CYBER SECURITY DIVISION

2013 PRINCIPAL INVESTIGATORS'

Gold Standard Benchmark for Static Source Code Analyzers

Kestrel Technology, LLC

Dr. Henny B. Sipma

Sep 16, 2013

Homeland Security

Science and Technology

Kestrel Technology

Founded: 2000

Location: Palo Alto, CA

Core activity: Sound static analysis (mathematical proof of safety)

Tool: CodeHawk

Languages supported: C, Java, x86 executables

Underlying technology: Abstract interpretation (Cousot,Cousot, 1977)

Properties

C: Language-level properties: memory safety, null-dereference (application-independent, mathematically well-defined properties)

Java: Information flow analysis, resource analysis, integer overflow, error handling

X86: Memory safety, information extraction

Software infrastructure: A need for software security metrics

Cement is everywhere. Our very lives depend on cement, yet we don't worry about it: such is the legacy of Joseph Bazalgette, who introduced **measurement and strict quality controls**

Physically safe in the largest buildings, but …

Like cement, software is everywhere: our very lives are becoming more and more dependent on software.

Unlike cement, for more than 50 years software has been continuously released and added to the national infrastructure, plagued by design and implementation defects that were largely detectable and preventable, but were not. **Why not?**

(from Geekonomics, David Rice)

not safe from continuous attacks from all over the world

Customer Need

We need the ability to measure software quality

We need the ability to measure the effectiveness of the tools that are used for software assurance

in terms of

False positives: bugs that turn out not to be bugs

AND

False negatives: bugs that are missed

Problems with measurement

From the 2012 Coverity Scan Report (May 2013):

TABLE 2: AVERAGE DEFECT DENSITY ACROSS ALL ACTIVE COVERITY SCAN PROJECTS, 2008-2012	
Coverity Scan Report Year	**Average Defect Density**
2008	.30
2009	.25
2010	.81
2011	.45
2012	.69

(number of defects per 1000 LOC)

Did software get worse?
Did detection technology get better?
Some combination of the two?

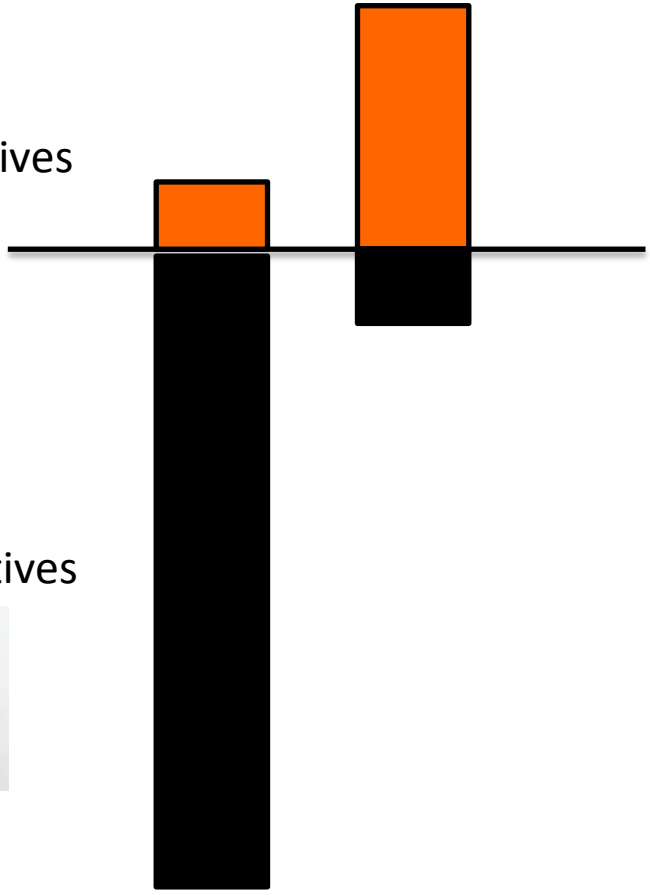
What is the "real" number?

What is a defect?

Compounding problem:
Economic incentives favor less assurance

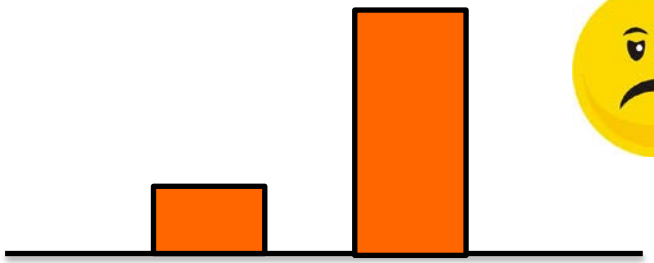
false positives

Reality



false negatives

Customer sees



.... and will choose Tool A

Our Solution: Measure memory-safety defects

Define Defect

mathematical definition based on

- C standard semantics (37 cases of memory-related undefined behavior)
- Semantic explication of C semantics by George Necula (Berkeley, 2002)
- ➢ **covers all of C plus gcc extensions**

Measure Defects

mathematical proof based on

- Primary proof obligations (for all memory accesses and related instructions)
- Sound static analysis to discharge proof obligations of safe memory accesses
- Demonstrate vulnerability in remaining cases

Our Solution: Measure memory-safety defects

Define Defect

mathematical definition based on

- C standard semantics (37 cases of memory-related undefined behavior)
- Semantic explication of C semantics by George Necula (Berkeley, 2002)
- ➢ **covers all of C plus gcc extensions**

Measure Defects

mathematical proof based on

- Primary proof obligations (for all memory accesses and related instructions)
- Sound static analysis to discharge proof obligations of safe memory accesses
- Demonstrate vulnerability in remaining cases

DIFFICULT

Approach: Capability and Technology

Kestrel Technology's Core Capability:

powerful and scalable, sound static analysis engine:

- based on theory of abstract interpretation (Cousot, Cousot 1977)
- generates invariants (over-approximation of program behaviors for all possible inputs

Technology: Distributed Proof Structure

- structural induction over the control flow graph of all functions
- assume-guarantee reasoning based on
  - function summaries (contracts)
  - data structure invariants
  - global data invariants
- hierarchy of primary and secondary proof obligations
- local invariant generation
- whole-program pointer analysis

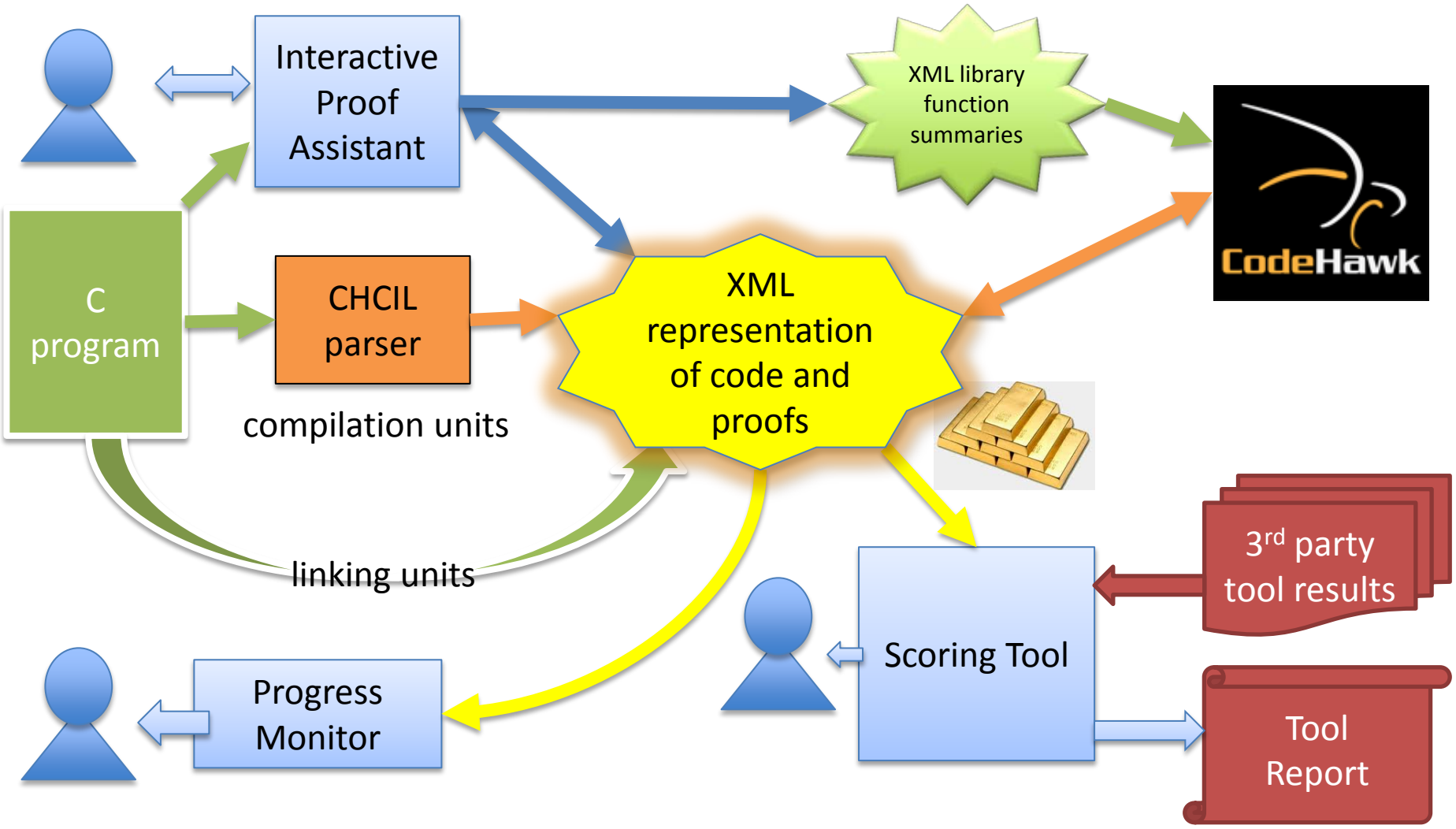
Apply to six benchmark programs

lighttpd	webserver designed and optimized for high-performance environments used by several popular websites, including YouTube and Wikipedia	144 files	38 KLOC
nagios	tool for IT-infrastructure monitoring with more than a million users, including Comcast, Philips, Siemens, Thales, McAfee	73 files	65 KLOC
naim	console chat client for AOL instant messaging (AIM) and internet relay chat (IRC)	44 files	23 KLOC
irssi	terminal-based IRC client for UNIX systems	347 files	53 KLOC
pvm	package that lets heterogeneous collections of Unix and Windows computers to be used as a single large parallel computer, developed by Oak Ridge Nat'l Labs	320 files	72 KLOC
dovecot	open-source IMAP and POP3 email server for Linux (written with security in mind)	1,111 files	119 KLOC

Apply to six benchmark programs

lighttpd (SATE 2008)	webserver designed and optimized for high-performance environments used by several popular websites, including YouTube and Wikipedia	144 files	38 KLOC
nagios (SATE 2008)	tool for IT-infrastructure monitoring with more than a million users, including Comcast, Philips, Siemens, Thales, McAfee	73 files	65 KLOC
naim (SATE 2008)	console chat client for AOL instant messaging (AIM) and internet relay chat (IRC)	44 files	23 KLOC
irssi (SATE 2009)	terminal-based IRC client for UNIX systems	347 files	53 KLOC
pvm (SATE 2009)	package that lets heterogeneous collections of Unix and Windows computers to be used as a single large parallel computer, developed by Oak Ridge Nat'l Labs	320 files	72 KLOC
dovecot (SATE 2010)	open-source IMAP and POP3 email server for Linux (written with security in mind)	1,111 files	119 KLOC

Approach: Architecture



Approach: Scoring tool

Scoring Tool:

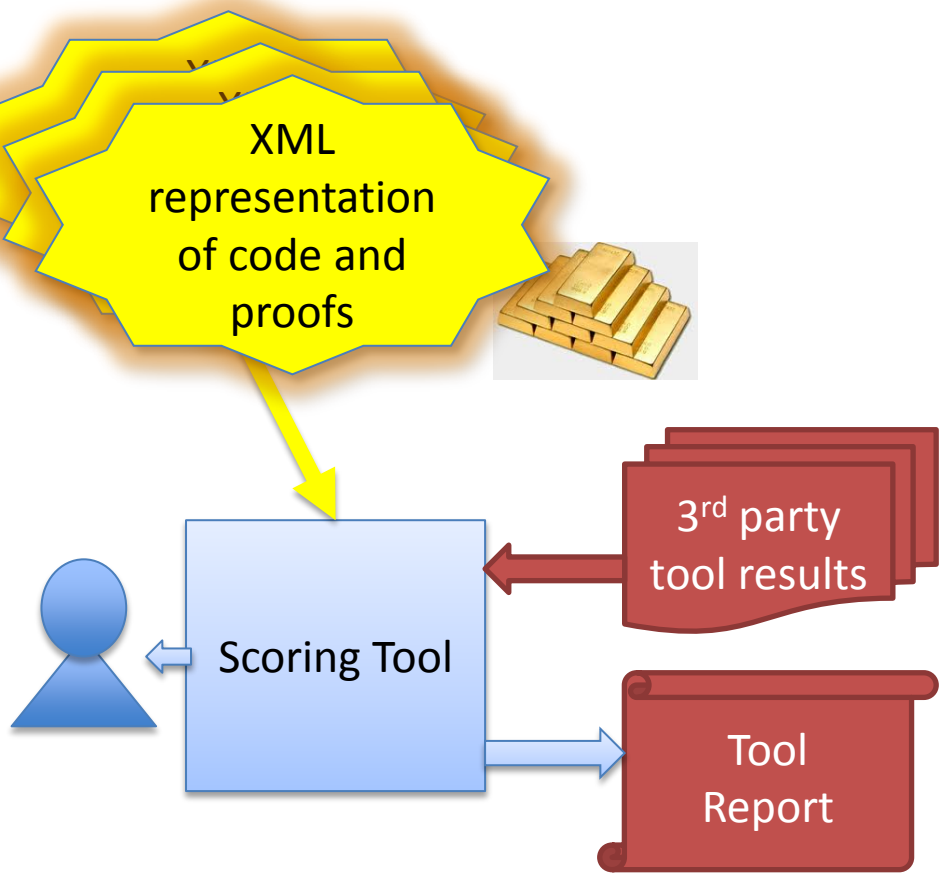
- Import results for all primary and secondary proof obligations
- Import 3rd party analysis tool results
- Compare
- Establish correspondence and report
  - false negatives
  - false positives
  - adequacy of bug placement
- Identify strengths and weaknesses of 3rd party analysis tool

Apply to results from **any** static analysis tool in the **SWAMP**

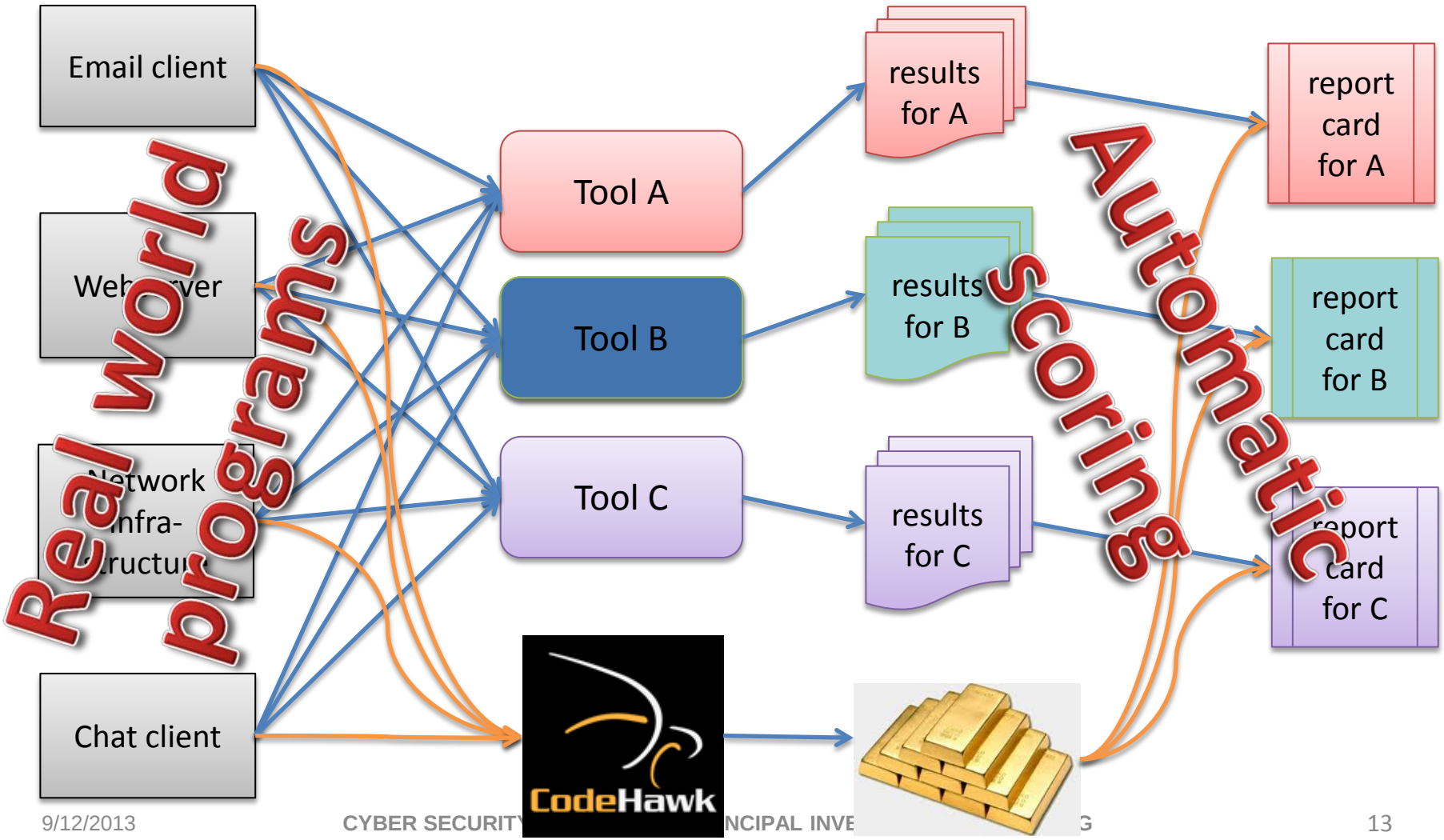
on

Programs:

- lighttpd
- nagios
- naim
- irssi
- pvm
- dovecot



Benefit: Ability to grade tools



Collaboration (rather than Competition)

Other government organizations:

- NIST (Dr. Paul E. Black)

Vendors of bug-finding tools:

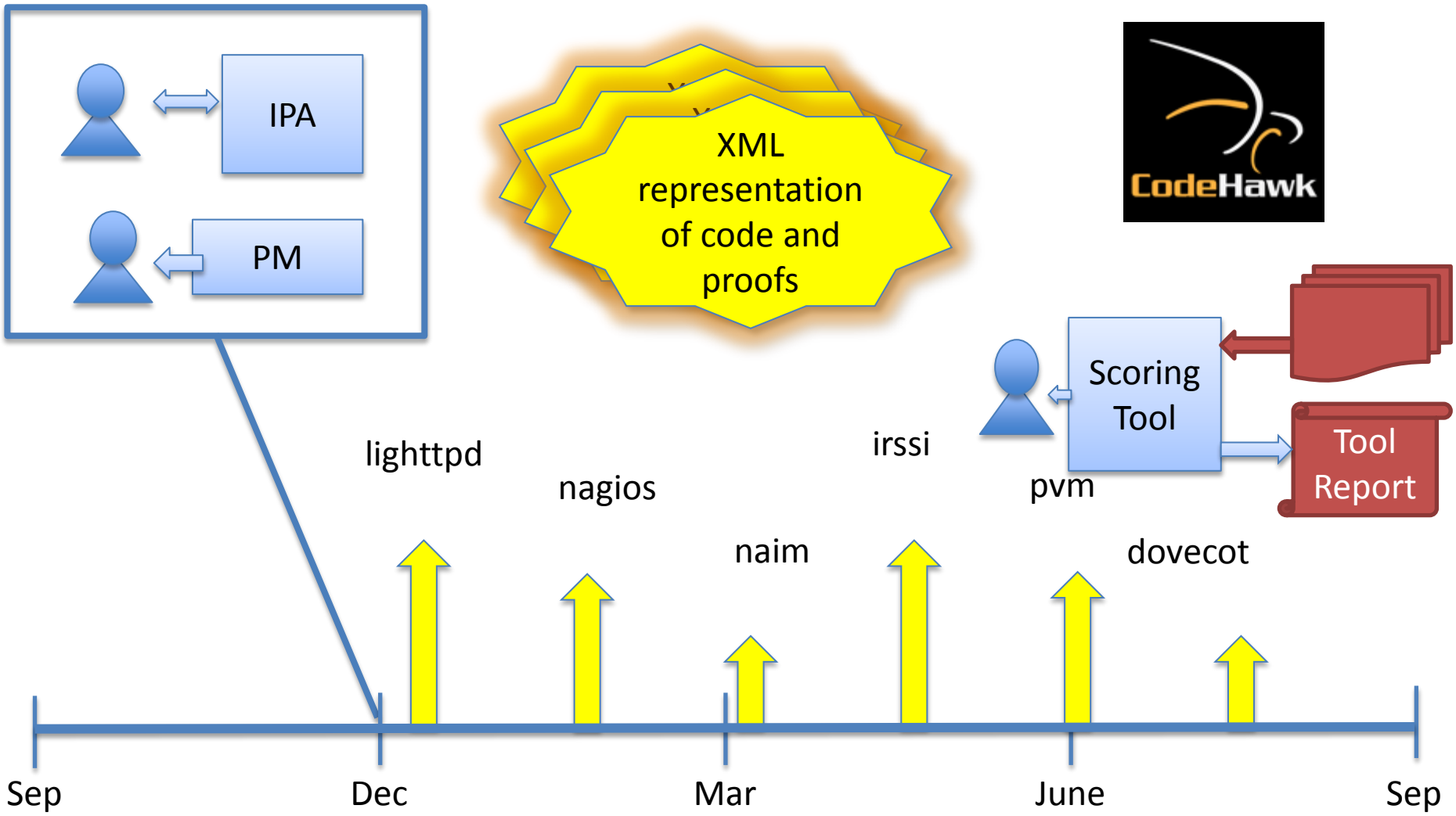
Offer scoring tool as a service to

- Identify strengths and weaknesses in bug-finding tools
- Improve bug-finding tools

Current Status: Accomplishments

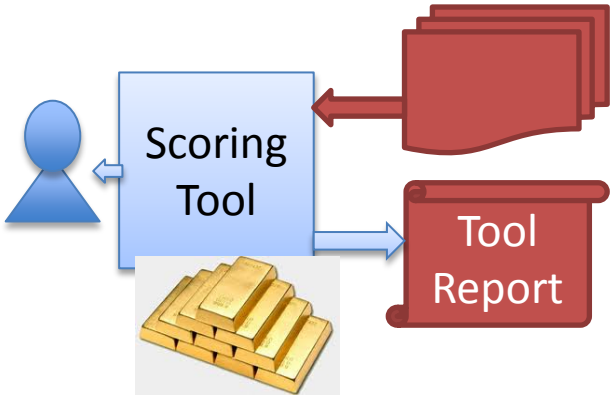
- ✓ Theoretical foundations of memory-safety defect measurement
  - shared and discussed with Dr. Paul E. Black (NIST)
  - relationship to 27 CWE's
  - illustrative application to lighttpd
- ❖ Software architecture and implementation (in progress)
  - xml representation of CIL semantic constructs
  - xml library function summaries
  - interactive proof assistant
  - progress monitor
- ❖ Benchmark programs (in progress)
  - xml representation of lighttpd

Schedule and Milestones

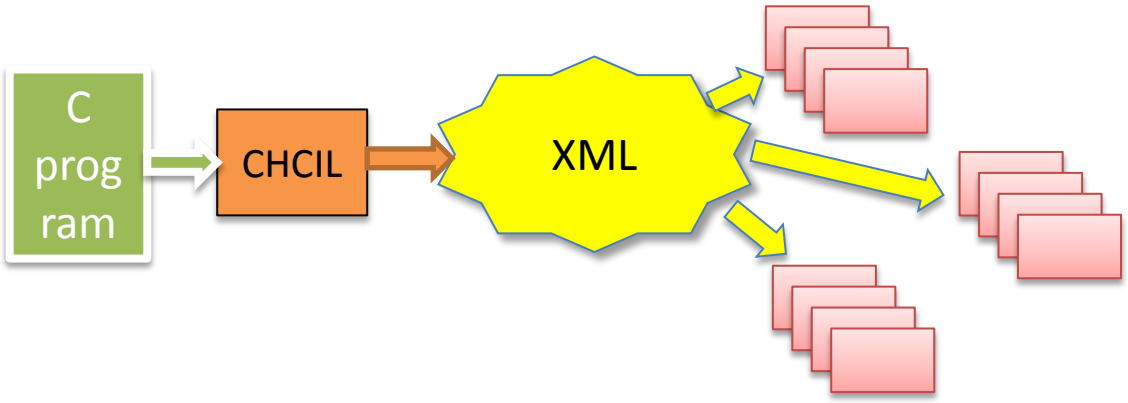


Technology Transition

SWAMP



Ability to score and evaluate static analysis tools



Preprocessed programs to

- lower entry barrier for new tool developers
- facilitate collaborative analysis

Contact Information

Team:

- Henny Sipma, Principal Investigator
- Rich Barry, Program Manager
- Eric Bush
- Arnaud Venet (Consultant)

Contact Information: sipma@kestreltechnology.com

Kestrel Technology, LLC
3260 Hillview Ave
Palo Alto, CA 94304
tel. (650) 320 8487