GRAMMATECH

Verifysoft TECHNOLOGY

PRODUCT DATA SHEET

# CODESONAR®
## Designed for zero-tolerance defect environments

CodeSonar analyzes source code and identifies serious programming errors that cause system crashes, memory corruption, leaks, data races, and security vulnerabilities.

### How We Are Different
GrammaTech makes source-code analysis for organizations that have zero-tolerance for defects and vulnerabilities. Our flagship product, CodeSonar, typically finds twice as many critical defects in software compared to other static-analysis tools.

### Pinpoints the Critical Problems
CodeSonar identifies problems that developers care about finding, like data races, deadlock, buffer overruns, leaks, null-pointer dereferences, and uninitialized variables.

### Improves Security
CodeSonar finds vulnerabilities and supports security-related standards like US-CERT's *Build Security In* and MITRE's *CWE*.

### Analyzes Millions of Lines of Code
CodeSonar can perform a whole-program analysis on 10M+ lines of code. Once an initial baseline analysis has been performed, CodeSonar's incremental analysis capability makes it fast to analyze daily changes to a codebase. The analysis can run in parallel to take advantage of multi-core environments.

### Works Out of the Box
No changes to the source code or existing build system are required.

### Shows Defect Trends
Graphs display data to help you manage development and testing efforts.

### Designed for High Assurance
CodeSonar is built for analyzing mission-critical applications, where reliability and security are paramount.

### Employs Sophisticated Algorithms
CodeSonar performs a unified dataflow and symbolic execution analysis that examines the computation of the entire program. The approach does not rely on pattern matching or similar approximations. CodeSonar's more general analysis naturally finds defects with new or unusual patterns.

### Supports Custom Checks
New checks can be created easily with the included C API.

### Shows Code-Level Metrics
CodeSonar is focused on finding critical defects, but it also provides code metrics. You can even define custom metrics.

### Provides Architecture Visualization
Smooth and scalable architecture visualization features make it easy to understand relationships between different elements in the code.

### Provides Workflow Automation
Automation features enable large teams to work together in a coordinated way. For example, it is easy to manage warnings across different project versions or development branches. A Python API supports customization and integration with other tools.

*"CodeSonar does a better job of finding the more serious problems, which are often buried deep in the code and sometimes hidden by unusual programming constructs that are hard for other static-analysis tools to parse."*

**– GE Aviation**

*"We were impressed by the depth of CodeSonar's analysis."*

**– Vivante**

*"The automated analysis provides a huge amount of leverage in a cost-effective way."*

**– Boston Scientific**

*"We tried the leading static-analysis tools. CodeSonar performed the deepest analysis and provided the most useful information."*

**– Adaptive Digital Systems**

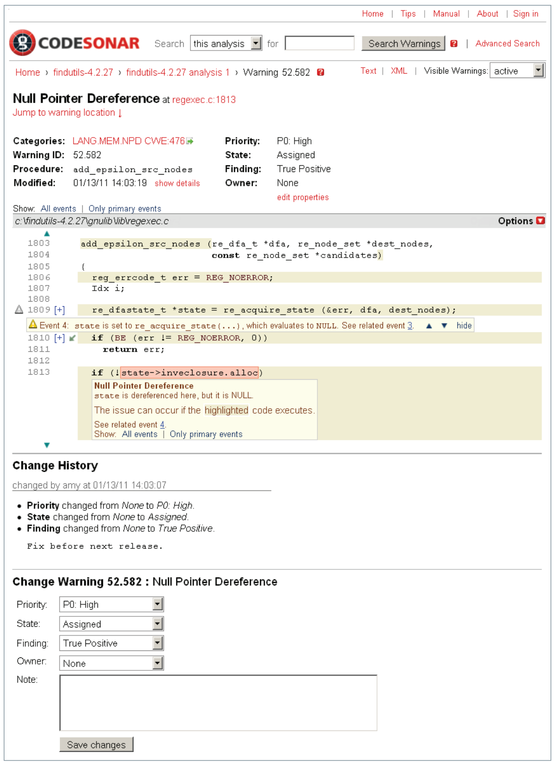
*"Especially good at inter-procedural analysis. It can be slow on large code bases, but is quite thorough and accurate. Highly recommended."*

**– Gerard Holzmann**
**SPIN Model Checker Creator**

*"In the last six years, we assessed and used several static-analysis tools. We assessed CodeSonar and we decided to purchase it because it gives valuable results easily and quickly."*

**– Électricité de France**

## **CodeSonar:** Code analysis for **zero-tolerance** defect environments


Home | Tips | Manual | About | Sign in
CODESONAR
Search this analysis for Search Warnings | Advanced Search
Home › findutils-4.2.27 › findutils-4.2.27 analysis 1 › Warning 52.582
Text | XML | Visible Warnings: active

**Null Pointer Dereference** at regexec.c:1813
Jump to warning location ↓

| | | | |
|---|---|---|---|
| **Categories:** | LANG.MEM.NPD CWE:476 | **Priority:** | P0: High |
| **Warning ID:** | 52.582 | **State:** | Assigned |
| **Procedure:** | add_epsilon_src_nodes | **Finding:** | True Positive |
| **Modified:** | 01/13/11 14:03:19 show details | **Owner:** | None |
| | | | edit properties |

Show: All events | Only primary events
c:\findutils-4.2.27\gnulib\lib\regexec.c Options

```
add_epsilon_src_nodes (re_dfa_t *dfa, re_node_set *dest_nodes,
                       const re_node_set *candidates)
{
  reg_errcode_t err = REG_NOERROR;
  Idx i;

  re_dfastate_t *state = re_acquire_state (&err, dfa, dest_nodes);
⚠ Event 4: state is set to re_acquire_state(...), which evaluates to NULL. See related event 3.  ▲ ▼ hide
  if (BE (err != REG_NOERROR, 0))
    return err;

  if (!state->inveclosure.alloc)
   Null Pointer Dereference
   state is dereferenced here, but it is NULL.
   The issue can occur if the highlighted code executes.
   See related event 4.
   Show: All events | Only primary events
```

**Change History**

changed by amy at 01/13/11 14:03:07

- **Priority** changed from *None* to *P0: High*.
- **State** changed from *None* to *Assigned*.
- **Finding** changed from *None* to *True Positive*.

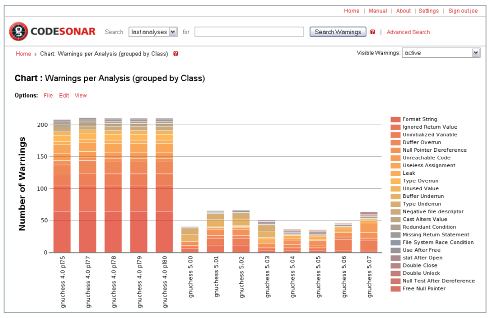
`Fix before next release.`

**Change Warning 52.582 : Null Pointer Dereference**

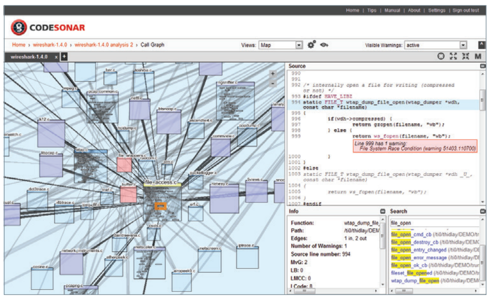
| | |
|---|---|
| Priority: | P0: High |
| State: | Assigned |
| Finding: | True Positive |
| Owner: | None |
| Note: | |

Save changes

*See the path to each flaw and how it can occur.*



*See quality trends by comparing analysis runs. Find out what types of defects are being introduced.*



*Understand your code with GrammaTech's award-winning software architecture visualization.*

## Some of the Checks

- Data Race
- Deadlock
- Buffer Overrun
- Null-Pointer Dereference
- Divide by Zero
- Uninitialized Variable
- Free Non-Heap Variable
- Use After Free
- Double Free/Close
- Format String Vulnerability
- Unreachable Code
- Resource Leak
- Return Pointer to Local
- Dangerous Function Cast
- *Misuse of Libraries*
- *Security Vulnerabilities*
- *User-Defined Checks*
- *Many More...*

## Technical Highlights

- Symbolic execution engine
- Scalable
- Incremental analysis capability
- Browser-based user interface
- Management reports
- Extensible analysis engine
- Integrates with other tools
- Easy setup no changes to build environment

## Free Trial

GrammaTech provides a cost-free means to evaluate CodeSonar on your own code so you can compare the results with those reported by other vendors. Request an evaluation copy at http://www.grammatech.com/free_trial

## About GrammaTech

Since its inception as a spin-off of Cornell University, GrammaTech has focused on providing static analysis for applications where reliability and security are paramount. Our staff includes fifteen PhD-level experts in static analysis and a superb engineering team, all focused on creating the most innovative and in-depth analysis algorithms. Our customers create software for avionics, medical, industrial control, and other mission-critical applications.

## System Requirements

### Supported languages

- C
- C++
- Java
  *See Java datasheet for more information.*

### Supported platforms

- Windows
- Linux
- Solaris

### Machine requirements

- 2 GHz CPU
- 2 GB of RAM*
- 15+ GB of free disk space

### Supported compilers

- Apple xcode
- ARM RealView
- CodeWarrior
- GCC
- G++
- Green Hills
- HI-TECH
- IAR
- Intel C/C++
- MS Visual Studio
- Renesas
- Sun C/C++
- Texas Instruments CodeComposer
- Wind River
- Most other compilers easily supported

### Output formats

- HTML
- XML
- Text (plain text and CSV)

**Requirements to run in serial mode. Parallel mode requires 512MB plus 512MB (and one core) per process.*

Verifysoft TECHNOLOGY

**Verifysoft Technology GmbH**
In der Spoeck 10-12
77656 Offenburg (Germany)
Phone: +49 781 127 8118-0
Web: www.verifysoft.com

GRAMMATECH

**GrammaTech, Inc.**
531 Esty Street
Ithaca, NY 14850
Phone: 607.273.7340
Web: www.grammatech.com