Software Assurance: Crippling Coming Cyberassaults

Dr. Paul E. Black
National Institute of Standards and Technology
http://samate.nist.gov/
paul.black@nist.gov
**Report Documentation Page**

Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

<table>
<thead>
<tr>
<th>1. REPORT DATE</th>
<th>APR 2010</th>
</tr>
</thead>
<tbody>
<tr>
<td>2. REPORT TYPE</td>
<td></td>
</tr>
<tr>
<td>3. DATES COVERED</td>
<td>00-00-2010 to 00-00-2010</td>
</tr>
<tr>
<td>4. TITLE AND SUBTITLE</td>
<td>Software Assurance: Crippling Coming Cyberassaults</td>
</tr>
<tr>
<td>5a. CONTRACT NUMBER</td>
<td></td>
</tr>
<tr>
<td>5b. GRANT NUMBER</td>
<td></td>
</tr>
<tr>
<td>5c. PROGRAM ELEMENT NUMBER</td>
<td></td>
</tr>
<tr>
<td>5d. PROJECT NUMBER</td>
<td></td>
</tr>
<tr>
<td>5e. TASK NUMBER</td>
<td></td>
</tr>
<tr>
<td>5f. WORK UNIT NUMBER</td>
<td></td>
</tr>
<tr>
<td>6. AUTHOR(S)</td>
<td></td>
</tr>
<tr>
<td>7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)</td>
<td>National Institute of Standards and Technology, 100 Bureau Dr # 1000, Gaithersburg, MD, 20899</td>
</tr>
<tr>
<td>8. PERFORMING ORGANIZATION REPORT NUMBER</td>
<td></td>
</tr>
<tr>
<td>9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)</td>
<td></td>
</tr>
<tr>
<td>10. SPONSOR/MONITOR’S ACRONYM(S)</td>
<td></td>
</tr>
<tr>
<td>11. SPONSOR/MONITOR’S REPORT NUMBER(S)</td>
<td></td>
</tr>
<tr>
<td>12. DISTRIBUTION/AVAILABILITY STATEMENT</td>
<td>Approved for public release; distribution unlimited</td>
</tr>
<tr>
<td>13. SUPPLEMENTARY NOTES</td>
<td>Presented at the 22nd Systems and Software Technology Conference (SSTC), 26-29 April 2010, Salt Lake City, UT.</td>
</tr>
<tr>
<td>14. ABSTRACT</td>
<td></td>
</tr>
<tr>
<td>15. SUBJECT TERMS</td>
<td></td>
</tr>
<tr>
<td>16. SECURITY CLASSIFICATION OF:</td>
<td></td>
</tr>
<tr>
<td>a. REPORT</td>
<td>unclassified</td>
</tr>
<tr>
<td>b. ABSTRACT</td>
<td>unclassified</td>
</tr>
<tr>
<td>c. THIS PAGE</td>
<td>unclassified</td>
</tr>
<tr>
<td>17. LIMITATION OF ABSTRACT</td>
<td>Same as Report (SAR)</td>
</tr>
<tr>
<td>18. NUMBER OF PAGES</td>
<td>29</td>
</tr>
<tr>
<td>19a. NAME OF RESPONSIBLE PERSON</td>
<td></td>
</tr>
</tbody>
</table>

Standard Form 298 (Rev. 8-98)  
Prescribed by ANSI Std Z39-18
What is NIST?

- U.S. National Institute of Standards and Technology
- A non-regulatory agency in Dept. of Commerce
- 3,000 employees + adjuncts
- Gaithersburg, Maryland and Boulder, Colorado
- Primarily research, not funding
- Over 100 years in standards and measurements: from dental ceramics to microspheres, from quantum computers to fire codes, from body armor to DNA forensics, from biometrics to text retrieval.
The NIST SAMATE Project

- Software Assurance Metrics And Tool Evaluation (SAMATE) project is sponsored in part by DHS
- Current areas of concentration
  - Web application scanners
  - Source code security analyzers
  - Static Analyzer Tool Exposition (SATE)
  - Software Reference Dataset
  - Software labels
  - Malware research protocols

- Web site http://samate.nist.gov/
Software Reference Dataset

- Public repository for software test cases
- Almost 1800 cases in C, C++, Java, and Python
- Search and compose custom Test Suites
- Contributions from Fortify, Defence R&D Canada, Klocwork, MIT Lincoln Laboratory, Praxis, Secure Software, etc.

26 April 2010

Paul E. Black
Software Facts Label

- **Software Facts should:**
  - Voluntary
  - Absolutely simple to produce
  - Have a standard format for other claims

- **What could be easily supplied?**
  - Source available? Yes/No/Escrowed
  - Default installation is secure?
  - Accessed: network, disk, ...
  - What configuration files? (registry, ...)
  - Certificates (eg, "No Severe weaknesses found by CodeChecker ver. 3.2")

- **Cautions**
  - A label can give false confidence.
  - A label shut out better software.
  - Labeling diverts effort from real improvements.

26 April 2010

Paul E. Black
Many people research malware, but there are no widely accepted protocols.

Biological research has defined levels with associated practices, safety equipment, and facilities.

Some approaches are

- Weakened programs (auxotrophs)
- Programs that ALERT
- Outgoing firewalls
- Isolated networks
Assurance that software is less vulnerable to coming cyberassaults

Static and dynamic analysis

Static Analysis Tool Exposition - 2009 outcomes and 2010 progress
Assurance from three sources

\[ A = f(p, s, e) \]

where \( A \) is functional assurance, \( p \) is process quality, \( s \) is assessed quality of software, and \( e \) is execution resilience.
$p$ is process quality

- High assurance software must be developed with care, for instance:
  - Validated requirements
  - Good system architecture
  - Security designed- and built in
  - Trained programmers
is assessed quality of software

Two general kinds of software assessment:

- Static analysis
  - e.g. code reviews and scanner tools
  - examines code

- Testing (dynamic analysis)
  - e.g. penetration testing, fuzzing, and red teams
  - runs code
e is execution resilience

- The execution platform can add assurance that the system will function as intended.
- Some techniques are:
  - Randomize memory allocation
  - Execute in a “sandbox” or virtual machine
  - Monitor execution and react to intrusions
  - Replicate processes and vote on output
Software analysis is vital

- Benefits are:
  - Provide feedback to development process
  - Build product assurance when process is less visible
    - contractors
    - open source
    - legacy software
  - Confirm minimum quality for execution
Analysis is like a seatbelt ...
• Assurance that software is less vulnerable to coming cyberassaults

• Static and dynamic analysis

• Static Analysis Tool Exposition - 2009 outcomes and 2010 progress
Comparing Static Analysis with Dynamic Analysis

**Static Analysis**
- Code review
- Binary, byte, or source code scanners
- Model checkers & property proofs
- Assurance case

**Dynamic Analysis**
- Execute code
- Simulate design
- Fuzzing, coverage, MC/DC, use cases
- Penetration testing
- Field tests
Strengths of Static Analysis

- Applies to many artifacts, not just code
- Independent of platform
- In theory, examines all possible executions, paths, states, etc.
- Can focus on a single specific property
Strengths of Dynamic Analysis

- No need for code
- Conceptually easier - “if you can run the system, you can run the test”.
- No (less) need to build or validate models or make assumptions.
- Checks installation and operation, along with end-to-end or whole-system.
Static and Dynamic Analysis
Complement Each Other

Static Analysis
- Handles unfinished code
- Can find backdoors, eg, full access for user name “JoshuaCaleb”
- Potentially complete

Dynamic Analysis
- Code not needed, eg, embedded systems
- Has few(er) assumptions
- Covers end-to-end or system tests
- Assess as-installed
• Assurance that software is less vulnerable to coming cyberassaults
• Static and dynamic analysis

• Static Analysis Tool Exposition - 2009 outcomes and 2010 progress
Static Analysis Tool Exposition (SATE) Overview

- Goal: advance research in, and improvement of, static analysis tools for security-relevant defects and speed tool adoption by demonstrating use on real software.

- Checkpoints
  - Participants run tools on Java and C programs we choose
  - NIST-led researchers analyze reports
  - Everyone shares results and observations at a workshop
  - Later release final report and all data

- http://samate.nist.gov/SATE.html

- Co-funded by NIST and DHS/NCSD
SATE Participants

- **2008:**
  - Aspect Security ASC
  - Checkmarx CxSuite
  - Flawfinder
  - Fortify SCA
  - Grammatech CodeSonar
  - HP DevInspect
  - SofCheck Inspector for Java
  - UMD FindBugs
  - Veracode SecurityReview

- **2009:**
  - Armorize CodeSecure
  - Checkmarx CxSuite
  - Coverity Prevent
  - Grammatech CodeSonar
  - Klocwork Insight
  - LDRA Testbed
  - SofCheck Inspector for Java
  - Veracode SecurityReview

26 April 2010
“Number of bugs” is undefined
Tangled Flow: 2 sources, 2 sinks, 4 paths

1503 free

808 use

819 use

2644 free
Summary of 2009 tool reports

- Reports from 18 tool runs
- About 20,000 total warnings
  - but tools prioritize by severity, likelihood
- Reviewed 521 warnings - 370 were not false
- Number of warnings varies a lot by tool and case
- 83 CWE ids/221 weakness names

26 April 2010
Tools don’t report same warnings

Overlap in Not-False Warnings

- 1 tool: 207
- 2 tools: 120
- 3 tools: 40
- 4 tools: 3
Some types have more overlap

Overlap in Not-False Buffer Errors

- 1 tool
- 2 tools
- 3 tools
- 4 tools
Why don’t tools find same things?

- Tools look for different weakness classes
- Tools are optimized differently
Tools find things that people find

IRSSI (3)

- Same or other: 1
- Coincidental: 1
- None: 1

Roller (10)

- Same or other: 4
- Coincidental: 1
- None: 5

Includes two access control issues – very hard for tools

26 April 2010
SATE 2010 tentative timeline

- Hold organizing workshop (12 Mar 2010)
- Recruit planning committee.
  - Revise protocol.
  - Choose test sets. Provide them to participants (17 May)
- Participants run their tools. Return reports (25 June)
- Analyze tool reports (27 Aug)
- Share results at workshop (October)
- Publish data (after Jan 2011)
Acronyms

- CWE - Common Weakness Enumeration
- DHS/NCSD - Department of Homeland Security/National Cyber Security Division
- MC/DC - Modified Condition/Decision Coverage
- SAMATE - Software Assurance Metrics And Tool Evaluation (project at NIST)
- SATE - Static Analysis Tool Exposition (annual event)
- NIST - National Institute of Standards and Technology