Android Public Tracker > Framework 65355068

repeatedly call registerContentObserver can result in system_server native crash global reference table overflow

+1 1 Hotlists (1) Mark as Duplicate

Comments (8) Dependencies Duplicates (0) Blocking (0) Resources (3)

Fixed Bug P3 + Add Hotlist [AOSP] assigned

STATUS UPDATE No update yet. Edit

Reporter	ai...@xiaor
Type	Bug
Priority	P3
Severity	S3
Status	Fixed
Access	Default acces
Assignee	am...@goo
Verifier	--
Collaborators	
CC	ai...@xiaomi.c
AOSP ID	--
ReportedBy	--
Found In	--
Targeted To	--
Verified In	--
In Prod	

DESCRIPTION ai...@xiaomi.com created issue #1 Sep 5, 2017 04:53PM

- Steps to reproduce the problem (including sample code if appropriate).

Call registerContentObserver again and again in my application

1) Open app-debug.apk in the attached file
2) Click "REBOOT CONTENT TRIGGER" and wait, you can check the register count in the logcat
3) The android system reboot

Here is my sample code:

```
public static void triggerWithContent(final Context context) {
    final ContentResolver contentResolver = context.getContentResolver();
    new Thread(new Runnable() {
        Uri uri = Uri.parse("content://sms/sent");
        @Override
        public void run() {
            while (true) {
                ContentObserver contentObserver = new MyObserver(handler);
                contentResolver.registerContentObserver(uri, true, contentObserver);
            }

        }
    }).start();
}

class MyObserver extends ContentObserver {
    public MyObserver(Handler handler) {
        super(handler);
    }

    @Override
    public void onChange(boolean selfChange) {
        this.onChange(selfChange, null);
    }

    @Override
    public void onChange(boolean selfChange, Uri uri) {
    }
}
```

- What happened.

The android system reboot after I have called registerContentObserver about 24,000 times.

For more details you can also check bugreport called "contentobserver.zip" in the attached files:
Here is the sample output:

```
09-05 14:20:38.899 19105 19105 F DEBUG   : *** *** *** *** *** *** *** *** *** *** *** *** *** *** *** ***
09-05 14:20:38.899 19105 19105 F DEBUG   : Build fingerprint: 'xiaomi/mido/mido:7.0/NRD90M/1.1.1:user/test-keys'
09-05 14:20:38.900 19105 19105 F DEBUG   : Revision: '0'
09-05 14:20:38.900 19105 19105 F DEBUG   : ABI: 'arm64'
09-05 14:20:38.900 19105 19105 F DEBUG   : pid: 1633, tid: 4450, name: Binder:1633_1B  >>> system_server <<<
09-05 14:20:38.900 19105 19105 F DEBUG   : signal 6 (SIGABRT), code -6 (SI_TKILL), fault addr --------
09-05 14:20:38.920 19105 19105 F DEBUG   : Abort message: 'art/runtime/indirect_reference_table.cc:116] JNI ERROR (app bug): global reference table overflow (max=51200)'
09-05 14:20:38.920 19105 19105 F DEBUG   :     x0   0000000000000000  x1   0000000000001162  x2   0000000000000006  x3   0000000000000008
09-05 14:20:38.920 19105 19105 F DEBUG   :     x4   000000000000008c  x5   0000007f78f83880  x6   4437f8787f000000  x7   0000007f78f83744
09-05 14:20:38.920 19105 19105 F DEBUG   :     x8   0000000000000083  x9   ffffffffffffffdf  x10  0000000000000000  x11  0000000000000001
09-05 14:20:38.921 19105 19105 F DEBUG   :     x12  ffffffffffffffff  x13  00000000ffffffff  x14  000000000008cc40  x15  0000000000001fe3
09-05 14:20:38.921 19105 19105 F DEBUG   :     x16  0000007f78fc1ed8  x17  0000007f78f6f4d0  x18  0000000000000000  x19  0000007f41b1d4f8
09-05 14:20:38.921 19105 19105 F DEBUG   :     x20  0000000000000006  x21  0000007f41b1d450  x22  000000000000000b  x23  000000000000389b
09-05 14:20:38.921 19105 19105 F DEBUG   :     x24  ffffffffffffffff  x25  0000007f779fc730  x26  0000007f77985ec8  x27  0000007f41b1b341
09-05 14:20:38.921 19105 19105 F DEBUG   :     x28  0000007f7794c09b  x29  0000007f41b1b270  x30  0000007f78f6c960
09-05 14:20:38.921 19105 19105 F DEBUG   :     sp   0000007f41b1b250  pc   0000007f78f6f4d8  pstate 0000000000000000
```

- What you think the correct behavior should be.

The number of registerContentObserver for each applications can call should be limited, since usually third-party application do not need to register so much observers, so that third-party cannot make android system reboot.

- Test device information
Android Model: hongmi note4X
Android Version: 7.0

The bug is also exist on other devices like Nexus 6P with Android 7.1

deleted Restricted
0 B

deleted Restricted
0 B

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

Mentioned issues (1)

-- -- " ...iginally caused by third party applications, after find the root cause, I made the app-debug.apk to trigger this bug 100 %." ai...... #5

Links (1)

"https://android-review.googlesource.com/#/c/platform/frameworks..." yo...@ #3, ai...@ #5

COMMENTS

All comments ▾ ↓ Oldest first

am...@google.com <am...@google.com> #2 Sep 5, 2017 08:27PM

Assigned to am...@google.com.

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.

yo...@gmail.com <yo...@gmail.com> #3 Sep 6, 2017 11:39AM

https://android-review.googlesource.com/#/c/platform/frameworks/base/+/476039/

na...@google.com <na...@google.com> #4 Sep 6, 2017 07:19PM

Have you observed any specific third party applications, or is this a theoretical concern ?

ai...@xiaomi.com <ai...@xiaomi.com> #5 Sep 6, 2017 09:50PM

Yep, this bug was originally caused by third party applications, after find the root cause, I made the app-debug.apk to trigger this bug 100%.
Please check https://android-review.googlesource.com/#/c/platform/frameworks/base/+/476039/, I solve it with limit count
Do you have better solution?

For example, can we optimize this code in ContentService.java:

```
public ObserverEntry(IContentObserver o, boolean n, Object observersLock,
                int _uid, int _pid, int _userHandle) {
    try {
        observer.asBinder().linkToDeath(this, 0);
    } catch (RemoteException e) {
        binderDied();
    }
}
```

When the same type of ContentObserver added into the observer's tree, the ContentService will create a new ObserverEntry and call linkToDeath again to the same ContentObserver. Which I think is unnecessary.
If we improve this, it can also avoid the situation that the native table exceed the maximum number easily.

am...@google.com <am...@google.com> #6 Nov 8, 2017 11:37PM

Marked as fixed.

Our engineering team has fixed this issue. It will be available in a future Android release.

yo...@gmail.com <yo...@gmail.com> #7 Nov 13, 2017 12:14PM

Hi, can I know how do the team solve this program? Is the patch already upload to the AOSP?

sh...@codeaurora.org <sh...@codeaurora.org> #8 Feb 15, 2018 09:36PM

Hi is the patch available on AOSP or OMR1? Or could you please share the gerrit link here?