uint16_t overflow in batterystats module.

DESCRIPTION qi...@xiaomi.corp-partner.google.com created issue #1

1. Problem Description

system server crash during startup after OTA:

```
*** *** *** *** *** *** *** *** *** *** *** *** *** *** *** ***
Build fingerprint: 'Redmi/munch/munch:13/TKQ1.220710.001/22.8.15:user/release-keys'
Revision: '0'
ABI: 'arm64'
Timestamp: 2022-08-15 09:44:26.304362364+0800
Process uptime: 2s
Cmdline: system_server
pid: 3566, tid: 3566, name: system_server  >>> system_server <<<
uid: 1000
signal 11 (SIGSEGV), code 2 (SEGV_ACCERR), fault addr 0x00000070a162d008
    x0  0000000000000000  x1  0000000000067424  x2  0000000000000004  x3  0000000000080000
    x4  0000006e2146b650  x5  000000715c756214  x6  0000000000010002  x7  0000000000010002
    x8  b4000070a162d000  x9  0000000000000000  x10 000000006e4265f0  x11 0000000000000006
    x12 00000000a15d2f30  x13 0000000000080010  x14 0000007fe29472e0  x15 0000000034155555
    x16 00000071417e0f00  x17 00000071454769ac  x18 000000715eabc000  x19 b400006ee1470b10
    x20 b400006e414f53d0  x21 b400006e71482490  x22 000000713b700d75  x23 000000713b70da4a
    x24 00000000000065f0  x25 000000715e5f5000  x26 000000000000bffe  x27 0000007fe2947308
    x28 0000007fe2947330  x29 0000007fe29471e0
    lr  000000713b8683b4  sp  0000007fe29471c0  pc  000000713b86839c  pst 0000000020001000

backtrace:
      #00 pc 00000000001d839c  /system/lib64/libandroid_runtime.so (android::native_initFromParcel(_JNIEnv*, _jclass*, _jobject*)+460) (BuildId: 2fd0487bf2f5
      #01 pc 00000000001cb988  /system/framework/arm64/boot-framework.oat (art_jni_trampoline+104) (BuildId: bc051b7895b41ed548911eff3686e23f083f68cd)
      #02 pc 0000000000209398  /apex/com.android.art/lib64/libart.so (nterp_helper+152) (BuildId: 4d3a87f8c3a770c16045d167de17db52)
      #03 pc 000000000040c86e  /system/framework/framework.jar (com.android.internal.os.LongMultiStateCounter.<init>+6)
      #04 pc 0000000000020a254  /apex/com.android.art/lib64/libart.so (nterp_helper+3924) (BuildId: 4d3a87f8c3a770c16045d167de17db52)
      #05 pc 000000000040c8a4  /system/framework/framework.jar (com.android.internal.os.LongMultiStateCounter.<init>+0)
      #06 pc 0000000000020a254  /apex/com.android.art/lib64/libart.so (nterp_helper+3924) (BuildId: 4d3a87f8c3a770c16045d167de17db52)
      #07 pc 000000000040c682  /system/framework/framework.jar (com.android.internal.os.LongMultiStateCounter$1.createFromParcel+6)
      #08 pc 0000000000020a254  /apex/com.android.art/lib64/libart.so (nterp_helper+3924) (BuildId: 4d3a87f8c3a770c16045d167de17db52)
      #09 pc 000000000040c6b4  /system/framework/framework.jar (com.android.internal.os.LongMultiStateCounter$1.createFromParcel+0)
      #10 pc 0000000000020b074  /apex/com.android.art/lib64/libart.so (nterp_helper+7540) (BuildId: 4d3a87f8c3a770c16045d167de17db52)
      #11 pc 00000000003e05b8  /system/framework/framework.jar (com.android.internal.os.BatteryStatsImpl$TimeMultiStateCounter.readFromParcel+4)
      #12 pc 0000000000209334  /apex/com.android.art/lib64/libart.so (nterp_helper+52) (BuildId: 4d3a87f8c3a770c16045d167de17db52)
      #13 pc 00000000003e0598  /system/framework/framework.jar (com.android.internal.os.BatteryStatsImpl$TimeMultiStateCounter.-$$Nest$smreadFromParcel+0)
      #14 pc 0000000000209334  /apex/com.android.art/lib64/libart.so (nterp_helper+52) (BuildId: 4d3a87f8c3a770c16045d167de17db52)
      #15 pc 00000000003dd312  /system/framework/framework.jar (com.android.internal.os.BatteryStatsImpl$ControllerActivityCounterImpl.readTimeMultiStateCoun
      #16 pc 0000000000020a254  /apex/com.android.art/lib64/libart.so (nterp_helper+3924) (BuildId: 4d3a87f8c3a770c16045d167de17db52)
      #17 pc 00000000003dd4e4  /system/framework/framework.jar (com.android.internal.os.BatteryStatsImpl$ControllerActivityCounterImpl.readSummaryFromParcel+
      #18 pc 0000000000846c20  /system/framework/arm64/boot-framework.oat (com.android.internal.os.BatteryStatsImpl.readSummaryFromParcel+4176) (BuildId: bc0
      #19 pc 0000000000020a2b0  /apex/com.android.art/lib64/libart.so (nterp_helper+4016) (BuildId: 4d3a87f8c3a770c16045d167de17db52)
      #20 pc 00000000003f4ec0  /system/framework/framework.jar (com.android.internal.os.BatteryStatsImpl.readLocked+140)
      #21 pc 0000000000020a254  /apex/com.android.art/lib64/libart.so (nterp_helper+3924) (BuildId: 4d3a87f8c3a770c16045d167de17db52)
      #22 pc 000000000030b52a  /system/framework/services.jar (com.android.server.am.ActivityManagerService.<init>+1398)
      #23 pc 0000000000020a254  /apex/com.android.art/lib64/libart.so (nterp_helper+3924) (BuildId: 4d3a87f8c3a770c16045d167de17db52)
      #24 pc 00000000002f7ece  /system/framework/services.jar (com.android.server.am.ActivityManagerService$Lifecycle.<init>+14)
      #25 pc 000000000021096c  /apex/com.android.art/lib64/libart.so (art_quick_invoke_stub+556) (BuildId: 4d3a87f8c3a770c16045d167de17db52)
      #26 pc 0000000000027b478  /apex/com.android.art/lib64/libart.so (art::ArtMethod::Invoke(art::Thread*, unsigned int*, unsigned int, art::JValue*, char co
      #27 pc 00000000006125d4  /apex/com.android.art/lib64/libart.so (art::InvokeConstructor(art::ScopedObjectAccessAlreadyRunnable const&, art::ArtMethod*,
      #28 pc 00000000005848dc  /apex/com.android.art/lib64/libart.so (art::Constructor_newInstance0(_JNIEnv*, _jobject*, _jobjectArray*)+468) (BuildId: 4d3a8
      #29 pc 0000000000095368  /system/framework/arm64/boot.oat (art_jni_trampoline+104) (BuildId: 418107fb1ea0ec9212f1b6d7cfaf912b4f648dc7)
      #30 pc 00000000002da3c8  /system/framework/arm64/boot.oat (java.lang.reflect.Constructor.newInstance+72) (BuildId: 418107fb1ea0ec9212f1b6d7cfaf912b4f64
      #31 pc 0000000000020a2b0  /apex/com.android.art/lib64/libart.so (nterp_helper+4016) (BuildId: 4d3a87f8c3a770c16045d167de17db52)
      #32 pc 0000000000278ae6  /system/framework/services.jar (com.android.server.SystemServiceManager.startService+166)
      #33 pc 000000000021096c  /apex/com.android.art/lib64/libart.so (art_quick_invoke_stub+556) (BuildId: 4d3a87f8c3a770c16045d167de17db52)
      #34 pc 0000000000027b478  /apex/com.android.art/lib64/libart.so (art::ArtMethod::Invoke(art::Thread*, unsigned int*, unsigned int, art::JValue*, char co
      #35 pc 00000000003e7af0  /apex/com.android.art/lib64/libart.so (art::interpreter::ArtInterpreterToCompiledCodeBridge(art::Thread*, art::ArtMethod*, art
      #36 pc 00000000003e2ac0  /apex/com.android.art/lib64/libart.so (bool art::interpreter::DoCall<false, true>(art::ArtMethod*, art::Thread*, art::ShadowFr
      #37 pc 0000000000024bbe8  /apex/com.android.art/lib64/libart.so (void art::interpreter::ExecuteSwitchImplCpp<true, false>(art::interpreter::SwitchImplCo
      #38 pc 000000000021cbd8  /apex/com.android.art/lib64/libart.so (ExecuteSwitchImplAsm+8) (BuildId: 4d3a87f8c3a770c16045d167de17db52)
```

```
#39 pc 00000000002f7e90  /system/framework/services.jar (com.android.server.am.ActivityManagerService$Lifecycle.startService+0)
#40 pc 00000000003daf94  /apex/com.android.art/lib64/libart.so (art::interpreter::Execute(art::Thread*, art::CodeItemDataAccessor const&, art::ShadowFr
#41 pc 000000000071fcfc  /apex/com.android.art/lib64/libart.so (artQuickToInterpreterBridge+740) (BuildId: 4d3a87f8c3a770c16045d167de17db52)
#42 pc 000000000021a488  /apex/com.android.art/lib64/libart.so (art_quick_to_interpreter_bridge+88) (BuildId: 4d3a87f8c3a770c16045d167de17db52)
#43 pc 0000000000209398  /apex/com.android.art/lib64/libart.so (nterp_helper+152) (BuildId: 4d3a87f8c3a770c16045d167de17db52)
#44 pc 0000000000274edc  /system/framework/services.jar (com.android.server.SystemServer.startBootstrapServices+440)
#45 pc 000000000020a254  /apex/com.android.art/lib64/libart.so (nterp_helper+3924) (BuildId: 4d3a87f8c3a770c16045d167de17db52)
#46 pc 0000000000274a48  /system/framework/services.jar (com.android.server.SystemServer.run+932)
#47 pc 000000000020a254  /apex/com.android.art/lib64/libart.so (nterp_helper+3924) (BuildId: 4d3a87f8c3a770c16045d167de17db52)
#48 pc 0000000000274506  /system/framework/services.jar (com.android.server.SystemServer.main+34)
#49 pc 0000000000210c00  /apex/com.android.art/lib64/libart.so (art_quick_invoke_static_stub+576) (BuildId: 4d3a87f8c3a770c16045d167de17db52)
#50 pc 000000000027b4ac  /apex/com.android.art/lib64/libart.so (art::ArtMethod::Invoke(art::Thread*, unsigned int*, unsigned int, art::JValue*, char co
#51 pc 0000000000610b40  /apex/com.android.art/lib64/libart.so (_jobject* art::InvokeMethod<(art::PointerSize)8>(art::ScopedObjectAccessAlreadyRunnable
#52 pc 000000000059065c  /apex/com.android.art/lib64/libart.so (art::Method_invoke(_JNIEnv*, _jobject*, _jobject*, _jobjectArray*)+52) (BuildId: 4d3a87
#53 pc 0000000000099148  /system/framework/arm64/boot.oat (art_jni_trampoline+120) (BuildId: 418107fb1ea0ec9212f1b6d7cfaf912b4f648dc7)
#54 pc 00000000007bddf0  /system/framework/arm64/boot-framework.oat (com.android.internal.os.RuntimeInit$MethodAndArgsCaller.run+144) (BuildId: bc051b7
#55 pc 00000000007c707c  /system/framework/arm64/boot-framework.oat (com.android.internal.os.ZygoteInit.main+3036) (BuildId: bc051b7895b41ed548911eff36
#56 pc 0000000000210c00  /apex/com.android.art/lib64/libart.so (art_quick_invoke_static_stub+576) (BuildId: 4d3a87f8c3a770c16045d167de17db52)
#57 pc 000000000027b4ac  /apex/com.android.art/lib64/libart.so (art::ArtMethod::Invoke(art::Thread*, unsigned int*, unsigned int, art::JValue*, char co
#58 pc 00000000006112c8  /apex/com.android.art/lib64/libart.so (art::JValue art::InvokeWithVarArgs<art::ArtMethod*>(art::ScopedObjectAccessAlreadyRunna
#59 pc 00000000006117b4  /apex/com.android.art/lib64/libart.so (art::JValue art::InvokeWithVarArgs<_jmethodID*>(art::ScopedObjectAccessAlreadyRunnable
#60 pc 00000000004fb21c  /apex/com.android.art/lib64/libart.so (art::JNI<true>::CallStaticVoidMethodV(_JNIEnv*, _jclass*, _jmethodID*, std::__va_list)+
#61 pc 00000000000c0c04  /system/lib64/libandroid_runtime.so (_JNIEnv::CallStaticVoidMethod(_jclass*, _jmethodID*, ...)+124) (BuildId: 2fd0487bf2f512f1
#62 pc 00000000000cd730  /system/lib64/libandroid_runtime.so (android::AndroidRuntime::start(char const*, android::Vector<android::String8> const&, boo
#63 pc 0000000000002610  /system/bin/app_process64 (main+1464) (BuildId: 82486a24ab8170e1b0c9080a100ddae4)
#64 pc 000000000004bc70  /apex/com.android.runtime/lib64/bionic/libc.so (__libc_init+100) (BuildId: 2594db063b42cfbc24ddfd230657b4cf)
```

2. Initial Analysis

Some key code after addr2line:

```
android::battery::MultiStateCounter<long>::setValue(unsigned short, long const&)
frameworks/native/libs/battery/MultiStateCounter.h:191
android::native_initFromParcel(_JNIEnv*, _jclass*, _jobject*)
frameworks/base/core/jni/com_android_internal_os_LongMultiStateCounter.cpp:157
```

```
36template <class T>
37class MultiStateCounter {
..
55    MultiStateCounter(uint16_t stateCount, const T& emptyValue);
```

The first argument is uint16_t, but the parameter is int32:

```
146static jlong native_initFromParcel(JNIEnv *env, jclass theClass, jobject jParcel) {
147    ndk::ScopedAParcel parcel(AParcel_fromJavaParcel(env, jParcel));
148
149    int32_t stateCount;
150    THROW_ON_READ_ERROR(AParcel_readInt32(parcel.get(), &stateCount));
151
152    battery::LongMultiStateCounter *counter = new battery::LongMultiStateCounter(stateCount, 0);
153
154    for (battery::state_t state = 0; state < stateCount; state++) {
155        int64_t value;
156        THROW_ON_READ_ERROR(AParcel_readInt64(parcel.get(), &value));
157        counter->setValue(state, value);
158    }
159
160    return reinterpret_cast<jlong>(counter);
161}
```

So when stateCount is larger than 2^16, heap buffer overflow occurs.

3. Suggestion on Fix

At least a check and exception thrown is required when stateCount is larger than 2^16. I think it's better to use int32 instead of uint16. I'll upload a patch later.

4. Steps to Reproduce

This is related to our OTA between daily builds. And I think it's not important to reproduce. The risk of overflow is obvious.

5. Possibility of Reproducing

High. More than 1%.

6. APK/App Name that you're developing and affected by this issue

None

8. Device Make and Model

Not device related.

9. Android OS version

Android 13

10. Related API or developer doc

None

11. Bugreports on Google Drive

https://drive.google.com/file/d/1B4MYiTs-BGxc0qBnHfiSGZKzlsejUvUQ/view?usp=sharing

✓ Links (2)

Links (2)

"https://drive.google.com/file/d/1B4MYiTs-BGxc0qBnHfiSGZKzlsejUvUQ… "
"Patch: https://android-review.googlesource.com/q/topic:243120…"

COMMENTS

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

Assigned to vi...@google.com.

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

vi...@google.com <vi...@google.com> #3

Meanwhile, you may upload the proposed patch and share with us, thanks.

qi...@xiaomi.corp-partner.google.com <qi...@xiaomi.corp-partner.google.com> #4

Comment has been deleted.

Message last modified on Aug 22, 2022 11:33AM

qi...@xiaomi.corp-partner.google.com <qi...@xiaomi.corp-partner.google.com> #5

Patch: https://android-review.googlesource.com/q/topic:243120067

I'm so sorry that the title is wrong. I created this issue from a similar one and forgot to change the title. Could you help change it to:

uint16_t overflow in batterystats module

Or just be aware of it. Thanks!

qi...@xiaomi.corp-partner.google.com <qi...@xiaomi.corp-partner.google.com> #6

Thanks for changing the title!

I've read more code in this module and now I know stateCount is always very small value. But I still think we should do two things at least:

1. Check the value stateCount after reading from parcel to ensure it is less than 2^16 to avoid overflow.
2. Use type state_t for both member stateCount and currentState in class MultiStateCounter. Like this:

```
//libs/battery/MultiStateCounter.h
typedef uint32_t state_t;  // If value from parcel is checked, uint16_t is also OK.

template <class T>
class MultiStateCounter {
    state_t stateCount;
    state_t currentState;
...
```

They're index and size of same array. So use same type is safer.

dp...@google.com <dp...@google.com> #7

FTR, if the state count is a large number (more than 10), this indicates that we are reading from a corrupt parcel. We also need to check the parcel for validity before creating a multi-state cou

vi...@google.com <vi...@google.com> #8

Marked as fixed.

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