SATE V Background

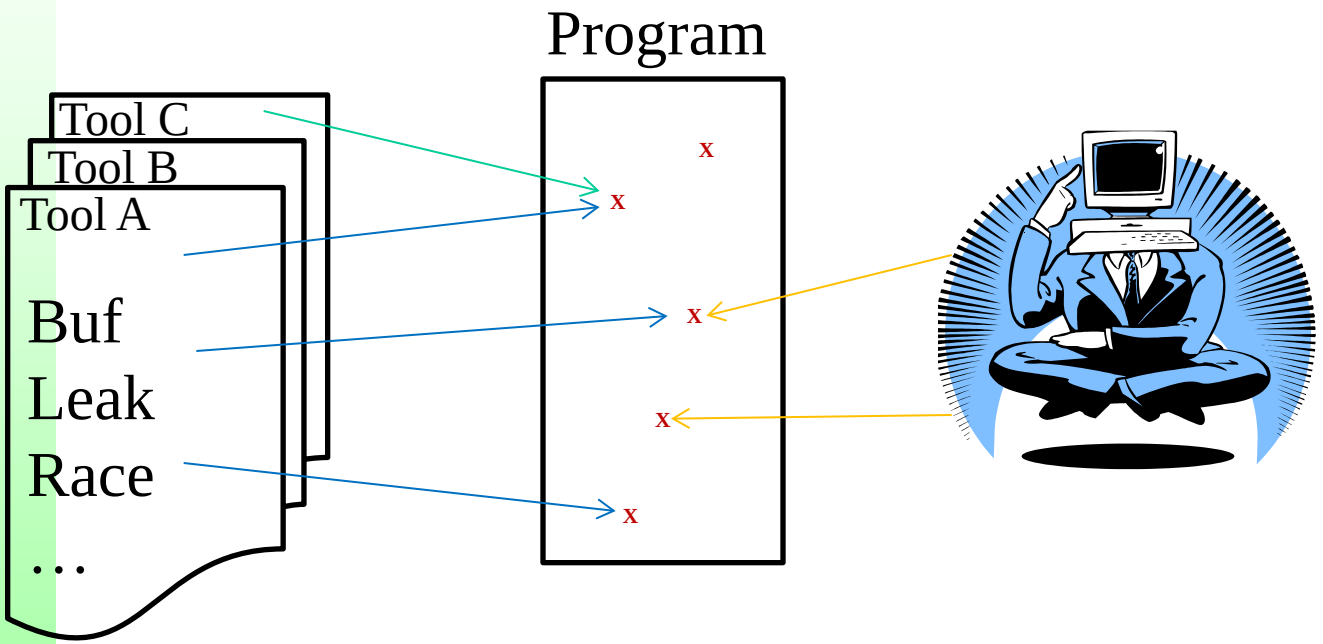
Vadim Okun, NIST

March 14, 2014

Cautions on Using SATE Data

- Our analysis procedure has limitations
- In practice, users write special rules, suppress false positives, and write code in certain ways to minimize tool warnings
- There are many other factors that we did not consider: user interface, integration, etc.
- So do NOT use our analysis to rate/choose tools

Analyzing Source Code Analyzers



Security *Quality* *Insignificant* *False* *Unknown*

Warning Selection Methods

1. Random subset
2. Related to CVEs
3. Synthetic test cases

SATE V Timeline

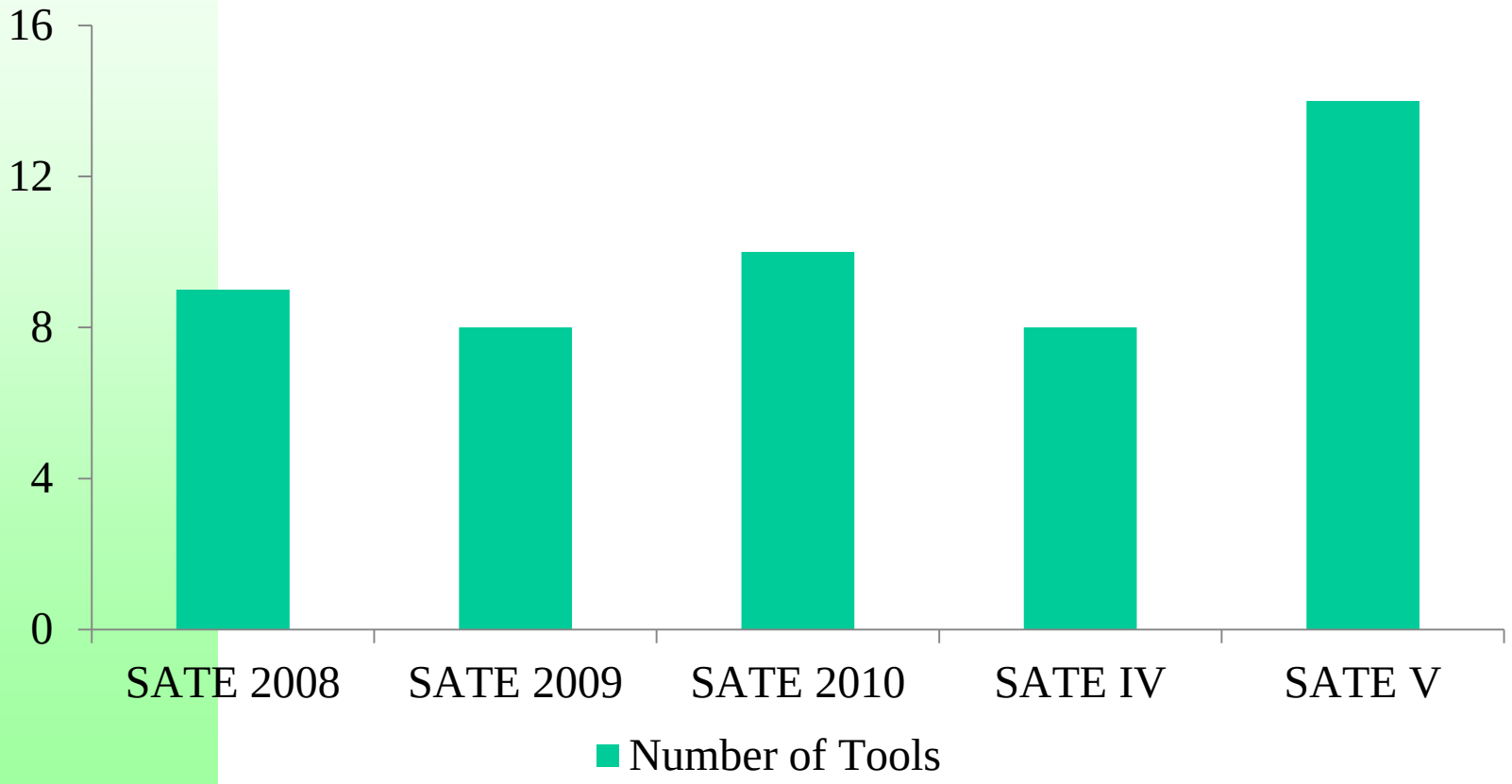
- Provide test sets to teams (June 2013)
- Teams run their tools, return tool outputs (Sep 2013)
- Analysis of tool outputs (mostly done)
- Experience workshop (here & now)
- Teams can submit a research paper (May 30, 2014)
- Publish SATE V report (Summer/Fall 2014)

Participating teams

- Buguroo
- Coverity
- Cppcheck
- Frama-C
- GrammaTech
- HP Fortify
- LDRA Testbed
- Parasoft
- PVS-Studio
- Red Lizard Goanna
- Clang
- FindBugs
- PMD

Certain instruments, software, materials, and organizations are identified in this paper to specify the exposition adequately. Such identification is not intended to imply recommendation or endorsement by NIST, nor is it intended to imply that the instruments, software, or materials are necessarily the best available for the purpose.

Participation over time



Test cases

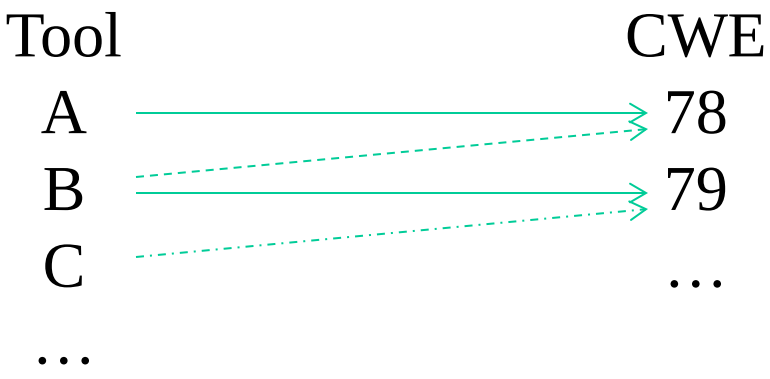
- CVE-selected vulnerable/fixed pairs:
  - Asterisk: telephone switching – C
  - Wireshark: network protocol analyzer – C
  - JSPWiki: wiki engine – Java
  - OpenFire: IM and groupchat server – Java
  - WordPress: blogging – PHP
  - All are open source programs
  - 24k LoC (WordPress) to 2.2M LoC (Wireshark)
- 87k synthetic C/C++ and Java test cases

Software Assurance Marketplace (SWAMP)

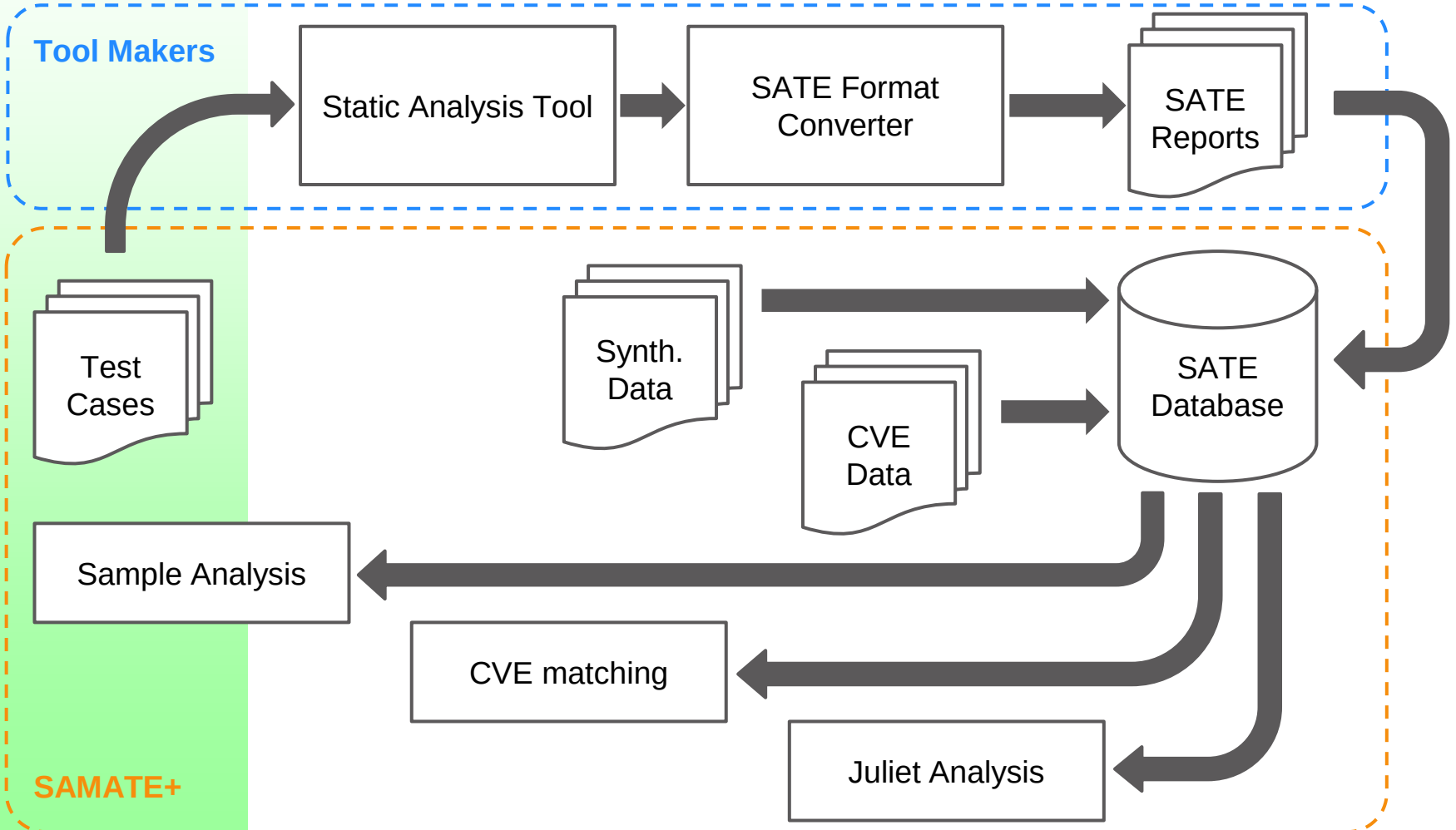
- Test cases in virtual machines hosted by SWAMP
  - VMs have the needed libraries for compiling test cases
  - Ubuntu Linux or Windows VM
- Teams install and run their tools
- Provides consistent environment

Coverage Claims Representation (CCR)

- XML format to tell what a tool looks for
  - Specific CWEs
  - Details of coverage
- Allow more accurate measurements
- Help us analyze tool outputs



SATE Procedure



Warning Sample Selection

For vulnerable versions only

- We assigned severity if a tool did not
- Avoid warnings with severity 5 (lowest)
- Select more warnings from higher severities
- Select 30 warnings from each of 30 tool reports
  - 1 report had only 9 warnings
- Total is 879

CVEs

- Identify the CVEs
  - Locations in code
- Find related warnings from tools
- Can tools discriminate between versions
  - Or report for a fixed version also?
- Goal: focus our analysis on real-life exploitable vulnerabilities

Differences from SATE IV

- SATE V Ockham sound analysis criteria
- PHP language track
- Test cases hosted in SWAMP
- Coverage Claims Representation (CCR)
- Will not make tool outputs public

- Still, much can be improved…

Thanks to teams!