Android Public Tracker > Libcore 109448553

Broken Runtime.exec

+1 2 Hotlists (3) Mark as Duplicate

Comments (8) Dependencies Duplicates (0) Blocking (0) Resources (5)

Infeasible Bug P2 + Add Hotlist [AOSP] assigned

STATUS UPDATE No update yet. Edit

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

Jun 4, 2018 10:40AM

Sometimes if you try start native process via Runtime.getRuntime().exec you can get very odd errors.

Code:

```
try {
      Runtime.getRuntime().exec("ls");
} catch (Throwable e) {
      Log.e(TAG, "ERROR", e);
}
```

Result in logcat:

```
ERROR
java.io.IOException: Cannot run program "ls": error=-1322135296, Unknown error -1322135296
      at java.lang.ProcessBuilder.start(ProcessBuilder.java:983)
      at java.lang.Runtime.exec(Runtime.java:691)
      at java.lang.Runtime.exec(Runtime.java:524)
      at java.lang.Runtime.exec(Runtime.java:421)
      at com.example.Main.onCreate(Main.java:198)
      at android.app.Activity.performCreate(Activity.java:6664)
      at android.app.Instrumentation.callActivityOnCreate(Instrumentation.java:1118)
      at android.app.ActivityThread.performLaunchActivity(ActivityThread.java:2599)
      at android.app.ActivityThread.handleLaunchActivity(ActivityThread.java:2707)
      at android.app.ActivityThread.-wrap12(ActivityThread.java)
      at android.app.ActivityThread$H.handleMessage(ActivityThread.java:1460)
      at android.os.Handler.dispatchMessage(Handler.java:102)
      at android.os.Looper.loop(Looper.java:154)
      at android.app.ActivityThread.main(ActivityThread.java:6077)
      at java.lang.reflect.Method.invoke(Native Method)
      at com.android.internal.os.ZygoteInit$MethodAndArgsCaller.run(ZygoteInit.java:865)
      at com.android.internal.os.ZygoteInit.main(ZygoteInit.java:755)
Caused by: java.io.IOException: error=-1322135296, Unknown error -1322135296
      at java.lang.UNIXProcess.forkAndExec(Native Method)
      at java.lang.UNIXProcess.<init>(UNIXProcess.java:133)
      at java.lang.ProcessImpl.start(ProcessImpl.java:128)
      at java.lang.ProcessBuilder.start(ProcessBuilder.java:964)
      ... 18 more
```

Error code can be any. It is different on each run app.

Source of error:
https://android.googlesource.com/platform/libcore/+/master/ojluni/src/main/native/UNIXProcess_md.c#747
if (moveDescriptor(p->fail[1], FAIL_FILENO) == -1)

exec use 4th pipe as way for send errno to parent:
https://android.googlesource.com/platform/libcore/+/master/ojluni/src/main/native/UNIXProcess_md.c#768

And for this used pipe. All ok on this moment. But later fd of pipe moved to number 3 (FAIL_FILENO).
It is source of mistake.
Before this move 3 fd will be used for socket in zygote (and all apps forked from it):

```
> ls -l /proc/240/fd
[...]
lrwx------ 1 root root 64 2018-06-02 10:54 3 -> socket:[10629]
[...]
```

With this socket can be associated buffer with data.
If buffer empty all ok. Exec work as must.
But if buffer contain data, then this data go to pipe and read in parent at:
https://android.googlesource.com/platform/libcore/+/master/ojluni/src/main/native/UNIXProcess_md.c#920
switch (readFully(fail[0], &errnum, sizeof(errnum))) {

And this cause throw IOException listed above:
throwIOException(env, errnum, "Exec failed");

Because data can be any, number errnum fully random and vary on each run.

Possible solution:
Do not move p->fail[1] at
https://android.googlesource.com/platform/libcore/+/master/ojluni/src/main/native/UNIXProcess_md.c#747
if (moveDescriptor(p->fail[1], FAIL_FILENO) == -1)

Reporter
Type
Priority
Severity
Status
Access
Assigne
Verifier
Collabor
CC

AOSP ID
Reporte
Found In
Targeted
Verified
In Prod

Just remember in variable and ignore for closeDescriptors(). Also use it in other places where now used FAIL_FILENO.

✓ Links (5) Hide all

"https://android.googlesource.com/platform/libcore/+/master/ojluni/src/main/native/U… "	en...@ #1
"https://android.googlesource.com/platform/libcore/+/master/ojluni/src/main/native/U… "	en...@ #1
"https://android.googlesource.com/platform/libcore/+/master/ojluni/src/main/native/U… "	en...@ #1
"It is for file: https://android.googlesource.com/platform/libcore/+/master/ojluni/src/main/native/UNIXPr… "	en...@ #4
"https://github.com/android-hacker/VirtualXposed/blob/b6075b9aaeae4ee691f8313e0f6406a2a01c6b04/VirtualApp/lib/src/main… "	en...@ #4

COMMENTS

All comments ▾ ↓ Oldest first

en...@gmail.com <en...@gmail.com> #2 Jun 5, 2018 11:51PM ⋮

This error happens if present any output into logcat inside child process before call exec, because this socket is socket for logcat.
So send to logcat write to fd = 3, but it is no more valid socket for logcat it is fail pipe.

ku...@google.com <ku...@google.com> Jun 6, 2018 03:11PM

Assigned to ku...@google.com.

ku...@google.com <ku...@google.com> #3 Jun 6, 2018 09:28PM ⋮

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. OPM1.171019.011)

Steps to reproduce
Please provide a sample application or apk to reproduce the issue.
Also kindly mention the steps to be followed for reproducing the issue with the given sample application.

Expected output
What is the expected output?

Current output
What is the current output?

Android bug report
After reproducing the issue, press the volume up, volume down, and power button simultaneously. This will capture a bug report on your device in the "bug reports" directory. Attach the bug report file to this issue.

Alternate method:
After reproducing the issue, navigate to developer settings, ensure 'USB debugging' is enabled, then enable 'Bug report shortcut'. To take bug report, hold the power button and select the 'Take bug report' option.

en...@gmail.com <en...@gmail.com> #4 Jun 7, 2018 10:03PM ⋮

I do not know what you were going to investigate there.

I've already investigate everything, pointed out specific lines of code in the master branch, wrote the reasons and even suggested a possible solution.
If the code was on github, I would already have done push request, as I cloned your code and made changes so that there was no problem.

If there is no output in logcat in child after fork, but before exec, then there will be no problem.
In conventional firmware, there is no such conclusion, so the bug remained unnoticed. However, if such output is added, for example, by an application of virtual space, then the problem will be.

I used VirtualXposed version 0.10.1 to show this problem on Genymotion with Android 7.0, API 24 and patches from August 5, 2016. Kernel 4.4.34. The build number is NRD90M.
Only I do not understand why you need it? I have already clearly written the source of the problem. Specified specific lines of code.

Here are some screenshots with kdiff, one of the possible solutions.
scr_1528371264.jpg
scr_1528371326.jpg
It is for file: https://android.googlesource.com/platform/libcore/+/master/ojluni/src/main/native/UNIXProcess_md.c

Made a test application. It needs to be installed in VirtualXposed 0.10.1 to reproduce the problem.
TestExec_signed.apk
VirtualXposed_0.10.1.apk

Here's the video with the problem.
exec_mute.mp4
First, I started directly - there is no problem.
Then in VirtualXposed - there is a problem.

VirtualXposed uses a hook for exec, which outputs a log to the logcat:
https://github.com/android-hacker/VirtualXposed/blob/b6075b9aaeae4ee691f8313e0f6406a2a01c6b04/VirtualApp/lib/src/main/jni/Foundation/IOUniformer.cpp#L600

Since this is not the only way VirtualXposed can do, the problem is of a general nature.

Also, the proposed changes are completely safe and can prevent not only the problem with the output of logcat, but also other possible problems associated with changing the descriptor.

For example, if tomorrow with handle #3 there will be no socket for logcat, but something else where the write will be made.

I do not quite understand why it was necessary to move it at all. Is it nice to group the processing in a loop. But it's not worth it.
Anyway, this descriptor will be closed, one way or another. Either at exec due to FD_CLOEXEC, or directly, after writing the error code to it.
Moving does nothing, except for possible errors, because the descriptor #3 could be used earlier.

deleted Restricted
0 B ?

deleted Restricted
0 B ?

deleted Restricted
0 B ?

deleted Restricted
0 B ?

deleted Restricted
0 B ?

ku...@google.com <ku...@google.com> #5 Jun 11, 2018 11:48PM

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.

to...@google.com <to...@google.com> #6 Feb 12, 2019 05:19AM

This issue only arises when running Android under a virtual environment such as VirtualXposed which alters file descriptors between the process fork and exec, in this case by opening a log stream. Such behaviour is unsupported, and the virtual environment should be fixed to avoid it.

en...@gmail.com <en...@gmail.com> #7 Feb 12, 2019 10:28AM

"which alters file descriptors between the process fork and exec"
Wrong. Issue source on try send info in logcat not in alter descriptor. Better avoid code (reuse logcat descriptor) which can cause problems.

ku...@google.com <ku...@google.com> #8 Feb 12, 2019 08:05PM

Status: Won't Fix (Infeasible)

Please check comment#6.

to...@google.com <to...@google.com> Feb 12, 2019 09:09PM

Status: Won't Fix (Infeasible)