

Comments (20)

Dependencies

Duplicates (0)

Blocking (0)

Resources (7)


Obsolete

Bug

P3

+ Add Hotlist

NeedsInfo

 STATUS UPDATE

No update yet.

Edit

 DESCRIPTION

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

<https://gist.github.com/nmr8acme/85a28d0bcce7664e4f46>

I can't tell where exactly the fault for this lies. I trimmed the repro code down as much as I could, and as far as I can tell all the nio code is kosher.

Crashes N6 5.1, N5 5.0, N9 5.1, does not crash N7 5.0, Adreno 330 Note 3 4.4 (all devices bone stock)

N9 is an Tegra K1, the rest of the crashes are various Adrenos.

To repro, run CrashActivity in a basic application, observe crashes where none are expected.

Typical crash looks as follows, but it varies between executions:

```
04-07 20:32:44.083 8041-8060/com.example.acmeaom.artplusglequalscrash A/libc : Fatal signal 11 (SIGSEGV), code 1, fault addr 0xe0 in tid 8060 (GLThread 3607)
04-07 20:32:44.169 850-902/? I/ActivityManager : Displayed com.example.acmeaom.artplusglequalscrash/.CrashActivity: +462ms
04-07 20:32:44.186 355-355/? I/DEBUG : *** *** *** *** *** *** *** *** *** *** *** *** *** *** ***
04-07 20:32:44.186 355-355/? I/DEBUG : Build fingerprint: 'google/shamu/shamu:5.1/LMY47D/1743759:user/release-keys'
04-07 20:32:44.186 355-355/? I/DEBUG : Revision: '33696'
04-07 20:32:44.186 355-355/? I/DEBUG : ABI: 'arm'
04-07 20:32:44.186 355-355/? I/DEBUG : pid: 8041, tid: 8060, name: GLThread 3607 >>> com.example.acmeaom.artplusglequalscrash <<<
04-07 20:32:44.186 355-355/? I/DEBUG : signal 11 (SIGSEGV), code 1 (SEGV_MAPERR), fault addr 0xe0
04-07 20:32:44.217 355-355/? I/DEBUG : r0 000000e0 r1 b359b00c r2 aec09688 r3 003fffff
04-07 20:32:44.218 355-355/? I/DEBUG : r4 000000e0 r5 00000004 r6 000000e0 r7 b6e2d600
04-07 20:32:44.218 355-355/? I/DEBUG : r8 b6e39fa0 r9 b359b00c sl 00000000 fp 00000004
04-07 20:32:44.218 355-355/? I/DEBUG : ip b6e2d63c sp b35797e8 lr b6e12f9f pc b6ddb682 cpsr 200f0030
04-07 20:32:44.218 355-355/? I/DEBUG : backtrace:
04-07 20:32:44.218 355-355/? I/DEBUG : #00 pc 00017682 /system/lib/libc.so (pthread_mutex_lock+7)
04-07 20:32:44.218 355-355/? I/DEBUG : #01 pc 0004ef9b /system/lib/libc.so (je_tcache_bin_flush_small+78)
04-07 20:32:44.218 355-355/? I/DEBUG : #02 pc 0004989d /system/lib/libc.so (ifree+448)
04-07 20:32:44.218 355-355/? I/DEBUG : #03 pc 00012caf /system/lib/libc.so (free+10)
04-07 20:32:44.218 355-355/? I/DEBUG : #04 pc 001d5f7d /system/lib/libart.so (art::JNI::ReleasePrimitiveArrayCritical(_JNIEnv*,_jarray*,void*,int)+876)
04-07 20:32:44.218 355-355/? I/DEBUG : #05 pc 000c0661 /system/lib/libart.so (art::CheckJNI::ReleasePrimitiveArrayCritical(_JNIEnv*,_jarray*,void*,int)+116)
04-07 20:32:44.218 355-355/? I/DEBUG : #06 pc 00063015 /system/lib/libandroid_runtime.so
04-07 20:32:44.219 355-355/? I/DEBUG : #07 pc 0006e1ad /system/lib/libandroid_runtime.so
04-07 20:32:44.219 355-355/? I/DEBUG : #08 pc 00b63485 /data/dalvik-cache/arm/system@framework@boot.oat
04-07 20:32:44.251 753-803/? I/Icing : Indexing 2C4C1AF75972075F064C58664019D4DD3C1838B3 from com.google.android.googlequicksearchbox
04-07 20:32:44.316 753-803/? I/Icing : Indexing done 2C4C1AF75972075F064C58664019D4DD3C1838B3
04-07 20:32:44.656 850-8063/? W/ActivityManager : Force finishing activity 1 com.example.acmeaom.artplusglequalscrash/.CrashActivity
04-07 20:32:44.661 355-355/? I/DEBUG : Tombstone written to: /data/tombstones/tombstone_09
04-07 20:32:44.662 850-898/? I/BootReceiver : Copying /data/tombstones/tombstone_09 to DropBox (SYSTEM_TOMBSTONE)
04-07 20:32:44.703 850-8063/? E/JavaBinder : !!! FAILED BINDER TRANSACTION !!!
```

✓ Links (7)

"<https://gist.github.com/nmr8acme/85a28d0bcce7664e4f46>..."

"Not sure why you are surprised that platform engineers don't use Android SDK when we basically use command like make, even to build APKs: <http://source.android.com/source/building-running.html>


"Okay, I see <https://play.google.com/store/apps/details?id=org.gemesys.android.dosbox> but it is too late tonight for me to look."

"<https://source.android.com/devices/tech/security/selinux>..."


"...ps, MMAP_MIN_ADDR has remained unchanged since Android 4.1. Applications are prohibited from mapping memory into the region 0x00000000-0x000080000. (commit <https://android.googlesource.com/>)

See all related links

COMMENTS



en...@google.com <en...@google.com>
Assigned to bd...@google.com.















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

nmr8acme, can you attach a test APK for CrashActivity?



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

	Can you elaborate? Is the linked Activity not crashing for you? Forgive me, I'm a little confused.
	ag...@google.com <ag...@google.com> #4 Internal tracking bug is 20150888. Re #3: bdc asked whether you could attach a full APK here.
	nm...@gmail.com <nm...@gmail.com> #5 Sorry, still a little confused, can you elaborate as to why, please?
	ag...@google.com <ag...@google.com> #6 Because that is easier for us then having to build a complete APK from the single Activity java file. None of the people CC-ed on this bug so far develop apps for a living...
	nm...@gmail.com <nm...@gmail.com> #7 Ahh, gotcha, sure thing.
	nm...@gmail.com <nm...@gmail.com> #8 You know, I spent some time on this. Couple hours out of my way, identifying the crash, and distilling a small repro for it. And, I "don't develop ART for a living..." ;) And I've got a halfway decer I think you could meet it 10% of the way, and compile it. The Android SDK is great, better every day. `android create project frob or something`, `cp ~/Downloads/CrashActivity.java frob/src/m That's all I've got.
	ha...@gmail.com <ha...@gmail.com> #9 Love your response @nmr8acme! This is surreal!
	bd...@google.com <bd...@google.com> #10 <i>Status: New</i> Not sure why you are surprised that platform engineers don't use Android SDK when we basically use command like make, even to build APKs: http://source.android.com/source/building-run
	nm...@gmail.com <nm...@gmail.com> #11 I don't think it would be fair to characterize anyone here as "surprised" ;)
	ge...@gmail.com <ge...@gmail.com> #12 Not sure if this is helpful, but here goes... I got to this page by doing a "bing" search on the error that I was able to find in the logcat trace on a Samsung Galaxy Tab "A", currently running And except for the keyboard, they all crash and die on Android 5.x (Lollipop) tablets. Since I cannot get the emulator to run on my development box, with API 24 SDK, I had to wait until I could ge The specific error is "F/libc (12524) Fatal signal 11(SIGSEGV), code 1, fault addr 0xdead4321 in tid 12539 (GLThread)" Not too dissimilar from what we have here, perhaps? If you want some APK's, you can search "GEMESYS" on the Google Play Store, and download any of my apps. They all crash and die on Android 5.x, and the error is in (GLthread). They are a DOS emulator, called "gDOSbox", a plotting package, called "GNUPlot", and four different APL interpreters: "Watcom APL", "IBM TryAPL2", "APLSE", and "sAPL". All these applicator and a Samsung Galaxy 3 cellphone, up to Samsung Tab 3 and Tab 4 and Tab S machines. Everything runs fine on Jellybean, Kitkat, and earlier. But Android 5.x is a no-go. If you want some - Mark Langdon GEMESYS Ltd. PS. I don't develop any of this stuff for a living either... :) It is all "pro bono". The "gDOSbox" was built mainly to get the APL's running, as a teaching and demo project. But folks are using the stuff now, and I feel its only fair to try to help them get their stuff working on the current O/S. Hope my APK's might help to illustrate the issue. If anyone wants more information, I will do what I can to provide it.
	nm...@gmail.com <nm...@gmail.com> #13 Mark, what leads you to believe this is the same crash? Are you also dying in JNI gl calls to glBufferSubData?
	ge...@gmail.com <ge...@gmail.com> #14 Hi nmr8a; Just the reference to GLthread. I am pretty much thrashing about and grasping at straws here. All my stuff is broken now, in all 5.x devices, near as I can tell, and it looks maybe related to tf 2 or 3 minutes effort. As I said, all my stuff runs fine on 4.x series devices. Your bug report looks important. Crashes may be related to the "improved" garbage-collection in ART. Or not. Me DOSemu to work on my CentOS Linux box - similar SELinux restrictions now in Android?) I don't yet know exactly where my stuff is dying. You've done a really nice job making a reproducabl quickly (which crashes in GLThread), might shed light. But maybe its not same crash.
	bd...@google.com <bd...@google.com> #15

Okay, I see <https://play.google.com/store/apps/details?id=org.gemesys.android.dosbox> but it is too late tonight for me to look.

selinux certainly is enabled. does adb logcat" or "adb shell cat /proc/kmsg" show any selinux "avc: denied" messages

<https://source.android.com/devices/tech/security/selinux/validate.html>

From external/sepolicy/domain.te:

```
# No process can map low memory (< CONFIG_LSM_MMAP_MIN_ADDR).
neverallow domain self:memprotect mmap_zero;
```

makes me worried for you regarding using low addresses. but that seems to be a different issue.

nn...@google.com <nn...@google.com> [#16](#)

For non-privileged apps, MMAP_MIN_ADDR has remained unchanged since Android 4.1. Applications are prohibited from mapping memory into the region 0x00000000-0x000080000. (comr SELinux extended this enforcement to processes running with privileges, but that shouldn't affect this bug.

nm...@gmail.com <nm...@gmail.com> [#17](#)

As luck would have it, I am at I/O this year. As an experienced app developer, I would like to extend my hand in help to anyone CCed on this bug. I would be pleased to sit down with you, buy y

ge...@gmail.com <ge...@gmail.com> [#18](#)

Hi Nmr8a,
Thanx for the offer. I would buy you a drink if you could shed any light on why all my apps now crash and die under Android 5. I've got a Samsung Galaxy Tab A, and was just messing about today, looking at the logcat output. I have "gDOSbox", (a DOS-emulator), "GNUplot37" (Android version of GNUplot), and a version of IBM's "TryAPL2", and a couple of other APL's, all running nicely on Kitkat 4.x Android. (Search "Gemesys" on the Playstore, and you can see these, if curious). All apps tombstone out under Android 5.x. I have GNUplot37 and a version of the IP-Sharp APL running on the Blackberry Playbook also, so this is the second time I have been thrown under the bus for attempting to build scientific software for these little tablet computers. I've pretty much given up now, as it's just not worth the grief. The Apple folks don't even allow emulators or programming languages on their stupid iPad, and it looks like Google is going down that path now also. I am so pissed that I am thinking I will go to Asia, find a tablet-maker, and commission the construction of a new, reduced-complexity tablet, running some standard, true open-source Linux variant. I want a programmable computer, not a locked-down Google appliance.
But thanx for the offer. Hope someone takes you up on it.
Android 5 has broken the JNI for me, and I guess that is that. If someone gets the JNI - GL / SDL stuff working, send me a link to what you did. Thanx.
Mark Langdon
GEMESYS Ltd.

ge...@gmail.com <ge...@gmail.com> [#19](#)

Thanx for this info. Yes, I've confirmed that the MMAP_MIN_ADDR value, which stops DOSemu on CentOS if set above zero, has nothing to do with DOSbox operation, and is not related to issues gDOSbox is having when it crashes GLThread. The gDOSbox stuff works fine on Android 4.4 series devices, but on Android 5.x devices, it crashes in GLThread after libart.so runs JNI call CallIntMethod, which then calls InvokeVirtualOrInterfaceWithArgs.
gDOSbox is using SDL 1.2 (Simple Directmedia Layer), which is using OpenGL.

I will post an update with a logcat screen capture, showing the crash details, from Samsung Galaxy Tab A.
- Mark Langdon

ge...@gmail.com <ge...@gmail.com> [#20](#)

Just to update:
Looks like my stuff is not at all related to this bug, which Nathan Ramsey has filed for the GLES20 JNI - GL crash.

My stuff all works now, was a stale pointer, caused by not using the NewGlobalRef feature for JNI calls, as is documented in detail in this 2011 post, for Ice Cream Sandwich Android version...
["http://android-developers.blogspot.ca/2011/11/jni-local-reference-changes-in-ics.html"](http://android-developers.blogspot.ca/2011/11/jni-local-reference-changes-in-ics.html)

Here is copy of note I sent to Nathan on this:

Hi Nathan;

I fixed all my code, and got all my apps working.

Thanx so much for your comments and pointers! The

reference to the Pelya/Commanergenius game on Github was particularly helpful, and actually provided the key.

If you are using the older SDL-1.2 or SDL-1.3 code, there is a program called SDL_androidvideo.c, in the src directory, subdir video/android, and this program has the JNI stuff defined to allow Java code which controls the screen, to be called from native c code. There is a jobject pointer which is static defined, local (I think), and it causes the whole thing to kack, if the

new Android 5.x is run, which has ART doing its hardcore GC stuff (garbage collect, clean up stray freed memory...I am sure you know all this)

I am not a skilled .c++ prgmr, being a old-school type. Basically, the

code change is to just set up the JavaRenderer pointer using the GlobalNewRef instead of just assigning it as a local memory reference. (I can send a detailed note if you are interested.) Basically, there is this 2011 blog post about this,

by Elliot Hughes, I think it was (a software guy on the Dalvik team..)

Actually, I just found the URL...

["http://android.developers.blogspot.ca/2011/11/jni-local-reference-changes-in-ics.html"](http://android.developers.blogspot.ca/2011/11/jni-local-reference-changes-in-ics.html)

The error documented in that blog post was exactly what was tripping me up.

Hope this info is of some use.

Thanx again for sending me the links you did. After some research, they helped me fix everything..!

- Mark Langdon,
GEMESYS Ltd.



bd...@google.com <bd...@google.com>

Status: Won't Fix (Obsolete)