System_server abort because of state check failed

Assigned Bug P3 + [AOSP] assigned

STATUS UPDATE No update yet. Edit

DESCRIPTION zj...@gmail.com created issue #1 Mar 13, 2020 08:58PM

My team has reproduced this issue twice on Android P。
Our ART code is updated to b151df35115c546fd72107de43dd5e3a4a7f6bd4
And security patch is updated to 2020-3-01

It looks like every time the abort is happen just after the jni call
the binder thread finish the jni stuff and is about to exit the call
but the state of the thread suddenly change to kRunnable, which lead to abort
we have two direction about this issue:
1、the memory is corrupted ,so the state changed.
but we anaylysis the coredump of system_server just to find out that
the memory around the state is all fine. So the possiblity of this direction is extremly low.
we will continue to do some work to find out .
2、we saw similiar issue like ours， is there any possibility that this issue could have some relation
to the art? have u guys tried more pressure test on jni call?

the issue is like:

first time:

ABI: 'arm64'

```
pid: 2362, tid: 2374, name: Binder:2362_2  >>> system_server <<<
signal 6 (SIGABRT), code -6 (SI_TKILL), fault addr --------
Abort message: 'thread-inl.h:115] Check failed: old_state_and_flags.as_struct.state != kRunnable
(old_state_and_flags.as_struct.state=67, kRunnable=Runnable) '
   x0  0000000000000000  x1  0000000000000946  x2  0000000000000006  x3  0000000000000008
   x4  000000737ee9ba40  x5  000000737ee9ba40  x6  000000737ee9ba40  x7  000000737ee9bb80
   x8  0000000000000083  x9  945feafa4384f3d2  x10 0000000000000000  x11 ffffffffc7ffffbdf
   x12 0000000000000001  x13 000000737ee9c440  x14 0000000000000000  x15 0000000000000000
   x16 000000741d3dc2c8  x17 000000741d30ddb8  x18 0000000000000001  x19 000000000000093a
   x20 0000000000000946  x21 000000737ee9b900  x22 000000739a685000  x23 000000739a5fc608
   x24 0000000000000000  x25 000000739a52a3a6  x26 000000739a52a3b0  x27 000000739a52a3d7
   x28 00000073814a194c  x29 00000073814a1800
   sp  00000073814a17c0  lr  000000741d2ffd64  pc  000000741d2ffd8c

backtrace:
   #00 pc 0000000000022d8c  /system/lib64/libc.so (abort+116)
   #01 pc 000000000048367c  /system/lib64/libart.so (art::Runtime::Abort(char const*)+1196)
   #02 pc 000000000055d6f4  /system/lib64/libart.so
(_ZNSt3__110__function6__funcIPFvPKcENS_9allocatorIS5_EES4_EclEOS3_+36)
   #03 pc 0000000000009104  /system/lib64/libbase.so (android::base::LogMessage::~LogMessage()+724)
   #04 pc 00000000000e1978  /system/lib64/libart.so (art::Thread::SetState(art::ThreadState)+376)
   #05 pc 0000000000106a28  /system/lib64/libart.so (art::(anonymous namespace)::CheckJNI::GetField(char const*, _JNIEnv*,
_jobject*, _jfieldID*, bool, art::Primitive::Type)+1900)
   #06 pc 00000000000f09e8  /system/lib64/libart.so (art::(anonymous namespace)::CheckJNI::GetLongField(_JNIEnv*,
_jobject*, _jfieldID*)+68)
   #07 pc 000000000016e170  /system/lib64/libandroid_runtime.so (android::Region_translate(_JNIEnv*, _jobject*, int, int,
_jobject*)+72)
   #08 pc 00000000003e89b8  /system/framework/arm64/boot-framework.oat (offset 0x3cd000)
(android.graphics.Region.translate [DEDUPED]+168)
   #09 pc 00000000009163ec  /system/framework/arm64/boot-framework.oat (offset 0x3cd000)
(android.graphics.Region.translate+44)
   #10 pc 00000000016b6ec0  /system/framework/oat/arm64/services.odex (offset 0x63f000)
```

second time:
ABI: 'arm64'

```
pid: 2240, tid: 2253, name: Binder:2240_2  >>> system_server <<<
signal 6 (SIGABRT), code -6 (SI_TKILL), fault addr --------
Abort message: 'thread-inl.h:115] Check failed: old_state_and_flags.as_struct.state != kRunnable
(old_state_and_flags.as_struct.state=67, kRunnable=Runnable) '
   x0  0000000000000000  x1  00000000000008cd  x2  0000000000000006  x3  0000000000000008
   x4  0000006fc35a1000  x5  0000006fc35a1000  x6  0000006fc35a1000  x7  00000000010070e8
   x8  0000000000000083  x9  4342d65bd821932a  x10 0000000000000000  x11 ffffffffc7ffffbdf
   x12 0000000000000001  x13 ffffffffa1abc9e5  x14 0000000279ecf61  x15 0000000000000000
   x16 0000006fc09182c8  x17 0000006fc0849db8  x18 0000000000000001  x19 00000000000008c0
```

Reporter	zj...@gmail.com
Type	Bug
Priority	P3
Severity	S3
Status	Assigned
Access	Default access View
Assignee	zj...@gmail.com
Verifier	--
Collaborators	
CC	ar...@google.com zj...@gmail.com
AOSP ID	--
ReportedBy	Developer
Found In	--
Targeted To	--
Verified In	--
In Prod	

```
    x20 00000000000008cd  x21 0000006f3d8f07c0  x22 0000006f3d885000  x23 0000006f3d73c608
    x24 0000000000000000  x25 0000006f3d66a3a6  x26 0000006f3d66a3b0  x27 0000006f3d66a3d7
    x28 0000006f2465b98c  x29 0000006f2465b840
    sp  0000006f2465b800  lr  0000006fc083bd64  pc  0000006fc083bd8c

  backtrace:
    #00 pc 0000000000022d8c  /system/lib64/libc.so (abort+116)
    #01 pc 000000000048367c  /system/lib64/libart.so (art::Runtime::Abort(char const*)+1196)
    #02 pc 000000000055d6f4  /system/lib64/libart.so
(_ZNSt3__110__function6__funcIPFvPKcENS_9allocatorIS5_EES4_EclEOS3_+36)
    #03 pc 0000000000009104  /system/lib64/libbase.so (android::base::LogMessage::~LogMessage()+724)
    #04 pc 00000000000e1978  /system/lib64/libart.so (art::Thread::SetState(art::ThreadState)+376)
    #05 pc 0000000000103cd0  /system/lib64/libart.so (art::(anonymous namespace)::CheckJNI::CallMethodV(char const*,
_JNIEnv*, _jobject*, _jclass*, _jmethodID*, std::__va_list, art::Primitive::Type, art::InvokeType)+3320)
    #06 pc 00000000000ee60c  /system/lib64/libart.so (art::(anonymous namespace)::CheckJNI::CallBooleanMethodV(_JNIEnv*,
_jobject*, _jmethodID*, std::__va_list)+92)
    #07 pc 00000000000c81a8  /system/lib64/libandroid_runtime.so (_JNIEnv::CallBooleanMethod(_jobject*, _jmethodID*,
...)+120)
    #08 pc 0000000000144328  /system/lib64/libandroid_runtime.so (JavaBBinder::onTransact(unsigned int, android::Parcel
const&, android::Parcel*, unsigned int)+156)
    #09 pc 0000000000050078  /system/lib64/libbinder.so (android::BBinder::transact(unsigned int, android::Parcel const&,
android::Parcel*, unsigned int)+152)
    #10 pc 000000000005d958  /system/lib64/libbinder.so (android::IPCThreadState::executeCommand(int)+520)
    #11 pc 000000000005d67c  /system/lib64/libbinder.so (android::IPCThreadState::getAndExecuteCommand()+172)
    #12 pc 000000000005ddd0  /system/lib64/libbinder.so (android::IPCThreadState::joinThreadPool(bool)+76)
    #13 pc 0000000000083564  /system/lib64/libbinder.so (android::PoolThread::threadLoop()+40)
    #14 pc 0000000000011774  /system/lib64/libutils.so (android::Thread::_threadLoop(void*)+280)
    #15 pc 00000000000b5744  /system/lib64/libandroid_runtime.so (android::AndroidRuntime::javaThreadShell(void*)+140)
    #16 pc 0000000000011034  /system/lib64/libutils.so (thread_data_t::trampoline(thread_data_t const*)+248)
    #17 pc 000000000008dd64  /system/lib64/libc.so (__pthread_start(void*)+36)
    #18 pc 0000000000024a90  /system/lib64/libc.so (__start_thread+68)
```

✓ Mentioned issues (1) ✓ Links (2) Hide all

Mentioned issues (1)

P3 system_server occur abort " issue 137036159 is same as mine." zj...@ #3, zj...@ #11

Links (2)

" … steps to capture a bug report, please refer: https://developer.android.com/studio/debug/bug-report#bugreportdevice" vi...@ #2
"http://b.ne" zj...@ #7, zj...@ #10

COMMENTS All comments ↓ Oldest first

vi...@google.com <vi...@google.com> #2 Mar 13, 2020 09:19PM

Assigned to vi...@google.com.

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

Android build
Which Android build are you using? (e.g. OPP1.170223.012)

Device used
Which device did you use to reproduce this issue?

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

Frequency
How frequently does this issue occur? (e.g 100% of the time, 10% of the time)

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

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

zj...@gmail.com <zj...@gmail.com> #3 Mar 13, 2020 10:45PM

sorry, some detail information i connot provide.

1、Android build "PPR1.180610.011
2、we reproduce this issue on our product, product detail i cannot provide.
3、this issue is reproduced during our daily Steady test
4、frequency is : 2 times in 15 days。 we run steady test every day，move than 50 times per day. The probability of problems is extremly low
5、i cannot provide tombstone and bugreport，u can refer to https://issuetracker.google.com/issues/137036159.

issue 137036159 is same as mine.

vi...@google.com <vi...@google.com> #4 Mar 16, 2020 05:00PM

We have passed this to the development team and will update this issue with more information as it becomes available.

zj...@gmail.com <zj...@gmail.com> #5 Mar 16, 2020 08:21PM

thanks for the updates

according to the code below： old_thread_state_ is read from the thread tls32. and it must not be kRunnable so that it can take the branch self_->SetState(new_thread_state);
but int the SetState， read state frome tls32 again， the state has suddenly changed to kRunnable, state check fail lead to abort.

our team is now working on pressure test by looping this commond： dumpsys meminfo com.android.phone as we think this would trigger binder call and jni call
u guys can send some test if u want to run on our devices

```
inline ScopedThreadStateChange::ScopedThreadStateChange(Thread* self, ThreadState new_thread_state)
    : self_(self), thread_state_(new_thread_state), expected_has_no_thread_(false) {
  if (UNLIKELY(self_ == nullptr)) {
    // Value chosen arbitrarily and won't be used in the destructor since thread_ == null.
    old_thread_state_ = kTerminated;
    Runtime* runtime = Runtime::Current();
    CHECK(runtime == nullptr || !runtime->IsStarted() || runtime->IsShuttingDown(self_));
  } else {
    DCHECK_EQ(self, Thread::Current());
    // Read state without locks, ok as state is effectively thread local and we're not interested
    // in the suspend count (this will be handled in the runnable transitions).
    old_thread_state_ = self->GetState();
    if (old_thread_state_ != new_thread_state) {
      if (new_thread_state == kRunnable) {
        self_->TransitionFromSuspendedToRunnable();
      } else if (old_thread_state_ == kRunnable) {
        self_->TransitionFromRunnableToSuspended(new_thread_state);
      } else {
        // A suspended transition to another effectively suspended transition, ok to use Unsafe.
        self_->SetState(new_thread_state);
      }
    }
  }
}
```

deleted
0 B

Restricted

ma...@google.com <ma...@google.com> #6 Mar 18, 2020 05:06AM

Reassigned to ma...@google.com.

I was looking at similar issues in other bugs and observed that it should be calling this instead of SetState:
self_->TransitionFromRunnableToSuspended(old_thread_state_);

My assumption is that the ScopedThreadStateChange is being corrupted on the stack since it must be that thread_state_ is no longer kRunnable. Based on the control flow, thread_state_ should always be set to kRunnable in ScopedObjectAccess.

zj...@gmail.com <zj...@gmail.com> #7 Mar 19, 2020 02:42AM

very good assumption， we are working on this by anlaysising coredump, and our pressure test hasn't reproduced the issue so far

accord to the Disassembly code, the w8 is load from [sp, #80],
seems like [sp,#80] is currupted， the content is 0xd5b79d85， is that right?
we are not sure which value is store is [sp, #84], but the value 5b is equal to kNative
let me know if u want some anlaysis on the coredump

```
~ScopedThreadStateChange():
art/runtime/scoped_thread_state_change-inl.h:58
  e325c:	f94027f4	ldr	x20, [sp,#72]
  e3260:	b4001bf4	cbz	x20, e35dc
<_ZN3art12_GLOBAL__N_18CheckJNI11DefineClassEP7_JNIEnvPKcP8_jobjectPKai+0x6b8>
art/runtime/scoped_thread_state_change-inl.h:65
  e3264:	294a57e8	ldp	w8, w21, [sp,#80]
  e3268:	6b0802bf	cmp	w21, w8
  e326c:	540015c0	b.eq	e3524
<_ZN3art12_GLOBAL__N_18CheckJNI11DefineClassEP7_JNIEnvPKcP8_jobjectPKai+0x600>
```

```
art/runtime/scoped_thread_state_change-inl.h:66
   e3270:       71010ebf        cmp     w21, #0x43
   e3274:       54000d81        b.ne    e3424
<_ZN3art12_GLOBAL__N_18CheckJNI11DefineClassEP7_JNIEnvPKcP8_jobjectPKai+0x500>

~ScopedThreadStateChange():
art/runtime/scoped_thread_state_change-inl.h:68
   e3424:       71010d1f        cmp     w8, #0x43
   e3428:       540000c0        b.eq    e3440
<_ZN3art12_GLOBAL__N_18CheckJNI11DefineClassEP7_JNIEnvPKcP8_jobjectPKai+0x51c>
art/runtime/scoped_thread_state_change-inl.h:72
   e342c:       aa1403e0        mov     x0, x20
   e3430:       2a1503e1        mov     w1, w21
   e3434:       97fff8f3        bl      e1800 <_ZN3art6Thread8SetStateENS_11ThreadStateE>
   e3438:       1400003b        b       e3524
<_ZN3art12_GLOBAL__N_18CheckJNI11DefineClassEP7_JNIEnvPKcP8_jobjectPKai+0x600>

(gdb) bt
#0  abort () at bionic/libc/bionic/abort.cpp:73
#1  0x000000739a41d680 in art::Runtime::Abort (msg=<optimized out>) at art/runtime/runtime.cc:588
#2  0x000000739a4f76f8 in std::__1::__invoke<void (*&)(char const*), char const*> (__args=<unknown type in
./symbols/system/lib64/libart.so, CU 0x4beea67, DIE 0x4bf88e5>, __f=<optimized out>) at
external/libcxx/include/type_traits:4480
#3  std::__1::__invoke_void_return_wrapper<void>::__call<void (*&)(char const*), char const*>(void (*&)(char const*), char
const*&&) (__args=<optimized out>, __args=<optimized out>) at external/libcxx/include/__functional_base:349
#4  std::__1::__function::__func<void (*)(char const*), std::__1::allocator<void (*)(char const*)>, void (char
const*)>::operator()(char const*&&) (this=<optimized out>,
    __arg=<unknown type in ./symbols/system/lib64/libart.so, CU 0x4beea67, DIE 0x4bf88a3>) at
external/libcxx/include/functional:1562
#5  0x000000741d40e108 in std::__1::function<void (char const*)>::operator()(char const*) const (this=<optimized out>,
    __arg=0x7394427900 "Check failed: old_state_and_flags.as_struct.state != kRunnable
(old_state_and_flags.as_struct.state=67, kRunnable=Runnable) ") at external/libcxx/include/functional:1916
#6  android::base::LogMessage::~LogMessage (this=0x73814a1940) at system/core/base/logging.cpp:458
#7  0x000000739a07b97c in art::Thread::SetState (this=<optimized out>, new_state=<optimized out>) at
art/runtime/thread-inl.h:115
#8  0x000000739a0a0a2c in art::ScopedThreadStateChange::~ScopedThreadStateChange (this=<optimized out>) at
art/runtime/scoped_thread_state_change-inl.h:72
#9  art::ScopedObjectAccessUnchecked::~ScopedObjectAccessUnchecked (this=<optimized out>) at
art/runtime/scoped_thread_state_change.h:153
#10 art::(anonymous namespace)::CheckJNI::GetField (function_name=0x739a52b7fa "GetLongField", env=<optimized
out>, obj=<optimized out>, fid=<optimized out>, is_static=false, type=art::Primitive::kPrimLong)
    at art/runtime/check_jni.cc:2951
#11 0x000000739a08a9ec in art::(anonymous namespace)::CheckJNI::GetLongField (env=0x946 <art::chains+1126>,
obj=0x6 <_DYNAMIC+6>, fid=0x8 <_DYNAMIC+8>) at art/runtime/check_jni.cc:2261
#12 0x000000741db43174 in _JNIEnv::GetLongField (this=0x739a6e2920, obj=0x946 <art::chains+1126>, fieldID=0x6
<_DYNAMIC+6>) at libnativehelper/include_jni/jni.h:710
#13 android::GetSkRegion (env=0x739a6e2920, regionObject=0x946 <art::chains+1126>) at
frameworks/base/core/jni/android/graphics/Region.cpp:37
#14 android::Region_translate (env=0x739a6e2920, region=0x946 <art::chains+1126>, x=<optimized out>, y=<optimized
out>, dst=<optimized out>) at frameworks/base/core/jni/android/graphics/Region.cpp:157
#15 0x0000000072eb49bc in android.graphics.Region.translate [DEDUPED] () from
./symbols/system/framework/arm64/boot-framework.oat
#16 0x00000000733e23f0 in android.graphics.Region.translate () from ./symbols/system/framework/arm64/boot-
framework.oat
#17 0x00000073835d8ec4 in ?? ()


(gdb) fr 8
#8  0x000000739a0a0a2c in art::ScopedThreadStateChange::~ScopedThreadStateChange (this=<optimized out>) at
art/runtime/scoped_thread_state_change-inl.h:72
72      in art/runtime/scoped_thread_state_change-inl.h

(gdb) info reg
x0             0x0      0
x1             0x946    2374
x2             0x6      6
x3             0x8      8
x4             0x737ee9ba40     496050485824
x5             0x737ee9ba40     496050485824
x6             0x737ee9ba40     496050485824
x7             0x737ee9bb80     496050486144
x8             0x83     131
x9             0x945feafa4384f3d2       -7755221672713194542
x10            0x0      0
x11            0xffffffffc7ffffbdf      -15032386593
x12            0x1      1
x13            0x737ee9c440     496050488384
x14            0x0      0
x15            0x0      0
x16            0x741d3dc2c8     498706793160
x17            0x741d30ddb8     498705948088
x18            0x1      1
x19            0x7371ed7fa0     495832629152
x20            0x739a685000     496511766528
x21            0x5b     91
```

```
x22            0x0     0
x23            0x6     6
x24            0x739a52b7fa    496510351354
x25            0x739a685000    496511766528
x26            0x739a529a63    496510343779
x27            0x73814a3588    496090363272
x28            0x43    67
x29            0x73814a1a90    496090356368
x30            0x739a0a0a2c    496505588268
sp            0x73814a19c0    0x73814a19c0
pc            0x739a0a0a2c    0x739a0a0a2c <art::(anonymous namespace)::CheckJNI::GetField(char const*, _JNIEnv*,
_jobject*, _jfieldID*, bool, art::Primitive::Type)+1904>
cpsr           0x60000000      1610612736
fpsr           0x0     0
fpcr           0x0     0


(gdb) x/40x 0x73814a19c0
0x73814a19c0:   0x718e8190      0x00000000      0x814a3588      0x00000073
0x73814a19d0:   0x00000000      0x00000000      0x71ed7fa0      0x00000073
0x73814a19e0:   0x9a52b7fa      0x00000073      0x00000000      0x00010000
0x73814a19f0:   0x9a685000      0x00000073      0x9a6e2920      0x00000073
0x73814a1a00:   0x9a71f2c0      0x00000073      0x9a685000      0x00000073
0x73814a1a10:   0xd5b79d85      0x0000005b      0x1d374300      0x00000074
0x73814a1a20:   0x9a6e2920      0x00000073      0x814a1b44      0x00000073
0x73814a1a30:   0x7114410c      0x00000000      0x4384f3d2      0x945feafa
0x73814a1a40:   0x12ca8230      0x00000000      0x00000000      0x00000000
0x73814a1a50:   0x00000000      0x00000000      0x1dc57000      0x00000074


(gdb) x/40x 0x73814a19c0 + 80
0x73814a1a10:   0xd5b79d85      0x0000005b      0x1d374300      0x00000074
0x73814a1a20:   0x9a6e2920      0x00000073      0x814a1b44      0x00000073
0x73814a1a30:   0x7114410c      0x00000000      0x4384f3d2      0x945feafa
0x73814a1a40:   0x12ca8230      0x00000000      0x00000000      0x00000000
0x73814a1a50:   0x00000000      0x00000000      0x1dc57000      0x00000074
0x73814a1a60:   0x814a3588      0x00000073      0x12ca93d0      0x00000000
0x73814a1a70:   0x9a6e2920      0x00000073      0x00000000      0x00000000
0x73814a1a80:   0x00000000      0x00000000      0x814a3588      0x00000073
0x73814a1a90:   0x814a1ac0      0x00000073      0x9a08a9ec      0x00000073
0x73814a1aa0:   0x00000000      0x00000000      0x4384f3d2      0x945feafa
```

---

ma...@google.com <ma...@google.com> #8 Mar 19, 2020 07:07AM

The fields in the ScopedThreadStateChange are:
```
  Thread* const self_ = nullptr;
  const ThreadState thread_state_ = kTerminated;
  ThreadState old_thread_state_ = kTerminated;
```

Based on the asm you provided, [sp + 72] is self_ and the cbz is probably the UNLIKELY(self_ == nullptr).

I suspect these are the values of the ScopedThreadStateChange class:
[sp + 72]: self_ = 0x000000739a685000
[sp + 80]: thread_state_ = 0xd5b79d85 (corrupted)
[sp + 84]: old_thread_state_ = 0x0000005b (kNative as expected)
[sp + 88]: expected_has_no_thread_ = 0x1d374300 (false, or maybe corrupted??)

---

ma...@google.com <ma...@google.com> #9 Mar 21, 2020 05:01AM

Reassigned to zj...@gmail.com.

For the first trace, the corruption happens after getting a field value through JNI. I find it surprising that this code path could corrupt the stack, maybe there is a hardware or kernel issue? What are your thoughts

backtrace:
```
    #00 pc 0000000000022d8c  /system/lib64/libc.so (abort+116)
    #01 pc 000000000048367c  /system/lib64/libart.so (art::Runtime::Abort(char const*)+1196)
    #02 pc 000000000055d6f4  /system/lib64/libart.so
(_ZNSt3__110__function6__funcIPFvPKcENS_9allocatorIS5_EES4_EclEOS3_+36)
    #03 pc 0000000000009104  /system/lib64/libbase.so (android::base::LogMessage::~LogMessage()+724)
    #04 pc 00000000000e1978  /system/lib64/libart.so (art::Thread::SetState(art::ThreadState)+376)
    #05 pc 0000000000106a28  /system/lib64/libart.so (art::(anonymous namespace)::CheckJNI::GetField(char const*,
_JNIEnv*, _jobject*, _jfieldID*, bool, art::Primitive::Type)+1900)
    #06 pc 00000000000f09e8  /system/lib64/libart.so (art::(anonymous namespace)::CheckJNI::GetLongField(_JNIEnv*,
_jobject*, _jfieldID*)+68)
```

---

zj...@gmail.com <zj...@gmail.com> #10 Mar 22, 2020 10:32PM

We check the data between the sp to sp + 72, the data is all correct
and our conclusion is the same, for the first trace, thread_state_ and expected_has_no_thread_ maybe corrupted

for the second trace， we only collected the tombstone， according to the tombstone

```
   #04  0000006f2465b980  0000006f3d8f0680  [anon:libc_malloc]
       ........  ........
   #05  0000006f2465ba00  4342d65bd821932a

art/runtime/check_jni.cc:3254
  102fd8:      a9ba6ffc       stp     x28, x27, [sp,#-96]!
  102fdc:      a90167fa       stp     x26, x25, [sp,#16]
  102fe0:      a9025ff8       stp     x24, x23, [sp,#32]
  102fe4:      a90357f6       stp     x22, x21, [sp,#48]
  102fe8:      a9044ff4       stp     x20, x19, [sp,#64]
  102fec:      a9057bfd       stp     x29, x30, [sp,#80]
  102ff0:      910143fd       add     x29, sp, #0x50


~ScopedThreadStateChange():
art/runtime/scoped_thread_state_change-inl.h:58
  103af4:      f85903b4       ldur    x20, [x29,#-112]
~VarArgs():
art/runtime/check_jni.cc:176
  103af8:      b85303a8       ldur    w8, [x29,#-208]
~ScopedThreadStateChange():
art/runtime/scoped_thread_state_change-inl.h:58
  103afc:      b4001bf4       cbz     x20, 103e78
<_ZN3art12_GLOBAL__N_18CheckJNI11CallMethodVEPKcP7_JNIEnvP8_jobjectP7_jclassP10_jmethodIDSt9__va_listNS_9
Primitive4TypeENS_10InvokeTypeE+0xea0>
art/runtime/scoped_thread_state_change-inl.h:65
  103b00:      297357a8       ldp     w8, w21, [x29,#-104]
  103b04:      6b0802bf       cmp     w21, w8
  103b08:      540015c0       b.eq    103dc0
<_ZN3art12_GLOBAL__N_18CheckJNI11CallMethodVEPKcP7_JNIEnvP8_jobjectP7_jclassP10_jmethodIDSt9__va_listNS_9
Primitive4TypeENS_10InvokeTypeE+0xde8>
art/runtime/scoped_thread_state_change-inl.h:66
  103b0c:      71010ebf       cmp     w21, #0x43
  103b10:      54000d81       b.ne    103cc0
<_ZN3art12_GLOBAL__N_18CheckJNI11CallMethodVEPKcP7_JNIEnvP8_jobjectP7_jclassP10_jmethodIDSt9__va_listNS_9
Primitive4TypeENS_10InvokeTypeE+0xce8>


self = [x29, #-112] = [0x6F2465B9E0] = 0000006f3d885000    the data is correct
thread_state_  = [x29, #-104] = 1  corrupted?
old_thread_state_ = [x29, #-100] = 0  corrupted?
expected_has_no_thread_  =  [x29, #-96]  = 2465bef0  corrupted?

memory near x28 (<anonymous:0000006f24561000>):
   0000006f2465b968 0000000000000043 0000006f2465b9f0  C.........e$o...
   0000006f2465b978 0000006f3d1bb97c 0000006f3d8f0680  |..=o......=o...
   0000006f2465b988 0000004300000043 004300002465b900  C...C.....e$..C.
   0000006f2465b998 4342d65bd821932a 0000006f2465bab0  *.!.[.BC..e$o...
   0000006f2465b9a8 0000006f2465c588 0000006f3d885000  ..e$o....P.=o...
   0000006f2465b9b8 0000006f3d66b0db 0000000000000001  ..f=o...........
   0000006f2465b9c8 0000006f2465bf00 0000000000000000  ..e$o...........
   0000006f2465b9d8 000000000000005b 0000006f3d885000  [........P.=o...
   0000006f2465b9e8 0000000000000001 0000006f2465bef0  ..........e$o...
   0000006f2465b9f8 0000006f3d1ddcd4 4342d65bd821932a  ...=o...*.!.[.BC
   0000006f2465ba08 4342d65bd821932a 0000006f2465c060  *.!.[.BC`.e$o...
   0000006f2465ba18 0000006f3d91d280 4342d65bd821932a  ...=o...*.!.[.BC
   0000006f2465ba28 ffffff80ffffffd8 0000006f2465c080  ..........e$o...
   0000006f2465ba38 0000006f2465c010 0000006f2465bfe0  ..e$o.....e$o...
   0000006f2465ba48 ffffff80ffffffd8 0000000000000009  ................
   0000006f2465ba58 0000006f2465c070 0000006f3d885000  p.e$o....P.=o...


the self_  is also stored in the stack, but it doest not cuppruted,  this is very strange to us
why alway the data after the self_  ?
ScopedThreadStateChange get  created and desotry very often and very fast ,
is there exist a software code that may currupt the stack, how can we monitor or grab it ?
```

---

zj...@gmail.com <zj...@gmail.com> #11 Mar 22, 2020 10:44PM

in issure https://issuetracker.google.com/issues/137036159
it looks like the same, this is really strange： there issues, and the self_ is all safe, but the data after self_ is currupted.

```
   #03  0000007d291c8e80  0000007d47cbb500  [anon:libc_malloc]        ........  ........
   #04  0000007d291c8f00  000000000000efb0

0000007d291c8f50 - 112 = 7D291C8EE0

self_  = 0000007d31164400  this it all right
```

but
thread_state_ = [x29, #-104] = 1 corrupted?
old_thread_state_ = [x29, #-100] = 0 corrupted?
expected_has_no_thread_ = [x29, #-96] = 291c8fd0 corrupted?

memory near x28 (<anonymous:0000007d290ce000>):
0000007d291c8e68 0000000000000043 0000007d291c8ef0 C.....)}...
0000007d291c8e78 0000007d63957640 0000007d47cbb500 @v.c}.....G}...
0000007d291c8e88 0000004300000043 0043000000000000 C...C.....C.
0000007d291c8e98 3d4383977359af56 0000007d291c9588 V.Ys..C=...)}...
0000007d291c8ea8 0000000000000043 0000007d63ddea83 C.....c}...
0000007d291c8eb8 0000007d63ddf351 0000007d63ddf320 Q..c}... ..c}...
0000007d291c8ec8 0000007d63eabc28 0000007d31164400 (..c}....D.1}...
0000007d291c8ed8 000000000000005b 0000007d31164400 [.....D.1}...
0000007d291c8ee8 0000000000000001 0000007d291c8fd0 .....)}...
0000007d291c8ef8 0000007d63bac148 000000000000efb0 H..c}.....
0000007d291c8f08 0000007de9af8930 0000007d291c9110 0...}.....)}...
0000007d291c8f18 0000007d291c90a0 0000007d291c9070 ...)}...p..)}...
0000007d291c8f28 ffffff80ffffffd8 0000007d291c9060 .....`..)}...
0000007d291c8f38 0000007d00430000 005b000000000000 ..C.}.....[.
0000007d291c8f48 0000007d31164400 0000007d40397d40 .D.1}...@}9@}...
0000007d291c8f58 0000007d644c8f40 0000007d31164400 @.Ld}....D.1}...

zh...@vivo.corp-partner.google.com <zh...@vivo.corp-partner.google.com> #12 May 16, 2020 06:42PM

i have reproduce this issue on Android Q.

pid: 1553, tid: 4180, name: AutobacklightTh >>> system_server <<<
uid: 1000
signal 6 (SIGABRT), code -1 (SI_QUEUE), fault addr --------
Abort message: 'Check failed: old_state_and_flags.as_struct.state != kRunnable (old_state_and_flags.as_struct.state=67, kRunnable=Runnable) '
x0 0000000000000000 x1 0000000000001054 x2 0000000000000006 x3 00000070a68de910
x4 000000726ff1b000 x5 000000726ff1b000 x6 000000726ff1b000 x7 00000000017bc472
x8 00000000000000f0 x9 d68c24e49f883bb1 x10 0000000000000001 x11 0000000000000000
x12 ffffffff0fffffbdf x13 ffffffffa1432b31 x14 000000003644d544 x15 0000007269f31000
x16 0000007269f27738 x17 0000007269f05f20 x18 00000070a5a86000 x19 0000000000000611
x20 0000000000001054 x21 00000000ffffffff x22 0000007173b36500 x23 00000071e7fc5e10
x24 00000071e7fa5e36 x25 00000071e84e4000 x26 00000071e857d258 x27 00000071e84e4000
x28 0000000000000043 x29 00000070a68de9b0
sp 00000070a68de8f0 lr 0000007269eb786c pc 0000007269eb7898

backtrace:
#00 pc 0000000000073898 /apex/com.android.runtime/lib64/bionic/libc.so (abort+160) (BuildId: 6aaa192fa70426ea767b3bcf55b19a30)
#01 pc 00000000004bb268 /apex/com.android.runtime/lib64/libart.so (art::Runtime::Abort(char const*)+2280) (BuildId: 79779e3609ad44f2dc180162e3e0dd20)
#02 pc 000000000000b458 /system/lib64/libbase.so (android::base::LogMessage::~LogMessage()+580) (BuildId: 200bade91ec98d3534054692f2cc1d30)
#03 pc 00000000001762b4 /apex/com.android.runtime/lib64/libart.so (art::Thread::SetState(art::ThreadState)+376) (BuildId: 79779e3609ad44f2dc180162e3e0dd20)
#04 pc 000000000038d5b8 /apex/com.android.runtime/lib64/libart.so (art::JNI::CallObjectMethodV(_JNIEnv*, _jobject*, _jmethodID*, std::__va_list)+1232) (BuildId: 79779e3609ad44f2dc180162e3e0dd20)
#05 pc 0000000000003fcc /apex/com.android.runtime/lib64/libnativehelper.so (_JNIEnv::CallObjectMethod(_jobject*, _jmethodID*, ...)+116) (BuildId: 86b4f5d7c011240efd5893b503eba78e)
#06 pc 00000000001a55bc /system/lib64/libandroid_runtime.so ((anonymous namespace)::Receiver::handleEvent(int, int, void*)+92) (BuildId: e4addbacdc15b7f400d579d82ac83914)
#07 pc 0000000000018370 /system/lib64/libutils.so (android::Looper::pollInner(int)+832) (BuildId: f7c8a354465b908ebfc4497b6d157cac)
#08 pc 0000000000017f90 /system/lib64/libutils.so (android::Looper::pollOnce(int, int*, int*, void**)+56) (BuildId: f7c8a354465b908ebfc4497b6d157cac)
#09 pc 000000000013b884 /system/lib64/libandroid_runtime.so (android::android_os_MessageQueue_nativePollOnce(_JNIEnv*, _jobject*, long, int)+44) (BuildId: e4addbacdc15b7f400d579d82ac83914)
#10 pc 00000000002b6c0c /system/framework/arm64/boot-framework.oat (art_jni_trampoline+140) (BuildId: af0f2d0ff398f6ee02386ce8ed32f393f605e591)
#11 pc 0000000000769534 /system/framework/arm64/boot-framework.oat (android.os.MessageQueue.next+228) (BuildId: af0f2d0ff398f6ee02386ce8ed32f393f605e591)
#12 pc 00000000007661a8 /system/framework/arm64/boot-framework.oat (android.os.Looper.loop+680) (BuildId: af0f2d0ff398f6ee02386ce8ed32f393f605e591)
#13 pc 0000000000764ee4 /system/framework/arm64/boot-framework.oat (android.os.HandlerThread.run+612) (BuildId: af0f2d0ff398f6ee02386ce8ed32f393f605e591)
#14 pc 0000000000137334 /apex/com.android.runtime/lib64/libart.so (art_quick_invoke_stub+548) (BuildId: 79779e3609ad44f2dc180162e3e0dd20)
#15 pc 0000000000169eac /apex/com.android.runtime/lib64/libart.so (art::ArtMethod::Invoke(art::Thread*, unsigned int*, unsigned int, art::JValue*, char const*)+244) (BuildId: 79779e3609ad44f2dc180162e3e0dd20)
#16 pc 00000000004b2b28 /apex/com.android.runtime/lib64/libart.so (art::(anonymous namespace)::InvokeWithArgArray(art::ScopedObjectAccessAlreadyRunnable const&, art::ArtMethod*, art::(anonymous namespace)::ArgArray*, art::JValue*, char const*)+104) (BuildId: 79779e3609ad44f2dc180162e3e0dd20)
#17 pc 00000000004b3c3c /apex/com.android.runtime/lib64/libart.so (art::InvokeVirtualOrInterfaceWithJValues(art::ScopedObjectAccessAlreadyRunnable const&, _jobject*, _jmethodID*, jvalue const*)+416) (BuildId: 79779e3609ad44f2dc180162e3e0dd20)
#18 pc 00000000004f477c /apex/com.android.runtime/lib64/libart.so (art::Thread::CreateCallback(void*)+1176)

(BuildId: 79779e3609ad44f2dc180162e3e0dd20)
#19 pc 00000000000d6eb0 /apex/com.android.runtime/lib64/bionic/libc.so (__pthread_start(void*)+36) (BuildId: 6aaa192fa70426ea767b3bcf55b19a30)
#20 pc 0000000000075314 /apex/com.android.runtime/lib64/bionic/libc.so (__start_thread+64) (BuildId: 6aaa192fa70426ea767b3bcf55b19a30)

deleted
0 B

Restricted

zj...@gmail.com <zj...@gmail.com> #13 May 30, 2020 12:26PM

update : problem sovled. memory curruption

[Deleted User] <[Deleted User]> #14 Jul 17, 2020 05:43PM

As you said "problem sovled. memory curruption", Could you show me why and how to fix it?

am...@google.com <am...@google.com> #15 Nov 10, 2020 07:31PM

deleted

Message last modified on Nov 11, 2020 06:03AM

zb...@gmail.com <zb...@gmail.com> #16 Mar 1, 2023 09:02PM

how to fix