__strlen_aarch64_mte frequently triggered problem with MTE

Assigned Bug P2 + Add Hotlist [AOSP] assigned

STATUS UPDATE No update yet. Edit

DESCRIPTION h8...@gmail.com created issue #1

To avoid the possibility of sharing private information, please share bugreports and screenshots from Google Drive. Share files with android-bugreport@google.com and include only Google drive

Disclaimer: Please note, by submitting this bug report, you acknowledge that Google may use information included in the bug report to diagnose technical issues and to improve our products and

Description of issue (what needs changing):
From our practical experiences, so many Android Applications could trigger the MTE syncerrors and bring collapses only if we open the MTE, since the Android standard libc would use the MTE v

```
#00 pc 00000000000937cc /apex/com.android.runtime/lib64/bionic/libc.so
(__strlen_aarch64_mte+12)
```

The developers' coding style or the compiler optimizations could make the string objects free ahead and of course, it is theoretically UAF (use after free). Something like the case below, in which

```
vector.push_back(toCppString(**).c_str())
```

I could understand the implementation consideration for this MTE version strlen. It is needed by completeness of theory and the MTE team may request the coverage of all dereerences from hea

But I still have no idea whether it is necessary to bring MTE to the strlen? I think the stirng object is greatly management inherently and there is no need to operate memory explicitly. Although the

So, is it possible to take some remedial actions or just disable the MTE strlen for better and more practical usage of MTE Android please?

✓ Links (4)

Links (4)

" ...g private information, please share bugreports and screenshots from Google Drive. Share files with android-bugreport@google.com and include only Google drive links in your bug. If attaching bug
"Disclaimer: Please note, by submitting this bug report, you acknowledge that Google may use information included in the bug report to diagnose technical issues and to improve our products and ser
"The example you mention, vector.push_back(toCppString(**).c_str()), is a great example of the type of footgun that exists in C++. Our other tools even specifically mention this as an example of a
"The example you mention, vector.push_back(toCppString(**).c_str()), is a great example of the type of footgun that exists in C++. Our other tools even specifically mention this as an example of a

COMMENTS

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

Assigned to ra...@google.com.

We have shared this with our product and engineering team and will update this issue with more information as it becomes available.

vi...@google.com <vi...@google.com> #3

Thanks again for the feedback! Our product and engineering teams have evaluated the request and responded:

Hello,

An MTE-specific `strlen` (and all `string.h` family of functions) is necessary and desirable to detect buffer-overflow bugs.

The example you mention, `vector.push_back(toCppString(**).c_str())`, is a great example of the type of footgun that exists in C++. Our other tools even specifically mention this

Some of those specific examples of use-after-free bugs are begnign, however they're still undefined behaviour. MTE catching them is the desired functionality.