N1860: Proposed new rule for TS 17961

Author: Clive Pygott 6/9/2014

Background

At the WG14 meeting in Parma, I presented a defect report on rule 5.21 – N1832. The issue was that the example didn't match the headline rule. A change to the example was agree, however, I don't believe that 5.21 either matches the drafting committee's original intension or indeed does anything particularly useful.

This proposal is therefore for a new rule, that I believe both matches what we intended to say and arguably, says something worthwhile.

The existing rule 5.21

Just as a reminder, rule 5.21 (in summary) says:

*A call to a standard memory allocation function is presumed to be intended for type T * when it appears in any of the following contexts:*

- *in the right operand of an assignment to an object of type T *, or*
- *in an initializer for an object of type T *, or*
- *in an expression that is passed as an argument of type T *, or*
- *in the expression of a return statement for a function returning type T *.*

*A call to a standard memory allocation function taking a size integer argument n and presumed to be intended for type T * shall be diagnosed when n < sizeof(T).*

Rationale

Returning insufficient memory from a memory allocation function is likely to result in undefined behaviour when that memory is accessed.

The proposed new rule

In summary, the proposed rule says that such an allocation should be treated as an array of n/sizeof(T) elements, and a diagnosis is required if the array is treated in a manner that violates this bound.

*A call to a standard memory allocation function is presumed to be intended for type T * when it appears in any of the following contexts:*

- *in the right operand of an assignment to an object of type T *, or*
- *in an initializer for an object of type T *, or*
- *in an expression that is passed as an argument of type T *, or*
- *in the expression of a return statement for a function returning type T *.*

*A call to a standard memory allocation function taking a size integer argument n and presumed to be intended for type T * shall be regarded as an array of N elements, where N = n / sizeof(T).*

Any attempt to use this array in a manner that causes its array bound to be violated shall be diagnosed.

The following are the standard memory allocation functions that take a size integer argument and return a pointer:

aligned_alloc
calloc
malloc
realloc

Rationale

Attempting to access an array with an index larger than its array bound (buffer overrun) leads to undefined behaviour and is a root cause of many security vulnerabilities

Example

```
wchar_t *f1(void) {
   const wchar_t *p = L"Hello, World!";
   const size_t n = (wcslen(p) + 1);     // n == 14
   wchar_t *q = (wchar_t *)malloc(n);
   wcscpy(q, p); // diagnostic required
                 // q is treated as wchar_t q[7];
                 // but 14 character are to be copied
   return q;
}
```

Additional points

1. The proposed rule could replace the current 5.21 by adding the requirement that n/sizeof(T) >= 1