JTC 1/SC 22/WG 23 N 0656

A Mapping from the New Vulnerabilities to CERT Secure Coding Rules

David Keaton
2016-05-16

Introduction

This is a list of CERT secure coding rules that can be referenced by the new vulnerabilities in TR 24772-1. For the concurrency rules, the CERT C Secure Coding Standard was used because that is the most mature. For the OOP rules, the CERT C++ Coding Standard was used.

Concurrency

6.60 [CGA] Activation

CERT:
ERR-33C. Detect and handle standard library errors

6.61 [CGT] Directed termination

CERT:
ERR-33C. Detect and handle standard library errors

6.62 [CGX] Concurrent data access

CERT:
CON32-C. Prevent data races when accessing bit-fields from multiple threads
CON33-C. Avoid race conditions when using library functions
CON34-C. Declare objects shared between threads with appropriate storage durations
CON40-C. Do not refer to an atomic variable twice in an expression

6.63 [CGS] Premature termination

CERT: N/A

6.64 [CGM] Protocol lock errors

CERT:
CON31-C. Do not destroy a mutex while it is locked
CON35-C. Avoid deadlock by locking in a predefined order
CON38-C. Preserve thread safety and liveness when using condition variables

OOP

6.41 [SYM] Templates and generics

CERT: N/A

6.42 [RIP] Inheritance

CERT:

OOP50-CPP. Do not invoke virtual functions from constructors or destructors

OOP51-CPP. Do not slice derived objects

OOP52-CPP. Do not delete a polymorphic object without a virtual destructor

6.43 [BLP] Violations of the Liskov principle or the contract model

CERT: N/A

6.44 [PPH] Redispatching

CERT: N/A

6.45 [BKK] Polymorphic variables

CERT:

OOP51-CPP. Do not slice derived objects

OOP52-CPP. Do not delete a polymorphic object without a virtual destructor