Fatal signal 11 (SIGSEGV), code 1 (SEGV_MAPERR), fault addr 0x4 in tid 15945

+1 Hotlists (10) Mark as Duplicate

Comments (20) Dependencies Duplicates (0) Blocking (0) Resources (6)

Infeasible Bug P3 + [AOSP] assigned adexe s nau

STATUS UPDATE No update yet. Edit

Reporter	va...@gmail.com
Type	Bug
Priority	P3
Severity	S3
Status	Won't fix (Infeasible)
Access	Default access View
Assignee	ad...@google.com
Verifier	--
Collaborators	
CC	ad...@google.com va...@gmail.com
AOSP ID	--
ReportedBy	Developer
Found In	--
Targeted To	--
Verified In	--
In Prod	

DESCRIPTION va...@gmail.com created issue #1 Feb 10, 2021 06:14PM

We are facing following crash after updating our app to Android 10. Our application is working fine with older versions of android.

code 1 (SEGV_MAPERR). Crash backtrace pointing to /apex/com.android.runtime/lib64/libart.so and /system/lib64/libhwui.so

Beginning of Crash 1:-

Fatal signal 11 (SIGSEGV), code 1 (SEGV_MAPERR), fault addr 0x4 in tid 15945

```
E/crash_dump64(16647): unknown process state: t
I/crash_dump64(16647): obtaining output fd from tombstoned, type: kDebuggerdTombstone
I/crash_dump64(16647): performing dump of process 15945 (target tid = 15945)
F/DEBUG   (16647): *** *** *** *** *** *** *** *** *** *** *** *** *** *** *** ***
F/DEBUG   (16647): Build fingerprint: 'samsung/star2ltexx/star2lte:10/QP1A.190711.020/G965FXXU7DTAA:user/release-keys'
F/DEBUG   (16647): Revision: '26'
F/DEBUG   (16647): ABI: 'arm64'
F/DEBUG   (16647): Timestamp: 2021-02-02 08:55:09+0300
F/DEBUG   (16647): pid: 15945, tid: 15945, name: z.tigo.tigoshop  >>> tz.tigo.tigoshop <<<
F/DEBUG   (16647): uid: 10534
F/DEBUG   (16647): signal 11 (SIGSEGV), code 1 (SEGV_MAPERR), fault addr 0x4
F/DEBUG   (16647): Cause: null pointer dereference
F/DEBUG   (16647): Abort message: 'Check failed: found_virtual Didn't find oat method index for virtual method: void
android.Manifest$permission.<init>()'
F/DEBUG   (16647):     x0  00000070f293b338  x1  0000000000000000  x2  000000706b0cc0b8  x3  0000000000000000
F/DEBUG   (16647):     x4  0000000000000000  x5  00000070569f31ed  x6  000000000000000a  x7  000000000000000a
F/DEBUG   (16647):     x8  0000007fecb63400  x9  0000000000000000  x10 0000007fecb63400  x11 00000070f293b360
F/DEBUG   (16647):     x12 000000000a055108  x13 3d7365786574756d  x14 00000000ffffffff  x15 0000000000000000
F/DEBUG   (16647):     x16 000000706aff6ff0  x17 00000070ef69e530  x18 00000070f273c000  x19 00000070f293b338
F/DEBUG   (16647):     x20 0000000000000000  x21 000000706b14b660  x22 00000070f14f9000  x23 0000000000000000
F/DEBUG   (16647):     x24 00000070f293b338  x25 0000000000000001  x26 00000070f1686020  x27 0000000000000000
F/DEBUG   (16647):     x28 0000000000000000  x29 00000070f293b300
F/DEBUG   (16647):     sp  00000070f293b0a0  lr  000000706b506000  pc  000000706b4e8ba4
F/DEBUG   (16647):
F/DEBUG   (16647): backtrace:
F/DEBUG   (16647):       #00 pc 00000000004ddba4  /apex/com.android.runtime/lib64/libart.so
(_ZN3art12StackVisitor9WalkStackILNS0_16CountTransitionsE0EEEvb+1556) (BuildId: 9073c75c7bcb19eca4fe361a4c68592f)
F/DEBUG   (16647):       #01 pc 00000000004faffc  /apex/com.android.runtime/lib64/libart.so
(art::Thread::DumpStack(std::__1::basic_ostream<char, std::__1::char_traits<char>>&, bool, BacktraceMap*, bool) const+468) (BuildId:
9073c75c7bcb19eca4fe361a4c68592f)
F/DEBUG   (16647):       #02 pc 0000000000515434  /apex/com.android.runtime/lib64/libart.so
(art::DumpCheckpoint::Run(art::Thread*)+820) (BuildId: 9073c75c7bcb19eca4fe361a4c68592f)
F/DEBUG   (16647):       #03 pc 000000000050e7a0  /apex/com.android.runtime/lib64/libart.so
(art::ThreadList::RunCheckpoint(art::Closure*, art::Closure*)+528) (BuildId: 9073c75c7bcb19eca4fe361a4c68592f)
F/DEBUG   (16647):       #04 pc 000000000050d94c  /apex/com.android.runtime/lib64/libart.so
(art::ThreadList::Dump(std::__1::basic_ostream<char, std::__1::char_traits<char>>&, bool)+1260) (BuildId:
9073c75c7bcb19eca4fe361a4c68592f)
F/DEBUG   (16647):       #05 pc 00000000004bac10  /apex/com.android.runtime/lib64/libart.so (art::Runtime::Abort(char const*)+1352)
(BuildId: 9073c75c7bcb19eca4fe361a4c68592f)
F/DEBUG   (16647):       #06 pc 000000000000c650  /system/lib64/libbase.so (android::base::LogMessage::~LogMessage()+608) (BuildId:
74e39b9e4bda61561a36377476803040)
F/DEBUG   (16647):       #07 pc 0000000000147090  /apex/com.android.runtime/lib64/libart.so
(_ZN3artL16FindOatMethodForEPNS_9ArtMethodENS_11PointerSizeEPb.llvm.6985246053800691335+608) (BuildId:
9073c75c7bcb19eca4fe361a4c68592f)
F/DEBUG   (16647):       #08 pc 0000000000146cc8  /apex/com.android.runtime/lib64/libart.so
(art::ArtMethod::GetOatQuickMethodHeader(unsigned long)+280) (BuildId: 9073c75c7bcb19eca4fe361a4c68592f)
F/DEBUG   (16647):       #09 pc 00000000001efae0  /apex/com.android.runtime/lib64/libart.so
(art::FaultManager::IsInGeneratedCode(siginfo*, void*, bool)+896) (BuildId: 9073c75c7bcb19eca4fe361a4c68592f)
F/DEBUG   (16647):       #10 pc 00000000001ef3c4  /apex/com.android.runtime/lib64/libart.so (art::FaultManager::HandleFault(int,
siginfo*, void*)+92) (BuildId: 9073c75c7bcb19eca4fe361a4c68592f)
F/DEBUG   (16647):       #11 pc 0000000000004dd4  /system/bin/app_process64 (art::SignalChain::Handler(int, siginfo*, void*)+588)
(BuildId: a569457735bdeff7f71efb40991cc89e)
```

Beginning of Crash 2:-

Fatal signal 11 (SIGSEGV), code 1 (SEGV_MAPERR), fault addr 0x4551d4927d7848 in tid 22924

```
E/crash_dump64(23333): unknown process state: t
I/crash_dump64(23333): obtaining output fd from tombstoned, type: kDebuggerdTombstone
```

```
I/crash_dump64(23333): performing dump of process 22860 (target tid = 22924)
F/DEBUG   (23333): *** *** *** *** *** *** *** *** *** *** *** *** *** *** *** ***
F/DEBUG   (23333): Build fingerprint: 'samsung/star2ltexx/star2lte:10/QP1A.190711.020/G965FXXU7DTAA:user/release-keys'
F/DEBUG   (23333): Revision: '26'
F/DEBUG   (23333): ABI: 'arm64'
F/DEBUG   (23333): Timestamp: 2021-02-02 11:55:06+0300
F/DEBUG   (23333): pid: 22860, tid: 22924, name: RenderThread  >>> tz.tigo.tigoshop <<<
F/DEBUG   (23333): uid: 10536
F/DEBUG   (23333): signal 11 (SIGSEGV), code 1 (SEGV_MAPERR), fault addr 0x4551d4927d7848
F/DEBUG   (23333):     x0  0000007056a04d80  x1  0000006fe1674290  x2  0000006fe1674b20  x3  0000000000000000
F/DEBUG   (23333):     x4  0000006fe16744a0  x5  2c01000090010000  x6  00000070f2941000  x7  00000000002bf056
F/DEBUG   (23333):     x8  00000070547e2a48  x9  0000000000000000  x10 0000000000000019  x11 000000000000001a
F/DEBUG   (23333):     x12 0000000000000033  x13 000000000000001a  x14 6c4551d4927d7068  x15 6c4551d4927d7068
F/DEBUG   (23333):     x16 0000000000000050  x17 00000070ed9adff0  x18 0000006fe14da000  x19 0000006fe1674290
F/DEBUG   (23333):     x20 0000006fe1675020  x21 6c4551d4927d7888  x22 6c4551d4927d7838  x23 00000070547e2a48
F/DEBUG   (23333):     x24 0000006fe1675020  x25 0000000000000001  x26 0000006fe1675020  x27 0000000000000000
F/DEBUG   (23333):     x28 0000006fe16745b0  x29 0000006fe1674250
F/DEBUG   (23333):     sp  0000006fe1674210  lr  00000070ed9ac3f4  pc  00000070ed9ac500
F/DEBUG   (23333):
F/DEBUG   (23333): backtrace:
F/DEBUG   (23333):       #00 pc 000000000039a500  /system/lib64/libhwui.so
(android::uirenderer::skiapipeline::SkiaDisplayList::updateChildren(std::__1::function<void (android::uirenderer::RenderNode*)>)+136)
(BuildId: 343ecb7aa186fa3af5b14eab30973292)
F/DEBUG   (23333):       #01 pc 000000000039a3f0  /system/lib64/libhwui.so
(android::uirenderer::RenderNode::syncDisplayList(android::uirenderer::TreeObserver&, android::uirenderer::TreeInfo*)+240) (BuildId:
343ecb7aa186fa3af5b14eab30973292)
F/DEBUG   (23333):       #02 pc 0000000000399874  /system/lib64/libhwui.so
(android::uirenderer::RenderNode::prepareTreeImpl(android::uirenderer::TreeObserver&, android::uirenderer::TreeInfo&, bool)+2020)
(BuildId: 343ecb7aa186fa3af5b14eab30973292)
F/DEBUG   (23333):       #03 pc 000000000039a8fc  /system/lib64/libhwui.so
(android::uirenderer::skiapipeline::SkiaDisplayList::prepareListAndChildren(android::uirenderer::TreeObserver&,
android::uirenderer::TreeInfo&, bool, std::__1::function<void (android::uirenderer::RenderNode*, android::uirenderer::TreeObserver&,
android::uirenderer::TreeInfo&, bool)>)+636) (BuildId: 343ecb7aa186fa3af5b14eab30973292)
F/DEBUG   (23333):       #04 pc 00000000003993d0  /system/lib64/libhwui.so
(android::uirenderer::RenderNode::prepareTreeImpl(android::uirenderer::TreeObserver&, android::uirenderer::TreeInfo&, bool)+832) (BuildId:
343ecb7aa186fa3af5b14eab30973292)
F/DEBUG   (23333):       #05 pc 000000000039a8fc  /system/lib64/libhwui.so
(android::uirenderer::skiapipeline::SkiaDisplayList::prepareListAndChildren(android::uirenderer::TreeObserver&,
android::uirenderer::TreeInfo&, bool, std::__1::function<void (android::uirenderer::RenderNode*, android::uirenderer::TreeObserver&,
android::uirenderer::TreeInfo&, bool)>)+636) (BuildId: 343ecb7aa186fa3af5b14eab30973292)
F/DEBUG   (23333):       #06 pc 00000000003993d0  /system/lib64/libhwui.so
(android::uirenderer::RenderNode::prepareTreeImpl(android::uirenderer::TreeObserver&, android::uirenderer::TreeInfo&, bool)+832) (BuildId:
343ecb7aa186fa3af5b14eab30973292)
F/DEBUG   (23333):       #07 pc 0000000000398ce8  /system/lib64/libhwui.so
(android::uirenderer::RenderNode::prepareTree(android::uirenderer::TreeInfo&)+152) (BuildId: 343ecb7aa186fa3af5b14eab30973292)
F/DEBUG   (23333):       #08 pc 0000000000153044  /system/lib64/libandroid_runtime.so
(android::RootRenderNode::prepareTree(android::uirenderer::TreeInfo&)+316) (BuildId: 97c11c0a9e40704eea4a584db87b34e1)
F/DEBUG   (23333):       #09 pc 0000000000407544  /system/lib64/libhwui.so
(android::uirenderer::renderthread::CanvasContext::prepareTree(android::uirenderer::TreeInfo&, long*, long,
android::uirenderer::RenderNode*)+316) (BuildId: 343ecb7aa186fa3af5b14eab30973292)
F/DEBUG   (23333):       #10 pc 00000000004071d8  /system/lib64/libhwui.so
(android::uirenderer::renderthread::DrawFrameTask::syncFrameState(android::uirenderer::TreeInfo&)+176) (BuildId:
343ecb7aa186fa3af5b14eab30973292)
F/DEBUG   (23333):       #11 pc 0000000000406cc8  /system/lib64/libhwui.so
(_ZNSt3__110__function6__funcIZN7android10uirenderer12renderthread13DrawFrameTask11postAndWaitEvE3$_0NS_9allocatorIS6_EEFv
vEEclEv$c303f2d2360db58ed70a2d0ac7ed911b+104) (BuildId: 343ecb7aa186fa3af5b14eab30973292)
F/DEBUG   (23333):       #12 pc 0000000000417a44  /system/lib64/libhwui.so (android::uirenderer::WorkQueue::process()+228) (BuildId:
343ecb7aa186fa3af5b14eab30973292)
F/DEBUG   (23333):       #13 pc 0000000000417770  /system/lib64/libhwui.so
(android::uirenderer::renderthread::RenderThread::threadLoop()+80) (BuildId: 343ecb7aa186fa3af5b14eab30973292)
F/DEBUG   (23333):       #14 pc 00000000000137a4  /system/lib64/libutils.so (android::Thread::_threadLoop(void*)+284) (BuildId:
e401a05bdd74f2cd876793e31ceba528)
```

Thanks & Regards,
Rahul Vaghani

✓ Links (5)

Hide all

“For steps to capture a bug report, please refer: https://developer.android.com/studio/debug/bug-report#bugreportdevice” ad...@ #2, ad...@ #9
“https://drive.google.com/file/d/1gFPSbwq9zNzggJnt-I5UIG8ZRiB_sNns… ” va...@ #4
“Bug Report - https://drive.google.com/file/d/1f100sboeAklCy0rPDc_swW4TKyqAa7Uo/view?usp… ” 07...@ #10
“https://drive.google.com/file/d/1Sbua29vCa0su2ERdKSV68sNW3imQ4IP… ” va...@ #11
“Please find link https://docs.google.com/document/d/1IYWRKZn7xfArCwyilwgviJwtB1xkVknDmuPZ1Vz-PQ4/edit?us…” va...@ #17

COMMENTS

All comments ▾ ↓ Oldest first

ad...@google.com <ad...@google.com> #2 Feb 10, 2021 07:12PM ⋮

Assigned to ad...@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 do others need to take in order to reproduce the issue themselves?

Are you able to reproduce the issue on Pixel device or Android Emulator as well?

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” option.

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

yo...@gmail.com <yo...@gmail.com> #3 Feb 10, 2021 07:36PM

So I am on the list on the phone for blacklisted for my email and I want to know what the hell happened and why I can't use half of my phone features and why this is popping up with all this crap I'm confused this living hell right now somebody please help me figure it out what is going on

va...@gmail.com <va...@gmail.com> #4 Feb 10, 2021 10:33PM

Hi,

Please find the link for bug report

https://drive.google.com/file/d/1gFPSbwq9zNzggJnt-I5UIG8ZRiB_sNns/view?usp=sharing

Thanks

ad...@google.com <ad...@google.com> #5 Feb 11, 2021 05:16PM

Also share steps & other info requested in comment #2.

va...@gmail.com <va...@gmail.com> #6 Feb 11, 2021 10:27PM

Hi,

We can able to reproduce an issue after connecting fingerprint scanner with application. Same thing is working fine upto android os version 9. Its crashing only in Android 10. Pixel device is not available with us and on Emulator we can not able to test as we need to connect external device with application.

Thanks

ad...@google.com <ad...@google.com> #7 Feb 12, 2021 08:24PM

Please provide sample project and apk to reproduce the issue. Also mention the steps to be followed for reproducing the issue with the given sample project and apk.

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

va...@gmail.com <va...@gmail.com> #8 Feb 12, 2021 08:44PM

Hi,

Thanks for your reply.

We can provide you the APK but it will not gonna help as it require fingerprint device too reproduce the issue. If you have Morpho fingerprint scanner than we can provide the APK as well.

If you want we can provide you debug logs as well as bug report with crash.

Thanks.

ad...@google.com <ad...@google.com> #9 Feb 15, 2021 09:00PM

Yes, please share the sample APK/ project. Also share :

Android build
Which Android build are you using? (e.g. KVT49L)

Device used
Which device did you use 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” option.

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

07...@gmail.com <07...@gmail.com> #10 Feb 15, 2021 10:51PM

I am also facing the issue with Android 10 while connecting the Fingerprint device with it. After connecting the fingerprint device I am not able to unlock the phone, it's getting freeze in unlock the screen. A bug report is attached please check.

Bug Report - https://drive.google.com/file/d/1f100sboeAklCy0rPDc_swW4TKyqAa7Uo/view?usp=sharing

Thanks

va...@gmail.com <va...@gmail.com> #11 Feb 15, 2021 11:07PM

Hi,

Please find below link for APK.

https://drive.google.com/file/d/1Sbua29vCa0su2ERdKSV68sNW3imQ4IPC/view?usp=sharing

Device used
Which device did you use to reproduce this issue?.
samsung Model: SM-G965F

Thanks,
Rahul.

ad...@google.com <ad...@google.com> #12 Feb 16, 2021 11:08PM

Also mention the steps to be followed for reproducing the issue with the given sample project and apk.

Is the bug report shared in comment #10 relevant to the APK shared in comment #11?

va...@gmail.com <va...@gmail.com> #13 Feb 17, 2021 06:32PM

Also mention the steps to be followed for reproducing the issue with the given sample project and apk.

To reproduce the steps you need to do registration to the app and required Morpho scanner device to attach with phone.
Can you please provide us E-Mail ID on which we can share credentials and Test APK?

Is the bug report shared in comment #10 relevant to the APK shared in comment #11?
No its not.

ad...@google.com <ad...@google.com> #14 Feb 18, 2021 03:03AM

You can share the credentials in a Google Doc & share it with android-bugreport@google.com. Also mention steps to be followed for reproducing the issue here OR in the doc itself.

Test APK is the same one as comment #11? If not, then please share it by putting it in a folder in Google Drive & sharing it with android-bugreport@google.com

Also share a bug report (see comment #9).

va...@gmail.com <va...@gmail.com> #15 Feb 18, 2021 04:35PM

Hi,

Shared details with given mail Id.

Thanks.

ad...@google.com <ad...@google.com> #16 Feb 19, 2021 02:13AM

Please also share the Google Drive & Google Doc link here.

va...@gmail.com <va...@gmail.com> #17 Feb 22, 2021 03:39PM

Hi,

Please find link https://docs.google.com/document/d/1IYWRKZn7xfArCwyilwgviJwtB1xkVknDmuPZ1Vz-PQ4/edit?usp=sharing

ad...@google.com <ad...@google.com> #18 Feb 24, 2021 05:59AM

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

ad...@google.com <ad...@google.com> #19 Feb 25, 2021 09:13PM

Response from the engineering team (regarding Crash 1) :

> We think this is an app issue that is unfortunately crashing ART. In all likelihood some JNI code in the app is raising an exception that is propagated through ART's fault handler code. The fault handler code assumes the thread is in a runnable state (`ThreadState::kRunnable`) rather than executing JNI code (`ThreadState::kNative`) and ends up with the null pointer dereference.
>
> Until we have an ART Mainline Module, we have no way of fixing issues like this in Android devices that have shipped.
>
> The developer will have to identify where the crash is triggered in the apps JNI code and fixing it there. That's going to be a case of adding instrumentation to the source code to find where in JNI code triggering this issue is.

ad...@google.com <ad...@google.com> #20 May 8, 2021 02:22AM

Status: Won't Fix (Infeasible)

Regarding Crash 2 :
Since the crash is from a Samsung device we can't symbolize the stack & we don't know the source. Unless this is reproducible in the emulator or on a Pixel device there's not much we can do.