devices?

ng...@google.com <ng...@google.com> #32

#31: I don't think so - currently the fix is in Pixel devices, but unclear about other devices.

fr...@gmail.com <fr...@gmail.com> #33

Receiving a crash whenever I skip the video on my Unity Application only on Samsung Devices (both Android 10 & 11)

```
#00 pc 000000000004e750  /apex/com.android.runtime/lib64/bionic/libc.so (abort+164) (BuildId: 2b4aa488d83a04c0c5cd917657ecd514)
#01 pc 000000000000f314  /system/lib64/libdatasource.so (android::NuCachedSource2::readInternal(long, void*, unsigned long)+536) (BuildId: 575cb89760bdebd
#02 pc 000000000000ef64  /system/lib64/libdatasource.so (android::NuCachedSource2::onRead(android::sp<android::AMessage> const&)+244) (BuildId: 575cb8976(
#03 pc 000000000000e85c  /system/lib64/libdatasource.so (BuildId: 575cb89760bdebd2ad589e1af4c06c49)
#04 pc 0000000000018814  /system/lib64/libstagefright_foundation.so (android::AHandler::deliverMessage(android::sp<android::AMessage> const&)+84) (BuildIc
#05 pc 000000000001e9cc  /system/lib64/libstagefright_foundation.so (android::AMessage::deliver()+188) (BuildId: 684ca5fa699d8995be76b3f9bfc86bd2)
#06 pc 0000000000019b48  /system/lib64/libstagefright_foundation.so (android::ALooper::loop()+572) (BuildId: 684ca5fa699d8995be76b3f9bfc86bd2)
#07 pc 0000000000015594  /system/lib64/libutils.so (android::Thread::_threadLoop(void*)+460) (BuildId: 4e69b93bf70ed592f0029dbd1097529e)
#08 pc 00000000000a5648  /system/lib64/libandroid_runtime.so (android::AndroidRuntime::javaThreadShell(void*)+144) (BuildId: 5541b6408ddf6d7f1b39761e30471
#09 pc 0000000000014d90  /system/lib64/libutils.so (BuildId: 4e69b93bf70ed592f0029dbd1097529e)
#10 pc 00000000000b0228  /apex/com.android.runtime/lib64/bionic/libc.so (__pthread_start(void*)+64) (BuildId: 2b4aa488d83a04c0c5cd917657ecd514)
#11 pc 000000000005070c  /apex/com.android.runtime/lib64/bionic/libc.so (__start_thread+64) (BuildId: 2b4aa488d83a04c0c5cd917657ecd514)
```

Please feel free to reach out for more information about this. Or if anyone has a resolution for this problem it would be great help.. Thanks..

il...@gmail.com <il...@gmail.com> #34

Is there any fix for this? Issue still persist on Android 10, 11 devices

pu...@gmail.com <pu...@gmail.com> #35

What if we don't use InMemoryDexClassloader, including third party SDKs? Is there any other way to reproduce or any other cause? Help...

so...@gmail.com <so...@gmail.com> #36

I am receiving crash of this in Android 10 and 11 Only. Didn't find any solution for it. I have a game developed in Unity3d. using Latest SDK's .
backtrace:

```
  #00  pc 0000000000089cd0  /apex/com.android.runtime/lib64/bionic/libc.so (abort+164)
  #00  pc 0000000000566728  /apex/com.android.art/lib64/libart.so (art::Runtime::Abort(char const*)+2308)
  #00  pc 0000000000013978  /system/lib64/libbase.so (android::base::SetAborter(std::__1::function<void (char const*)>&&)::$_3::__invoke(char const*)+76)
  #00  pc 0000000000012fa4  /system/lib64/libbase.so (android::base::LogMessage::~LogMessage()+320)
  #00  pc 00000000005bb568  /apex/com.android.art/lib64/libart.so (art::Thread::AssertNoPendingException() const+1808)
  #00  pc 00000000001d26cc  /apex/com.android.art/lib64/libart.so (art::ClassLinker::FindClass(art::Thread*, char const*, art::Handle<art::mirror::ClassLoader>)+64)
  #00  pc 000000000039c094  /apex/com.android.art/lib64/libart.so (art::JNI<false>::FindClass(_JNIEnv*, char const*)+1048)
  #00  pc 00000000003a593c  /data/app/~~j9mDWQWCzLzoLHW_BE4LRA==/com.twizar.chef.cooking.city.mad.crazy.game-4nlR5DAq2ZI--AgN4nohvg==/lib/arm64/libunity.so
  #00  pc 00000000003d04a8  /data/app/~~j9mDWQWCzLzoLHW_BE4LRA==/com.twizar.chef.cooking.city.mad.crazy.game-4nlR5DAq2ZI--AgN4nohvg==/lib/arm64/libunity.so
  #00  pc 00000000000c770c  /data/app/~~j9mDWQWCzLzoLHW_BE4LRA==/com.twizar.chef.cooking.city.mad.crazy.game-4nlR5DAq2ZI--AgN4nohvg==/oat/arm64/base.odex (art_jni_tram
  #00  pc 00000000001faadc  /data/app/~~j9mDWQWCzLzoLHW_BE4LRA==/com.twizar.chef.cooking.city.mad.crazy.game-4nlR5DAq2ZI--AgN4nohvg==/oat/arm64/base.odex
(com.unity3d.player.UnityPlayer$e$1.handleMessage+620)
  #00  pc 00000000006b6698  /system/framework/arm64/boot-framework.oat (android.os.Handler.dispatchMessage+136)
  #00  pc 00000000006b9bbc  /system/framework/arm64/boot-framework.oat (android.os.Looper.loop+1516)
  #00  pc 00000000001fb718  /data/app/~~j9mDWQWCzLzoLHW_BE4LRA==/com.twizar.chef.cooking.city.mad.crazy.game-4nlR5DAq2ZI--AgN4nohvg==/oat/arm64/base.odex (com.unity3c
  #00  pc 0000000000134564  /apex/com.android.art/lib64/libart.so (art_quick_invoke_stub+548)
  #00  pc 00000000001a9a78  /apex/com.android.art/lib64/libart.so (art::ArtMethod::Invoke(art::Thread*, unsigned int*, unsigned int, art::JValue*, char const*)+200)
  #00  pc 000000000055e278  /apex/com.android.art/lib64/libart.so (art::JValue art::InvokeVirtualOrInterfaceWithJValues<art::ArtMethod*>(art::ScopedObjectAccessAlreadyRunnable const
const*)+460)
  #00  pc 00000000005ada3c  /apex/com.android.art/lib64/libart.so (art::Thread::CreateCallback(void*)+1308)
  #00  pc 00000000000eb7a8  /apex/com.android.runtime/lib64/bionic/libc.so (__pthread_start(void*)+64)
  #00  pc 000000000008bc8c  /apex/com.android.runtime/lib64/bionic/libc.so (__start_thread+64)
```

yr...@gmail.com <yr...@gmail.com> #37

I have same issue without using Facebook SDK integration

Getting this kind of Application not respond error:

```
#00 pc 0x000000000004b20c /apex/com.android.runtime/lib64/bionic/libc.so (syscall+28)
#1 pc 0x00000000001ad8cc /apex/com.android.art/lib64/libart.so (art::Mutex::ExclusiveLock(art::Thread*)+400)
#2 pc 0x000000000055218c /apex/com.android.art/lib64/libart.so (art::Runtime::Abort(char const*)+712)
#3 pc 0x0000000000013990 /system/lib64/libbase.so (android::base::SetAborter(std::__1::function<void (char const*)>&&)::$_3::__invoke(char const*)+76)
#4 pc 0x0000000000012fb4 /system/lib64/libbase.so (android::base::LogMessage::~LogMessage()+320)
#5 pc 0x000000000038410c /apex/com.android.art/lib64/libart.so (art::JavaVMExt::AddGlobalRef(art::Thread*, art::ObjPtrart: 🪞 :Object)+268)
#6 pc 0x000000000039671c /apex/com.android.art/lib64/libart.so (art::JNI::NewGlobalRef(_JNIEnv*, _jobject*)+632)
#7 pc 0x0000000000122ebc /system/lib64/libandroid_runtime.so (JavaBBinder::JavaBBinder(_JNIEnv*, _jobject*)+152)
#8 pc 0x0000000000121d10 /system/lib64/libandroid_runtime.so (JavaBBinderHolder::get(_JNIEnv*, _jobject*)+116)
#9 pc 0x0000000000121be0 /system/lib64/libandroid_runtime.so (android::ibinderForJavaObject(_JNIEnv*, _jobject*)+116)
#10 pc 0x00000000001127e8 /system/lib64/libandroid_runtime.so (android::android_os_Parcel_writeStrongBinder(_JNIEnv, _jclass, long, _jobject*)+56)
#11 pc 0x0000000000204834 /system/framework/arm64/boot-framework.oat (art_jni_trampoline+180)
#12 pc 0x0000000002054714 /memfd:jit-cache (com.google.android.gms.ads.internal.client.d.writeToParcel+836)
#13 pc 0x00000000020f76e4 /memfd:jit-cache (com.google.android.gms.ads.nonagon.slot.rewarded.c.a+244)
#14 pc 0x00000000020df7cc /memfd:jit-cache (com.google.android.gms.ads.nonagon.slot.common.p.a+108)
#15 pc 0x0000000002027448 /memfd:jit-cache (com.google.android.gms.ads.nonagon.slot.rewarded.w.bo+120)
#16 pc 0x000000000202a314 /memfd:jit-cache (com.google.android.gms.ads.internal.util.future.c.execute+196)
#17 pc 0x00000000020584e0 /memfd:jit-cache (com.google.android.gms.ads.nonagon.slot.rewarded.ac.a+832)
#18 pc 0x00000000020390f0 /memfd:jit-cache (bru.run+704)
#19 pc 0x000000000064223c /system/framework/arm64/boot-framework.oat (android.os.Handler.dispatchMessage+76)
#20 pc 0x0000000002089340 /memfd:jit-cache (com.google.android.gms.ads.internal.util.f.a+48)
#21 pc 0x000000000205eb80 /memfd:jit-cache (atp.dispatchMessage+160)
#22 pc 0x000000000064579c /system/framework/arm64/boot-framework.oat (android.os.Looper.loop+1516)
#23 pc 0x0000000000404d58 /system/framework/arm64/boot-framework.oat (android.app.ActivityThread.main+1672)
```

```
#24 pc 0x00000000001337e8 /apex/com.android.art/lib64/libart.so (art_quick_invoke_static_stub+568)
#25 pc 0x00000000001a8a88 /apex/com.android.art/lib64/libart.so (art::ArtMethod::Invoke(art::Thread, unsigned int, unsigned int, art::JValue*, char const*)+228)
#26 pc 0x000000000054aefc /apex/com.android.art/lib64/libart.so (art::InvokeMethod(art::ScopedObjectAccessAlreadyRunnable const&, jobject*, jobject, _jobject, unsigned long)+1364)
#27 pc 0x00000000004cbef0 /apex/com.android.art/lib64/libart.so (art::Method_invoke(_JNIEnv, _jobject, jobject*, jobjectArray*)+52)
#28 pc 0x00000000000896f4 /apex/com.android.art/javalib/arm64/boot.oat (art_jni_trampoline+180)
#29 pc 0x0000000000897b28 /system/framework/arm64/boot-framework.oat (com.android.internal.os.RuntimeInit$MethodAndArgsCaller.run+136)
#30 pc 0x00000000008a0258 /system/framework/arm64/boot-framework.oat (com.android.internal.os.ZygoteInit.main+2280)
#31 pc 0x00000000001337e8 /apex/com.android.art/lib64/libart.so (art_quick_invoke_static_stub+568)
#32 pc 0x00000000001a8a88 /apex/com.android.art/lib64/libart.so (art::ArtMethod::Invoke(art::Thread, unsigned int, unsigned int, art::JValue*, char const*)+228)
#33 pc 0x0000000000549930 /apex/com.android.art/lib64/libart.so (art::JValue art::InvokeWithVarArgsart::ArtMethod*(art::ScopedObjectAccessAlreadyRunnable const&, _jobject, art::ArtM
#34 pc 0x0000000000549dec /apex/com.android.art/lib64/libart.so (art::JValue art::InvokeWithVarArgs<_jmethodID*>(art::ScopedObjectAccessAlreadyRunnable const&, jobject*, jmethodID
#35 pc 0x00000000004327d0 /apex/com.android.art/lib64/libart.so (art::JNI::CallStaticVoidMethodV(_JNIEnv, _jclass, _jmethodID*, std::__va_list)+636)
#36 pc 0x0000000000099428 /system/lib64/libandroid_runtime.so (JNIEnv::CallStaticVoidMethod(_jclass*, jmethodID*, ...)+124)
#37 pc 0x00000000000a0880 /system/lib64/libandroid_runtime.so (android::AndroidRuntime::start(char const*, android::Vectorandroid::String8 const&, bool)+836)
#38 pc 0x0000000000003574 /system/bin/app_process64 (main+1324)
#39 pc 0x000000000004973c /apex/com.android.runtime/lib64/bionic/libc.so (__libc_init+108)

"AdWorker(Defaul" tid=3668 Native
#00 pc 0x000000000004b20c /apex/com.android.runtime/lib64/bionic/libc.so (syscall+28)
#1 pc 0x00000000001ad8cc /apex/com.android.art/lib64/libart.so (art::Mutex::ExclusiveLock(art::Thread*)+400)
#2 pc 0x000000000055218c /apex/com.android.art/lib64/libart.so (art::Runtime::Abort(char const*)+712)
#3 pc 0x0000000000013990 /system/lib64/libbase.so (android::base::SetAborter(std::__1::function<void (char const*)>&&)::$_3::__invoke(char const*)+76)
#4 pc 0x0000000000012fb4 /system/lib64/libbase.so (android::base::LogMessage::~LogMessage()+320)
#5 pc 0x000000000038410c /apex/com.android.art/lib64/libart.so (art::JavaVMExt::AddGlobalRef(art::Thread*, art::ObjPtrart: 🪞 :Object)+268)
#6 pc 0x000000000039671c /apex/com.android.art/lib64/libart.so (art::JNI::NewGlobalRef(_JNIEnv*, _jobject*)+632)
#7 pc 0x0000000000034c38 /data/data/com.appinvent.MysteryPlaceHiddenObject/app_pccache/5/7AF31A62EAA77EE667FAF8E832B8CDDEF880916B/pcam.jar

"AdWorker(Defaul" tid=4592 Native
#00 pc 0x000000000004b20c /apex/com.android.runtime/lib64/bionic/libc.so (syscall+28)
#1 pc 0x0000000000597198 /apex/com.android.art/lib64/libart.so (art::Thread::Park(bool, long)+444)
#2 pc 0x00000000004d288c /apex/com.android.art/lib64/libart.so (art::Unsafe_park(_JNIEnv, _jobject, unsigned char, long)+640)
#3 pc 0x00000000000837bc /apex/com.android.art/javalib/arm64/boot.oat (art_jni_trampoline+140)
#4 pc 0x00000000001b4b9c /apex/com.android.art/javalib/arm64/boot.oat (java.util.concurrent.SynchronousQueue$TransferStack.awaitFulfill+684)
#5 pc 0x00000000001b57c0 /apex/com.android.art/javalib/arm64/boot.oat (java.util.concurrent.SynchronousQueue$TransferStack.transfer+1744)
#6 pc 0x00000000002dbca0 /apex/com.android.art/javalib/arm64/boot.oat (java.util.concurrent.SynchronousQueue.take+80)
#7 pc 0x000000000029baa4 /apex/com.android.art/javalib/arm64/boot.oat (java.util.concurrent.ThreadPoolExecutor.getTask+468)
#8 pc 0x000000000029cfec /apex/com.android.art/javalib/arm64/boot.oat (java.util.concurrent.ThreadPoolExecutor.runWorker+236)
#9 pc 0x000000000029a4d0 /apex/com.android.art/javalib/arm64/boot.oat (java.util.concurrent.ThreadPoolExecutor$Worker.run+64)
#10 pc 0x000000000015aaf8 /apex/com.android.art/javalib/arm64/boot.oat (java.lang.Thread.run+72)
#11 pc 0x0000000000133564 /apex/com.android.art/lib64/libart.so (art_quick_invoke_stub+548)
#12 pc 0x00000000001a8a6c /apex/com.android.art/lib64/libart.so (art::ArtMethod::Invoke(art::Thread, unsigned int, unsigned int, art::JValue*, char const*)+200)
#13 pc 0x000000000054a470 /apex/com.android.art/lib64/libart.so (art::JValue art::InvokeVirtualOrInterfaceWithJValuesart::ArtMethod*(art::ScopedObjectAccessAlreadyRunnable const&,
#14 pc 0x0000000000598bcc /apex/com.android.art/lib64/libart.so (art::Thread::CreateCallback(void*)+1288)
#15 pc 0x00000000000af97c /apex/com.android.runtime/lib64/bionic/libc.so (__pthread_start(void*)+64)
#16 pc 0x00000000000500d0 /apex/com.android.runtime/lib64/bionic/libc.so (__start_thread+64)

"Thread-262481" tid=5292 Native
#00 pc 0x000000000009b2c4 /apex/com.android.runtime/lib64/bionic/libc.so (nanosleep+4)
#1 pc 0x0000000000262edc /data/app/~~uf_mEpcw9jMmIg-Cr-9JAg==/com.appinvent.MysteryPlaceHiddenObject-bes5BO6-7Ci4bJFptgOJlw==/lib/arm64/libunity.so
#2 pc 0x00000000002f9334 /data/app/~~uf_mEpcw9jMmIg-Cr-9JAg==/com.appinvent.MysteryPlaceHiddenObject-bes5BO6-7Ci4bJFptgOJlw==/lib/arm64/libunity.so
#3 pc 0x0000000000000628 [vdso] (__kernel_rt_sigreturn)
#4 pc 0x000000000004e174 /apex/com.android.runtime/lib64/bionic/libc.so (abort+164)
#5 pc 0x0000000000552798 /apex/com.android.art/lib64/libart.so (art::Runtime::Abort(char const*)+2260)
#6 pc 0x0000000000013990 /system/lib64/libbase.so (android::base::SetAborter(std::__1::function<void (char const*)>&&)::$_3::__invoke(char const*)+76)
#7 pc 0x0000000000012fb4 /system/lib64/libbase.so (android::base::LogMessage::~LogMessage()+320)
#8 pc 0x000000000038410c /apex/com.android.art/lib64/libart.so (art::JavaVMExt::AddGlobalRef(art::Thread*, art::ObjPtrart: 🪞 :Object)+268)
#9 pc 0x000000000039671c /apex/com.android.art/lib64/libart.so (art::JNI::NewGlobalRef(_JNIEnv*, _jobject*)+632)
#10 pc 0x00000000000e6d58 /data/app/~~uf_mEpcw9jMmIg-Cr-9JAg==/com.appinvent.MysteryPlaceHiddenObject-bes5BO6-7Ci4bJFptgOJlw==/lib/arm64/libunity.so
#11 pc 0x000000000108cd1c /data/app/~~uf_mEpcw9jMmIg-Cr-9JAg==/com.appinvent.MysteryPlaceHiddenObject-bes5BO6-7Ci4bJFptgOJlw==/lib/arm64/libil2cpp.so
#12 pc 0x0000000001090c70 /data/app/~~uf_mEpcw9jMmIg-Cr-9JAg==/com.appinvent.MysteryPlaceHiddenObject-bes5BO6-7Ci4bJFptgOJlw==/lib/arm64/libil2cpp.so
#13 pc 0x000000000108f6dc /data/app/~~uf_mEpcw9jMmIg-Cr-9JAg==/com.appinvent.MysteryPlaceHiddenObject-bes5BO6-7Ci4bJFptgOJlw==/lib/arm64/libil2cpp.so
#14 pc 0x00000000006fbcd4 /data/app/~~uf_mEpcw9jMmIg-Cr-9JAg==/com.appinvent.MysteryPlaceHiddenObject-bes5BO6-7Ci4bJFptgOJlw==/lib/arm64/libil2cpp.so
#15 pc 0x0000000000cea9e8 /data/app/~~uf_mEpcw9jMmIg-Cr-9JAg==/com.appinvent.MysteryPlaceHiddenObject-bes5BO6-7Ci4bJFptgOJlw==/lib/arm64/libil2cpp.so
#16 pc 0x0000000000cf01bc /data/app/~~uf_mEpcw9jMmIg-Cr-9JAg==/com.appinvent.MysteryPlaceHiddenObject-bes5BO6-7Ci4bJFptgOJlw==/lib/arm64/libil2cpp.so
#17 pc 0x0000000000561490 /data/app/~~uf_mEpcw9jMmIg-Cr-9JAg==/com.appinvent.MysteryPlaceHiddenObject-bes5BO6-7Ci4bJFptgOJlw==/lib/arm64/libil2cpp.so
#18 pc 0x00000000005613b4 /data/app/~~uf_mEpcw9jMmIg-Cr-9JAg==/com.appinvent.MysteryPlaceHiddenObject-bes5BO6-7Ci4bJFptgOJlw==/lib/arm64/libil2cpp.so
#19 pc 0x0000000000561840 /data/app/~~uf_mEpcw9jMmIg-Cr-9JAg==/com.appinvent.MysteryPlaceHiddenObject-bes5BO6-7Ci4bJFptgOJlw==/lib/arm64/libil2cpp.so
#20 pc 0x00000000005d502c /data/app/~~uf_mEpcw9jMmIg-Cr-9JAg==/com.appinvent.MysteryPlaceHiddenObject-bes5BO6-7Ci4bJFptgOJlw==/lib/arm64/libil2cpp.so
#21 pc 0x0000000000da0648 /data/app/~~uf_mEpcw9jMmIg-Cr-9JAg==/com.appinvent.MysteryPlaceHiddenObject-bes5BO6-7Ci4bJFptgOJlw==/lib/arm64/libil2cpp.so
#22 pc 0x000000000108d3a8 /data/app/~~uf_mEpcw9jMmIg-Cr-9JAg==/com.appinvent.MysteryPlaceHiddenObject-bes5BO6-7Ci4bJFptgOJlw==/lib/arm64/libil2cpp.so
#23 pc 0x0000000001092410 /data/app/~~uf_mEpcw9jMmIg-Cr-9JAg==/com.appinvent.MysteryPlaceHiddenObject-bes5BO6-7Ci4bJFptgOJlw==/lib/arm64/libil2cpp.so
#24 pc 0x00000000004b755c /data/app/~~uf_mEpcw9jMmIg-Cr-9JAg==/com.appinvent.MysteryPlaceHiddenObject-bes5BO6-7Ci4bJFptgOJlw==/lib/arm64/libil2cpp.so
#25 pc 0x0000000000561514 /data/app/~~uf_mEpcw9jMmIg-Cr-9JAg==/com.appinvent.MysteryPlaceHiddenObject-bes5BO6-7Ci4bJFptgOJlw==/lib/arm64/libil2cpp.so
#26 pc 0x00000000005613b4 /data/app/~~uf_mEpcw9jMmIg-Cr-9JAg==/com.appinvent.MysteryPlaceHiddenObject-bes5BO6-7Ci4bJFptgOJlw==/lib/arm64/libil2cpp.so
#27 pc 0x0000000000288834 /data/app/~~uf_mEpcw9jMmIg-Cr-9JAg==/com.appinvent.MysteryPlaceHiddenObject-bes5BO6-7Ci4bJFptgOJlw==/lib/arm64/libunity.so
#28 pc 0x00000000002965d0 /data/app/~~uf_mEpcw9jMmIg-Cr-9JAg==/com.appinvent.MysteryPlaceHiddenObject-bes5BO6-7Ci4bJFptgOJlw==/lib/arm64/libunity.so
#29 pc 0x00000000000e6260 /data/app/~~uf_mEpcw9jMmIg-Cr-9JAg==/com.appinvent.MysteryPlaceHiddenObject-bes5BO6-7Ci4bJFptgOJlw==/lib/arm64/libunity.so
#30 pc 0x0000000000034aa0 /data/app/~~uf_mEpcw9jMmIg-Cr-9JAg==/com.appinvent.MysteryPlaceHiddenObject-bes5BO6-7Ci4bJFptgOJlw==/oat/arm64/base.odex (art_jni_trampoline
#31 pc 0x00000000020dbb38 /memfd:jit-cache (com.unity3d.player.ReflectionHelper$1.invoke+216)
#32 pc 0x000000000016559c /apex/com.android.art/javalib/arm64/boot.oat (java.lang.reflect.Proxy.invoke+92)
#33 pc 0x00000000001337e8 /apex/com.android.art/lib64/libart.so (art_quick_invoke_static_stub+568)
#34 pc 0x00000000001a8a88 /apex/com.android.art/lib64/libart.so (art::ArtMethod::Invoke(art::Thread, unsigned int, unsigned int, art::JValue*, char const*)+228)
#35 pc 0x0000000000549fd0 /apex/com.android.art/lib64/libart.so (art::JValue art::InvokeWithJValuesart::ArtMethod*(art::ScopedObjectAccessAlreadyRunnable const&, _jobject, art::ArtMe
#36 pc 0x00000000003c0e10 /apex/com.android.art/lib64/libart.so (art::JNI::CallStaticObjectMethodA(_JNIEnv, _jclass, _jmethodID*, jvalue const*)+624)
#37 pc 0x0000000000620f30 /apex/com.android.art/lib64/libart.so (art::InvokeProxyInvocationHandler(art::ScopedObjectAccessAlreadyRunnable&, char const, _jobject, _jobject*, std::__1::v
#38 pc 0x000000000065f6c8 /apex/com.android.art/lib64/libart.so (artQuickProxyInvokeHandler+1212)
```

```
#39 pc 0x000000000013ca5c /apex/com.android.art/lib64/libart.so (art_quick_proxy_invoke_handler+76)
#40 pc 0x00000000020fd4cc /memfd:jit-cache (com.google.unity.ads.Interstitial$1$1$4.run+124)
#41 pc 0x000000000015aaf8 /apex/com.android.art/javalib/arm64/boot.oat (java.lang.Thread.run+72)
#42 pc 0x0000000000133564 /apex/com.android.art/lib64/libart.so (art_quick_invoke_stub+548)
#43 pc 0x00000000001a8a6c /apex/com.android.art/lib64/libart.so (art::ArtMethod::Invoke(art::Thread, unsigned int, unsigned int, art::JValue*, char const*)+200)
#44 pc 0x000000000054a470 /apex/com.android.art/lib64/libart.so (art::JValue art::InvokeVirtualOrInterfaceWithJValuesart::ArtMethod*(art::ScopedObjectAccessAlreadyRunnable const&,
#45 pc 0x0000000000598bcc /apex/com.android.art/lib64/libart.so (art::Thread::CreateCallback(void*)+1288)
#46 pc 0x00000000000af97c /apex/com.android.runtime/lib64/bionic/libc.so (__pthread_start(void*)+64)
#47 pc 0x00000000000500d0 /apex/com.android.runtime/lib64/bionic/libc.so (__start_thread+64)

"Signal Catcher" tid=20184 Native
#00 pc 0x000000000004b20c /apex/com.android.runtime/lib64/bionic/libc.so (syscall+28)
#1 pc 0x00000000005b89bc /apex/com.android.art/lib64/libart.so (art::ThreadList::SuspendAllInternal(art::Thread, art::Thread, art::Thread*, art::SuspendReason)+756)
#2 pc 0x00000000005b8ec4 /apex/com.android.art/lib64/libart.so (art::ThreadList::SuspendAll(char const*, bool)+204)
#3 pc 0x0000000000299520 /apex/com.android.art/lib64/libart.so (art::gc::Heap::GetObjectsAllocated() const+500)
#4 pc 0x00000000002a4488 /apex/com.android.art/lib64/libart.so (art::gc::Heap::DumpForSigQuit(std::__1::basic_ostream<char, std::__1::char_traits >&)+312)
#5 pc 0x0000000000562998 /apex/com.android.art/lib64/libart.so (art::Runtime::DumpForSigQuit(std::__1::basic_ostream<char, std::__1::char_traits >&)+92)
#6 pc 0x00000000005776c4 /apex/com.android.art/lib64/libart.so (art::SignalCatcher::HandleSigQuit()+1396)
#7 pc 0x00000000005766b8 /apex/com.android.art/lib64/libart.so (art::SignalCatcher::Run(void*)+348)
#8 pc 0x00000000000af97c /apex/com.android.runtime/lib64/bionic/libc.so (__pthread_start(void*)+64)
#9 pc 0x00000000000500d0 /apex/com.android.runtime/lib64/bionic/libc.so (__start_thread+64)

"Jit thread pool" tid=20185 Native
#00 pc 0x000000000004b20c /apex/com.android.runtime/lib64/bionic/libc.so (syscall+28)
#1 pc 0x00000000001af840 /apex/com.android.art/lib64/libart.so (art::ConditionVariable::WaitHoldingLocks(art::Thread*)+136)
#2 pc 0x00000000005bf24c /apex/com.android.art/lib64/libart.so (art::ThreadPool::GetTask(art::Thread*)+120)
#3 pc 0x00000000005be4f8 /apex/com.android.art/lib64/libart.so (art::ThreadPoolWorker::Run()+144)
#4 pc 0x00000000005bdfc8 /apex/com.android.art/lib64/libart.so (art::ThreadPoolWorker::Callback(void*)+192)
#5 pc 0x00000000000af97c /apex/com.android.runtime/lib64/bionic/libc.so (__pthread_start(void*)+64)
#6 pc 0x00000000000500d0 /apex/com.android.runtime/lib64/bionic/libc.so (__start_thread+64)

"HeapTaskDaemon" tid=20186 Native
#00 pc 0x000000000004b20c /apex/com.android.runtime/lib64/bionic/libc.so (syscall+28)
#1 pc 0x00000000001af840 /apex/com.android.art/lib64/libart.so (art::ConditionVariable::WaitHoldingLocks(art::Thread*)+136)
#2 pc 0x00000000002e3658 /apex/com.android.art/lib64/libart.so (art::gc::TaskProcessor::GetTask(art::Thread*)+532)
#3 pc 0x00000000002e3f74 /apex/com.android.art/lib64/libart.so (art::gc::TaskProcessor::RunAllTasks(art::Thread*)+92)
#4 pc 0x000000000001260c /apex/com.android.art/javalib/arm64/boot-core-libart.oat (art_jni_trampoline+124)
#5 pc 0x0000000000058844 /apex/com.android.art/javalib/arm64/boot-core-libart.oat (java.lang.Daemons$HeapTaskDaemon.runInternal+196)
#6 pc 0x0000000000024b90 /apex/com.android.art/javalib/arm64/boot-core-libart.oat (java.lang.Daemons$Daemon.run+160)
#7 pc 0x000000000015aaf8 /apex/com.android.art/javalib/arm64/boot.oat (java.lang.Thread.run+72)
#8 pc 0x0000000000133564 /apex/com.android.art/lib64/libart.so (art_quick_invoke_stub+548)
#9 pc 0x00000000001a8a6c /apex/com.android.art/lib64/libart.so (art::ArtMethod::Invoke(art::Thread, unsigned int, unsigned int, art::JValue*, char const*)+200)
#10 pc 0x000000000054a470 /apex/com.android.art/lib64/libart.so (art::JValue art::InvokeVirtualOrInterfaceWithJValuesart::ArtMethod*(art::ScopedObjectAccessAlreadyRunnable const&,
#11 pc 0x0000000000598bcc /apex/com.android.art/lib64/libart.so (art::Thread::CreateCallback(void*)+1288)
#12 pc 0x00000000000af97c /apex/com.android.runtime/lib64/bionic/libc.so (__pthread_start(void*)+64)
#13 pc 0x00000000000500d0 /apex/com.android.runtime/lib64/bionic/libc.so (__start_thread+64)

"ReferenceQueueD" tid=20187 Native
#00 pc 0x000000000004b20c /apex/com.android.runtime/lib64/bionic/libc.so (syscall+28)
#1 pc 0x00000000001af840 /apex/com.android.art/lib64/libart.so (art::ConditionVariable::WaitHoldingLocks(art::Thread*)+136)
#2 pc 0x000000000049190c /apex/com.android.art/lib64/libart.so (art::Monitor::Wait(art::Thread*, long, int, bool, art::ThreadState)+532)
#3 pc 0x0000000000493108 /apex/com.android.art/lib64/libart.so (art::Monitor::Wait(art::Thread*, art::ObjPtrart: 🪞 :Object, long, int, bool, art::ThreadState)+280)
#4 pc 0x00000000000803fc /apex/com.android.art/javalib/arm64/boot.oat (art_jni_trampoline+140)
#5 pc 0x000000000008fca0 /apex/com.android.art/javalib/arm64/boot.oat (java.lang.Object.wait+48)
#6 pc 0x00000000000588e0 /apex/com.android.art/javalib/arm64/boot-core-libart.oat (java.lang.Daemons$ReferenceQueueDaemon.runInternal+112)
#7 pc 0x0000000000024b90 /apex/com.android.art/javalib/arm64/boot-core-libart.oat (java.lang.Daemons$Daemon.run+160)
#8 pc 0x000000000015aaf8 /apex/com.android.art/javalib/arm64/boot.oat (java.lang.Thread.run+72)
#9 pc 0x0000000000133564 /apex/com.android.art/lib64/libart.so (art_quick_invoke_stub+548)
#10 pc 0x00000000001a8a6c /apex/com.android.art/lib64/libart.so (art::ArtMethod::Invoke(art::Thread, unsigned int, unsigned int, art::JValue*, char const*)+200)
#11 pc 0x000000000054a470 /apex/com.android.art/lib64/libart.so (art::JValue art::InvokeVirtualOrInterfaceWithJValuesart::ArtMethod*(art::ScopedObjectAccessAlreadyRunnable const&,
#12 pc 0x0000000000598bcc /apex/com.android.art/lib64/libart.so (art::Thread::CreateCallback(void*)+1288)
#13 pc 0x00000000000af97c /apex/com.android.runtime/lib64/bionic/libc.so (__pthread_start(void*)+64)
#14 pc 0x00000000000500d0 /apex/com.android.runtime/lib64/bionic/libc.so (__start_thread+64)

"FinalizerDaemon" tid=20188 Native
#00 pc 0x000000000004b20c /apex/com.android.runtime/lib64/bionic/libc.so (syscall+28)
#1 pc 0x00000000001af840 /apex/com.android.art/lib64/libart.so (art::ConditionVariable::WaitHoldingLocks(art::Thread*)+136)
#2 pc 0x000000000049190c /apex/com.android.art/lib64/libart.so (art::Monitor::Wait(art::Thread*, long, int, bool, art::ThreadState)+532)
#3 pc 0x0000000000493108 /apex/com.android.art/lib64/libart.so (art::Monitor::Wait(art::Thread*, art::ObjPtrart: 🪞 :Object, long, int, bool, art::ThreadState)+280)
#4 pc 0x00000000000803fc /apex/com.android.art/javalib/arm64/boot.oat (art_jni_trampoline+140)
#5 pc 0x00000000000a3aac /apex/com.android.art/javalib/arm64/boot.oat (java.lang.ref.ReferenceQueue.remove+348)
#6 pc 0x00000000000a392c /apex/com.android.art/javalib/arm64/boot.oat (java.lang.ref.ReferenceQueue.remove+44)
#7 pc 0x0000000000057fd4 /apex/com.android.art/javalib/arm64/boot-core-libart.oat (java.lang.Daemons$FinalizerDaemon.runInternal+356)
#8 pc 0x0000000000024b90 /apex/com.android.art/javalib/arm64/boot-core-libart.oat (java.lang.Daemons$Daemon.run+160)
#9 pc 0x000000000015aaf8 /apex/com.android.art/javalib/arm64/boot.oat (java.lang.Thread.run+72)
#10 pc 0x0000000000133564 /apex/com.android.art/lib64/libart.so (art_quick_invoke_stub+548)
#11 pc 0x00000000001a8a6c /apex/com.android.art/lib64/libart.so (art::ArtMethod::Invoke(art::Thread, unsigned int, unsigned int, art::JValue*, char const*)+200)
#12 pc 0x000000000054a470 /apex/com.android.art/lib64/libart.so (art::JValue art::InvokeVirtualOrInterfaceWithJValuesart::ArtMethod*(art::ScopedObjectAccessAlreadyRunnable const&,
#13 pc 0x0000000000598bcc /apex/com.android.art/lib64/libart.so (art::Thread::CreateCallback(void*)+1288)
#14 pc 0x00000000000af97c /apex/com.android.runtime/lib64/bionic/libc.so (__pthread_start(void*)+64)
#15 pc 0x00000000000500d0 /apex/com.android.runtime/lib64/bionic/libc.so (__start_thread+64)

"FinalizerWatchd" tid=20189 Native
#00 pc 0x000000000004b20c /apex/com.android.runtime/lib64/bionic/libc.so (syscall+28)
#1 pc 0x00000000001af840 /apex/com.android.art/lib64/libart.so (art::ConditionVariable::WaitHoldingLocks(art::Thread*)+136)
#2 pc 0x000000000049190c /apex/com.android.art/lib64/libart.so (art::Monitor::Wait(art::Thread*, long, int, bool, art::ThreadState)+532)
```

```
#3 pc 0x0000000000493108 /apex/com.android.art/lib64/libart.so (art::Monitor::Wait(art::Thread*, art::ObjPtrart: 🪞 :Object, long, int, bool, art::ThreadState)+280)
#4 pc 0x00000000000803fc /apex/com.android.art/javalib/arm64/boot.oat (art_jni_trampoline+140)
#5 pc 0x000000000008fca0 /apex/com.android.art/javalib/arm64/boot.oat (java.lang.Object.wait+48)
#6 pc 0x00000000000582c4 /apex/com.android.art/javalib/arm64/boot-core-libart.oat (java.lang.Daemons$FinalizerWatchdogDaemon.sleepUntilNeeded+68)
#7 pc 0x0000000000058634 /apex/com.android.art/javalib/arm64/boot-core-libart.oat (java.lang.Daemons$FinalizerWatchdogDaemon.runInternal+68)
#8 pc 0x0000000000024b90 /apex/com.android.art/javalib/arm64/boot-core-libart.oat (java.lang.Daemons$Daemon.run+160)
#9 pc 0x000000000015aaf8 /apex/com.android.art/javalib/arm64/boot.oat (java.lang.Thread.run+72)
#10 pc 0x0000000000133564 /apex/com.android.art/lib64/libart.so (art_quick_invoke_stub+548)
#11 pc 0x00000000001a8a6c /apex/com.android.art/lib64/libart.so (art::ArtMethod::Invoke(art::Thread, unsigned int, unsigned int, art::JValue*, char const*)+200)
#12 pc 0x000000000054a470 /apex/com.android.art/lib64/libart.so (art::JValue art::InvokeVirtualOrInterfaceWithJValuesart::ArtMethod*(art::ScopedObjectAccessAlreadyRunnable const&,
#13 pc 0x0000000000598bcc /apex/com.android.art/lib64/libart.so (art::Thread::CreateCallback(void*)+1288)
#14 pc 0x00000000000af97c /apex/com.android.runtime/lib64/bionic/libc.so (__pthread_start(void*)+64)
#15 pc 0x00000000000500d0 /apex/com.android.runtime/lib64/bionic/libc.so (__start_thread+64)
```

I used Admob version 7.3.0
Is there any fix for this ?? Guys please help me...

vi...@grabtaxi.com <vi...@grabtaxi.com> #38

Any update for other devices