Visual analysis of code security

John R. Goodall
Oak Ridge National Lab • goodalljr@ornl.gov • 865 576 5943

Hassan Radwan
Applied Visions, Inc. • hassanr@avi.com • 518 207 3106

Lenny Halseth
Applied Visions, Inc. • lennyh@avi.com • 518 207 3108
This effort was performed at Applied Visions, Inc. Secure Decisions division for DHS Science & Technology
The Problem
More than **98% of all PCs** have one or more vulnerable programs

Software Assurance: poorly written software is at the root of all of our security problems

Doug Maughan, CACM 53(2)
Top 10 Hard Problems in Cyber Security
More than 98% of all PCs have one or more vulnerable programs.

Lots of Bad Code

Software Assurance: poorly written software is at the root of all of our security problems.

Doug Maughan, CACM 53(2)
Top 10 Hard Problems in Cyber Security

http://secunia.com/blog/56/
...everybody should be using static analysis tools today. And if you are not using them, then basically you are negligent, and you should prepare to be sued by the army of lawyers that have already hit the beach.

Cigital's CTO Gary McGraw

enterprises must adopt SAST [Static Application Security Testing]

Gartner
...everybody should be using static analysis tools today. And if you are not using them, then basically you are negligent, and you should prepare to be sued by the army of lawyers that have already hit the beach.

Tools Exist Today

enterprises must adopt SAST
[Static Application Security Testing]

Gartner
No tool stands out as an uber-tool. Each has its strengths and weaknesses.

84% of the vulnerabilities were identified by one tool and one tool alone.

Kris Britton, Technical Director
NSA’s Center for Assured Software
No tool stands out as an uber-tool. Each has its strengths and weaknesses.

Kris Britton, Technical Director
NSA’s Center for Assured Software

84% of the vulnerabilities were identified by one tool and one tool alone.
Tools find different vulnerabilities

Non-overlap: Hits reported by one tool and no others (84%)

Overlap: Hits reported by more than one tool (16%)

- 2 tools
- 3 tools
- 4 tools
- All 5 tools

from MITRE
Tools find different vulnerabilities

Non-overlap: Hits reported by one tool and no others (84%)

Overlap: Hits reported by more than one tool (16%)

Very Little Overlap

from MITRE
... with different semantics
working with different tool vendors is a confusing and challenging and time-consuming process: the engines work differently, which is good since they catch different types of problems...

Jim Bird, Building Real Software
… with different semantics

working with different tool vendors is a confusing and challenging process: the engines work differently, which is good since they catch different types of problems…

Jim Bird, Building Real Software
Tools present **vulnerabilities**

Developers think in **code**

*(namespace/class/method)*
Tools present **vulnerabilities**

Developers think in **code**

*(namespace/class/method)*
50,000 vulnerabilities ... Now what?
No Big Picture

50,000 vulnerabilities ...

Now what?
Better Tools ≠ Total Security

- Tool results have very little overlap
- Tools use different semantics for results
- Tools present a vulnerability-centric view
- Tools offer no big picture overviews
Better Tools ≠ Total Security

- Tool results have very little overlap
- Tools use different semantics for results
- Tools present a vulnerability-centric view
- Tools offer no big picture overviews

Better analysis tools are only a part of improving code security
Technical Approach
Technical Approach

Software Assurance Visual Analysis

Provide a workflow for developers to
bring together disparate security analysis results
visually analyze and prioritize those results
explore those results to uncover hidden trends
use code context to assess the impact of those results
see who is responsible for vulnerabilities
assign vulnerabilities to developers responsible
data

integrate / correlate / normalize

context

visual analysis

workflow

SDLC Tools
Test Data

- Three software analysis tools
- Two test data sets
SwA Tool Output

- Tool output is in varying XML schemas
- Results are parsed and correlated
- Severity and category are normalized
Use case: Triage

- Which vulnerabilities are noise / most important
- What vulnerability categories are most common
- What vulnerabilities are found by multiple tools
- Where in the code are the vulnerabilities
- Who do confirmed vulnerabilities get assigned to
Visualization

- Each source code file is represented as a block
- Each block aggregates the vulnerabilities found
- Very compact, space filling method
- Flexible (color/sort) data > visual mappings
Demonstration
Overview first, zoom & filter, details on demand…

– Ben Schneiderman
StandardWrapperValve.java

Severity Score: 8.25
Vulnerabilities: 2
Minimum Severity: 6
Maximum Severity: 10

Vulnerabilities | Severity Score | Distribution | Objects
--- | --- | --- | ---
2 | 8.25 | | - StandardWrapperValve
2 | 8.25 | | - public invoke(Request request, Response response, ValveContext val
1 | 10.00 | | - Line 192 - Errors::NPE_COND (Klocwork)
1 | 6.00 | | - Line 192 - Errors::Null Dereference (Fortify)
Created bug #15 and assigned it to remm
Before

After
Starting point 33,895 vulnerabilities

2 clicks later ... 227 vulnerabilities
Benefits

• Increased vulnerability coverage through the integration of multiple tools
• Overview of large number of vulnerabilities
• Visual prioritization of vulnerabilities
• Traceability of developer responsibility
• Remediation via integration with SDLC
Benefits

• Increased vulnerability coverage through the integration of multiple tools
• Overview of large number of vulnerabilities
• Visual prioritization of vulnerabilities
• Traceability of developer responsibility
• Remediation via integration with SDLC

Enhances the coverage & speed for detection & remediation of vulnerabilities
Traceability
Remediation
Visual analysis of code security

Questions

John R. Goodall
Oak Ridge National Laboratory • goodalljr@ornl.gov • 865 576 5943

Hassan Radwan
Applied Visions, Inc. • hassanr@avi.com • 518 207 3106

Lenny Halseth
Applied Visions, Inc. • lennyh@avi.com • 518 207 3108