Securely Taking on New Executable Software of Uncertain Provenance (STONESOUP)

Neutralizing Exploitable Vulnerabilities in Software

Program Manager: Mr. W. Konrad Vesey; E-mail: william.vesey@iarpa.gov

Software risk assessment is ad hoc, labor-intensive, and relies on tools of uncertain effectiveness

- Commercial tools have either high false positive rates (finding flaws that aren't there), high false negative rates (missing flaws that are there), or both
- Security evaluation tools are designed for security-savvy software developers and are difficult for end users or IT administrators to configure or use effectively

- Software changes so fast that security evaluations are often out of date the moment the report is issued

Each research team addresses successively larger programs and more weakness types in their targeted software class

Phase 1	Phase 2	Phase 3
Neutralize 75% of vulnerabilities of 2 weakness types in 10,000 SLOC programs	Neutralize 80%+ of vulnerabilities of 4 weakness types in 100,000 SLOC programs	Neutralize 90%+ of vulnerabilities of 6 weakness types in 500,000 SLOC programs

- STONESOUP neutralizes vulnerabilities in:

Java

```
class MyClass {
    String s = "Hello";
    ... }
```

Java source code
Kestrel Institute,
Palo Alto, CA

C

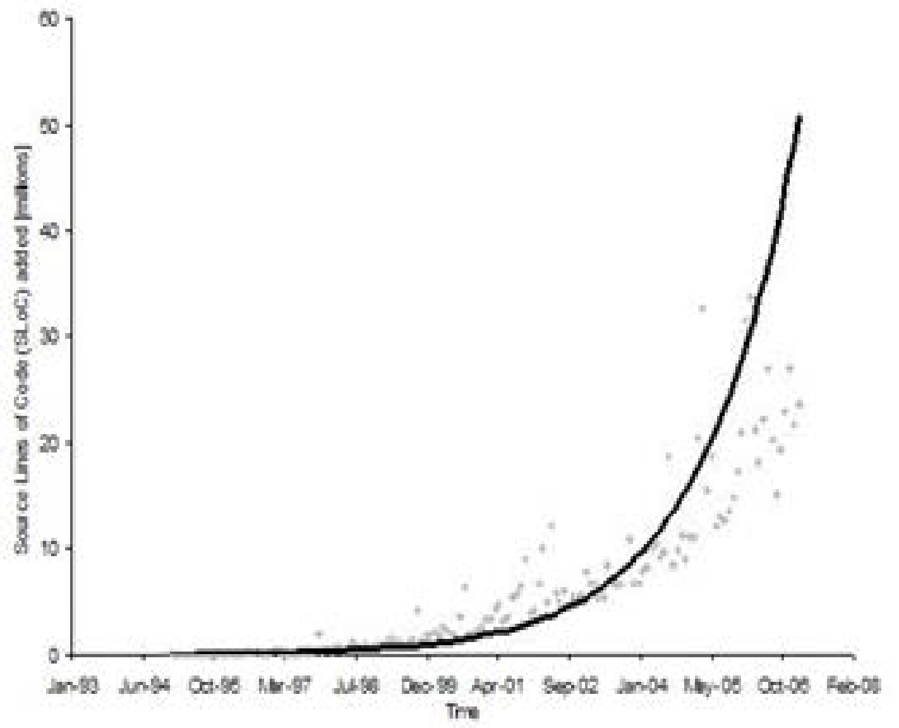
```
int main (
    int argc,
    char ** argv)
{ ... }
```

C source code
Columbia University,
New York, NY

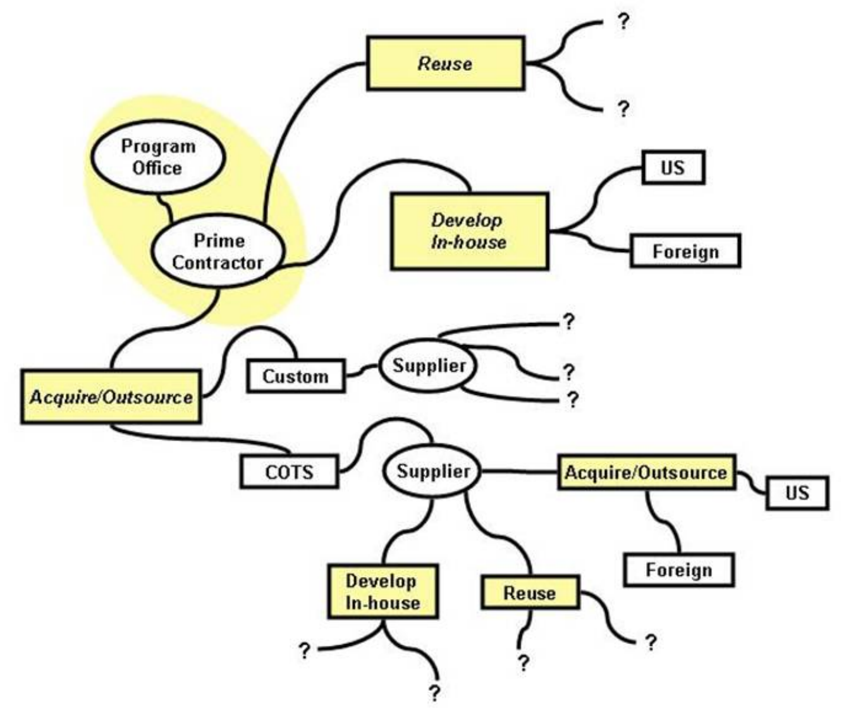
Linux binary executables
GrammaTech,
Ithaca, NY

GRAMMATECH

How can we benefit from highly functional software produced by a globalized industry without putting the enterprise at risk?



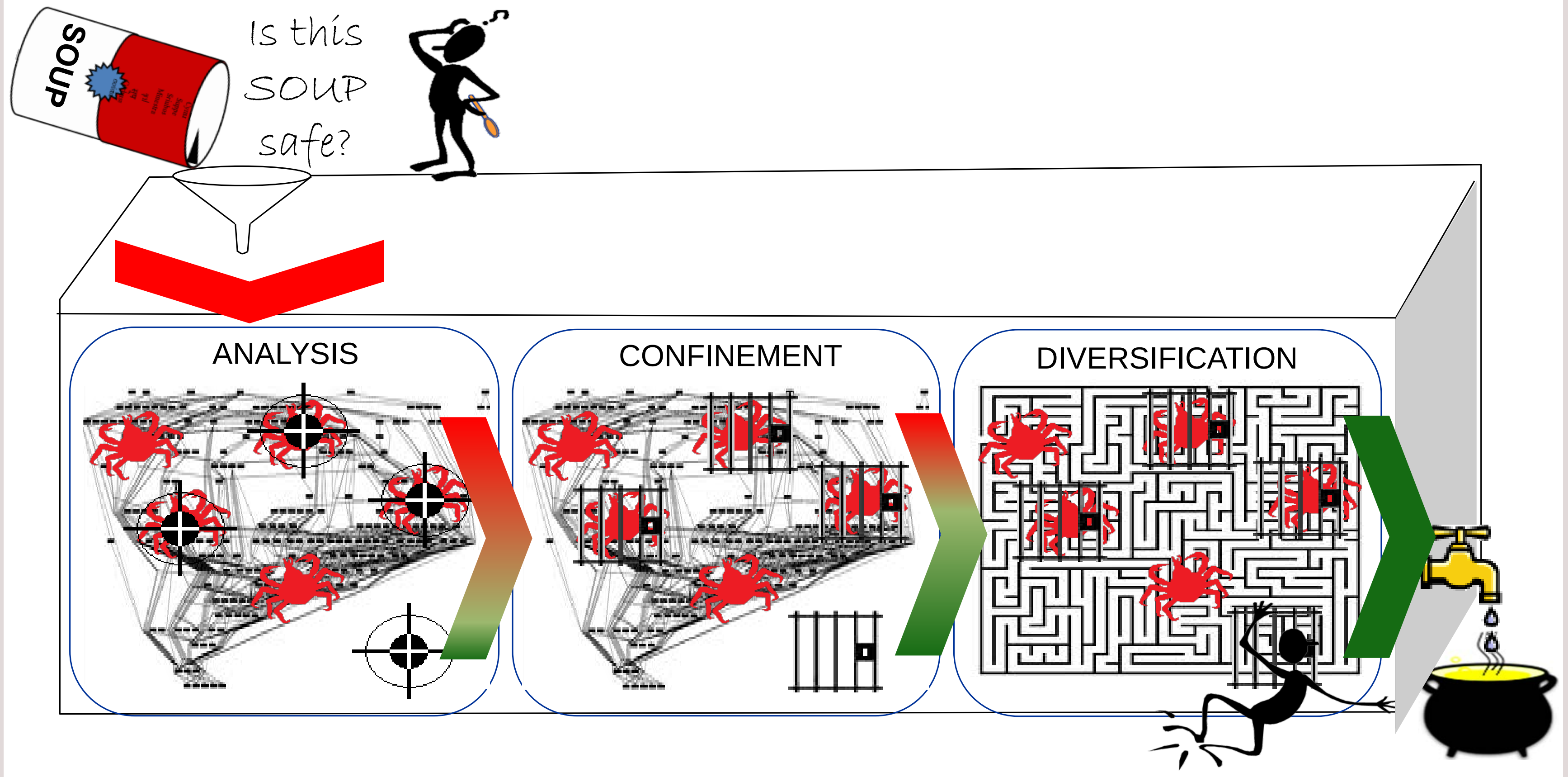
The volume of software deployed in the IC is growing faster each year



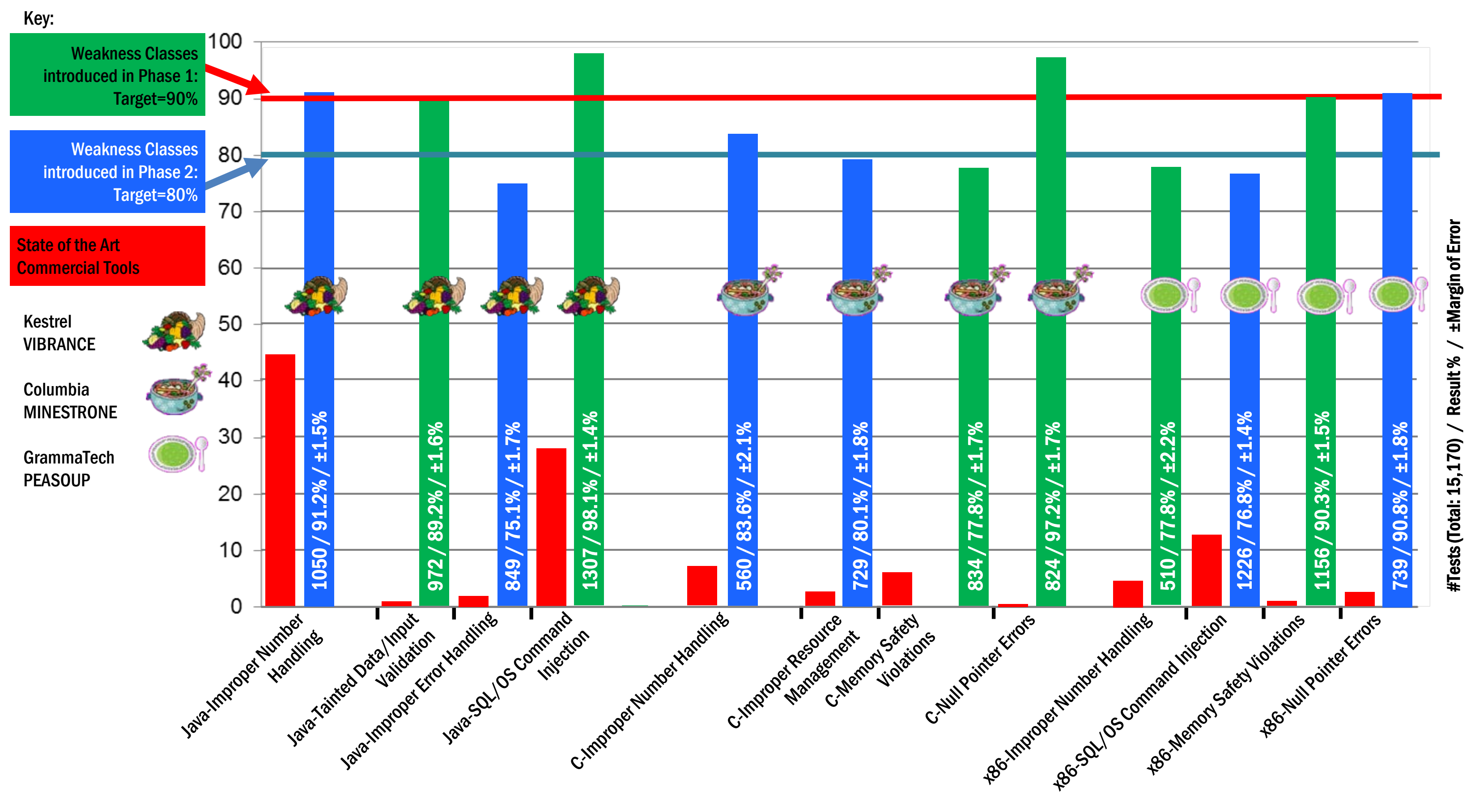
Provenance is increasingly irrelevant to risk assessment as supplier networks grow in complexity and globalized

STONESOUP automatically finds and mitigates exploitable security vulnerabilities in software

- Analyzes programs, not the data processed by programs
- Looks for coding flaws, not attack patterns
- Protected programs launch and run as expected
- Works with software as-built–no input from the supplier
- Protection is applied automatically–users need no special knowledge



STONESOUP technologies effectively address the most common implementation weaknesses in software



Weakness Types	
Insecure Number Handling	Insecure Handling of Tainted Data
Insecure Error Handling	Resource Drains
SQL or Command Injection	Concurrency Errors
Memory Safety Violations	Null Pointer Dereferences

STONESOUP prototypes are available for evaluation. Test data will be disseminated to stimulate further cyber research

NIST The SAMATE test data repository at NIST will host STONESOUP T&E data

Questions for future research:

- Can security transformations be coupled with performance optimizations so that secure code is actually faster than insecure code?
- STONESOUP technologies see a 6- to 8-fold increase in effectiveness over static analysis alone by tracking actual user inputs at run time. Can static analysis techniques be improved to close this gap and identify more vulnerabilities earlier in the software life cycle?