Coverity Results and Experiences for SATE V

Peter Henriksen, Sr Director of Analysis @ Coverity
March 14, 2014
Agenda

• Coverity Overview
• Static Analysis Future
• Feedback for Juliet
• Coverity Analysis Results
• Triage Criteria
• Feedback on Coverage Claim Representation
• Feedback for SATE
• SATE: Something Wrong?
• Thanks!
Coverity Overview

• Has passed its 10 year anniversary
  • In the process of being acquired by Synopsys, forming its own Development Testing business unit

• Focused on Static & Dynamic Analysis
  • Quality & Security Defects in C/C++, Java & C#
  • Test Coverage Policy

• 1200+ customers
  • Many large company-wide engagements (Cisco, Samsung, SAP etc)

• 1000+ Open Source packages scanned regularly
  • Improving the software supply chain
Static Analysis Future

• Ensure successfully deployed solutions
  • Post-sales / Support / Responsive R&D
• Moving earlier and earlier in the development process
  • From
    • Auditing to
    • QA to
    • Development
  • From
    • Once Per Release Cycle to
    • Multiple Times to
    • Nightly Build to
    • Developer Desktop (without any loss of precision)
• Never ending list of new languages and frameworks
• More precise and more evidence based
• Help solving a broader array of the software quality problems
Feedback for Juliet Test Suite(s)

- Best attempt at a Static Analysis synthetic test suite
- Maintainers listen
- Does not build out-of-the-box (on any platform?)
- Many tests are too simple/synthetic
  - Example: Unsalted Hash
    - String literal
    - Salt from random number generator
    - String not stored
    - Salt not stored
    - Salt’ing is no longer the recommended solution (should use HMAC)
- Tool to calculate metrics from the standard static analysis report format
  - Include tool with Juliet test suite?
Coverity Analysis Results

• Overall FP rate (estimate from Coverity triage of 120 defects): ~15%

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>Language</th>
<th>Defects Reports</th>
</tr>
</thead>
<tbody>
<tr>
<td>Asterisk 10.2.0 &amp; 10.12.2</td>
<td>C</td>
<td>1864 &amp; 1190</td>
</tr>
<tr>
<td>Wireshark 1.8.0 &amp; 1.8.7</td>
<td>C</td>
<td>569 &amp; 551</td>
</tr>
<tr>
<td>Juliet</td>
<td>C</td>
<td></td>
</tr>
<tr>
<td>JSPWiki 2.5.124 &amp; 2.5.139</td>
<td>Java</td>
<td>147 &amp; 147</td>
</tr>
<tr>
<td>Openfire 3.6.0 &amp; 3.6.4</td>
<td>Java</td>
<td>254 &amp; 251</td>
</tr>
<tr>
<td>Juliet</td>
<td>Java</td>
<td></td>
</tr>
</tbody>
</table>
Triage Criteria: Quality Defects

• FORWARD_NULL: Null dereference
  • SATE: ... the only place the function is called ...

• REVERSE_INULL: Null check after dereference
  • SATE: ... transmit_response() is only called from ...
  • SATE: Searched the codebase and concluded that the two functions will not be called in the way which will trigger the defect => False Positive

• Coverity: The function has potential problem and is worth fixing => (at least) insignificant
Coverage Claim Representation (CCR)
High Level Feedback

• Poorly designed/ill defined
• Delayed & limited feedback to questions
• Purpose? How/where will it be used?
  • Should clearly have been marked as optional
• Coverity attempted generating the data
  • Had to build a whole application for the purpose
  • Significant effort
  • Yet, could not finish because of outstanding questions
Coverage Claim Representation (CCR)
Detailed Feedback

- Match_Accuracy: Definition needs to allow results to be computable
  - Exact
  - CWE-more-abstract
  - CWE-more-specific
  - CWE-partial
- Question A: For most class/base CWE, the description is very high-level followed by several examples
  - Do we have match ‘Exact’ if all examples are covered?
- Questions B: For a CWE with children CWEs where we match the children, do we
  - Still claim the parent CWE?
  - If so, what accuracy do we use?
- Question C: When matching a CWE that has children CWE’s, do we
  - Also claim matching on the children CWE’s?
Positive SATE Feedback

• Live Virtual Machine: good!
• Benchmarks available inside VM: good!
• README for each benchmark: good!
• Multiple benchmarks for each language: good!
• Multiple versions of each benchmark: good!
• List of known defects: good... (but where are they?)
Areas for Improvement of SATE

• Standard static analysis report format
  • Great, keep at it!
  • But, it means that it is better to
    • Freeze the format
    • Evolve in big & rare steps
    • ... rather than constantly changing/improving

• Provide known defects up front
  • Allowing vendors to present their True Positive result rates
The SATE Format: Something Wrong?

• Where are all the Static Analysis vendors?
  • Need to ask them why they are not participating?

• Some ideas: Additional ‘Soft’ benefits of Final Report
  • Use more effort on the report
  • Publicize it widely (at least within federal agencies)
  • List one or more positive findings for each vendor
  • List and thank all participating vendors for helping to advance state-of-the-art Static Analysis
Thanks to the SATE V Team

- Lots of hard work went into
  - Organizing/running the benchmark VM’s
  - Triaging the results
- Coverity appreciates
  - The ease of building/scanning the benchmarks
  - The careful triage performed
- SATE listens
  - Changes in 2013 for how results are shared: very much appreciated!
Questions?