strace seems to ignore some calls to rt_sigaction

DESCRIPTION pa...@lunarg.com created issue #1

Hi

I am observing a behavior on Android 12 where `strace` seems to ignore/not print calls to `rt_sigaction` that are related to `SIGSEGV`. I've been observing that behavior with a large/"real world" a

```
#include <jni.h>
#include <string>

#include <android/log.h>
#include <signal.h>
#include <sys/mman.h>
#include <unistd.h>

#define  LOGE(...)  __android_log_print(ANDROID_LOG_ERROR, "LOG_TAG", __VA_ARGS__)

const size_t size = 4 * 1024;

static void sigsegv_handler(int id, siginfo_t *info, void *data)
{
    LOGE("%s()\n", __func__);
    LOGE("fault address: %p\n", info->si_addr);

    mprotect(info->si_addr, size, PROT_READ | PROT_WRITE);
}

extern "C" JNIEXPORT jstring JNICALL
Java_com_example_myapplication_MainActivity_stringFromJNI(
        JNIEnv* env,
        jobject /* this */) {

    sleep(2);

    LOGE("%s()\n", __func__);
    LOGE("sigsegv_handler: %p\n", sigsegv_handler);

    struct sigaction sa = {};
    sa.sa_flags = SA_SIGINFO;
    sigemptyset(&sa.sa_mask);
    sigaddset(&sa.sa_mask, SIGSEGV);
    sa.sa_sigaction = sigsegv_handler;
    int result = sigaction(SIGBUS, &sa, NULL);
    if (result == -1)
        LOGE("sigaction failed\n");

    result = sigaction(SIGSEGV, &sa, NULL);
    if (result == -1)
        LOGE("sigaction failed\n");

    void *addr = mmap(NULL, size, PROT_READ | PROT_WRITE, MAP_PRIVATE | MAP_ANONYMOUS, -1, 0);
    if (addr == MAP_FAILED)
        printf("mmap failed\n");

    mprotect(addr, size, PROT_NONE);
    int *segfault = (int *)addr;
    *segfault = 0;

    const std::string hello = "Hello from C++";
    return env->NewStringUTF(hello.c_str());
}
```

Running this and cherry-picking only relevant information from logcat (`adb shell logcat | grep -e LOG_TAG -e sigaction -e SIGSEGV`) the output is the following:

```
08-16 11:24:05.021 10211 10211 I wrap.sh : 11:24:05.021012 rt_sigaction(SIGABRT, {sa_handler=0x7ac499cd1c, sa_mask=~[], sa_flags=SA_ONSTACK|SA_RESTART|SA_SIG
08-16 11:24:05.021 10211 10211 I wrap.sh : 11:24:05.021059 rt_sigaction(SIGBUS, {sa_handler=0x7ac499cd1c, sa_mask=~[], sa_flags=SA_ONSTACK|SA_RESTART|SA_SIGI
08-16 11:24:05.021 10211 10211 I wrap.sh : 11:24:05.021098 rt_sigaction(SIGFPE, {sa_handler=0x7ac499cd1c, sa_mask=~[], sa_flags=SA_ONSTACK|SA_RESTART|SA_SIGI
```

```
08-16 11:24:05.021 10211 10211 I wrap.sh : 11:24:05.021137 rt_sigaction(SIGILL, {sa_handler=0x7ac499cd1c, sa_mask=~[], sa_flags=SA_ONSTACK|SA_RESTART|SA_SIGI
08-16 11:24:05.021 10211 10211 I wrap.sh : 11:24:05.021178 rt_sigaction(SIGSEGV, {sa_handler=0x7ac499cd1c, sa_mask=~[], sa_flags=SA_ONSTACK|SA_RESTART|SA_SIG
08-16 11:24:05.021 10211 10211 I wrap.sh : 11:24:05.021217 rt_sigaction(SIGSTKFLT, {sa_handler=0x7ac499cd1c, sa_mask=~[], sa_flags=SA_ONSTACK|SA_RESTART|SA_S
08-16 11:24:05.021 10211 10211 I wrap.sh : 11:24:05.021256 rt_sigaction(SIGSYS, {sa_handler=0x7ac499cd1c, sa_mask=~[], sa_flags=SA_ONSTACK|SA_RESTART|SA_SIGI
08-16 11:24:05.021 10211 10211 I wrap.sh : 11:24:05.021293 rt_sigaction(SIGTRAP, {sa_handler=0x7ac499cd1c, sa_mask=~[], sa_flags=SA_ONSTACK|SA_RESTART|SA_SIG
08-16 11:24:05.021 10211 10211 I wrap.sh : 11:24:05.021332 rt_sigaction(SIGRT_3, {sa_handler=0x7ac499cd1c, sa_mask=~[], sa_flags=SA_ONSTACK|SA_RESTART|SA_SIG
08-16 11:24:05.523 10211 10211 I wrap.sh : 11:24:05.523641 rt_sigaction(SIGRT_7, {sa_handler=0x7abe68dfbc, sa_mask=[], sa_flags=SA_ONSTACK|SA_RESTART}, {sa_h
08-16 11:24:05.525 10211 10211 I wrap.sh : 11:24:05.525496 rt_sigaction(SIGRT_4, {sa_handler=0x7abe677f3c, sa_mask=[], sa_flags=SA_RESTART|SA_SIGINFO}, NULL,
08-16 11:24:05.525 10211 10211 I wrap.sh : 11:24:05.525591 rt_sigaction(SIGRT_6, {sa_handler=SIG_IGN, sa_mask=[], sa_flags=SA_ONSTACK|SA_RESTART}, {sa_handle
08-16 11:24:05.711 10211 10211 I wrap.sh : 11:24:05.711434 rt_sigaction(SIGSEGV, NULL, {sa_handler=0x7ac499cd1c, sa_mask=~[KILL STOP], sa_flags=SA_ONSTACK|SA
08-16 11:24:05.985 10211 10211 I wrap.sh : 11:24:05.985900 rt_sigaction(SIGSEGV, {sa_handler=0x7ab6c7208c, sa_mask=~[], sa_flags=SA_ONSTACK|SA_RESTART|SA_SIG
08-16 11:24:05.986 10211 10211 I wrap.sh : 11:24:05.985960 rt_sigaction(SIGSEGV, NULL, {sa_handler=0x7ab6c7208c, sa_mask=~[KILL STOP], sa_flags=SA_ONSTACK|SA
08-16 11:24:06.182 10211 10211 I wrap.sh : 11:24:06.182752 rt_sigaction(SIGRT_2, {sa_handler=0x7801eba200, sa_mask=[], sa_flags=0}, NULL, 8) = 0 <0.000014>
08-16 11:24:06.189 10211 10211 I wrap.sh : 11:24:06.189637 rt_sigaction(SIGRT_2, {sa_handler=0x7801e63b9c, sa_mask=[], sa_flags=0}, {sa_handler=0x7801eba200,
08-16 11:24:06.220 10211 10211 I wrap.sh : [pid 10218] 11:24:06.220721 rt_sigaction(SIGRT_6, {sa_handler=0x77f2069d34, sa_mask=[], sa_flags=SA_RESTART|SA_SIG
08-16 11:24:06.668 10211 10211 I wrap.sh : [pid 10218] 11:24:06.668012 rt_sigaction(SIGBUS, {sa_handler=0x7aa31cba18, sa_mask=[], sa_flags=SA_SIGINFO}, {sa_h
08-16 11:24:06.969 10211 10211 I wrap.sh : [pid 10218] 11:24:06.968728 rt_sigaction(SIGCHLD, {sa_handler=SIG_DFL, sa_mask=[], sa_flags=SA_RESTART|SA_NOCLDSTO
08-16 11:24:07.476 10211 10211 I wrap.sh : [pid 10218] 11:24:07.476017 --- SIGSEGV {si_signo=SIGSEGV, si_code=SEGV_MAPERR, si_addr=NULL} ---




08-16 11:24:09.551 10218 10218 E LOG_TAG : Java_com_example_myapplication_MainActivity_stringFromJNI()
08-16 11:24:09.553 10211 10211 I wrap.sh : [pid 10218] 11:24:09.552166 writev(3, [{iov_base="\0\352'9b\373b?j\336 ", iov_len=11}, {iov_base="\6", iov_len=1},
08-16 11:24:09.555 10211 10211 I wrap.sh : [pid 10218] 11:24:09.554485 writev(4, [{iov_base="1X\0v'\352'", iov_len=7}, {iov_base="\0\352'9b\373b?j\336 ", iov
08-16 11:24:09.555 10218 10218 E LOG_TAG : sigsegv_handler: 0x7796e4f4b8
08-16 11:24:09.557 10211 10211 I wrap.sh : [pid 10218] 11:24:09.556518 writev(3, [{iov_base="\0\352'9b\373b\371\317\34!", iov_len=11}, {iov_base="\6", iov_le
08-16 11:24:09.558 10211 10211 I wrap.sh : [pid 10218] 11:24:09.557681 writev(4, [{iov_base="1:\0v'\352'", iov_len=7}, {iov_base="\0\352'9b\373b\371\317\34!"
08-16 11:24:09.558 10211 10211 I wrap.sh : [pid 10218] 11:24:09.557828 rt_sigaction(SIGBUS, {sa_handler=0x7796e4f4b8, sa_mask=[SEGV], sa_flags=SA_SIGINFO}, N
08-16 11:24:09.560 10211 10211 I wrap.sh : [pid 10218] 11:24:09.560128 --- SIGSEGV {si_signo=SIGSEGV, si_code=SEGV_ACCERR, si_addr=0x7abbda0000} ---
08-16 11:24:09.562 10218 10218 E LOG_TAG : sigsegv_handler()
08-16 11:24:09.564 10211 10211 I wrap.sh : [pid 10218] 11:24:09.563066 writev(3, [{iov_base="\0\352'9b\373ba\256\205!", iov_len=11}, {iov_base="\6", iov_len=
08-16 11:24:09.565 10211 10211 I wrap.sh : [pid 10218] 11:24:09.564309 writev(4, [{iov_base="1.\0v'\352'", iov_len=7}, {iov_base="\0\352'9b\373ba\256\205!",
08-16 11:24:09.564 10218 10218 E LOG_TAG : fault address: 0x7abbda0000
08-16 11:24:09.566 10211 10211 I wrap.sh : [pid 10218] 11:24:09.566506 writev(3, [{iov_base="\0\352'9b\373b\0046\253!", iov_len=11}, {iov_base="\6", iov_len=
08-16 11:24:09.568 10211 10211 I wrap.sh : [pid 10218] 11:24:09.567749 writev(4, [{iov_base="18\0v'\352'", iov_len=7}, {iov_base="\0\352'9b\373b\0046\253!",
```

The gap in the above extract is intentional and manually inserted in order to make more visible when the custom native codes starts to execute.

So it is visible that `rt_sigaction(SIGBUS, ...` is printed from `strace` but the expected `rt_sigaction(SIGSEGV, ...)` is not, although it is obvious that the `SIGSEGV` handler is installed as it

I don't know if this affects any other signals/system calls. I've been focusing only on the `SIGSEGV` and that's what I have observed so far.

Relevant information: Phone: Pixel 6 Android version: 12(SQ1D.220105.007) custom build from AOSP sources

COMMENTS