An ANR BUG triggered by clicking on a malformed string

+1 7 Hotlists (3) Mark as Duplicate

Comments (2) Dependencies Duplicates (0) Blocking (0) Resources (0)

Assigned Bug P2 + Add Hotlist

STATUS UPDATE No update yet. Edit

DESCRIPTION li...@gmail.com created issue #1

Description

I am using a script to generate a malformed string and edit it as an SMS to send to my phone. When I clicked on this message, it triggered an ANR. From the logs recorded by tomestone, I firmly I native method `android.graphics.Paint.nGetRunAdvance`. And it's a generic bug that affects similar scenarios.

My test Enviroment

- Manufacturer： Samsung
- Android version : Android R
- crash app : com.samsung.android.messaging

How to Reproduce

Use the following code to generate a malformed string and write it to the clipboard.

```
import pyperclip
import binascii


def str2hex(s):
    sbin = s.encode("utf-8")
    return binascii.hexlify(sbin).decode('utf-8')

n = 1000


a1="\u0f19\u0f19\u0f87\u0f87\ufb50\u0020\u093a\u200d\ufb50\u064d\u0651\u0636\u0020\u206a"
a2="\u0f19\u0f19\u0f87\u0f87\ufb50\u0020\u093a\u200d\ufb50\u064d\u0651\u0636\u000d\u000a"
a01="\u0737"
a02="\u0736"

ret = f"{a1}{a01*n}{a02*n}{a2}"
print("len : "+str(len(ret)))
print("content : "+ str2hex(ret))
print("content : ",ret)
pyperclip.copy(f"{a1}{a01*n}{a02*n}{a2}")
```

Send it to the phone as an SMS (I believe any Android phone can reproduce this bug). Finally, multiple clicks on the SMS content will reveal that the phone has thrown an ANR bug.

What cause the BUG

Through the stacktrace recorded in the ANR bugreport, I found the following information.

```
"main" prio=5 tid=1 Native
  | group="main" sCount=1 dsCount=0 flags=1 obj=0x74503528 self=0xb4000070700c4c00
  | sysTid=22121 nice=-10 cgrp=default sched=0/0 handle=0x70717c7500
  | state=S schedstat=( 321334091897 2760806019 28862 ) utm=31807 stm=325 core=7 HZ=100
  | stack=0x7ff0f22000-0x7ff0f24000 stackSize=8192KB
  | held mutexes=
  native: #00 pc 0000000000089c8c  /apex/com.android.runtime/lib64/bionic/libc.so (syscall+28)
  native: #01 pc 000000000019fd34  /apex/com.android.art/lib64/libart.so (art::ConditionVariable::WaitHoldingLocks(art::Thread*)+148)
  native: #02 pc 00000000003d567c  /apex/com.android.art/lib64/libart.so (art::JNI<false>::ReleaseCharArrayElements(_JNIEnv*, _jcharArray*, unsigned short*,
  native: #03 pc 00000000003439d8  /system/lib64/libhwui.so (android::PaintGlue::getRunAdvance___CIIIIZI_F(_JNIEnv*, _jclass*, long, _jcharArray*, int, int,
  at android.graphics.Paint.nGetRunAdvance(Native method)
  at android.graphics.Paint.getRunAdvance(Paint.java:3024)
  at android.graphics.Paint.getRunAdvance(Paint.java:3056)
  at android.text.TextLine.getRunAdvance(TextLine.java:827)
  at android.text.TextLine.handleText(TextLine.java:878)
  at android.text.TextLine.handleRun(TextLine.java:1125)
  at android.text.TextLine.measureRun(TextLine.java:510)
  at android.text.TextLine.measureAllOffsets(TextLine.java:439) // bug here;
  at android.text.Layout.getLineHorizontals(Layout.java:1271)
  at android.text.Layout.access$000(Layout.java:59)
  at android.text.Layout$HorizontalMeasurementProvider.init(Layout.java:1669)
```

```
    at android.text.Layout$HorizontalMeasurementProvider.<init>(Layout.java:1660)
    at android.text.Layout.getOffsetForHorizontal(Layout.java:1566)
    at android.text.Layout.getOffsetForHorizontal(Layout.java:1541)
    at com.samsung.android.messaging.ui.view.bubble.item.ah.e(BubbleMotionActionHelper.java:238)
    at ...
```

The string I created has the characteristic of containing more than 2000 characters in a single Horizontals line. So when processing to the `android.text.TextLine.measureAllOffsets` metho

```
// from https://cs.android.com/android/platform/superproject/+/master:frameworks/base/core/java/android/text/TextLine.java;l=438?q=android.text.TextLine.meas

    public float[] measureAllOffsets(boolean[] trailing, FontMetricsInt fmi) {
        float[] measurement = new float[mLen + 1];

        int[] target = new int[mLen + 1];
        for (int offset = 0; offset < target.length; ++offset) {
            target[offset] = trailing[offset] ? offset - 1 : offset;
        }
        if (target[0] < 0) {
            measurement[0] = 0;
        }

        float h = 0;
        for (int runIndex = 0; runIndex < mDirections.getRunCount(); runIndex++) {
            final int runStart = mDirections.getRunStart(runIndex);
            if (runStart > mLen) break;
            final int runLimit = Math.min(runStart + mDirections.getRunLength(runIndex), mLen);  // fix here maybe;
            final boolean runIsRtl = mDirections.isRunRtl(runIndex);

            int segStart = runStart;
            for (int j = mHasTabs ? runStart : runLimit; j <= runLimit; ++j) {
                if (j == runLimit || charAt(j) == TAB_CHAR) {
                    final  float oldh = h;
                    final boolean advance = (mDir == Layout.DIR_RIGHT_TO_LEFT) == runIsRtl;
                    final float w = measureRun(segStart, j, j, runIsRtl, fmi);
                    h += advance ? w : -w;

                    final float baseh = advance ? oldh : h;
                    FontMetricsInt crtfmi = advance ? fmi : null;
                    for (int offset = segStart; offset <= j && offset <= mLen; ++offset) {
                    // This loop will be run `mLen` times. the `j` and `mLen` are both the length of my string
                        if (target[offset] >= segStart && target[offset] < j) {
                            measurement[offset] =
                                    baseh + measureRun(segStart, offset, j, runIsRtl, crtfmi);
                        }
                    }

                    if (j != runLimit) {  // charAt(j) == TAB_CHAR
                        if (target[j] == j) {
                            measurement[j] = h;
                        }
                        h = mDir * nextTab(h * mDir);
                        if (target[j + 1] == j) {
                            measurement[j + 1] =  h;
                        }
                    }

                    segStart = j + 1;
                }
            }
        }
        if (target[mLen] == mLen) {
            measurement[mLen] = h;
        }

        return measurement;
    }
```

And it will trigger 2000+ calls to `android.graphics.Paint.nGetRunAdvance` method. The elapsed time here is much greater than 5s and therefore triggers an ANR.

Impact

ANR will be triggered when any app handles a similar click event in a similar way. Especially for some SMS or IM APP.

Solutions

Perhaps this can be solved by limiting the number of loops in the `android.text.TextLine.measureAllOffsets` method.

Attachment

- anr log file;

- trigger script(Python);
- Frida Hook script for explore this issue;

Attachment.zip
117 KB Download

COMMENTS

All

si...@google.com <si...@google.com>

Reassigned to no...@google.com.

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

Assigned to an...@google.com.

Thank you for the report and the details/investigations you have added. Will take a look as soon as possible.