



← ↻ ☆ AlarmManagerService: Global reference table overflow

+1²

Hotlists (2)

Mark as Duplicate

🔔

⋮

Comments (6)

Dependencies

Duplicates (0)

Blocking (0)

Resources (0)

Fixed

Bug

P3

+ Add Hotlist

[AOSP] assigned

👤 STATUS UPDATE

No update yet.

Edit

📄 DESCRIPTION

ww...@gmail.com created issue #1

```
done
→ AppWithCpp adb logcat -b crash
----- beginning of crash
12-14 18:55:06.571 514 586 F libc : Fatal signal 6 (SIGABRT), code -1 (SI_QUEUE) in tid 586 (Binder:514_4), pid 514 (Binder:514_3)
12-14 18:55:06.874 6205 6205 F DEBUG : *** *** *** *** *** *** *** *** *** *** *** *** *** *** ***
12-14 18:55:06.874 6205 6205 F DEBUG : Build fingerprint: 'google/sdk_gphone_x86/generic_x86_arm:11/RSR1.201013.001/6903271:user/release-keys'
12-14 18:55:06.874 6205 6205 F DEBUG : Revision: '0'
12-14 18:55:06.874 6205 6205 F DEBUG : ABI: 'x86'
12-14 18:55:06.875 6205 6205 F DEBUG : Timestamp: 2020-12-14 18:55:06+0800
12-14 18:55:06.875 6205 6205 F DEBUG : pid: 514, tid: 586, name: Binder:514_4 >>> system_server <<<
12-14 18:55:06.875 6205 6205 F DEBUG : uid: 1000
12-14 18:55:06.875 6205 6205 F DEBUG : signal 6 (SIGABRT), code -1 (SI_QUEUE), fault addr -----
12-14 18:55:06.875 6205 6205 F DEBUG : Abort message: 'JNI ERROR (app bug): global reference table overflow (max=51200)global reference table dump:
12-14 18:55:06.875 6205 6205 F DEBUG : Last 10 entries (of 51200):
12-14 18:55:06.875 6205 6205 F DEBUG : 51199: 0x132f6da0 com.android.server.AlarmManagerService$2
12-14 18:55:06.875 6205 6205 F DEBUG : 51198: 0x132f6da0 com.android.server.AlarmManagerService$2
12-14 18:55:06.875 6205 6205 F DEBUG : 51197: 0x132f6da0 com.android.server.AlarmManagerService$2
12-14 18:55:06.875 6205 6205 F DEBUG : 51196: 0x132f6da0 com.android.server.AlarmManagerService$2
12-14 18:55:06.875 6205 6205 F DEBUG : 51195: 0x132f6da0 com.android.server.AlarmManagerService$2
12-14 18:55:06.875 6205 6205 F DEBUG : 51194: 0x132f6da0 com.android.server.AlarmManagerService$2
12-14 18:55:06.875 6205 6205 F DEBUG : 51193: 0x132f6da0 com.android.server.AlarmManagerService$2
12-14 18:55:06.875 6205 6205 F DEBUG : 51192: 0x132f6da0 com.android.server.AlarmManagerService$2
12-14 18:55:06.875 6205 6205 F DEBUG : 51191: 0x132f6da0 com.android.server.AlarmManagerService$2
12-14 18:55:06.875 6205 6205 F DEBUG : 51190: 0x132f6da0 com.android.server.AlarmManagerService$2
12-14 18:55:06.875 6205 6205 F DEBUG : Summary:
12-14 18:55:06.875 6205 6205 F DEBUG : 49119 of com.android.server.AlarmManagerService$2 (1 unique instances)
12-14 18:55:06.875 6205 6205 F DEBUG : 366 of java.lang.Class (275 unique instances)
12-14 18:55:06.875 6205 6205 F DEBUG : 288 of java.nio.DirectByteBuffer (288 unique instances)
...
12-14 18:55:06.875 6205 6205 F DEBUG : '
12-14 18:55:06.875 6205 6205 F DEBUG : eax 00000000 ebx 00000202 ecx 0000024a edx 00000006
12-14 18:55:06.875 6205 6205 F DEBUG : edi e93ad81e esi b9083240
12-14 18:55:06.875 6205 6205 F DEBUG : ebp ee3afb90 esp b90831e8 eip ee3afb99
12-14 18:55:07.082 6205 6205 F DEBUG : backtrace:
12-14 18:55:07.082 6205 6205 F DEBUG : #00 pc 00000b99 [vdso] (__kernel_vsyscall+9)
12-14 18:55:07.082 6205 6205 F DEBUG : #01 pc 0005ad68 /apex/com.android.runtime/lib/bionic/libc.so (syscall+40) (BuildId: 6e3a0180fa6637b68c0d181)
12-14 18:55:07.082 6205 6205 F DEBUG : #02 pc 00076511 /apex/com.android.runtime/lib/bionic/libc.so (abort+209) (BuildId: 6e3a0180fa6637b68c0d181c)
12-14 18:55:07.082 6205 6205 F DEBUG : #03 pc 00639a4d /apex/com.android.art/lib/libart.so (art::Runtime::Abort(char const*)+2477) (BuildId: 81915)
12-14 18:55:07.082 6205 6205 F DEBUG : #04 pc 00025a23 /apex/com.android.art/lib/libartbase.so (std::_1::__function::__func<void (*) (char const*)>::__call) (BuildId: 81915)
12-14 18:55:07.082 6205 6205 F DEBUG : #05 pc 0001588f /system/lib/libbase.so (android::base::SetAborter(std::_1::function<void (char const*)>&&) (BuildId: bbac430fc6349b937996bb914e7)
12-14 18:55:07.082 6205 6205 F DEBUG : #06 pc 00006291 /system/lib/liblog.so (__android_log_call_aborter+33) (BuildId: bbac430fc6349b937996bb914e7)
12-14 18:55:07.082 6205 6205 F DEBUG : #07 pc 00014d14 /system/lib/libbase.so (android::base::LogMessage::~LogMessage()+436) (BuildId: 3abc3ce4c3b)
12-14 18:55:07.083 6205 6205 F DEBUG : #08 pc 00407a7d /apex/com.android.art/lib/libart.so (art::JavaVMExt::AddGlobalRef(art::Thread*, art::ObjPtr (BuildId: 81915)
12-14 18:55:07.083 6205 6205 F DEBUG : #09 pc 0041d5d1 /apex/com.android.art/lib/libart.so (art::JNI<false>::NewGlobalRef(_JNIEnv*, _jobject*)+753 (BuildId: 81915)
12-14 18:55:07.083 6205 6205 F DEBUG : #10 pc 003f662b /apex/com.android.art/lib/libart.so (art::(anonymous namespace)::CheckJNI::NewRef(char cons (BuildId: 81915)
12-14 18:55:07.083 6205 6205 F DEBUG : #11 pc 003df43b /apex/com.android.art/lib/libart.so (art::(anonymous namespace)::CheckJNI::NewGlobalRef(_JN (BuildId: 81915)
12-14 18:55:07.083 6205 6205 F DEBUG : #12 pc 00123612 /system/lib/libandroid_runtime.so (JavaDeathRecipient::JavaDeathRecipient(_JNIEnv*, _jobjec (BuildId: 81915)
12-14 18:55:07.083 6205 6205 F DEBUG : #13 pc 00123157 /system/lib/libandroid_runtime.so (android_os_BinderProxy_linkToDeath(_JNIEnv*, _jobject*, (BuildId: 81915)
12-14 18:55:07.083 6205 6205 F DEBUG : #14 pc 00210984 /system/framework/x86/boot-framework.oat (art_jni_trampoline+148) (BuildId: 9a9778e61b43d34)
12-14 18:55:07.083 6205 6205 F DEBUG : #15 pc 019ae2cc /system/framework/oat/x86/services.odex (com.android.server.AlarmManagerService.setImpl+284 (BuildId: 81915)
12-14 18:55:07.083 6205 6205 F DEBUG : #16 pc 016d00d9 /system/framework/oat/x86/services.odex (com.android.server.AlarmManagerService$.set+1001) (BuildId: 81915)
12-14 18:55:07.083 6205 6205 F DEBUG : #17 pc 0053a13c /system/framework/x86/boot-framework.oat (android.app.IAlarmManager$Stub.onTransact+1596) (BuildId: 81915)
12-14 18:55:07.083 6205 6205 F DEBUG : #18 pc 006d7acb /system/framework/x86/boot-framework.oat (android.os.Binder.execTransactInternal+619) (BuildId: 81915)
12-14 18:55:07.083 6205 6205 F DEBUG : #19 pc 006d7730 /system/framework/x86/boot-framework.oat (android.os.Binder.execTransact+288) (BuildId: 81915)
12-14 18:55:07.083 6205 6205 F DEBUG : #20 pc 0013b922 /apex/com.android.art/lib/libart.so (art_quick_invoke_stub+338) (BuildId: 81915)
12-14 18:55:07.083 6205 6205 F DEBUG : #21 pc 001d0381 /apex/com.android.art/lib/libart.so (art::ArtMethod::Invoke(art::Thread*, unsigned int*, un (BuildId: 81915)
12-14 18:55:07.083 6205 6205 F DEBUG : #22 pc 0062f81c /apex/com.android.art/lib/libart.so (art::JValue art::InvokeVirtualOrInterfaceWithVarArgs<a (BuildId: 81915)
12-14 18:55:07.083 6205 6205 F DEBUG : #23 pc 0062fa35 /apex/com.android.art/lib/libart.so (art::JValue art::InvokeVirtualOrInterfaceWithVarArgs< (BuildId: 81915)
12-14 18:55:07.083 6205 6205 F DEBUG : #24 pc 00426151 /apex/com.android.art/lib/libart.so (art::JNI<false>::CallBooleanMethodV(_JNIEnv*, _jobject (BuildId: 81915)
12-14 18:55:07.083 6205 6205 F DEBUG : #25 pc 003f841d /apex/com.android.art/lib/libart.so (art::(anonymous namespace)::CheckJNI::CallMethodV(char (BuildId: 81915)
12-14 18:55:07.083 6205 6205 F DEBUG : #26 pc 003e33a9 /apex/com.android.art/lib/libart.so (art::(anonymous namespace)::CheckJNI::CallBooleanMetho (BuildId: 81915)
12-14 18:55:07.083 6205 6205 F DEBUG : #27 pc 00091c6e /system/lib/libandroid_runtime.so (_JNIEnv::CallBooleanMethod(_jobject*, _jmethodID*, ...) (BuildId: 81915)
12-14 18:55:07.083 6205 6205 F DEBUG : #28 pc 001215aa /system/lib/libandroid_runtime.so (JavaBinder::onTransact(unsigned int, android::Parcel co (BuildId: 81915)
```

```
12-14 18:55:07.083 6205 6205 F DEBUG : #29 pc 0004723b /system/lib/libbinder.so (android::BBinder::transact(unsigned int, android::Parcel const&,
12-14 18:55:07.083 6205 6205 F DEBUG : #30 pc 000518ab /system/lib/libbinder.so (android::IPCThreadState::executeCommand(int)+1451) (BuildId: d43e
12-14 18:55:07.083 6205 6205 F DEBUG : #31 pc 000511f9 /system/lib/libbinder.so (android::IPCThreadState::getAndExecuteCommand()+169) (BuildId: d4
12-14 18:55:07.083 6205 6205 F DEBUG : #32 pc 00051cb8 /system/lib/libbinder.so (android::IPCThreadState::joinThreadPool(bool)+72) (BuildId: d43e4
12-14 18:55:07.083 6205 6205 F DEBUG : #33 pc 0007e309 /system/lib/libbinder.so (android::PoolThread::threadLoop()+41) (BuildId: d43e4e5a9165778bb
12-14 18:55:07.083 6205 6205 F DEBUG : #34 pc 00015116 /system/lib/libutils.so (android::Thread::_threadLoop(void*)+374) (BuildId: ab4be013cda31e8
12-14 18:55:07.083 6205 6205 F DEBUG : #35 pc 00098fee /system/lib/libandroid_runtime.so (android::AndroidRuntime::javaThreadShell(void*)+174) (Bu
12-14 18:55:07.083 6205 6205 F DEBUG : #36 pc 000147d9 /system/lib/libutils.so (thread_data_t::trampoline(thread_data_t const*)+457) (BuildId: ab4
12-14 18:55:07.083 6205 6205 F DEBUG : #37 pc 000e6974 /apex/com.android.runtime/lib/bionic/libc.so (__pthread_start(void*)+100) (BuildId: 6e3a018
12-14 18:55:07.083 6205 6205 F DEBUG : #38 pc 00078567 /apex/com.android.runtime/lib/bionic/libc.so (__start_thread+71) (BuildId: 6e3a0180fa6637b6
12-14 18:55:08.437 987 987 E AndroidRuntime: FATAL EXCEPTION: main
```

Reproduce steps:

1. Install the test app: adb install mtbf-debug.apk
2. Start the test app: adb shell am start -n com.miui.mtbf/.AlarmDosAttackActivity
3. Click the "Begin Attack"
4. Observe the phone will reboot in 5~10 minutes

 **deleted**
0 B 

 **deleted**
0 B 

COMMENTS

 **vi...@google.com** <vi...@google.com>
Assigned to vi...@google.com.

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


Thank you for reporting this issue. We have shared this with our product and engineering team and will update this issue with more information as it becomes available.

 **vi...@google.com** <vi...@google.com> [#3](#)

Could you please provide the mtbf-debug.apk file attached in [comment #1](#) for us to further investigate this issue.

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

See the attachments, plz.

 **mtbf-realApp-debug.apk**
2.3 MB [Download](#)

 **vi...@google.com** <vi...@google.com> [#5](#)

Thanks for providing the apk file again.

 **vi...@google.com** <vi...@google.com> [#6](#)

Marked as fixed.

The development team has fixed the issue that you have reported and it will be available in a future build.