# CWE Detail – CWE-543

## Description

The product uses the singleton pattern when creating a resource within a multithreaded environment.

## Extended Description

The use of a singleton pattern may not be thread-safe.

## Threat-Mapped Scoring

Score: 1.8

Priority: P4 - Informational (Low)

## Modes of Introduction

**•** Implementation: N/A

## Common Consequences

**•** Impact: Other, Modify Application Data — Notes:

## Potential Mitigations

**•** Architecture and Design: Use the Thread-Specific Storage Pattern. See References. (Effectiveness: N/A)

**•** Implementation: Do not use member fields to store information in the Servlet. In multithreading environments, storing user data in Servlet member fields introduces a data access race condition. (Effectiveness: N/A)

**•** Implementation: Avoid using the double-checked locking pattern in language versions that cannot guarantee thread safety. This pattern may be used to avoid the overhead of a synchronized call, but in certain versions of Java (for example), this has been shown to be unsafe because it still introduces a race condition (CWE-209). (Effectiveness: Limited)

## Applicable Platforms

**•** Java (Class: None, Prevalence: Undetermined)

**•** C++ (Class: None, Prevalence: Undetermined)

## Demonstrative Examples

**•** Consider the following course of events: