google / **sanitizers**

AddressSanitizerOnAndroid

Evgeniy Stepanov edited this page on 19 Aug 2017 · 6 revisions

NOTE: *this document is about running Android applications built with the NDK under AddressSanitizer. For information about using AddressSanitizer on Android platform components, see the android documentation.*

NOTE: *ASan is supported in all versions of Android starting with KitKat, with the exception of the initial release of Lollipop (Lollipop MR1 update is fine).*

NOTE: *AddressSanitizer on Android requires a rooted device (either -eng or -userdebug build, or any other setup that allows editing the contents of the /system partition).*

Android NDK supports AddressSanitizer on arm, armv7 and x86 ABIs starting with version r10d.

Building

To build your app's native (JNI) code with AddressSanitizer, add the following to Android.mk:

```
LOCAL_CFLAGS    := -fsanitize=address -fno-omit-frame-pointer
LOCAL_LDFLAGS   := -fsanitize=address
LOCAL_ARM_MODE := arm
```

Note that AddressSanitizer in NDK r10d does not support 64-bit ABIs, and compilation with `APP_ABI := all` will fail. Use

```
APP_ABI := armeabi armeabi-v7a x86
```

or a subset of those. This is not an issue with newer NDKs.

Stack traces

AddressSanitizer needs to unwind stack on every `malloc` / `realloc` / `free` call. There are 2 options here:

- A "fast" frame pointer-based unwinder.

> It requires

```
LOCAL_CFLAGS=-fno-omit-frame-pointer
LOCAL_ARM_MODE := arm
APP_STL := gnustl_shared  # or any other _shared option
```

> Implementations of operators new and delete in libstdc++ are usually built without frame pointer. ASan brings its own implementation, but it can't be used if the C++ stdlib is linked statically into the application.

- A "slow" CFI unwinder.

> In this mode ASan simply calls `_Unwind_Backtrace` . It requires only `-funwind-tables` , which is normally enabled by default. Warning: the "slow" unwinder is SLOW (10x or more, depending on how ofter you call `malloc` / `free`).

Fast unwinder is the default for malloc/realloc/free. Slow unwinder is the default for "fatal" stack traces, i.e. in the case when an error is detected. Slow unwinder can be enabled for all stack traces with

AddressSanitizerCompileTimeOptimizations

AddressSanitizerContainerOverflow

AddressSanitizerExampleGlobalOutOfBounds

AddressSanitizerExampleHeapOutOfBounds

Show 59 more pages…

Clone this wiki locally

`https://github.com/google`

```
ASAN_OPTIONS=fast_unwind_on_malloc=0
```

Running

Device setup step is required before applications built with AddressSanitizer can be started. Android NDK includes a script `asan_device_setup` that will do this for you. It must be run once for any device you want to install ASan applications to.

Once this is done, applications built with ASan can be installed and executed as any other.

Native binaries built with ASan must be prefixed with `LD_PRELOAD=libclang_rt.asan-arm-android.so` (replace `arm` with the target arch: `x86` , etc).

Under the hood

This script uploads the ASan runtime library to `/system/lib` and sets up the Zygote binary (`/system/bin/app_process`) in such a way that ASan runtime library is `LD_PRELOAD` -ed into the Zygote This guarantees that ASan runtime library is the first DSO in the global symbol lookup order, which is necessary for ASan libc interception to work. It also ensures that the memory allocator (`malloc` / `realloc` / `free` etc) is replaced with ASan implementation very early in the process lifetime.

To preserve memory, most AddressSanitizer features are disabled until the first DSO with ASan-instrumented code is loaded into the process memory. Normally, this happens in individual application processes (when the Zygote forks and loads the application-specific code), and does not affect the rest of applications on the device.

Run-time flags

```
asan_device_setup --extra-options=<options text>
```

> this flag can be used to set default ASAN_OPTIONS for the Zygote process. Note that this will affect all applications on the device!

Additional per-application options can be specified in `/data/local/tmp/asan.options.%b` , where `%b` stands for the "nice name" of the application as seen in `ps` output. These additional options that are evaluated at the moment the first instrumented DSO is loaded into the Zygote process. Not all possible `ASAN_OPTIONS` can be set here. Setting options through `asan_device_setup` is preferable.

See AddressSanitizerFlags for the list of supported flags.