Teste e Verificação de Software

Carlos Pereira
João Pereira
Pedro Couto

Static Analysis

Analysis of software development artifacts, e.g., requirements or code, carried out without execution of these software development artifacts. Static analysis is usually carried out by means of a supporting tool. [ISTQB]

”

Static Testing

- Testing done in a non-runtime environment
- **Review**
  - Errors in user requirements, architecture, design or use cases
- **Static Analysis**
  - The code is structurally tested without being run
  - Generally done with a support tool

Why is Static Analysis important?

- According to ESI International, approximately 50% of project failures result from poor definition of the user requirements
- Goes hand in hand with proper agile methodology implementation
- Helps setting the project for success from the very first stages
- Documentation analysis, otherwise neglected

Why is Static Analysis important?

- Whitebox visibility
- Cost-efficient
- Structural analysis
- Generate internal tests
- Early error detection
- Automated scan on all code
- Exact location error detection

Goals of Static Analysis

- Detect errors that go silent through dynamic analysis
- Code refactoring suggestions
- Enforce attention to artifacts often neglected by developers

Dynamic vs Static Analysis

- Designed to test the code during runtime
- Only verifies the part that is executed
- Misleading security sense
- Exact error locations are hard to find
- Time consuming in certain situations

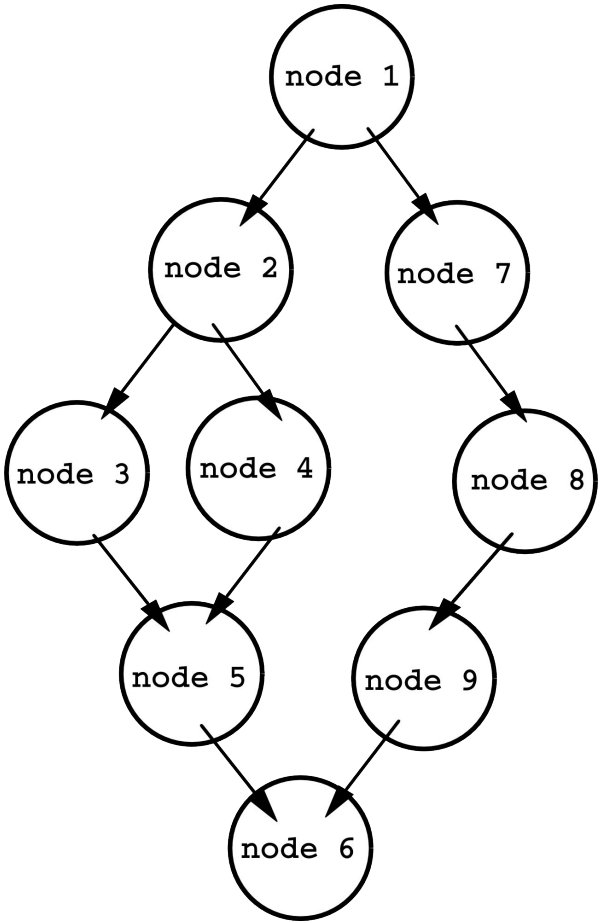
Techniques to analyze static source code

Data Flow Analysis

- Collects runtime information about the data in a program while in its static state
- Code divided into blocks
- Determine input and output for the blocks in order to retrieve data flow

Control Flow Graph

- Used in data flow analysis.
- Representation of the blocks (nodes representing instructions) and data flow (edges representing transitions/paths) in a computer program.



Taint Analysis/Checking

- Checks variables potentially influenced by outside input.
- Creates variable list
- Determines if there's a risk in using those variables (e.g. SQL Injection)
- Feature in some languages

Lexical Analysis

- Transforms code syntax into tokens of information.
- Abstract source code to make it easier to manipulate.

```
<?php $name = "Ryan"; ?>
```

```
T_OPEN_TAG
T_VARIABLE
=
T_CONSTANT_ENCAPSED_STRING
;
T_CLOSE_TAG
```

Strengths

- Early error detection
- Faster non-runtime error detection
- Handles time consuming tasks for developers
- Pinpoints errors against the entire codebase
  - Does not depend on the ran parts of the code

Strengths

- Scales well
- For things that such tools can automatically find with high confidence, such as buffer overflows or SQL Injection Flaws they are great.

Weaknesses

- Static testing done without support tools is very time consuming
- Does not detect runtime errors
- Not enough tools for the variety of programming languages used

Weaknesses

- Limited range in security vulnerabilities
- High number of false positives
- Configuration issues dismissed
- Hard to prove that issues represent vulnerabilities
- Difficulty analysis of uncompilable code

Features of Static Analysis Tools

Coding Standards

- Set of programming rules, naming conventions and layout specifications

Why?

Code Metrics

- Measurement of depth of nesting, cyclomatic number and number of lines of code

Why?

Code Structure

- **Control flow structure** - sequence in which the instructions are executed
- **Data flow structure** - follows the track of the data item as it is accessed and modified by the code
- **Data structure** - refers to the organization of the data itself, independent of the program

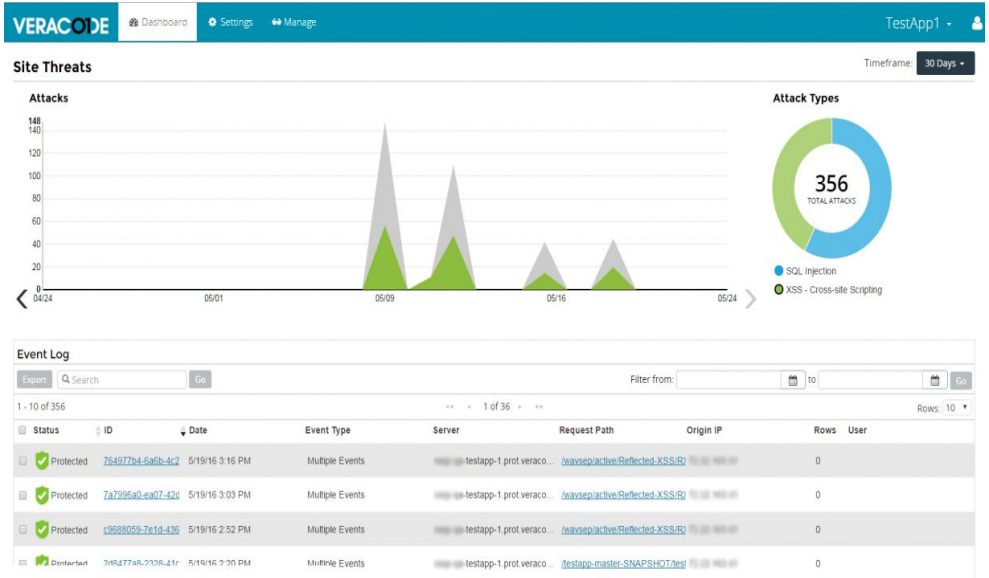
Tools

Criteria when choosing a static analysis tool

- Coverage
- Completeness
- Accuracy of results
- Customizability
- Sustainability / repeatability
- Deployment model
- Usability
- Reporting
- Security
- Cost
- Integration
- Scan performance

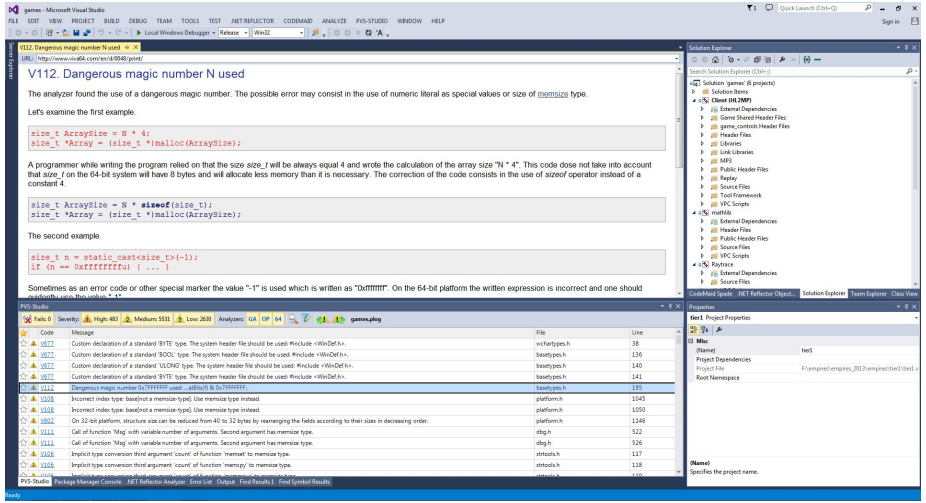
Veracode

- SaaS based
- Static security analysis
- Binary code testing



PVS-Studio

- C, C++ and C#
- Integration with Visual Studio
- Allows for automated static testing
- Windows and Linux



Parasoft

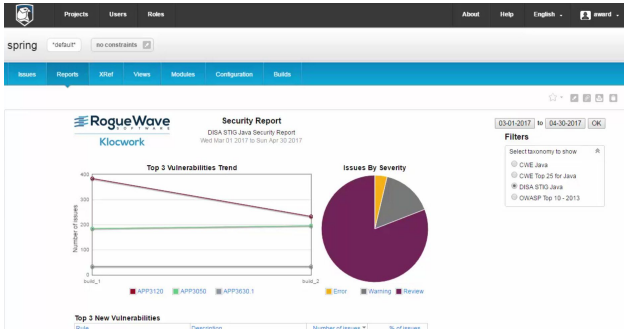
- Supports several static testing methods
- Feature to prevent future static errors

SnappyTrick

- Security oriented
- Faster code scanning
- Platform independent
  - Eliminates OS compatibility issues

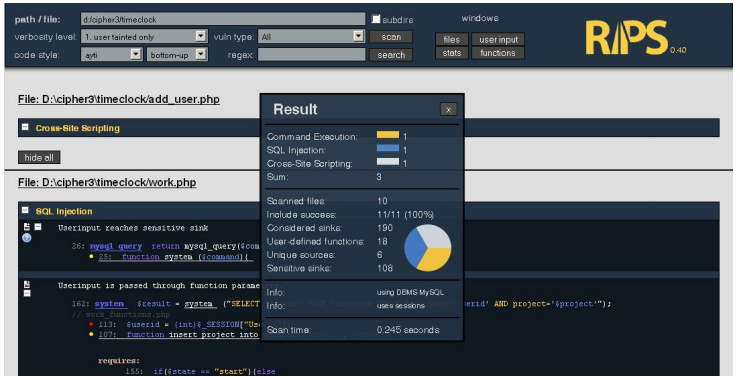
Klocwork

- Finds code vulnerabilities besides semantic related issues
- Runs along the code creation
  - Provides immediate addressing to detected errors



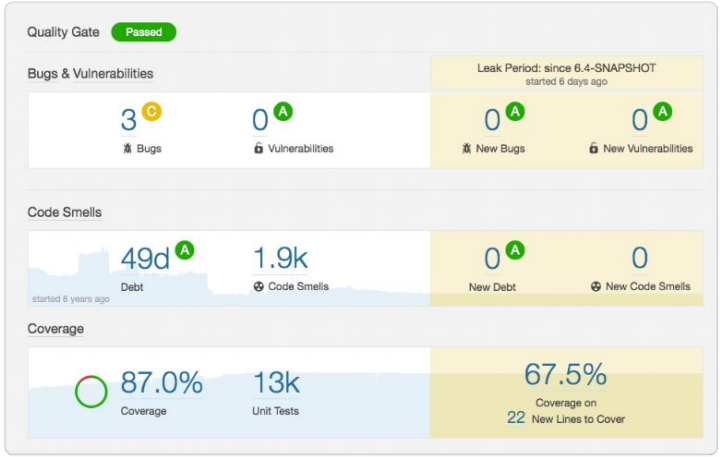
RIPS

- PHP oriented tool
- Support for the most popular PHP frameworks
- Detects complex PHP defects



SonarQube

- Web based platform
- Supports over 20 languages
- Open source
- Provides several quality reports
- Automated



Video

https://github.com/JoaoPere/TVVS-StaticAnalysis