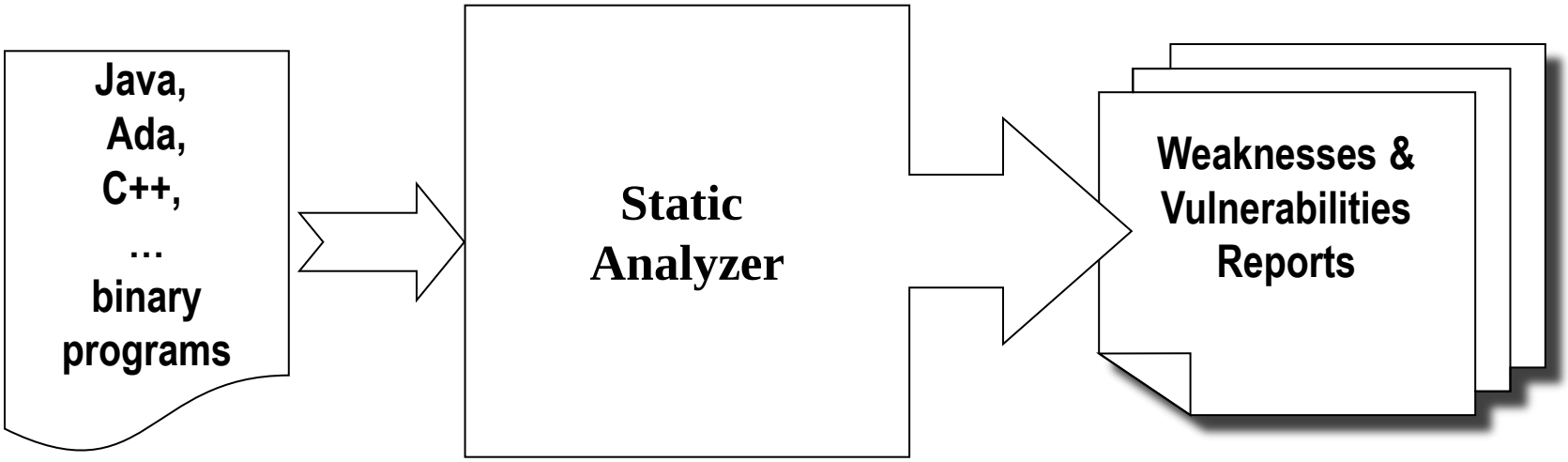
Toward CWE Compatibility Effectiveness

Paul E. Black

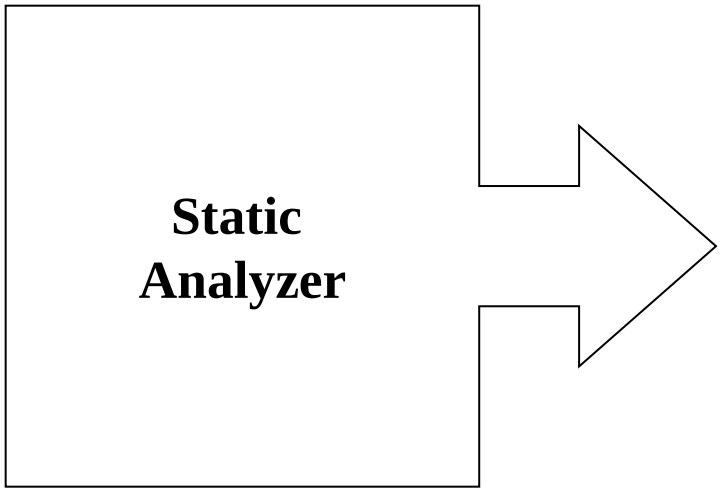
paul.black@nist.gov

What is Static Analysis?



- Examine design, source code, or binary for weaknesses, adherence to guidelines, etc.

Does this Static Analyzer Work?



- In particular, does it find the weaknesses (Common Weakness Enumeration - CWE) that it claims (Coverage Claims Representation - CCR) to find?

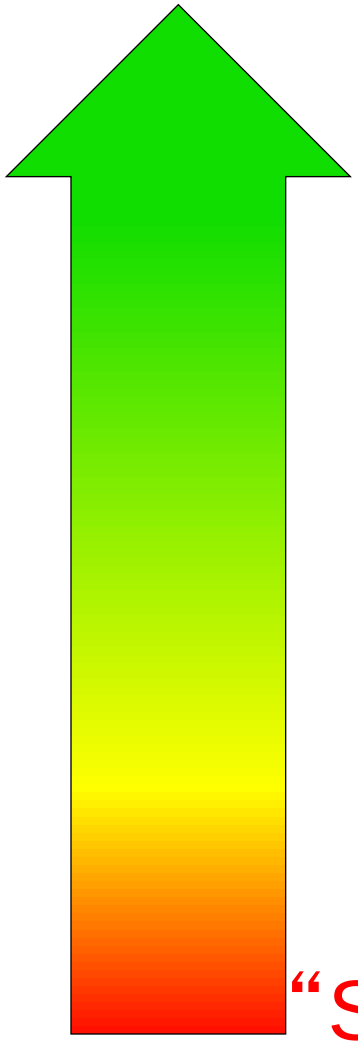
MITRE's CWE Compatibility and Effectiveness Program

- **Phase 1 – Declare compatibility**
- **Phase 2 – Verify mapping to CWEs**
- **Phase 3 – Test cases show effectiveness**
  - **tool/service effectively locates CWEs**
  - **tool/service deals with code complexities**

  http://cwe.mitre.org/compatible/program.html
- **We propose (1) what acceptable test cases are, and (2) that the SAMATE Reference Dataset (SRD) be the test case repository**

The Problem

- **Sound analysis is not perfect anyway.**
- **No test set can show all possible bugs, heuristics, variants, and corner cases.**
- ***How thorough should the test set be?***



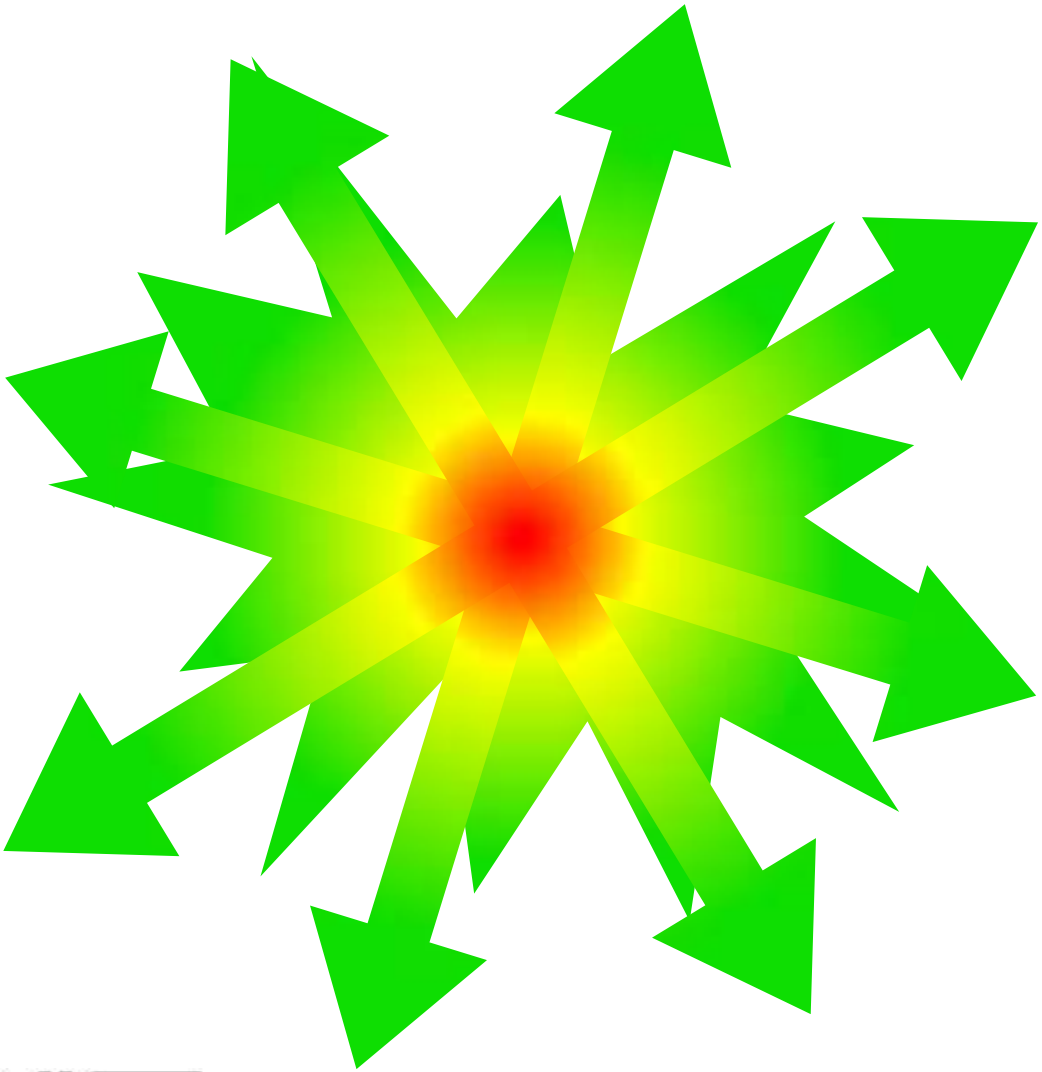
Sound

Useful

Ok

"Snake oil"

Measurement is Multidimensional



What are the Questions?

1. **Who should decide the criteria?**
   - **NIST,**
   - **MITRE, and**
   - **tool users**
2. **What should the content be like?**
3. **Can sets be cheat-proofed?**
4. **Who creates the test cases? How?**
5. **What about versions?**

What should the content be like?

- **Each CWE has one or more tests**
  - **short (this is not about handling megacode)**
  - **code is vulnerable, i.e., exploitable**
  - **(usually) synthetic**
  - **fairly "clean", but not necessarily pristine; meet SRD "accepted" standard**
  - **standard code; no language extensions**
- **Test cases have corresponding "fixed" test cases, to measure false positives**

What is a Code Complexity?

```
char data;

data = 'C';




data = 'Z';
printHexChar(data);
```

```
char data;
if (1) {
  data = 'C';
} else {
  data = 'C';
  printHexChar(data);
}
if (1) {
  data = 'Z';
  printHexChar(data);
} else {
  printHexChar(data);
}
```

after Juliet test set CWE563_Unused_Variable__unused_value_char_01.c and …_02.c

What about code complexities?

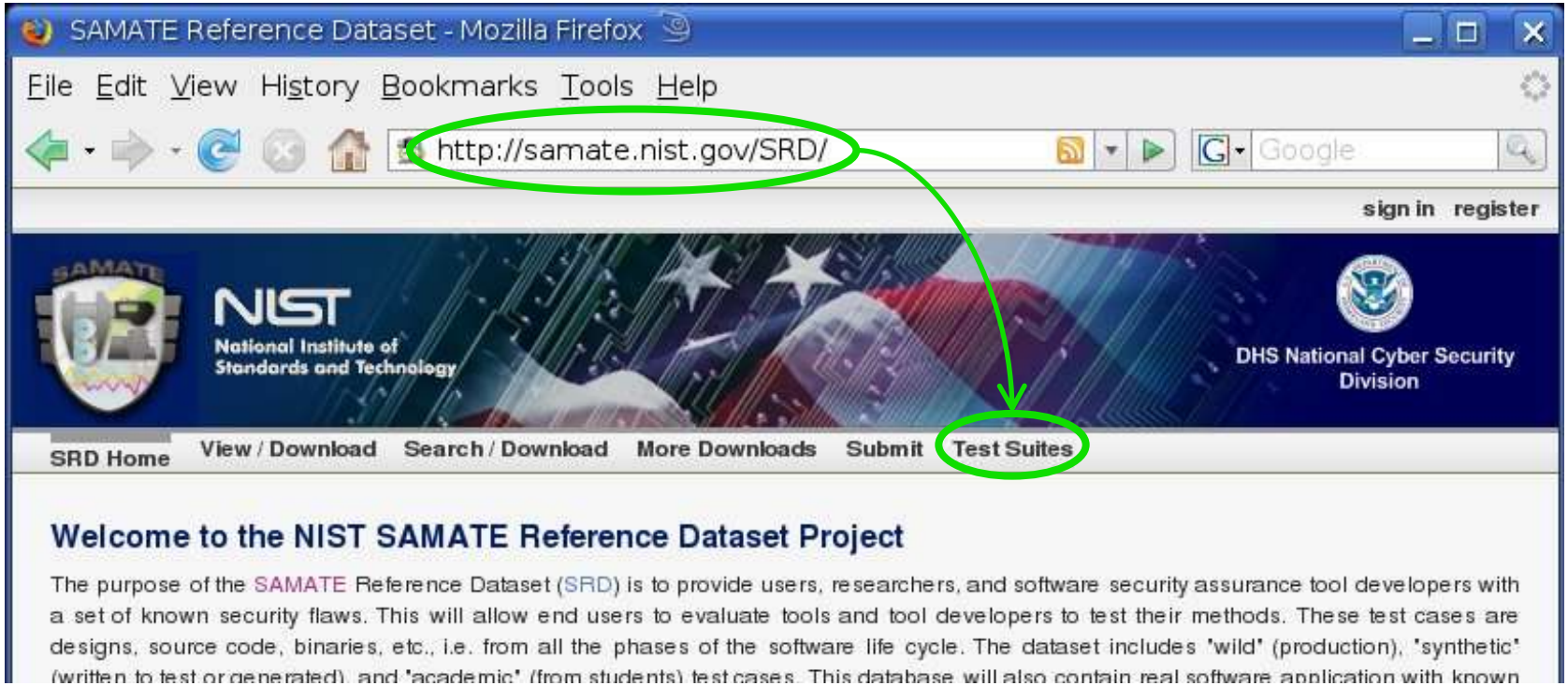
- ***Code complexities*** **are complications that do not affect the CWE.**

- **The fundamental set for each CWE has minimal complexities, perhaps none.**

Proposal:

- **A broad test set to explore complexities.**
  - **maybe several, but not one for each CWE**

Juliet Test Suite



- **59,493 small C/C++ and Java programs**
- **covering 128 CWEs**
- **in dozens of subtle variations**
- **within dozens of code complexities.**

Can sets be cheat-proofed?

- **A public, static set is susceptible to cheating. (A secret set has other problems.)**
- **Maybe change comments and identifier names for every download?**
- **Maybe add innocuous statements?**
- **Maybe transform code, like unroll loops?**

Proposal:

- **If concerns, privately corroborate results.**

Who creates the test cases?

- **Test cases may come from anywhere.**
- **Some cases chosen from SRD, especially Juliet test set.**
- **MITRE is working on a case generator.**
- **Contributions welcome from researchers.**

What about versions?

- **Test sets may have programs added, removed, or replaced as knowledge of measuring CWE effectiveness improves.**
- **Posted results include version of test sets and version of tool.**
- **Participants may retake the effectiveness tests and post *additional* results for new versions of tool or tests.**

Where should we begin?

- **3 or 4 fundamental test cases for covered CWEs (and corresponding "fixed" cases).**
- **A set of test cases for code complexities in one or two most common CWEs, e.g., Stack-based Buffer Overflow (CWE-121) or OS Injection (CWE-78).**

Phase 3 – Effectiveness Phase

- The major aspect of the CWE Effectiveness phase is:
  - focused on providing your prospective customers with an understanding of which specific CWE identifiers your capability reviews artifacts for;
    - e.g. though Claims Coverage Representation (CCR)
  - to provide a public collection of test results that will allow a prospective customer to understand which CWE identifiers your capability is effective in locating; and,
  - to articulate what types of complexity in software your capability is most successful at dealing with when looking for CWE identifier labeled weaknesses.
- The posting of the test results on the CWE Web site will conclude the CWE Effectiveness Phase and an appropriate CWE-Effective logo and brand will be made available for your use.

Questions or suggestions?

Paul E. Black paul.black@nist.gov