



TECHNICAL SPECIFICATION ISO/IEC TS 17961:2013
TECHNICAL CORRIGENDUM 1

Published 2016-08-15

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION • МЕЖДУНАРОДНАЯ ОРГАНИЗАЦИЯ ПО СТАНДАРТИЗАЦИИ • ORGANISATION INTERNATIONALE DE NORMALISATION
INTERNATIONAL ELECTROTECHNICAL COMMISSION • МЕЖДУНАРОДНАЯ ЭЛЕКТРОТЕХНИЧЕСКАЯ КОМИССИЯ • COMMISSION ÉLECTROTECHNIQUE INTERNATIONALE

Information technology — Programming languages, their environments and system software interfaces — C secure coding rules

TECHNICAL CORRIGENDUM 1

Technologies de l'information — Langages de programmation, leur environnement et interfaces des logiciels de systèmes — Règles de programmation sécurisée en C

RECTIFICATIF TECHNIQUE 1

Technical Corrigendum 1 to ISO/IEC TS 17961:2013 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 22, *Programming languages, their environments and system software interfaces*.

In rule 5.21, Rule section, replace

A call to a standard memory allocation function taking a size integer argument n and presumed to be intended for type τ * shall be diagnosed when $n < \text{sizeof}(\tau)$.

with

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

Any allocation where $N == 0$ shall be diagnosed (i.e. where $n < \text{sizeof}(\tau)$). Also, any attempt to use this array in a manner that causes its array bound to be violated shall be diagnosed.

In rule 5.21, replace

EXAMPLE In this noncompliant example, a diagnostic is required because the value of n that is used in the `malloc()` call has been possibly miscalculated.

```
wchar_t *f1(void) {
    const wchar_t *p = L"Hello, World!";
    const size_t n = sizeof(p) * (wcslen(p) + 1);
    wchar_t *q = (wchar_t *)malloc(n); // diagnostic required

    /* ... */
    return q;
}
```

with

EXAMPLE 1

```
struct S1 {
    unsigned int x;
    float      y;
    struct S1  *z;
};

struct S1 *f1(void) {
    struct S1 *p = (struct S1*)malloc(sizeof(p)); // diagnostic required
    return p;
}
```