Arseniy Zaostrovnykh

Boy Who Cried Wolf

Many FPs decrease

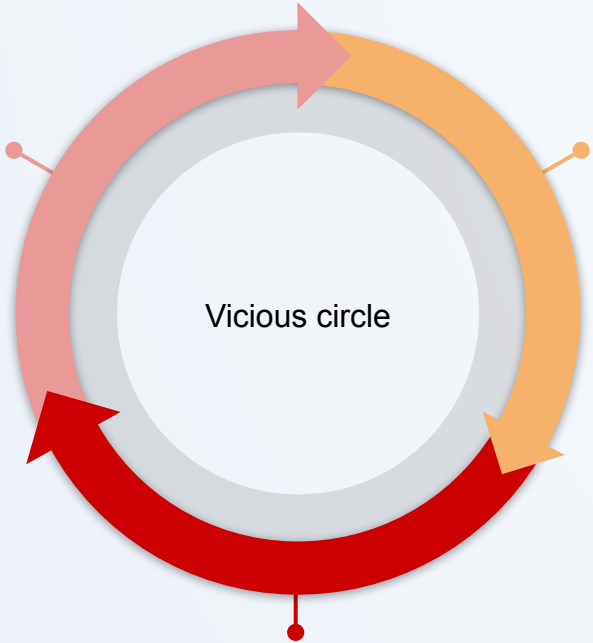
trust in the analyzer

sonar

Vicious Circle

Low trust

In the analyzer



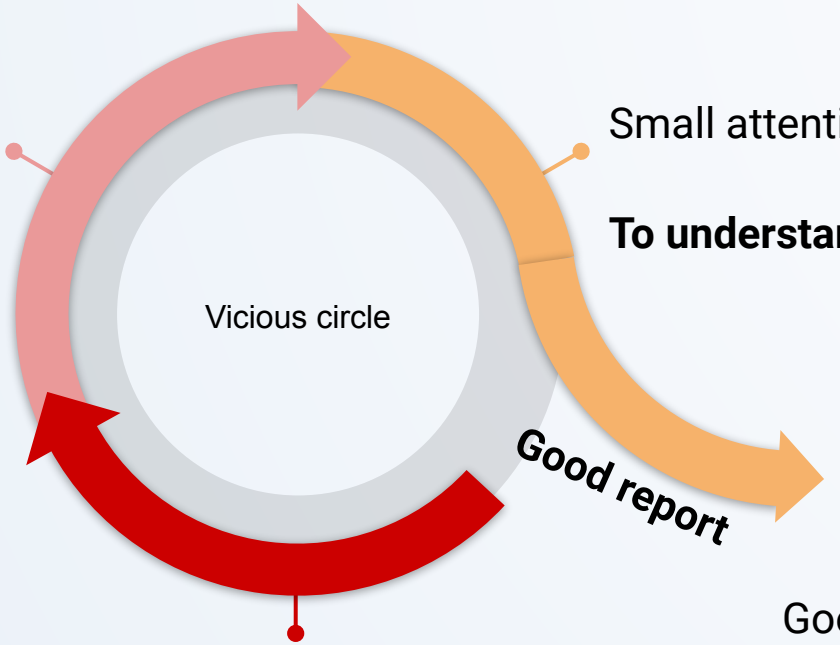
Small attention budget

To understand the report

Poor understanding

This must be a false positive

Break the Vicious Circle via Better Reporting



Good understanding

Higher trust

sonar

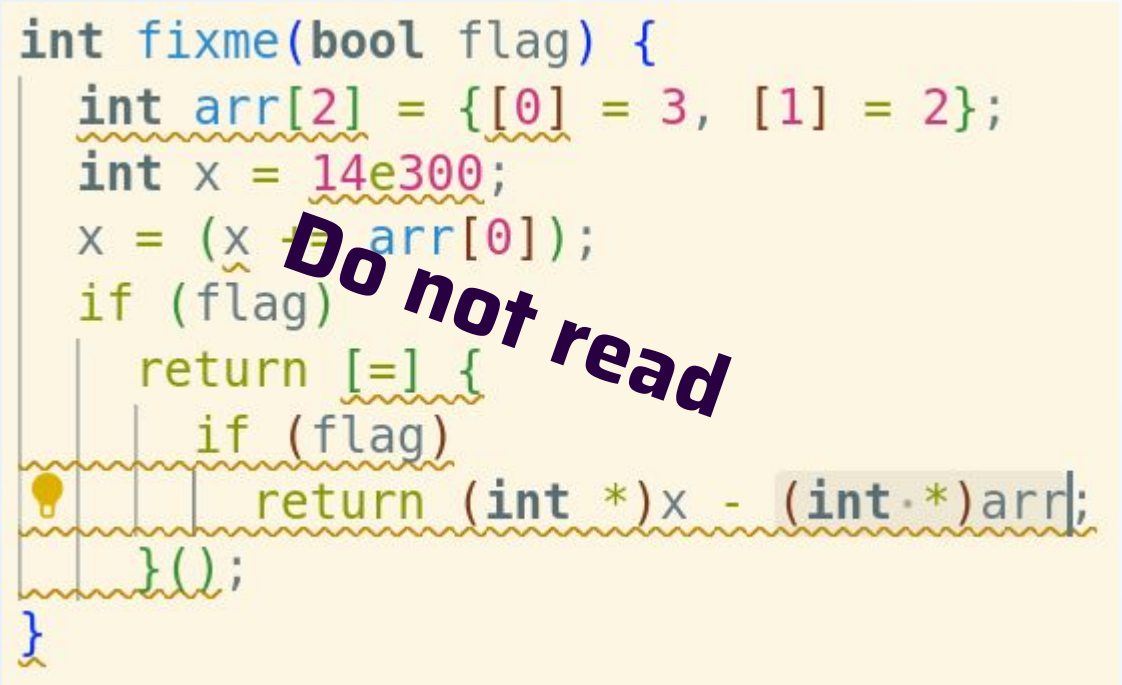
Interactive Reports:
Understandable & Complete

sonar

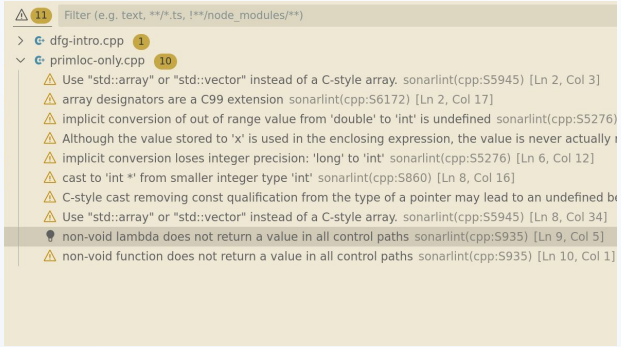
Issues
by Complexity

sonar

Simple Issues: Single Source Location




```
6 warnings generated.
/home/arseniy/proj/eurollvm-talk/primloc-only.cpp:2:17: warning: array designators are a C99 extension [clang-diagnostic-c99-designator]
  int arr[2] = {[0] = 3, [1] = 2};
                ^~
/home/arseniy/proj/eurollvm-talk/primloc-only.cpp:3:11: warning: implicit conversion of out of range value from 'double' to 'int' is undefined [clang-diagnostic-literal-conversion]
  int x = 14e300;
      ~   ^~~~~
/home/arseniy/proj/eurollvm-talk/primloc-only.cpp:4:8: warning: Although the value stored to 'x' is used in the enclosing expression, the value is never actually read from 'x' [clang-analyzer-deadcode.DeadStores]
  x = (x += arr[0]);
       ^  ~~~~~~
/home/arseniy/proj/eurollvm-talk/primloc-only.cpp:8:16: warning: cast to 'int *' from smaller integer type 'int' [clang-diagnostic-int-to-pointer-cast]
        return (int *)x - (int *)arr;
               ^~~~~~~~
/home/arseniy/proj/eurollvm-talk/primloc-only.cpp:9:5: warning: non-void lambda does not return a value in all control paths [clang-diagnostic-return-type]
    }();
    ^
/home/arseniy/proj/eurollvm-talk/primloc-only.cpp:10:1: warning: non-void function does not return a value in all control paths [clang-diagnostic-return-type]
}
^
```



sonar

Simple Issues: Plain-Text

```
6 warnings generated.
/home/arseniy/proj/eurollvm-talk/primloc-only.cpp:2:17: warning: array designators are a C99 extension [clang-diagnostic-c99-
designator]
  int arr[2] = {[0] = 3, [1] = 2};
                ^~~
/home/arseniy/proj/eurollvm-talk/primloc-only.cpp:3:11: warning: implicit conversion of out of range value from 'double' to '
int' is undefined [clang-diagnostic-literal-conversion]
  int x = 14e300;
      ~   ^~~~~
/home/arseniy/proj/eurollvm-talk/primloc-only.cpp:4:8: warning: Although the value stored to 'x' is used in the enclosing exp
ression, the value is never actually read from 'x' [clang-analyzer-deadcode.DeadStores]
  x = (x += arr[0]);
       ^    ~~~~~~
/home/arseniy/proj/eurollvm-talk/primloc-only.cpp:8:16: warning: cast to 'int *' from smaller integer type 'int' [clang-diagn
ostic-int-to-pointer-cast]
        return (int *)x - (int *)arr;
               ^~~~~~~~
/home/arseniy/proj/eurollvm-talk/primloc-only.cpp:9:5: warning: non-void lambda does not return a value in all control paths
[clang-diagnostic-return-type]
    }();
    ^
/home/arseniy/proj/eurollvm-talk/primloc-only.cpp:10:1: warning: non-void function does not return a value in all control pat
hs [clang-diagnostic-return-type]
}
^
```

Do not read

sonar

Simple Issues: GUI

sonar

Involved Issues: Multiple Locations

```cpp
#include <vector>
#include <string>

void myvec() {
  std::vector<std::string> vs{1, 2};
}
```

```
many-secondaries-irrelevant.cpp:5:28: error: no matching constructor for initialization of
      'std::vector<std::string>' (aka 'vector<basic_string<char>>')
  std::vector<std::string> vs{1, 2};
                           ^ ~~~~~~
/usr/bin/../lib/gcc/x86_64-linux-gnu/11/../../../../include/c++/11/bits/stl_vector.h:510:7: note:
      candidate constructor not viable: no known conversion from 'int' to 'const allocator_type' (aka 'con
      std::allocator<std::basic_string<char>>') for 2nd argument
      vector(size_type __n, const allocator_type& __a = allocator_type())
      ^
/usr/bin/../lib/gcc/x86_64-linux-gnu/11/../../../../include/c++/11/bits/stl_vector.h:522:7: note:
      candidate constructor not viable: no known conversion from 'int' to 'const value_type'
      (aka 'const std::basic_string<char>') for 2nd argument
      vector(size_type __n, const value_type& __value,
      ^
/usr/bin/../lib/gcc/x86_64-linux-gnu/11/../../../../include/c++/11/bits/stl_vector.h:575:7: note:
      candidate constructor not viable: no known conversion from 'int' to 'const vector<basic_string<char>
      vector(const vector& __x, const allocator_type& __a)
      ^
/usr/bin/../lib/gcc/x86_64-linux-gnu/11/../../../../include/c++/11/bits/stl_vector.h:607:7: note:
      candidate constructor not viable: no known conversion from 'int' to 'vector<basic_string<char>>' for 1st argument
      vector(vector&& __rv, const allocator_type& __m)
      ^
/usr/bin/../lib/gcc/x86_64-linux-gnu/11/../../../../include/c++/11/bits/stl_vector.h:625:7: note:
      candidate constructor not viable: no known conversion from 'int' to 'initializer_list<value_type>' (aka
      'initializer_list<std::basic_string<char>>') for 1st argument
      vector(initializer_list<value_type> __l,
      ^
/usr/bin/../lib/gcc/x86_64-linux-gnu/11/../../../../include/c++/11/bits/stl_vector.h:653:2: note:
      candidate template ignored: substitution failure [with _InputIterator = int]: no type named 'iterator_category' in
      'std::iterator_traits<int>'
        vector(_InputIterator __first, _InputIterator __last,
        ^
/usr/bin/../lib/gcc/x86_64-linux-gnu/11/../../../../include/c++/11/bits/stl_vector.h:497:7: note:
      candidate constructor not viable: requires single argument '__a', but 2 arguments were provided
      vector(const allocator_type& __a) _GLIBCXX_NOEXCEPT
      ^
/usr/bin/../lib/gcc/x86_64-linux-gnu/11/../../../../include/c++/11/bits/stl_vector.h:553:7: note:
      candidate constructor not viable: requires single argument '__x', but 2 arguments were provided
      vector(const vector& __x)
      ^
/usr/bin/../lib/gcc/x86_64-linux-gnu/11/../../../../include/c++/11/bits/stl_vector.h:572:7: note:
      candidate constructor not viable: requires 1 argument, but 2 were provided
      vector(vector&&) noexcept = default;
      ^
/usr/bin/../lib/gcc/x86_64-linux-gnu/11/../../../../include/c++/11/bits/stl_vector.h:585:7: note:
      candidate constructor not viable: requires 3 arguments, but 2 were provided
      vector(vector&& __rv, const allocator_type& __m, true_type) noexcept
      ^
      vector() = default;
      ^
1 error generated.
```

Do not read

Do not read

sonar

Multiple Locations: GUI

"InheritableAttr", and remove the ones you manually duplicated.

Code Smell +3

Add a using-declaration to this derived class to inherit the constructors of "InheritableParamAttr", and remove the ones you manually duplicated.

Code Smell +3

clang/include/clang/AST/StmtIterator.h

Add a using-declaration to this derived class to inherit the constructors of "StmtIteratorImpl", and remove the ones you manually duplicated.

Code Smell +7

1. The derived class
2. Removable constructor
3. Matching constructor in base class
4. Removable constructor
5. Matching constructor in base class
6. Removable constructor
7. Matching constructor in base class

Clang | clang/include/clang/AST/StmtIterator.h | See all issues in this file

```
127
128  struct [1] StmtIterator : public StmtIteratorImpl<StmtIterator, Stmt*&> {
```

Add a using-declaration to this derived class to inherit the constructors of "StmtIteratorImpl", and remove the ones you manually duplicated.

```
129      explicit StmtIterator() = default;
130  [2] StmtIterator(Stmt** S) : StmtIteratorImpl<StmtIterator, Stmt*&>(S) {}
131  [4] StmtIterator(Decl** dgi, Decl** dge)
132          : StmtIteratorImpl<StmtIterator, Stmt*&>(dgi, dge) {}
133  [6] StmtIterator(const VariableArrayType *t)
134          : StmtIteratorImpl<StmtIterator, Stmt*&>(t) {}
135
136      private:
137      explicit StmtIterator(const StmtIteratorBase &RHS)
138          : StmtIteratorImpl<StmtIterator, Stmt *&>(RHS) {}
```

Do not read

```
84  …    using difference_type = std::ptrdiff_t;
85       using pointer = REFERENCE;
86       using reference = REFERENCE;
87
88       StmtIteratorImpl() = default;
89   [3] StmtIteratorImpl(Stmt **s) : StmtIteratorBase(s) {}
90   [5] StmtIteratorImpl(Decl **dgi, Decl **dge) : StmtIteratorBase(dgi, dge) {}
91   [7] StmtIteratorImpl(const VariableArrayType *t) : StmtIteratorBase(t) {}
```

Path-Sensitive Issues: Flow

```
int getInt();
int *getPtr();

void fixme() {
  int x = getInt();
int y = getInt();
  1 int *p = getPtr();
  3 if ( 2 x < y) {
    5 if ( 4 p)
      x = 8;
    7 if ( 6 x == 3) {
      y = 8 *p;
    }
  }
}
```

- Dereference of null pointer (loaded from varia
  - Flow 1
    1. 'p' initialized here [7, 2]
    2. Assuming 'x' is < 'y' [8, 6]
    3. Taking true branch [8, 2]
    4. Assuming 'p' is null [9, 8]
    5. Taking false branch [9, 4]
    6. Assuming 'x' is equal to 3 [11, 8]
    7. Taking true branch [11, 4]
    8. Dereference of null pointer (loaded from variable 'p') [...

sonar

Path-Sensitive Issues: Flow

```
int getInt();
int *getPtr();

void fixme() {
  int x = getInt();
int y = getInt();
  1 int *p = getPtr();
  3 if ( 2 x < y) {
    5 if ( 4 p)
      x = 8;
    7 if ( 6 x == 3) {
      y =  8 *p;
    }
  }
}
```

Dereference c
- Flow 1
  - 1: 'p' initialized
  - 2: Assuming 'x
  - 3: Taking true
  - 4: Assuming 'p
  - 5: Taking false
  - 6: Assuming 'x
  - 7: Taking true
  - 8: Dereference

```
4  void fixme() {
5    int x = getInt();
6    int y = getInt();
7    int *p = getPtr();
       1 'p' initialized here →
8    if (x < y) {
       2 ← Assuming 'x' is < 'y' →
       3 ← Taking true branch →
9      if (p)
         4 ← Assuming 'p' is null →
         5 ← Taking false branch →
10       x = 8;
11     if (x == 3) {
         6 ← Assuming 'x' is equal to 3 →
         7 ← Taking true branch →
12       y = *p;
           8 ← Dereference of null pointer (loaded from variable 'p')
13     }
14   }
15 }
```

Null pointer passed as read buffer "ntdev.Buffer" in call to "wcsncasecmp"

Get permalink

Parameter values should be appropriate cpp:S3807

1 month ago L269

Bug Critical Open Not assigned 5min effort 0 comments symbolic-execution

Full execution flow +39

1. Null pointer passed as read buffer "ntdev.Buffer" in call to "wcsncasecmp"
2. Assuming the condition is false
3. Taking false branch
4. 'trailing' is null
5. Taking false branch
6. Assuming 'trailing' is null
7. Taking true branch
8. Assuming 'ntdev.Length' is > 'tgtdev.Length'
9. Taking true branch
10. Assuming the condition is true
11. Returning from 'RtlEqualUnicodePathPrefix'
12. Returning without writing to 'path->Buffer'
13. '?' condition is false
14. Assuming 'prefix->Length' is >= 'path->Length'

Where is the issue? | Why is this an issue?

msys2-runtime winsup/utils/cygpath.cc See all issues in this file

```
262                 ans.MaximumLength - ans.Length);
263           ans.Buffer[ans.MaximumLength - 1] = '\0';
264           got_one = true;
265           /* Special case for local disks:  It's most feasible if the
266          DOS device name reflects the DOS drive, so we check for this
267          explicitly and only return prematurely if so. */
268           if ( 2 ntdev.Length < wcslen (HARDDISK_PREFIX)
269           || 1 wcsncasecmp (ntdev.Buffer, HARDDISK_PREFIX, 8) != 0
270           || (odi->ObjectName.Length == 2 * sizeof (WCHAR)
271               && odi->ObjectName.Buffer[1] == L':'))
272         {
273           if (trailing)
274             {
275               /* If there's a trailing path, it's a perfectly valid
276              DOS pathname without the \\.\ prefix.  Unless it's
277              longer than MAX_PATH - 1 in which case it needs
278              the \\?\ prefix. */
```

msys2-runtime winsup/cygwin/local_includes/ntdll.h See all issues in this file

Null pointer passed as read buffer "ntdev.Buffer" in call to "wcsncasecmp"

Get permalink

Parameter values should be appropriate cpp:S3807

1 month ago L269

Bug Critical Open Not assigned 5min effort 0 comments symbolic-execution

Full execution flow +39

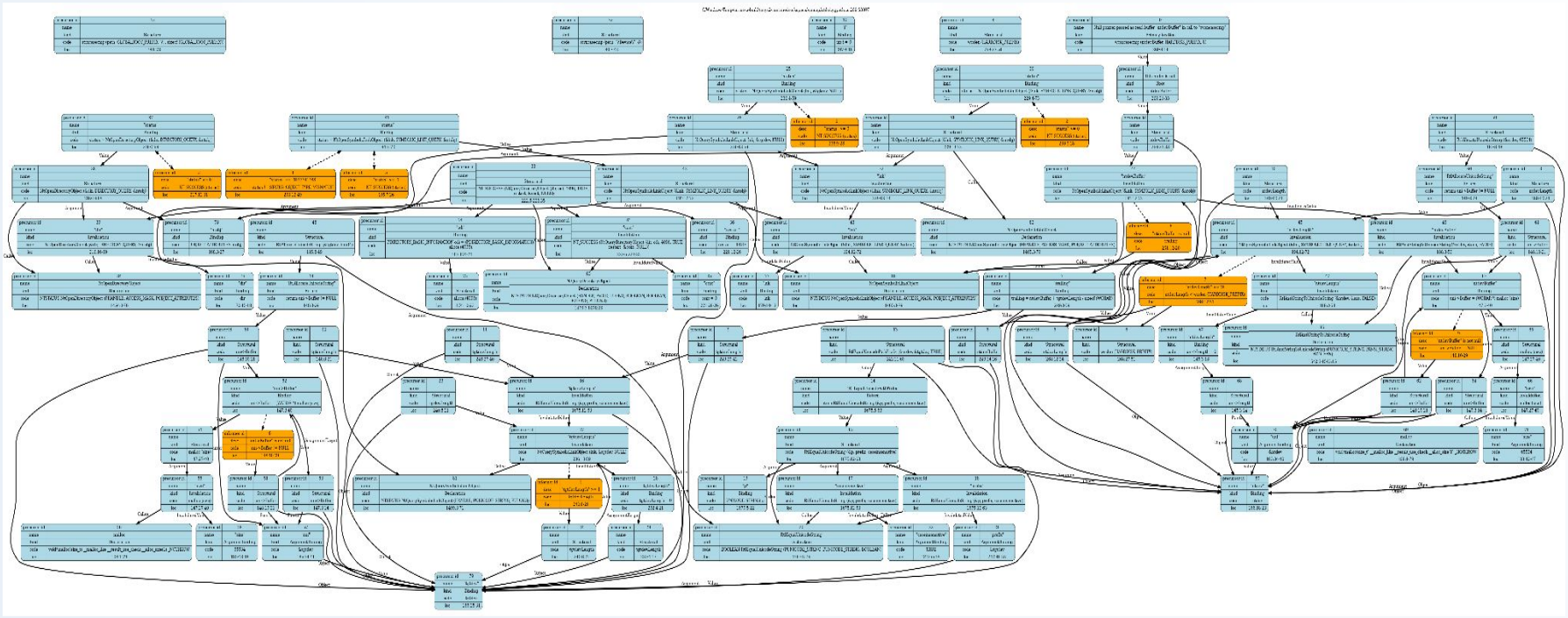
1. Null pointer passed as read buffer "ntdev.Buffer" in call to "wcsncasecmp"
2. Assuming the condition is false
3. Taking false branch
4. 'trailing' is null
5. Taking false branch
6. Assuming 'trailing' is null
7. Taking true branch
8. Assuming 'ntdev.Length' is > 'tgtdev.Length'
9. Taking true branch
10. Assuming the condition is true
11. Returning from 'RtlEqualUnicodePathPrefix'
12. Returning without writing to 'path->Buffer'
13. '?' condition is false
14. Assuming 'prefix->Length' is >= 'path->Length'

Where is the issue? | Why is this an issue?

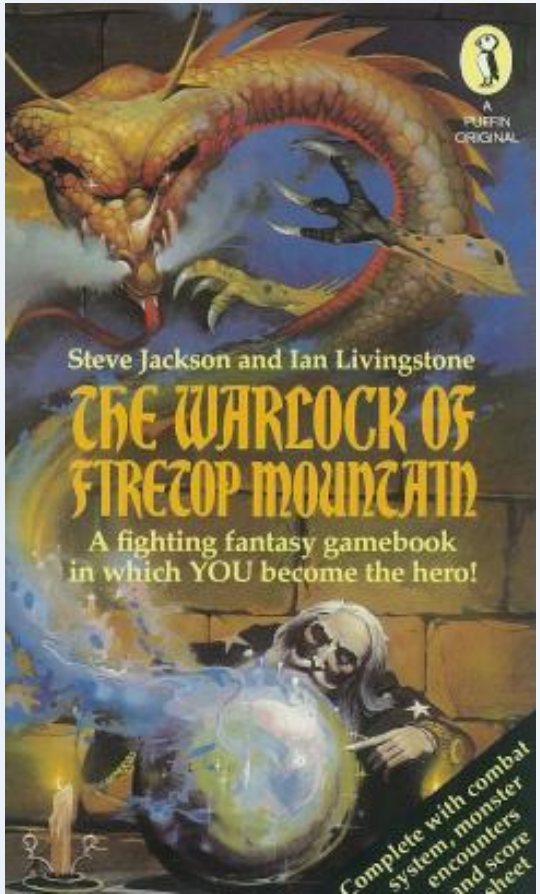
msys2-runtime winsup/utils/cygpath.cc See all issues in this file

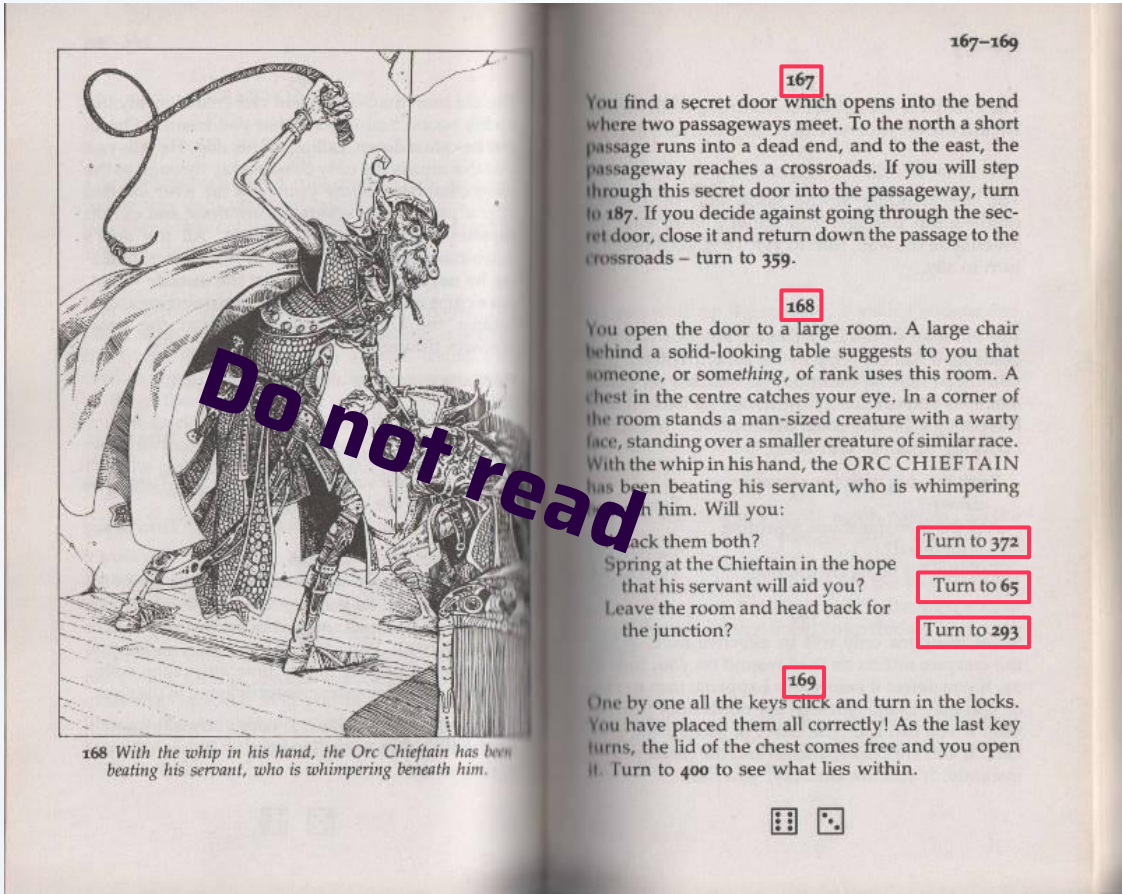
```
243         {
244           /* If the comparison succeeds, the name of the directory entry is
245            a valid DOS device name, if prepended with "\\.\".  Return that
246            valid DOS path. */
247           wchar_t *trailing = NULL;
248         7 if ( 8 ntdev.Length > tgtdev.Length)
249           trailing = ntdev.Buffer + tgtdev.Length / sizeof (WCHAR);
250           ULONG len = RtlUnicodeStringToAnsiSize (&odi->ObjectName);
251         5 if ( 6 trailing)
252           len += my_wcstombs (NULL, trailing, 0);
253           free (ret);
254           ret = (char *) malloc (len + 4);
255           strcpy (ret, "\\\\.\\");
256           ans.Length = 0;
257           ans.MaximumLength = len;
258           ans.Buffer = ret + 4;
259           RtlUnicodeStringToAnsiString (&ans, &odi->ObjectName, FALSE);
260         3 if ( 4 trailing)
261           my_wcstombs (ans.Buffer + ans.Length, trailing,
```

DFG



Choose Your Own Adventure





168 *With the whip in his hand, the Orc Chieftain has been beating his servant, who is whimpering beneath him.*

167

You find a secret door which opens into the bend where two passageways meet. To the north a short passage runs into a dead end, and to the east, the passageway reaches a crossroads. If you will step through this secret door into the passageway, turn to 187. If you decide against going through the secret door, close it and return down the passage to the crossroads – turn to 359.

168

You open the door to a large room. A large chair behind a solid-looking table suggests to you that someone, or something, of rank uses this room. A chest in the centre catches your eye. In a corner of the room stands a man-sized creature with a warty face, standing over a smaller creature of similar race. With the whip in his hand, the ORC CHIEFTAIN has been beating his servant, who is whimpering beneath him. Will you:

Attack them both?	Turn to 372
Spring at the Chieftain in the hope that his servant will aid you?	Turn to 65
Leave the room and head back for the junction?	Turn to 293

169

One by one all the keys click and turn in the locks. You have placed them all correctly! As the last key turns, the lid of the chest comes free and you open it. Turn to 400 to see what lies within.

Data Flow Based Interactive Report

sonar

Null pointer passed as read buffer "ntdev.Buffer" in call to "wcsncasecmp".

Relevant values:

This buffer is null:
Use: ntdev.Buffer
Def: ntdev.Buffer

```
239             continue;
240           if (tgtdev.Length /* There's actually a symlink pointing to an
241                                empty string: \??\GLOBALROOT -> "" */
242               && RtlEqualUnicodePathPrefix (&ntdev, &tgtdev, TRUE))
243             {
244               /* If the comparison succeeds, the name of the directory entry is
245                  a valid DOS device name, if prepended with "\\.\".  Return that
246                  valid DOS path. */
247               wchar_t *trailing = NULL;
248               if (ntdev.Length > tgtdev.Length)
249                 trailing = ntdev.Buffer + tgtdev.Length / sizeof (WCHAR);
250               ULONG len = RtlUnicodeStringToAnsiSize (&odi->ObjectName);
251               if (trailing)
252                 len += my_wcstombs (NULL, trailing, 0);
253               free (ret);
254               ret = (char *) malloc (len + 4);
255               strcpy (ret, "\\\\.\\");
256               ans.Length = 0;
257               ans.MaximumLength = len;
258               ans.Buffer = ret + 4;
259               RtlUnicodeStringToAnsiString (&ans, &odi->ObjectName, FALSE);
260               if (trailing)
261                 my_wcstombs (ans.Buffer + ans.Length, trailing,
262                              ans.MaximumLength - ans.Length);
263               ans.Buffer[ans.MaximumLength - 1] = '\0';
264               got_one = true;
265               /* Special case for local disks:  It's most feasible if the
266                  DOS device name reflects the DOS drive, so we check for this
267                  explicitly and only return prematurely if so. */
268               if (ntdev.Length < wcslen (HARDDISK_PREFIX)
269                   || wcsncasecmp (ntdev.Buffer, HARDDISK_PREFIX, 8) != 0
270                   || (odi->ObjectName.Length == 2 * sizeof (WCHAR)
271                       && odi->ObjectName.Buffer[1] == L':'))
272                 {
273                   if (trailing)
274                     {
275                       /* If there's a trailing path, it's a perfectly valid
276                          DOS pathname without the \\.\ prefix.  Unless it's
277                          longer than MAX_PATH - 1 in which case it needs
278                          the \\?\ prefix. */
279                       if ((len = strlen (ret + 4)) >= MAX_PATH)
280                         ret[2] = '?';
281                       else
282                         memmove (ret, ret + 4, strlen (ret + 4) + 1);
```

Null pointer passed as read buffer "ntdev.Buffer" in call to "wcsncasecmp".

Explored points:

This buffer is null:
Use: ntdev.Buffer
Def: ntdev.Buffer

ntdev.Buffer

Relevant values:

"ntdev.Buffer":
Use: ntdev.Buffer
Invalidation Def: NtOpenSymbolicLinkObject (&lnk, SYMBOLIC_LINK_QUERY, &ntobj)

"ntdev":
Use: ntdev
Def: ntdev

```
239             continue;
240           if (tgtdev.Length /* There's actually a symlink pointing to an
241                                empty string: \??\GLOBALROOT -> "" */
242               && RtlEqualUnicodePathPrefix (&ntdev, &tgtdev, TRUE))
243             {
244               /* If the comparison succeeds, the name of the directory entry is
245                  a valid DOS device name, if prepended with "\\.\".  Return that
246                  valid DOS path. */
247               wchar_t *trailing = NULL;
248               if (ntdev.Length > tgtdev.Length)
249                 trailing = ntdev.Buffer + tgtdev.Length / sizeof (WCHAR);
250               ULONG len = RtlUnicodeStringToAnsiSize (&odi->ObjectName);
251               if (trailing)
252                 len += my_wcstombs (NULL, trailing, 0);
253               free (ret);
254               ret = (char *) malloc (len + 4);
255               strcpy (ret, "\\\\.\\");
256               ans.Length = 0;
257               ans.MaximumLength = len;
258               ans.Buffer = ret + 4;
259               RtlUnicodeStringToAnsiString (&ans, &odi->ObjectName, FALSE);
260               if (trailing)
261                 my_wcstombs (ans.Buffer + ans.Length, trailing,
262                              ans.MaximumLength - ans.Length);
263               ans.Buffer[ans.MaximumLength - 1] = '\0';
264               got_one = true;
265               /* Special case for local disks:  It's most feasible if the
266                  DOS device name reflects the DOS drive, so we check for this
267                  explicitly and only return prematurely if so. */
268               if (ntdev.Length < wcslen (HARDDISK_PREFIX)
269                   || wcsncasecmp (ntdev.Buffer, HARDDISK_PREFIX, 8) != 0
270                   || (odi->ObjectName.Length == 2 * sizeof (WCHAR)
271                       && odi->ObjectName.Buffer[1] == L':'))
272                 {
273                   if (trailing)
274                     {
275                       /* If there's a trailing path, it's a perfectly valid
276                          DOS pathname without the \\.\ prefix.  Unless it's
277                          longer than MAX_PATH - 1 in which case it needs
278                          the \\?\ prefix. */
279                       if ((len = strlen (ret + 4)) >= MAX_PATH)
280                         ret[2] = '?';
281                       else
```

Null pointer passed as read buffer "ntdev.Buffer" in call to "wcsncasecmp".

Explored points:

This buffer is null:
Use: ntdev.Buffer
Def: ntdev.Buffer

ntdev.Buffer

"ntdev.Buffer":
Use: ntdev.Buffer
Invalidation Def: NtOpenSymbolicLinkObject (&lnk, SYMBOLIC_LINK_QUERY, &ntobj)

Relevant values:

"ntdev.Buffer":
Use: NtOpenSymbolicLinkObject (&lnk, SYMBOLIC_LINK_QUERY, &ntobj)
Invalidation Def: RtlAnsiStringToUnicodeString (&ntdev, &ans, FALSE)

NtOpenSymbolicLinkObject:
Use: NtOpenSymbolicLinkObject
Def: NTSTATUS NtOpenSymbolicLinkObject (PHANDLE, ACCESS_MASK, POBJECT_ATTRIBUTES)

Assumptions about "ntdev.Buffer":

trailing
"ntdev.Buffer" is null

```
164 get_device_name (char *path)
165 {
166   UNICODE_STRING ntdev, tgtdev, ntdevdir;
167   ANSI_STRING ans;
168   OBJECT_ATTRIBUTES ntobj;
169   NTSTATUS status;
170   HANDLE lnk, dir;
171   bool got_one = false;
172   char *ret = strdup (path);
173   PDIRECTORY_BASIC_INFORMATION odi = (PDIRECTORY_BASIC_INFORMATION)
174                                      alloca (4096);
175   BOOLEAN restart;
176   ULONG cont;
177
178   if (!strncasecmp (path, GLOBALROOT_PREFIX "\\", sizeof (GLOBALROOT_PREFIX)))
179     path += sizeof (GLOBALROOT_PREFIX) - 1;
180   if (strncasecmp (path, "\\Device\\", 8))
181     return ret;
182
183   if (!RtlAllocateUnicodeString (&ntdev, 65534))
184     return ret;
185   if (!RtlAllocateUnicodeString (&tgtdev, 65534))
186     return ret;
187   RtlInitAnsiString (&ans, path);
188   RtlAnsiStringToUnicodeString (&ntdev, &ans, FALSE);
189
190   /* First check if the given device name is a symbolic link itself.  If so,
191      query it and use the new name as actual device name to search for in the
192      DOS device name directory.  If not, just use the incoming device name. */
193   InitializeObjectAttributes (&ntobj, &ntdev, OBJ_CASE_INSENSITIVE, NULL, NULL);
194   status = NtOpenSymbolicLinkObject (&lnk, SYMBOLIC_LINK_QUERY, &ntobj);
195   if (NT_SUCCESS (status))
196     {
197       status = NtQuerySymbolicLinkObject (lnk, &tgtdev, NULL);
198       NtClose (lnk);
199       if (!NT_SUCCESS (status))
200         goto out;
201       RtlCopyUnicodeString (&ntdev, &tgtdev);
202     }
203   else if (status != STATUS_OBJECT_TYPE_MISMATCH
204            && status != STATUS_OBJECT_PATH_SYNTAX_BAD)
205     goto out;
206
```

Null pointer passed as read buffer "ntdev.Buffer" in call to "wcsncasecmp".

Explored points:

This buffer is null:
Use: ntdev.Buffer
Def: ntdev.Buffer

ntdev.Buffer

"ntdev.Buffer":
Use: ntdev.Buffer
Invalidation Def: NtOpenSymbolicLinkObject (&lnk, SYMBOLIC_LINK_QUERY, &ntobj)

trailing
"ntdev.Buffer" is null

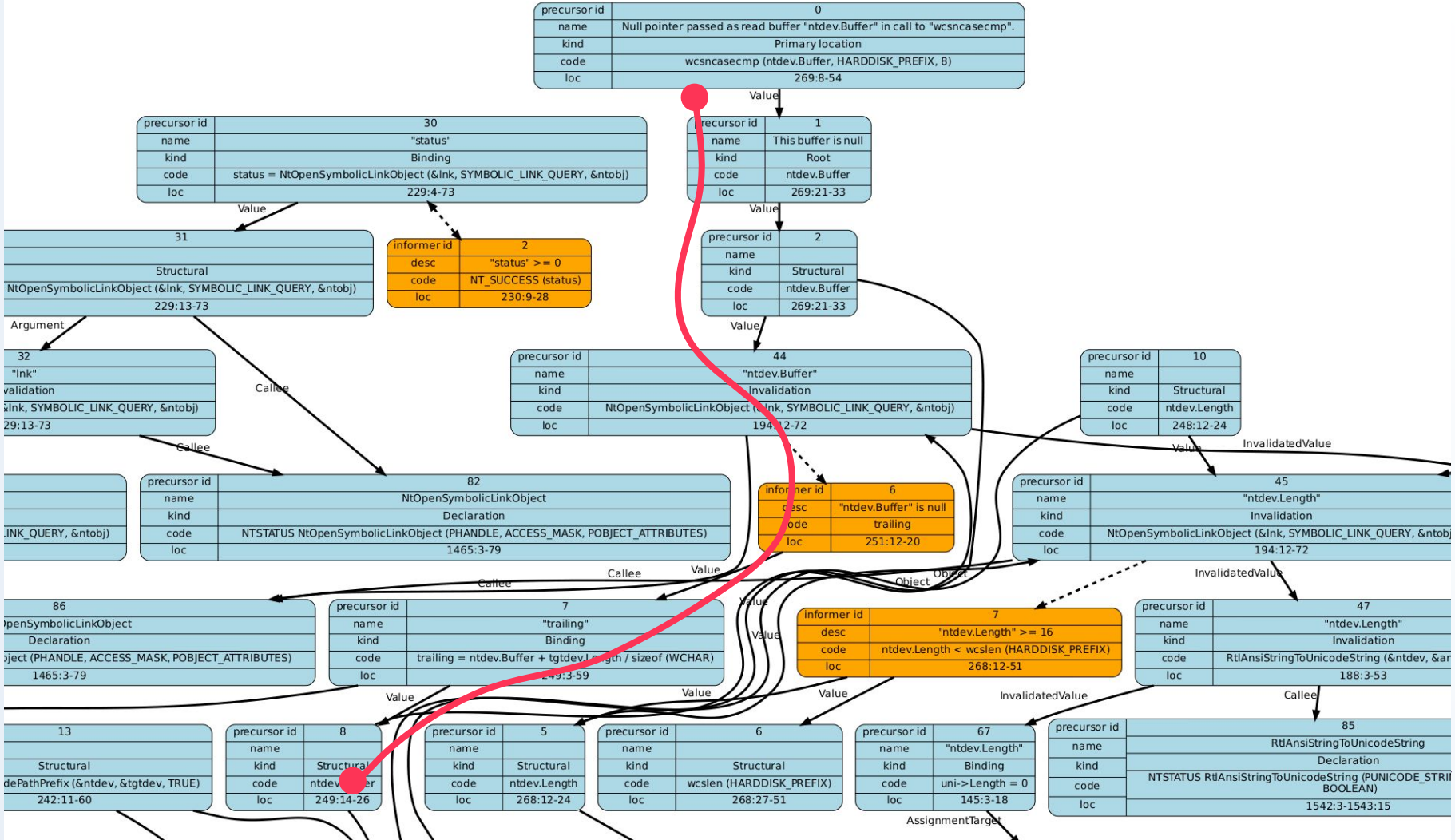
"trailing":
Use: trailing
Def: trailing = ntdev.Buffer + tgtdev.Length / sizeof (WCHAR)

Relevant values:

ntdev.Buffer

tgtdev.Length

```
219
220       /* ...and scan it. */
221       for (restart = TRUE, cont = 0;
222            NT_SUCCESS (NtQueryDirectoryObject (dir, odi, 4096, TRUE,
223                                                restart, &cont, NULL));
224            restart = FALSE)
225         {
226           /* For each entry check if it's a symbolic link. */
227           InitializeObjectAttributes (&ntobj, &odi->ObjectName,
228                                       OBJ_CASE_INSENSITIVE, dir, NULL);
229           status = NtOpenSymbolicLinkObject (&lnk, SYMBOLIC_LINK_QUERY, &ntobj);
230           if (!NT_SUCCESS (status))
231             continue;
232           tgtdev.Length = 0;
233           tgtdev.MaximumLength = 512;
234           /* If so, query it and compare the target of the symlink with the
235              incoming device name. */
236           status = NtQuerySymbolicLinkObject (lnk, &tgtdev, NULL);
237           NtClose (lnk);
238           if (!NT_SUCCESS (status))
239             continue;
240           if (tgtdev.Length /* There's actually a symlink pointing to an
241                                empty string: \??\GLOBALROOT -> "" */
242               && RtlEqualUnicodePathPrefix (&ntdev, &tgtdev, TRUE))
243             {
244               /* If the comparison succeeds, the name of the directory entry is
245                  a valid DOS device name, if prepended with "\\.\".  Return that
246                  valid DOS path. */
247               wchar_t *trailing = NULL;
248               if (ntdev.Length > tgtdev.Length)
249                 trailing = ntdev.Buffer + tgtdev.Length / sizeof (WCHAR);
250               ULONG len = RtlUnicodeStringToAnsiSize (&odi->ObjectName);
251               if (trailing)
252                 len += my_wcstombs (NULL, trailing, 0);
253               free (ret);
254               ret = (char *) malloc (len + 4);
255               strcpy (ret, "\\\\.\\");
256               ans.Length = 0;
257               ans.MaximumLength = len;
258               ans.Buffer = ret + 4;
259               RtlUnicodeStringToAnsiString (&ans, &odi->ObjectName, FALSE);
260               if (trailing)
261                 my_wcstombs (ans.Buffer + ans.Length, trailing,
```


precursor id 0
name Null pointer passed as read buffer "ntdev.Buffer" in call to "wcsncasecmp".
kind Primary location
code wcsncasecmp (ntdev.Buffer, HARDDISK_PREFIX, 8)
loc 269:8-54
Value
precursor id 30
name "status"
kind Binding
code status = NtOpenSymbolicLinkObject (&lnk, SYMBOLIC_LINK_QUERY, &ntobj)
loc 229:4-73
precursor id 1
name This buffer is null
kind Root
code ntdev.Buffer
loc 269:21-33
Value
Value
31
Structural
NtOpenSymbolicLinkObject (&lnk, SYMBOLIC_LINK_QUERY, &ntobj)
229:13-73
informer id 2
desc "status" >= 0
code NT_SUCCESS (status)
loc 230:9-28
precursor id 2
name
kind Structural
code ntdev.Buffer
loc 269:21-33
Argument
Value
32
"lnk"
Callee
Callee
precursor id 44
name "ntdev.Buffer"
kind Invalidation
code NtOpenSymbolicLinkObject (&lnk, SYMBOLIC_LINK_QUERY, &ntobj)
loc 194:12-72
precursor id 10
name
kind Structural
code ntdev.Length
loc 248:12-24
Value
InvalidatedValue
precursor id 82
name NtOpenSymbolicLinkObject
kind Declaration
code NTSTATUS NtOpenSymbolicLinkObject (PHANDLE, ACCESS_MASK, POBJECT_ATTRIBUTES)
loc 1465:3-79
informer id 6
desc "ntdev.Buffer" is null
code trailing
loc 251:12-20
precursor id 45
name "ntdev.Length"
kind Invalidation
code NtOpenSymbolicLinkObject (&lnk, SYMBOLIC_LINK_QUERY, &ntobj)
loc 194:12-72
Callee
Callee
Value
Object
InvalidatedValue
86
Declaration
1465:3-79
precursor id 7
name "trailing"
kind Binding
code trailing = ntdev.Buffer + tgtdev.Length / sizeof (WCHAR)
loc 249:3-59
Value
Value
Value
Value
informer id 7
desc "ntdev.Length" >= 16
code ntdev.Length < wcslen (HARDDISK_PREFIX)
loc 268:12-51
precursor id 47
name "ntdev.Length"
kind Invalidation
code RtlAnsiStringToUnicodeString (&ntdev, &ar
loc 188:3-53
Value
Value
Value
InvalidatedValue
Callee
13
Structural
242:11-60
precursor id 8
name
kind Structural
code ntdev.Buffer
loc 249:14-26
precursor id 5
name
kind Structural
code ntdev.Length
loc 268:12-24
precursor id 6
name
kind Structural
code wcslen (HARDDISK_PREFIX)
loc 268:27-51
precursor id 67
name "ntdev.Length"
kind Binding
code uni->Length = 0
loc 145:3-18
precursor id 85
name RtlAnsiStringToUnicodeString
Declaration
code
loc 1542:3-1543:15
AssignmentTarget

Conclusion

- Clarity is as important as precision
- GUI overcomes limitations of plain-text
- GUI capabilities are underutilized
- The *completeness* *vs* *brevity* tradeoff is rudimentary

sonar