A checker for dangling string pointers in C++

in the Clang Static Analyzer

Réka Kovács

Eötvös Loránd University, Budapest, Hungary
rekanikolett@gmail.com

Mentors:
Artem Dergachev
Gábor Horváth

Real-world example

```
return std::to_string(size).c_str();
```

Real-world example

```
return std::to_string(size).c_str();
```

`std::to_string()` creates a temporary object

the caller will receive a pointer to an already deallocated character buffer

Real-world example

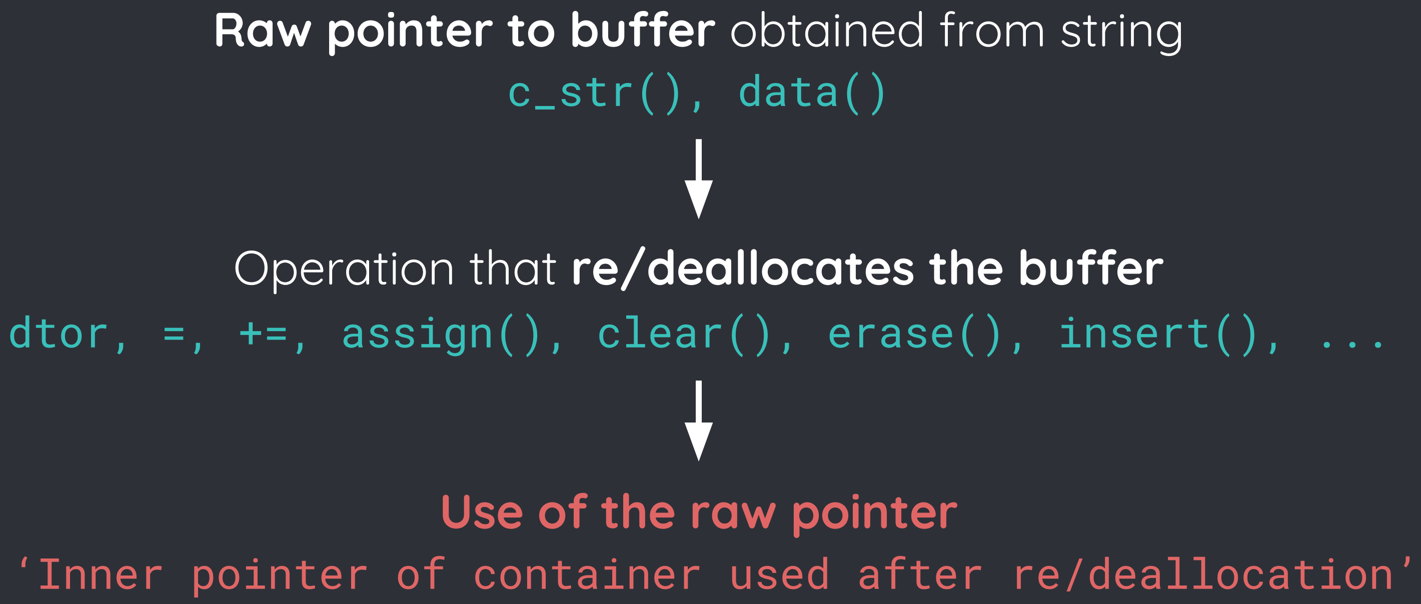
```
return std::to_string(size).c_str();
```
*

`std::to_string()` creates a temporary object

the caller will receive a pointer to an already deallocated character buffer

* found code like this in popular open-source projects

cplusplus.InnerPointer



Raw pointer to buffer obtained from string

`c_str(), data()`

Operation that **re/deallocates the buffer**

`dtor, =, +=, assign(), clear(), erase(), insert(), ...`

Use of the raw pointer

`‘Inner pointer of container used after re/deallocation’`

cplusplus.InnerPointer

Evaluated on a couple of open-source projects (+ dependencies):
Bitcoin, Ceph, Harfbuzz, ICU, LibreOffice, LLVM, qBittorrent

Found **3 true positives**
in Ceph, GPGME and Facebook’s RocksDB
Reported & fixed within a day

Found **0 false positives** in these projects!
Please try it out and give feedback!

Future plans

other STL / non-STL containers

`std::string_view`

How to use

Analyze a project:

```
$ scan-build
```

Enabled by default

Analyze one file:

```
$ clang --analyze a.cpp
```

Enabled by default

Thanks!

Final report:

rnkovacs.github.io/gsoc2018

Réka Kovács / rekanikolett@gmail.com