CYBER SECURITY DIVISION
2013 PRINCIPAL INVESTIGATORS’

SWAMP – The Software Assurance Market Place

Morgridge Institute for Research and University of Wisconsin - Madison
Miron Livny and Bart Miller

09/15/13
Building and Operating the SWAMP is a joint effort of four research institutions – Morgridge Institute for Research (lead), Indiana University, University of Illinois Urbana Champaign and University of Wisconsin – Madison
A Facility for the SwA community

Our target customers are all the members of the Software Assurance (SwA) community – tool developers, software developers, facility managers, researchers and educators.

The community needs a **continuous assurance** facility that will enable significant improvement in the quality of SwA tools and that will lead to a broader adoption of SwA tools and SwA methodologies.

The SWAMP will:

- **Profile** the ability of your SwA tool to identify (possible) software defect every time you commit a change
- **Identify** new (possible) defects in your software every time you commit a change
- **Identify** new (possible) defects in a software/library/module you are using every time a new version is released
- **Expose** your tools and software to the SwA community
Continuous Assurance Requires

- **Automation** – perform assessment runs with as many SwA tools as possible for every commit
- Compute and storage **resources** to meet the demand
- **Scheduling** capabilities to support priorities and deadlines
- **Integration** of assessment results from different tools and from different versions
- **Analytics** to profile impact of software engineering practices and quality of SwA tools and methodologies
- For tool developers, a **rich set** of software packages and reference suites
Technologies needed

Some Key Technology Areas:

- Components that simplify the job of applying SwA tools to arbitrary software packages (the “plumbing”).
- Centralized and unified repository of assessment tool results.
- Automation and capacity to provide continuous assurance capabilities.
- A full spectrum of tools: static/dynamic/hybrid, source/byte code/binary, mobile web.
- Infrastructure that preserves privacy and confidentiality.
Customer Need

For SwA tool developers: Enabling technologies
• Managing the realities of modern, complex builds
• Supporting for the small tool project

For SwA tool users: Simplifying access to tool technologies
• Once you get your software package to build, we automate the rest.

For the whole community: Assessment data in a single archive in a single format.
Approach

Building Software: The Ideal

- One directory, one executable
- Source code
  - Few files
  - Standard include files
- Standard libraries
- One tool
  - Source as input
  - Weaknesses as output
Building Software: The Complexities of the Real World

• Build generators, multi-level makefiles, custom scripts, complex conditional compiling criteria.

• Source files
  – From a variety of directories
  – Contents controlled by options, macros and includes
  – Generated by frameworks and tools

• Multiple executables

• Not obvious what to assess (no easily obtained list)

• Binaries:
  – Difficult to determine what source files were used in the creation of the executable
  – Also difficult to determine which object and library files used.
Approach

Plumbing to Track All: An Open Source SWAMP Product

Handles C, C++, and Java builds
Approach

Plumbing to Track All: An Open Source SWAMP Product

Diagram:
- Tool Driver
- SwA Tool
- Build Metadata
- Raw Assessment Results
- Software package, as downloaded
- Package, as built
Approach

Interactive Access to Supporting Building in the SWAMP

Build Scripts

Build Metadata

Software package, as downloaded

Package, as built

Handles C, C++, and Java builds
Benefits

Applying SwA Tools to Software Packages

• For the tool developer, you need to only understand the simple case of applying your tool to a known file. *The SWAMP plumbing does the rest for you.*

• For the software package developer, you only need to get your code to build. *The SWAMP plumbing automates the steps of apply tools to your package.*

• For the tool researcher, your results will get wide use. *The SWAMP allows you to be good at one thing, providing the context for the tool user.*
Current Status

- Building the environment in which to run assessments. Virtual machines containers, identity management, web interfaces.
- Selected initial set of five tools and run them in the SWAMP Findbugs, PMD, cppcheck, Clang, and Oink.
- The Java version of our “plumbing” has been released to the implement team (supporting Ant and Maven).
- Initial package selection in progress, with over 60 chosen so far.
  Goal is a variety of languages, frameworks, services, and usage communities.
- Active outreach, with TTA-1 performers, SwA tool developers, and analysis/viz tool developers.
Next Steps

• Pre-IOC (1/2014)
  – Recruiting and engaging beta users – fourth quarter of 2013.
  – Results visualization: integrating external tools and developing internal one.

• Post IOC
  – Handling multi-language builds
  – SwA tools for binary code
  – Dynamic tools
  – Tools for mobile code
  – …
Contact Information

Pat Beyer
Project Manager
pbeyer@ContinuousAssurance.org
(608) 316-4664

Miron Livny
Director and CTO
miron@ContinuousAssurance.org
(608) 316-4336