### Codemind Configurations

### Codemind CSI/CQI Server System

- Supports Source Code Version Control System such as SVN and Git
- Supports Analysis on User-Uploaded Codes
- · Provides Web based Management System
- Management of Permission for Admin and Developers
- · Automatic/Manual Analysis Control

### **Codemind Developer**

- Stand-Alone Static Analysis System on Desk
- · Similar to Standard Development Tools
- · Comes with Defect Tracking Graphs



### **Software Verification Experts Company**

- Group of professionals specialized in programming language code analysis and verification Affiliated partner enterprise of Seoul National University Research on Software Analysis for Error-free Computing Center
- Partner enterprise of Korea University Security Software Research Center

### Patents and Certifications

- · Patented: How to detect information outflow from applications and equipment.
- · Patented: On-the-fly static analysis mode and device
- · Patent applied for: Program analysis method and device
- · Patent applied for: Defect tracking method and device
- Common Criteria Certification
- · Good Software Certification
- · Perfomance Certification
- · CWE Compatible



h

Detailed

Report

Developer with Codemind

Developer

Version Control

**Q** 

Survey

Report

 $\overline{\mathbf{r}}$ 

·\*\*\*\*\*\*\*\*\*





CODEMIND Into the Source 0



CODEMIND Corporation \$02-859-2633 @www.codemind.co.kr ♥ #807, Digital-ro 285, Guro-gu, Seoul 08381, Korea

# **THE SOURCE**

Innovative Static Analysis Technology from Software Analysis and Verification Experts

INTO

### **Codemind** Static Code Analysis Tool





66 The static analysis tool detects security vulnerabilities and runtime errors in software source code beforehand. This comes from the cutting-edge technology that traces all execution paths without running software. Now static analysis tools are essential to obtain security and safety of software.

In contrast to its predecessors, Codemind provides static analysis results using 'on-the-fly' mode. Existing tools use batch processing mode to acquire analysis results after analyzing the entire source code thoroughly. In on-the-fly mode you can review the results as soon as they come out even though not every part of the source code has been analyzed. Hence, you can find and fix defects without having to wait for several hours. 99

Codemind Technology for Next Generation	Latest Static Analysis Technology	<ul> <li>Up-to-date Semantic Analysis Framework</li> <li>Balance between Accuracy and Speed</li> <li>Jointed Analysis of Value and Memory</li> <li>Analysis based on Symbolic Execution</li> <li>Faster Analysis using Machine Learning Technique</li> <li>High Speed Interprocedural Analysis</li> </ul>
	Query based Static Analysis	<ul> <li>Graph DB based Analysis Framework</li> <li>Query based Source Code Analysis</li> <li>Query based Metric Analysis</li> <li>Query based Infrastructure Analysis</li> <li>Query based Code Conventions Inspection</li> </ul>
	On-the-Fly Static Analysis	<ul> <li>On-the-Fly Analysis Framework</li> <li>On-the-Fly Defect Tracking System</li> <li>Code and Graph Co-Browsing</li> <li>Interoperation of Analyzing and Defect Tracking</li> <li>Fast Discovery of the Causes of Defects</li> </ul>
Administration Features		<ul> <li>Web based Analysis Management System</li> <li>Dashboard for Statistics Monitoring</li> <li>Survey Report for Administrator</li> <li>Detailed Report for Developer</li> <li>Ruleset Settings</li> <li>Interoperating with Source Code Version Control System</li> <li>Permission Control and History Management</li> <li>Various File Formats for Reports</li> </ul>
1448 xF 1448 xF 144		Supported· Linux, Windows, OS XOSs,· Java, JSP, Javascript, ASP.NET, PHP,Languages,HTML, C/C++, C#, Android(Java),FrameworksiOS (Objective C) atc

## Codemind<sup>™</sup> CO

**Code Quality Inspector** 

### **Analysis Features**

- Runtime Errors Detection
- · Flawed Code Detection
- Latest Abstract Interpretation Framework
- · Optimized Value Analysis Engine
- Memory Analysis Engine with Maximum Performance
- High Speed Interprocedural Analysis
- Faster Analysis using Machine Learning Technique
- Slide Bar to Adjust between Accuracy and Speed

### Secure Coding Standards

- · OWASP Top 10 Security Vulnerabilities
- · CWE/SANS Top 25 Security Vulnerabilities
- Power of 10 Rules
- · CERT Secure Coding Standard
- · LDAP Injection
  - · Cross-site Request Forgery

  - Integer Overflow



### Main Rules of Code Quality Inspection

- Buffer Overrun/Underrun
- $\cdot$  Use after Free
- Unused Value
- · Unreachable Code
- Incorrect Numeric Casting
- Uninitialized Variable
- Type Overrun/Underrun
- $\cdot$  Null Pointer Dereference
- · Memory Leak
- · Double Free
- $\cdot$  Divide by Zero
- Return Pointer to Local
- · Null Check after Dereference
- Mismatched Memory Management

 $(\alpha\circ f)(c)\sqsubseteq_A(f^{\#}\circ\alpha)(c), \text{ for all } c\in$ 

 $(f \circ \gamma)(a) \equiv_C (\gamma \circ f^{\#})(a)$ , for all  $a \in A$ 

(f(c))

iff



### **Code Security Inspector**

### Main Rules of Secure Coding Inspection

- $\cdot$  SQL Injection
- · Resource Injection
- · Cross-site Script
- · OS Command Injection
- · Relative/Absolute Path Traversal

- Weak Encryption
- Hard-coded Password
- Null Pointer Dereference
- · Data Leak between Sessions
- · Information Leak of System Data
- · Improper Resource Shutdown/Release
- · API abuse