The Challenge of Artifacts and Summaries for Analysis Tool Execution and Qualification

Paul E. Black

paul.black@nist.gov

October 2021

Executive Order on Improving Cybersecurity

- Mandated many specific tasks with short deadlines.
- OMB will require for Federal Acquisition; add to FARS and DFARS
- Deadline to publish is 6 February 2022

Subsection 4(e)(iv)

“employing automated tools, or comparable processes, that check for known and potential vulnerabilities and remediate them, which shall operate regularly, or at a minimum prior to product, version, or update release;”

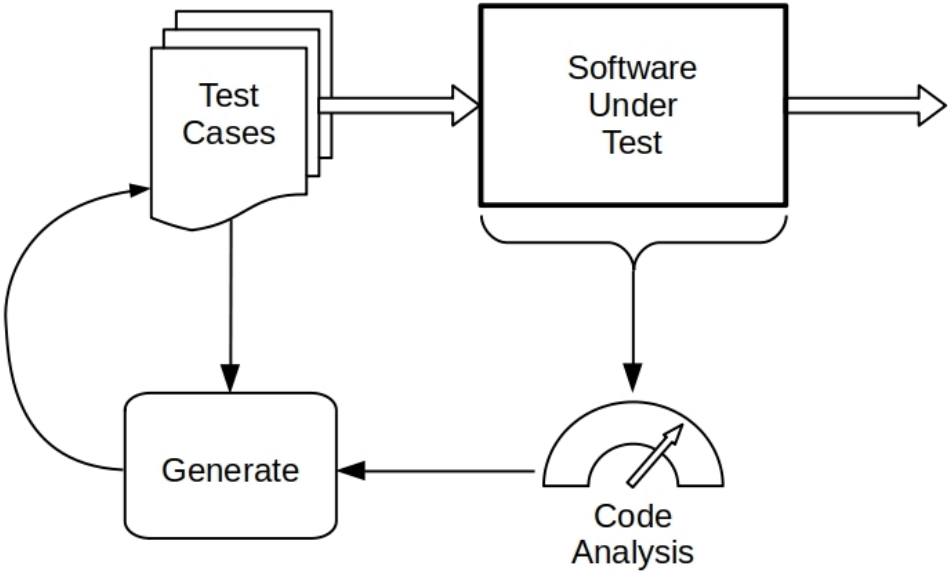
“… check for … vulnerabilities …”

- Do we hope to find *lots* of vulnerabilities?
  - No, that indicates poor software.
- Then if we don’t find many, is the software good quality or is the tool poor??
- We don’t want to dictate particular tools.
- Challenge: tool qualification
  - Some way to assure that a tool does what we want.

How Developers Might Qualify Tools

Types of tools

- Static analyzers to find bugs
- Fuzzers and web app scanners
- Software Composition Analyzers
- Test case generators

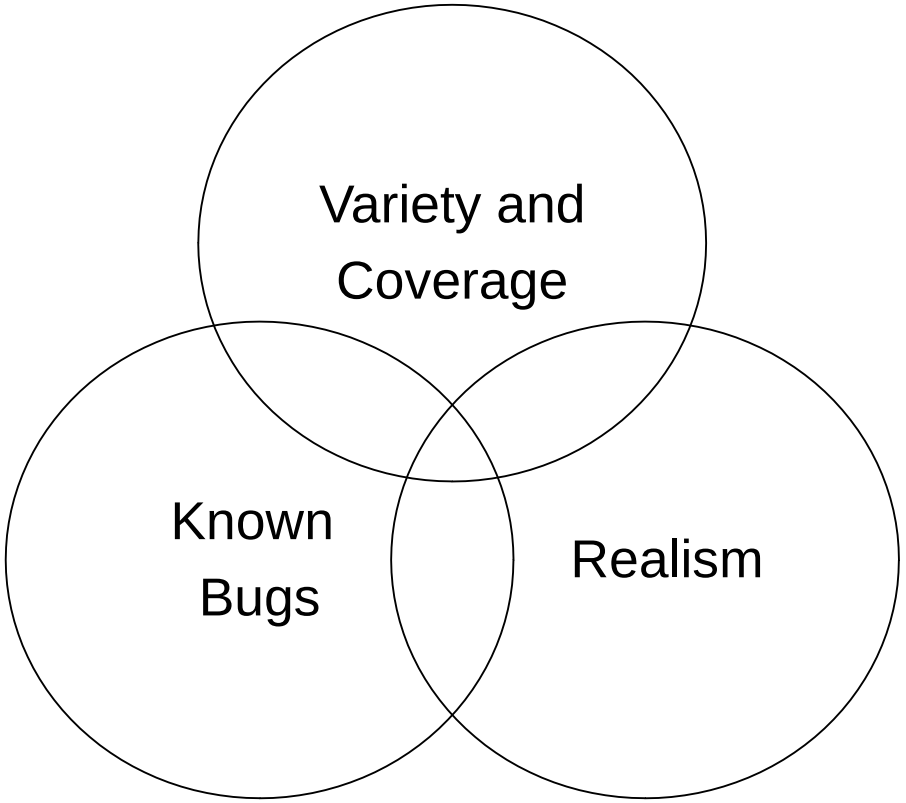


We Want to Find All(?) Bugs

- Decide what classes of bugs are most important.
  - What are “all” the kinds of bugs you want to find?
    Top 25, CWE, BF, and historical bugs may help.
  - Threat assessment may help here.
- Determine the most practical way to preclude, detect & remove, or mitigate each class.

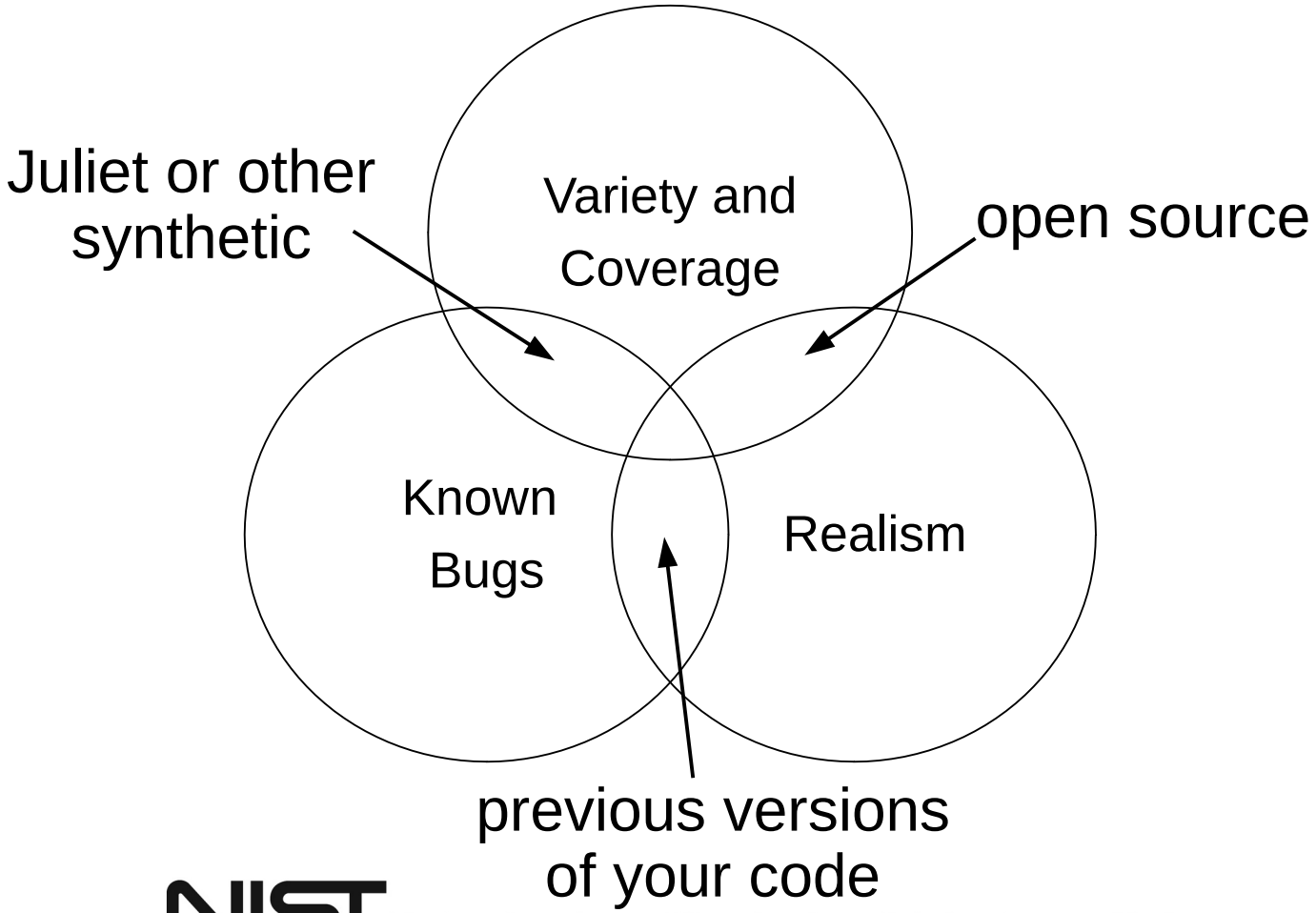
Qualification Test Suites

- Choose qualification suites

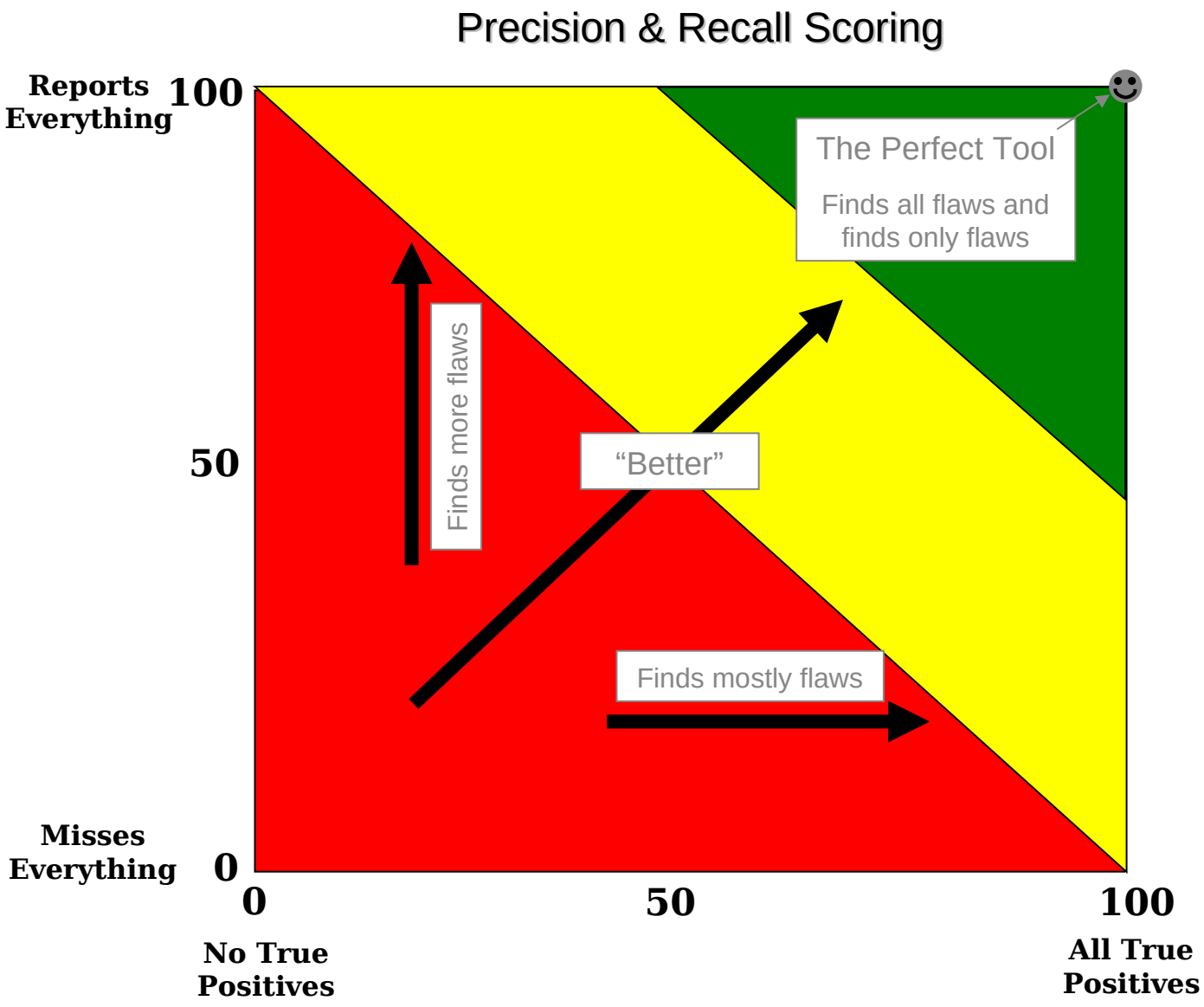


Qualification Test Suites

- Choose qualification suites

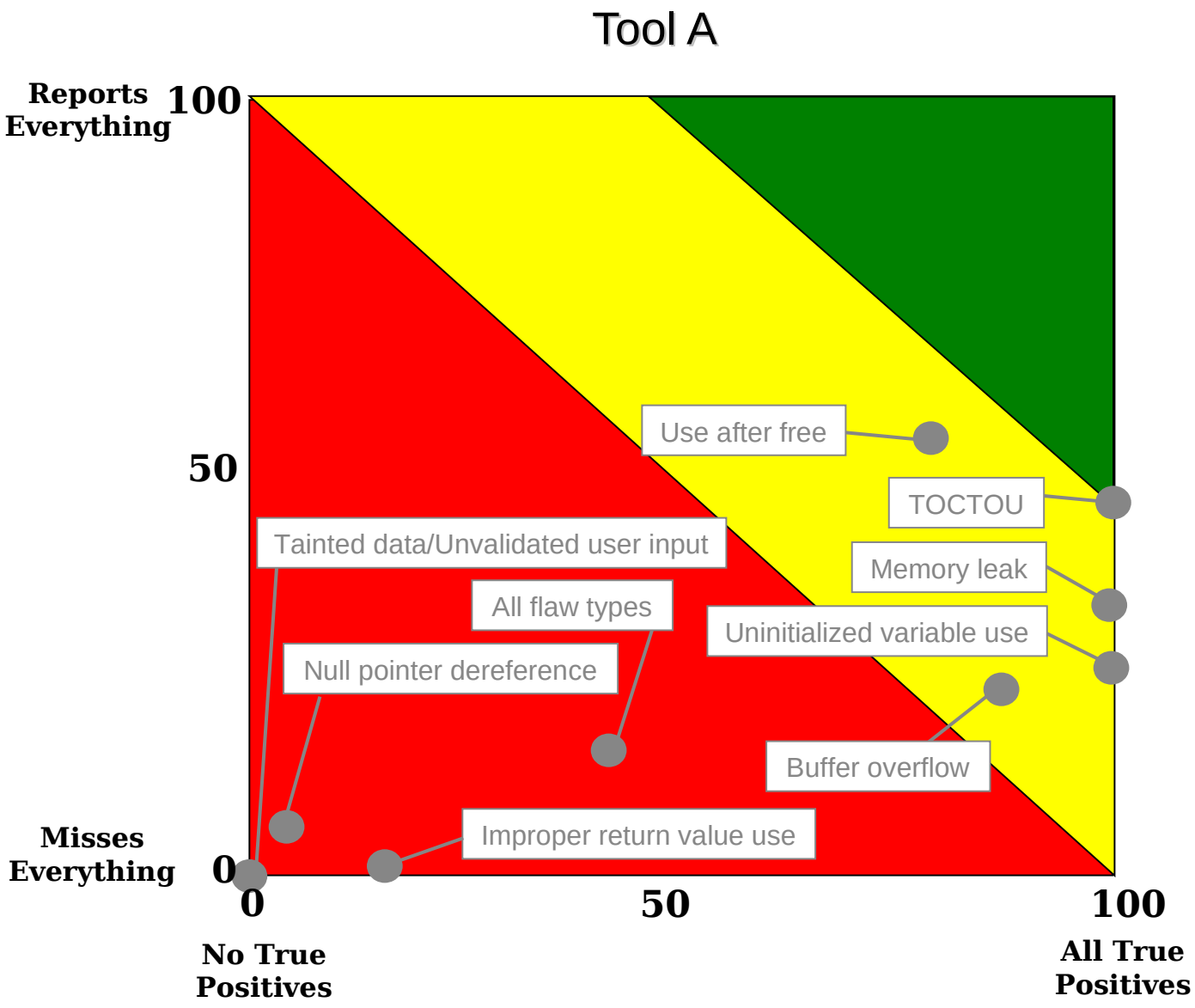


Tools Differ in Response Profiles



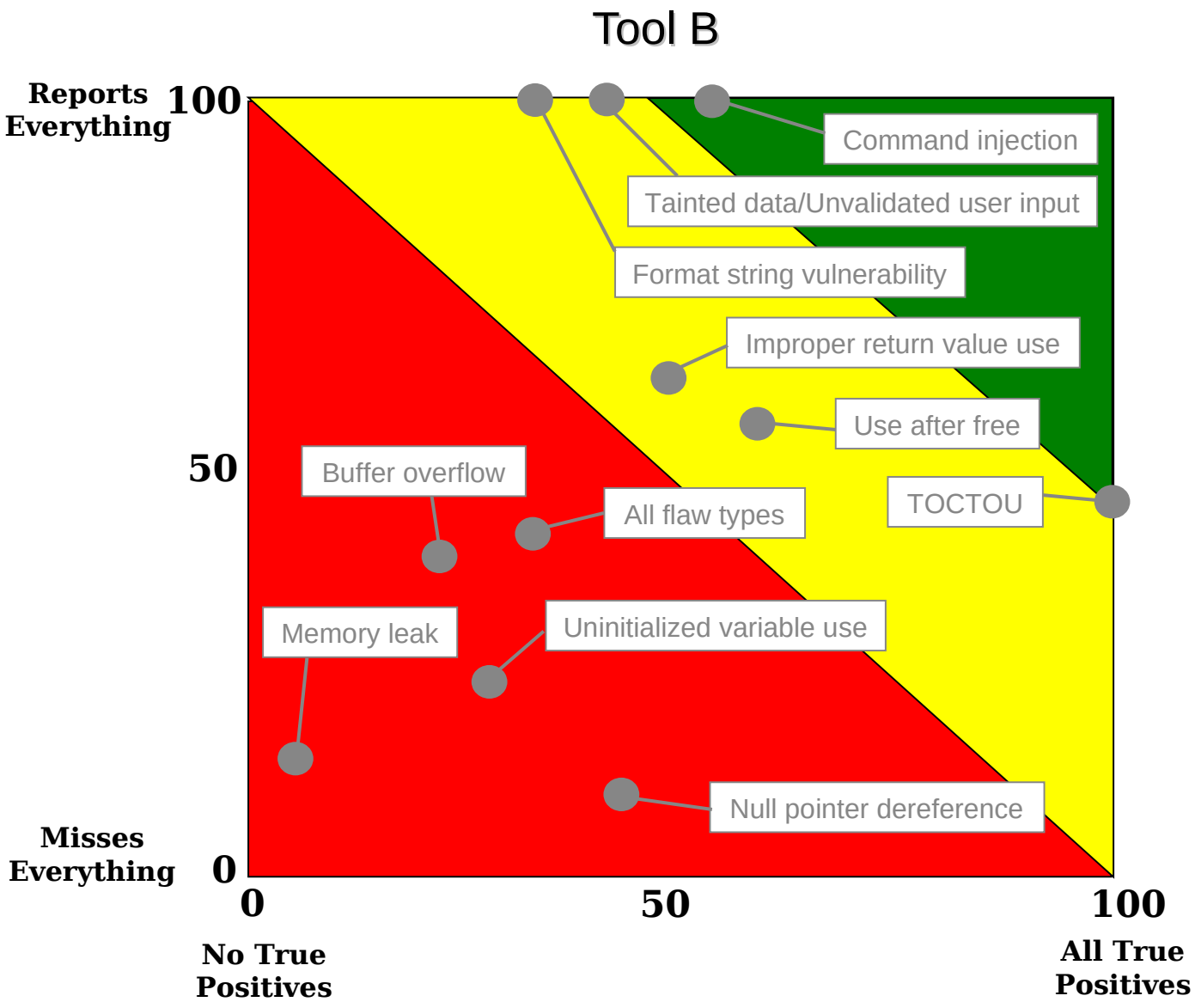
from NSA

Tools Differ in Response Profiles



from NSA

Tools Differ in Response Profiles



from NSA

Subsection 4(e)(v)

“providing … artifacts of the execution of the tools and processes described …, and making publicly available summary information on completion of these actions, to include a summary description of the risks assessed and mitigated;”

Subsection 4(e)(v)

"providing … artifacts of the execution of the tools and processes described …, and making publicly available summary information on completion of these actions, to include a summary description of the risks assessed and mitigated;"

Artifacts and Summaries of Execution Goals

We want information that is

- Effective — leads to more secure software,
- Efficient — high benefit/cost ratio,
- Flexible — for current variations and future innovation, and
- Applicable to small shops, say 3 or 4 people, not just Google, IBM, and Microsoft.

Challenge of 4(e)(v)

What “artifacts of execution” and “summary information” will

1) Communicate assurance,
2) Be reasonable to produce *and check*,
3) Won’t disclose (much) proprietary information, and
4) Accommodate future innovation?

Note

- Self-attestation
  - 3rd-party certification is too time-consuming
- Attestation does not *prove* some level of assurance.
  - It reassures acquirer that gross incompetence or crude deceptions are discovered (not Volkswagen emissions cheating revealed in 2015).

Potential Artifacts of Execution

Threats considered

Attack vectors considered

Architecture—domains (for fault isolation)

Software Bill of Material (SBOM) is accurate and latest versions are used

- both open source and proprietary software

What was done or run?

- Tool name, version, execution date, options used, etc.—SARIF can inform

What was checked for?

- Patterns, bug classes, etc.—e.g. MITRE Coverage Claims Representation

Potential Artifacts of Execution II

Fuzzer and web app scanner input/generation models

Coverage

Overall %, modules/files/functions with low coverage

For testing: statement coverage; combinatorial input space coverage

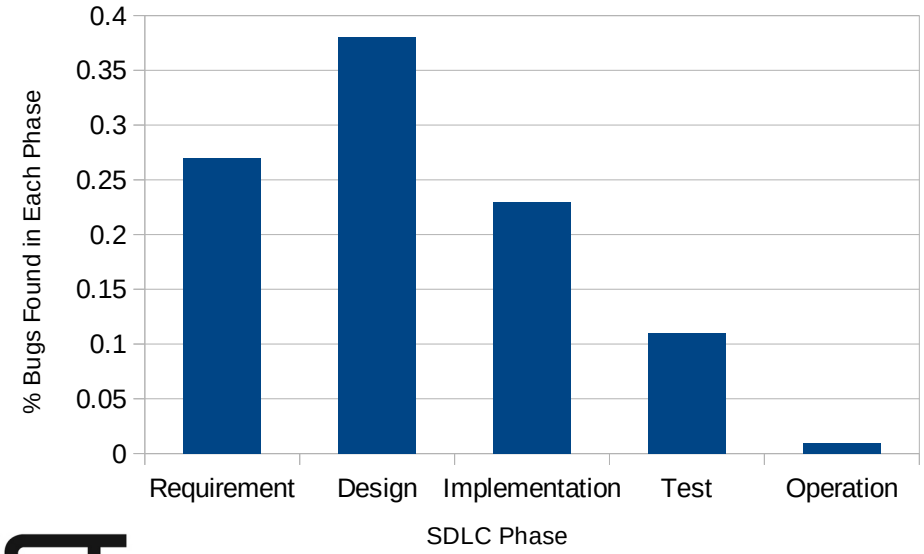
For static analysis: # sites for each bug type; % sites examined

Potential Artifacts of Execution III

Weaknesses, bugs, or vulnerabilities found

- Origin? (Root cause)
- If fixed, then when (any process change?)
- If not fixed, estimated severity; triggering circumstance

Relative Numbers of Bugs Found in Each Phase



Public Summary Information—Potential

Summary of the risks found and mitigated

Tools run

Overall coverage

- % statements
- % sites
- n-way input space

test cases in regression suite

(or hours) of fuzzer/web app scanner runs

Summary of Challenges

- Ways for developers to qualify tools
- Development and verification process artifacts
  - *What artifacts and information assure you?*
  - *What would you examine to decide?*
- Publicly available information (software “labels”)

Contact Paul E. Black paul.black@nist.gov