Resolving the almost decade old checker dependency issue in the Clang Static Analyzer

Kristóf Umann
dkszelethus@gmail.com

Eötvös Loránd University, Budapest

Ericsson Hungary

April 8., 2019

The original problem: easy-to-mess-up command line interface

```
clang -cc1 -analyze myfile.cpp \
  -analyzer-checker=cplusplus.InnerPointer \
  -analyzer-config note-as-warning=true
```

...meant to be notes-as-warnings

```
clang -cc1 -analyze myfile.cpp \
  -analyzer-checker=cplusplus.InnerPointer \
  -analyzer-config unix.Malloc:Optimist=true
```

...meant to be Optimistic

No warnings, no errors, the analyzer simply doesn't do what you intended...

Bug unearthed: ”The Checker Naming Bug”

- Multiple checker objects could receive the same name
- Incorrect checker names in bug reports
- Errors while parsing checker configurations

Real-life problems coming from the Checker Naming Bug

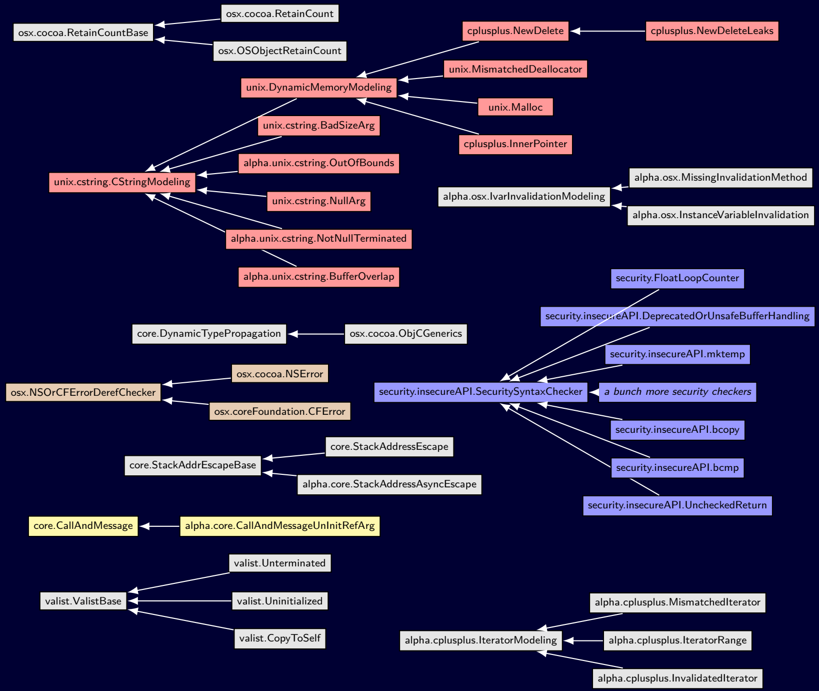
```
clang -cc1 -analyze myfile.cpp \
  -analyzer-checker=cplusplus.InnerPointer \
  -analyzer-config unix.Malloc:Optimistic=true
```

Real-life problems coming from the Checker Naming Bug

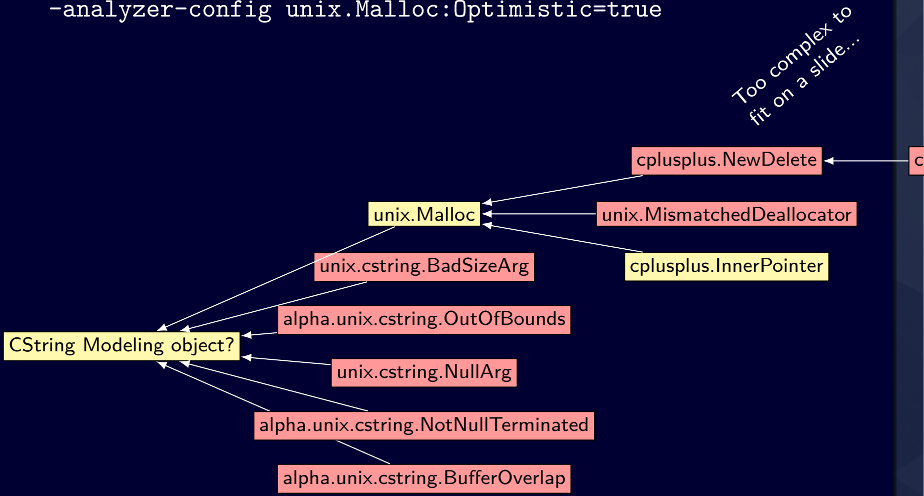
```
clang -cc1 -analyze myfile.cpp \
  -analyzer-checker=cplusplus.InnerPointer \
  -analyzer-config cplusplus.InnerPointer:Optimistic=true
```

InnerPointerChecker and MallocChecker have the same name!

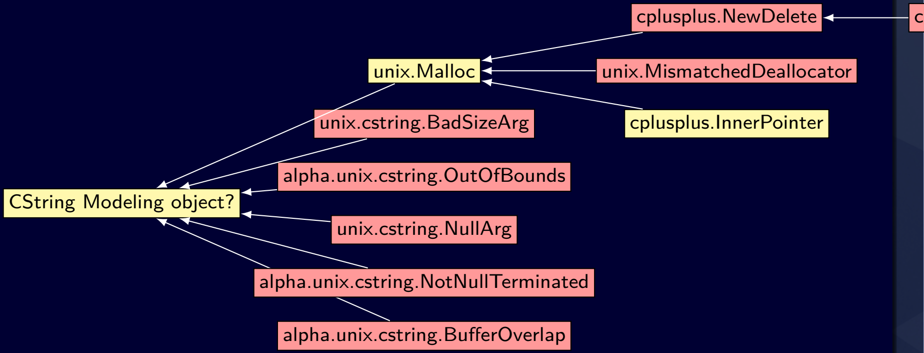
- Turns out InnerPointerChecker depends on MallocChecker!
- InnerPointerChecker enables both itself and MallocChecker
- Fixing this bug implies the need to reimplement dependencies...


osx.cocoa.RetainCount
osx.cocoa.RetainCountBase
osx.OSObjectRetainCount
cplusplus.NewDelete
cplusplus.NewDeleteLeaks
unix.MismatchedDeallocator
unix.DynamicMemoryModeling
unix.Malloc
unix.cstring.BadSizeArg
cplusplus.InnerPointer
alpha.unix.cstring.OutOfBounds
unix.cstring.CStringModeling
unix.cstring.NullArg
alpha.osx.MissingInvalidationMethod
alpha.osx.IvarInvalidationModeling
alpha.osx.InstanceVariableInvalidation
alpha.unix.cstring.NotNullTerminated
alpha.unix.cstring.BufferOverlap
security.FloatLoopCounter
security.insecureAPI.DeprecatedOrUnsafeBufferHandling
core.DynamicTypePropagation
osx.cocoa.ObjCGenerics
security.insecureAPI.mktemp
osx.cocoa.NSError
osx.NSOrCFErrorDerefChecker
security.insecureAPI.SecuritySyntaxChecker
a bunch more security checkers
osx.coreFoundation.CFError
security.insecureAPI.bcopy
core.StackAddressEscape
core.StackAddrEscapeBase
alpha.core.StackAddressAsyncEscape
security.insecureAPI.bcmp
security.insecureAPI.UncheckedReturn
core.CallAndMessage
alpha.core.CallAndMessageUnInitRefArg
valist.Unterminated
valist.ValistBase
valist.Uninitialized
alpha.cplusplus.MismatchedIterator
alpha.cplusplus.IteratorModeling
alpha.cplusplus.IteratorRange
valist.CopyToSelf
alpha.cplusplus.InvalidatedIterator

```
clang -cc1 -analyze myfile.cpp \
  -analyzer-checker=cplusplus.InnerPointer \
  -analyzer-config unix.Malloc:Optimistic=true
```

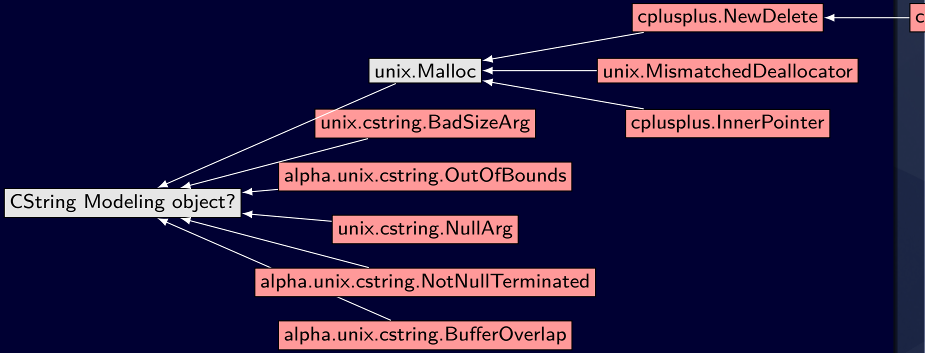

Too complex to fit on a slide...
cplusplus.NewDelete
c
unix.Malloc
unix.MismatchedDeallocator
cplusplus.InnerPointer
unix.cstring.BadSizeArg
alpha.unix.cstring.OutOfBounds
CString Modeling object?
unix.cstring.NullArg
alpha.unix.cstring.NotNullTerminated
alpha.unix.cstring.BufferOverlap

```
clang -cc1 -analyze myfile.cpp \
  -analyzer-checker=cplusplus.InnerPointer \
  -analyzer-config cplusplus.InnerPointer:Optimistic=true
```

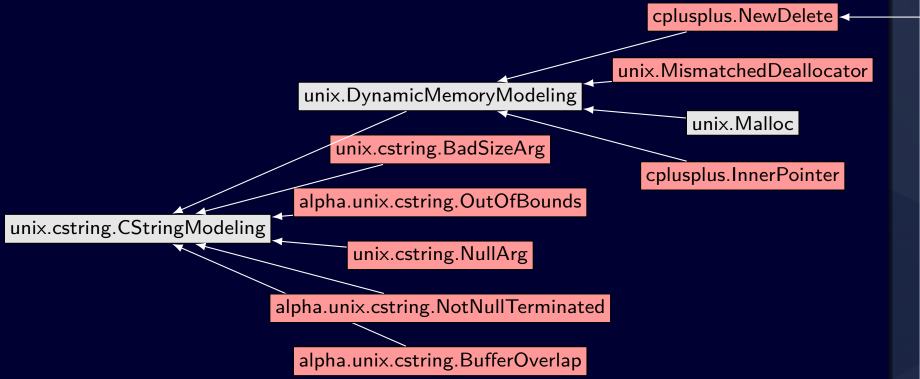


cplusplus.NewDelete
c
unix.Malloc
unix.MismatchedDeallocator
cplusplus.InnerPointer
unix.cstring.BadSizeArg
alpha.unix.cstring.OutOfBounds
CString Modeling object?
unix.cstring.NullArg
alpha.unix.cstring.NotNullTerminated
alpha.unix.cstring.BufferOverlap

How do we solve this?


cplusplus.NewDelete
c
unix.Malloc
unix.MismatchedDeallocator
cplusplus.InnerPointer
unix.cstring.BadSizeArg
alpha.unix.cstring.OutOfBounds
CString Modeling object?
unix.cstring.NullArg
alpha.unix.cstring.NotNullTerminated
alpha.unix.cstring.BufferOverlap

Be able to represent dependencies with a directed tree


cplusplus.NewDelete
unix.MismatchedDeallocator
unix.DynamicMemoryModeling
unix.Malloc
unix.cstring.BadSizeArg
cplusplus.InnerPointer
alpha.unix.cstring.OutOfBounds
unix.cstring.CStringModeling
unix.cstring.NullArg
alpha.unix.cstring.NotNullTerminated
alpha.unix.cstring.BufferOverlap

Resolve dependencies at a higher level

- Declare dependencies in TableGen
- Don't allow checkers to enable more than one checker
- Make sure dependencies are enabled in the correct order

```
def InnerPointerChecker : Checker<"InnerPointer">,
  HelpText<"Looks for pointers to temp. strings">,
  Dependencies<[DynamicMemoryModeling]>,
  Documentation<NotDocumented>;
```

Conclusion

- We are able to list checker dependencies
- We can now list and verify checker options
- Checker names won't depend on how we invoke the analyzer
- Plugins can now depend on builtin checkers
- Already in trunk!

osx.cocoa.RetainCount
osx.cocoa.RetainCountBase
osx.OSObjectRetainCount
cplusplus.NewDelete
cplusplus.NewDeleteLeaks
unix.MismatchedDeallocator
unix.DynamicMemoryModeling
unix.Malloc
cplusplus.InnerPointer
unix.cstring.BadSizeArg
alpha.unix.cstring.OutOfBounds
unix.cstring.CStringModeling
unix.cstring.NullArg
alpha.unix.cstring.NotNullTerminated
alpha.unix.cstring.BufferOverlap
alpha.osx.MissingInvalidationMethod
alpha.osx.IvarInvalidationModeling
alpha.osx.InstanceVariableInvalidation
security.FloatLoopCounter
osx.NSOrCFErrorDerefChecker
osx.coreFoundation.CFError
security.insecureAPI.SecuritySyntaxChecker
a bunch more security checkers
security.insecureAPI.bcopy
security.insecureAPI.bcmp
security.insecureAPI.UncheckedReturn
core.StackAddressEscape
core.StackAddrEscapeBase
alpha.core.StackAddressAsyncEscape
core.CallAndMessage
alpha.core.CallAndMessageUnInitRefArg
valist.Unterminated
valist.ValistBase
valist.Uninitialized
valist.CopyToSelf
alpha.cplusplus.MismatchedIterator
alpha.cplusplus.IteratorModeling
alpha.cplusplus.IteratorRange
alpha.cplusplus.InvalidatedIterator

Thank you for your attention!