



Proactive Defect Detection

Coverity's Software Analysis Toolset (SWAT) provides a powerful combination of out of the box defect detection and an extensible architecture that enables the rapid deployment of reliable, secure software. Using SWAT's comprehensive analysis framework, you can extend the built-in capabilities of SWAT with a near limitless variety of company specific and project specific analyses of C code.

Our source code analysis technology was developed by a team of researchers in the Computer Science Department at Stanford University. The analysis engine as it stands today is based on powerful patent pending statistical inference technology, combined with a comprehensive abstract dataflow engine. This enabling technology allows SWAT to achieve critical feature milestones, such as:

- Scalability to millions of lines of code within a fraction of the build time.
- Precise error reporting pinpointing the line numbers and root cause of each detected defect.
- 100% path coverage using a powerful fixed point analysis, enabling defect detection in parts of the code that are difficult, if not impossible, to test.
- Powerful semantic analysis allowing SWAT to find dangerous defects without restrictions on coding style or conventions.

Our defect detection analyses focus on bugs that present substantial security and reliability risks to any enterprise class software project. SWAT's integrated user interface provides intuitive and precise defect diagnosis and defect tracking. Unlike any other source code analysis solution, SWAT is able to distinguish stylistic rules that may not apply to all code from critical defects.

Defect Detection out of the Box

SWAT includes a suite of analyses that detect numerous categories of critical defects. SWAT uses patent pending statistical inference technology to automatically customize each analysis to each client's source code enabling powerful defect detection without any of the burden of annotation, scripting, or manual configuration. The early adopter release of SWAT targets defects of the following types:

- **NULL pointer misuse**
 - Statistical API inference to check NULL pointer handling across interfaces.
 - Patent pending consistency analysis to ensure consistent treatment of NULL pointers.
 - Interprocedural dataflow analysis to track dangerous NULL pointers along each path through the code including those that follow function calls and function pointers.

- **Memory leaks**
 - Statistical inference to check system specific memory and resource allocation and deallocation.
 - Interprocedural dataflow analysis to follow allocated memory across function pointers and function calls.
- **Memory corruption**
 - Statistical inference to detect illegal uses of deallocated resources.
 - Interprocedural dataflow analysis to detect array/buffer overruns.
 - Detection of incorrect size parameters to allocation routines.
- **Generic API checking**
 - Statistical inference to detect function pairings.
 - Configurable templates for generic temporal ordering restrictions.
 - Statistical inference to detect method-call ordering restrictions.
- **Error handling**
 - Statistical inference to identify functions that can exit in failure modes that the calling code must detect.
 - Interprocedural analysis to ensure that if an error does occur, the error code is passed up the call chain to the error handling code.
- **Concurrency errors**
 - Statistical inference and checking of lock hierarchies to detect potential deadlocks.
 - Statistical inference of which items of shared data must be protected by which mutexes to detect race conditions.
 - Consistency checking to ensure that locking routines are always paired with appropriate unlocking routines.
 - Interprocedural dataflow analysis to identify potential deadlocks due to recursive acquisitions of the same lock.
- **Performance degradation errors**
 - Forgetting to re-enable interrupts.
 - Dangerous or slow operations with critical resources/locks held.
- **Redundant operations**
 - Unintentionally unused data.
 - Unintended no-op operations.
 - Interprocedural dataflow analysis to detect dead code.
- **Embedded devices/kernel code**
 - Interprocedural dataflow analysis to detect stack overflow on architectures with a limited stack size/kernel stack size.
 - 64-bit compliance for clients with multiple target platforms and to assist in upgrading legacy code.

- **Coding policies**
 - Incorrect fall-through in switch statements.
 - Code complexity measurements.

- **Security holes**
 - Illegal uses of tainted data
 - Buffer overruns
 - Other vulnerabilities leading to denial of service attacks and compromising of control/data.
 - Format string vulnerabilities.

Extensible Architecture

In addition to standard software errors, SWAT's extensible architecture allows Coverity's clients to optionally turn custom coding policies, rules, and standards that were previously written on paper into powerful analyses that can be applied effectively to each client's source code.

A suite of custom analyses is as essential to the development of reliable, secure code as a generic test suite. In particular, SWAT gives QA engineers the ability to translate defects detected with traditional testing techniques into custom source code analyses. To enable this translation, SWAT includes the Metal scripting language, compiler, and associated interfaces to enable fast development and deployment of custom analyses.

About Coverity

Coverity, Inc. is a leading provider of source code analysis solutions that help organizations produce reliable, secure software while significantly improving time to market. Coverity's quickly growing customer base includes a wide range of companies, from startups to Fortune 100 enterprises.

Coverity's patent pending technology was originally developed by a team of researchers in the Computer Systems Lab at Stanford University. Preliminary applications of the technology resulted in the successful detection of over 2000 defects and hundreds of exploitable security holes in the Linux and OpenBSD kernels.

Coverity, Inc. is headquartered in Menlo Park, California. For more information or a free trial, contact us at:

Tel: 1-650-980-3408

E-mail: info@coverity.com