PRE09-C. Do not replace secure functions with deprecated or obsolescent functions

Macros are frequently used in the remediation of existing code to globally replace one identifier with another, for example, when an existing API changes. Although some risk is always involved, this practice becomes particularly dangerous if a function name is replaced with the function name of a deprecated or obsolescent function. Deprecated functions are defined by the C Standard and Technical Corrigenda. Obsolescent functions are defined by MSC24-C. Do not use deprecated or obsolescent functions.

Although compliance with rule MSC24-C. Do not use deprecated or obsolescent functions guarantees compliance with this recommendation, the emphasis of this recommendation is the extremely risky and deceptive practice of replacing functions with less secure alternatives.

Noncompliant Code Example

The Internet Systems Consortium's (ISC) Dynamic Host Configuration Protocol (DHCP) contained a vulnerability that introduced several potential buffer overflow conditions [VU#654390]. ISC DHCP makes use of the `vsnprintf()` function for writing various log file strings; `vsnprintf()` is defined in the Portable Operating System Interface (POSIX®), Base Specifications, Issue 7 [IEEE Std 1003.1:2013] as well as in the C Standard. For systems that do not support `vsnprintf()`, a C include file was created that defines the `vsnprintf()` function to `vsprintf()`, as shown in this noncompliant code example:

```
#define vsnprintf(buf, size, fmt, list) \
vsprintf(buf, fmt, list)
```

The `vsprintf()` function does not check bounds. Consequently, `size` is discarded, creating the potential for a buffer overflow when untrusted data is used.

Compliant Solution

The solution is to include an implementation of the missing function `vsnprintf()` to eliminate the dependency on external library functions when they are not available. This compliant solution assumes that `__USE_ISOC11` is not defined on systems that fail to provide a `vsnprintf()` implementation:

```
#include <stdio.h>
#ifndef __USE_ISOC11
  /* Reimplements vsnprintf() */
  #include "my_stdio.h"
#endif
```

Risk Assessment

Replacing secure functions with less secure functions is a very risky practice because developers can be easily fooled into trusting the function to perform a security check that is absent. This may be a concern, for example, as developers attempt to adopt more secure functions, such as the C11 Annex K functions, that might not be available on all platforms. (See STR07-C. Use the bounds-checking interfaces for string manipulation.)

Recommendation	Severity	Likelihood	Remediation Cost	Priority	Level
PRE09-C	High	Likely	Medium	**P18**	**L1**

Automated Detection

Tool	Version	Checker	Description
Astrée	19.04		Supported, but no explicit checker
Axivion Bauhaus Suite	6.9.0	**CertC-PRE09**	
Polyspace Bug Finder	R2019b	CERT C: Rec. PRE09-C	Checks for use of dangerous standard function (rec. fully covered)
PRQA QA-C	9.7	**5003**	Fully implemented

Related Vulnerabilities

Search for vulnerabilities resulting from the violation of this rule on the CERT website.

Related Guidelines

SEI CERT C++ Coding Standard	VOID PRE09-CPP. Do not replace secure functions with less secure functions
ISO/IEC TR 24772:2013	Executing or Loading Untrusted Code [XYS]
MITRE CWE	CWE-684, Failure to provide specified functionality

Bibliography

[IEEE Std 1003.1:2013]	XSH, System Interfaces, vsnprintf, `vsprintf`
[Seacord 2013]	Chapter 6, "Formatted Output"
[VU#654390]	