SWAMP

SOFTWARE ASSURANCE MARKETPLACE

Getting to Know the SWAMP (Software Assurance Marketplace)

Continuous Assurance:

Do it Early and Do it Often

SWAMP
SOFTWARE ASSURANCE MARKETPLACE

Software Assurance (SwA) Challenges

- Cybersecurity is on everyone's mind.
  - We immediately think about the network and the system (firewalls, software updates, pen tests, etc.), but we should remember that this also includes the software itself!
- The world is software-centric.
  - There are numerous entry points for a variety of attacks against confidential data and physical resources. Many software vulnerabilities and weaknesses exist while more continue to emerge.
- Software developers need effective continuous software assurance capabilities to integrate into their development workflows.
  - Find and fix problems with your code on a continuous basis.
- Consumers of software components need services to evaluate the quality of the components they deploy or integrate into their software stack.
- Challenges with software assessment tools:
  - Each tool has its strengths, but no single tool is good at everything.
  - Configuring, maintaining, and using tools is cumbersome and time-consuming.

What if we could…

- Simplify and automate the task of applying a broad spectrum of software analysis tools to software packages throughout the development lifecycle
- Deliver assessment results to the user in a way that is easy to understand
- Lower the obstacles to performing software security assessments
- Provide a resource for organizations and open-source developers to institute software assurance practices
- Promote software quality before deployment
- Foster more secure deployed software
- Allow users to collaborate and share SwA products and methodologies
- Serve as a testing and evaluation ground for new and mature software assurance tools and technologies

Continuous Assurance:

Do it Early and Do it Often

What Is SWAMP?

- The SWAMP, or Software Assurance Marketplace, is a no-cost resource available to the software community to promote a more stable and secure software ecosystem.
- We currently host 19 static analysis tools to check your code for weaknesses.
- Our results viewer, Code Dx, allows you to view the results from multiple tools in one place, making it easier to identify and address the most important problems in your code.
- We support continuous software assurance, the practice of scheduling assessments of your code throughout the development lifecycle and whenever code changes are made.

Continuous Assurance:

Do it Early and Do it Often

About Us

- Operational since February 2014
- Funded by a 5-year grant from the U.S. Department of Homeland Security
- A joint effort of 4 research institutions:
  - Morgridge Institute for Research (infrastructure, web application, local instances, testing)
  - University of Illinois Urbana-Champaign-NCSA (identity management, testing)
  - University of Wisconsin-Madison (framework: tools, languages, platforms)
  - Indiana University (cybersecurity, infrastructure monitoring, 24/7 support)
- Secure and dependable facility hosted at the Morgridge Institute for Research
- Principal Investigators:
  - Jim Basney
  - Miron Livny
  - Bart Miller
  - Von Welch

Welcome to the SWAMP

- Support for 5 languages: C/C++, Java source, Java bytecode, Python, Ruby
- Support for 10 platforms: 9 varieties of Linux plus Android
- 19 static software analysis tools are available for public use:
  - C/C++: Clang Static Analyzer, CppCheck, GCC warnings, Parasoft C/C++test
  - Java: Checkstyle, error-prone, FindBugs with Find Security Bugs, Parasoft Jtest, PMD
  - Python: Bandit, Flake8, Pylint
  - Android: Android Lint, Revealdroid
  - Ruby: Reek, RuboCop, ruby-lint, Brakeman, Dawnscanner
- Working with 4 commercial tool vendors to add their tools:
  - C/C++test and Jtest (Parasoft) are available now
  - Code Sonar (GrammaTech), Goanna (Red Lizard), and SAST (Veracode)
  - 550+ software packages are available for public use:
  - NIST Juliet and SATE test suites for C/C++ and Java
- Supported platforms, tools, and packages are maintained by the SWAMP

Welcome to the SWAMP (continued)

- The fully-integrated results viewer, CodeDx (Secure Decisions), consolidates and prioritizes vulnerabilities from multiple tools to significantly simplify remediation
- Support for GitHub identities, uploading packages from repositories, and pulling packages from public repositories
- Powerful high-throughput computing capabilities
- Scheduling feature for automated continuous software assurance
- Maintain confidentiality of software and results at the discretion of the user
  - Managed sharing of tools, software packages, and results
- Audience:
  - Software Developers
  - Software Assurance Tool Developers
  - Software Assurance Tool Researchers
  - Infrastructure Operators
  - Educators and Students

Continuous Assurance:

Do it Early and Do it Often

Key Attributes

- Highly automated
  - If you can compile your tool in the SWAMP, all else is automated.
- Secure
  - Strong sandboxing: all executions in single-use virtual machines
- Private (if you wish)
  - Share your tool, app, or data if and when you choose.
- Open
  - Lots of tools, lots of apps, lots of assessment data
- A resource
  - Software to help make your job easier; people to advise you
- A community
  - Join and leverage other like-minded users online and in person.

What SWAMP Can Do for Developers

- Automates building packages on SWAMP platforms
- Automates assessing software packages in C/C++, Java, Python, or Ruby with tools in the SWAMP
- Free access to commercial tools for open-source developers, students, & educators
- Analyzes Results
  - View weakness results
  - View integrated multi-tool results from the same version of a package
  - Compare results between package versions
  - Inter-tool result viewing
- Protects privacy of results
- Support third-party assessments, so SWAMP can provide assurance evidence to acquirer

Run the Tools Early, Run Them Often

Built in assurance from day one, or the task becomes overwhelming for the programmer :

```
dthread.h: In constructor 'ScopeLock::ScopeLock(Mutex&)':
dthread.h:132: warning: unused variable 'result'
dthread.h: In constructor 'ScopeLock::ScopeLock(CondVar&)':
dthread.h:140: warning: unused variable 'result'
src/irpc.C: In member function 'void
int_iRPC::setState(int_iRPC::State)':
src/irpc.C:118: warning: unused variable 'old_state'
src/irpc.C:119: warning: unused variable 'new_state'
src/irpc.C: In member function 'bool int_iRPC::saveRPCState()':
src/irpc.C:714: warning: unused variable 'result'
src/irpc.C:723: warning: unused variable 'result'
src/irpc.C:736: warning: unused variable 'result'
src/irpc.C:1030: warning: unused variable 'result'
src/irpc.C:1041: warning: unused variable 'result'
src/irpc.C:1081: warning: unused variable 'result'
dyninst/proccontrol/src/response.h:35,
dyninst/proccontrol/src/int_process.h:39,
dyninst/proccontrol/src/mailbox.C:33:
dthread.h: In constructor 'ScopeLock::ScopeLock(Mutex&)':
dthread.h:132: warning: unused variable 'result'
dthread.h: In constructor 'ScopeLock::ScopeLock(CondVar&)':
dthread.h:140: warning: unused variable 'result'
```

Continuous Assurance:

Do it Early and Do it Often

Open & Open-Source

- Managed as an open-source project
- All software is developed under Apache license
- No-cost software assurance resource
- Integration of open-source software analysis tools and platforms
- Active interaction with the community to identify trends, promote adoption, and collect feedback
- User needs and input drive SWAMP development!

- **SWAMP-in-a-Box, an on-premise solution, will be available on GitHub starting September 27th, 2016. Search for mirswamp on GitHub and look for the gear** (as seen below)**!**

Long Term Vision

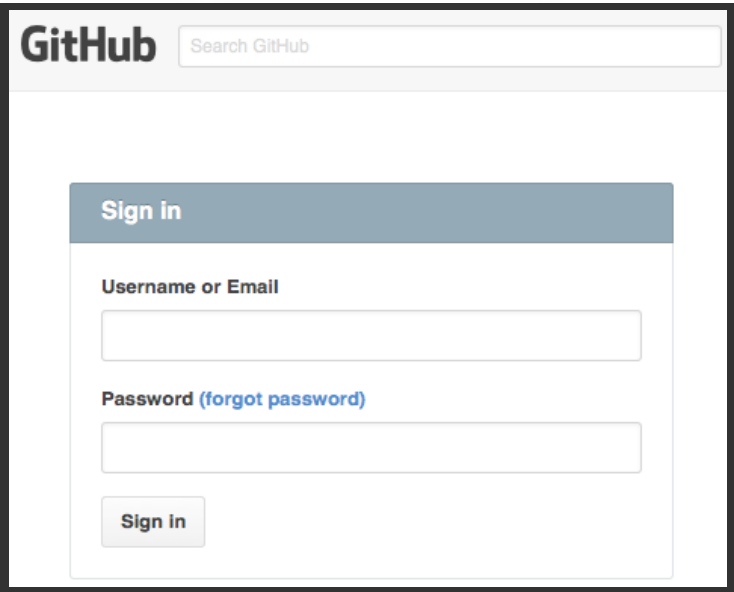
- SWAMP API
- Support for iOS, MacOSX, and Windows platforms
- Plug-ins for IDEs: Eclipse, IntelliJ IDEA, BlueJ
- Support for a large number of programming languages
- Integration with multiple code repositories
- Automated scheduled software analysis of the latest code version in a public or private repository

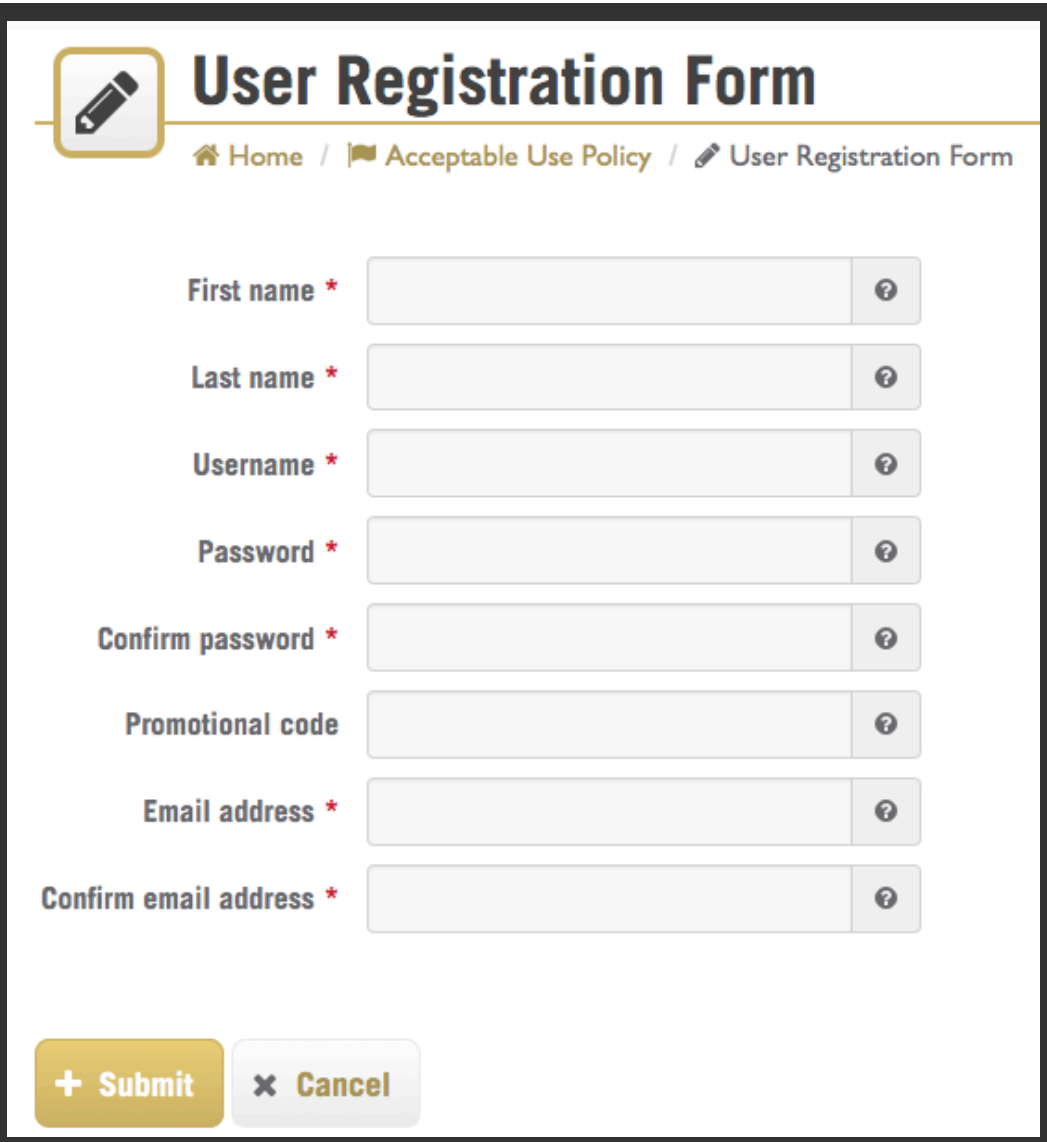
Continuous Assurance:

Do it Early and Do it Often

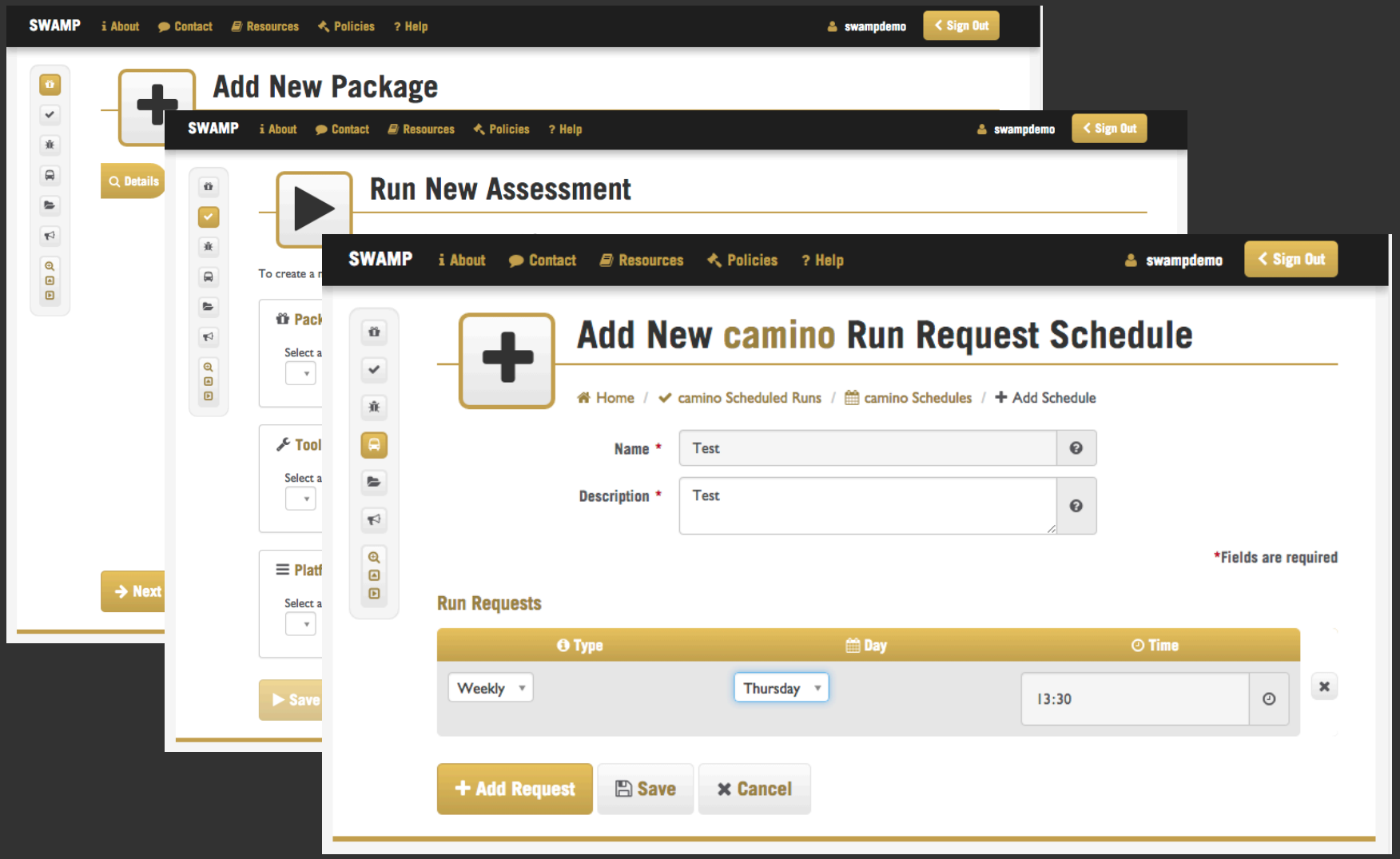
Tour


Welcome to the SWAMP
The Software Assurance Marketplace (SWAMP) is a service that provides continuous software assurance capabilities to developers and researchers.
This no-cost code analysis service is open to the public. Let the SWAMP help you to build better, safer, and more secure code today!
Sign Up!
Sign Up with GitHub!

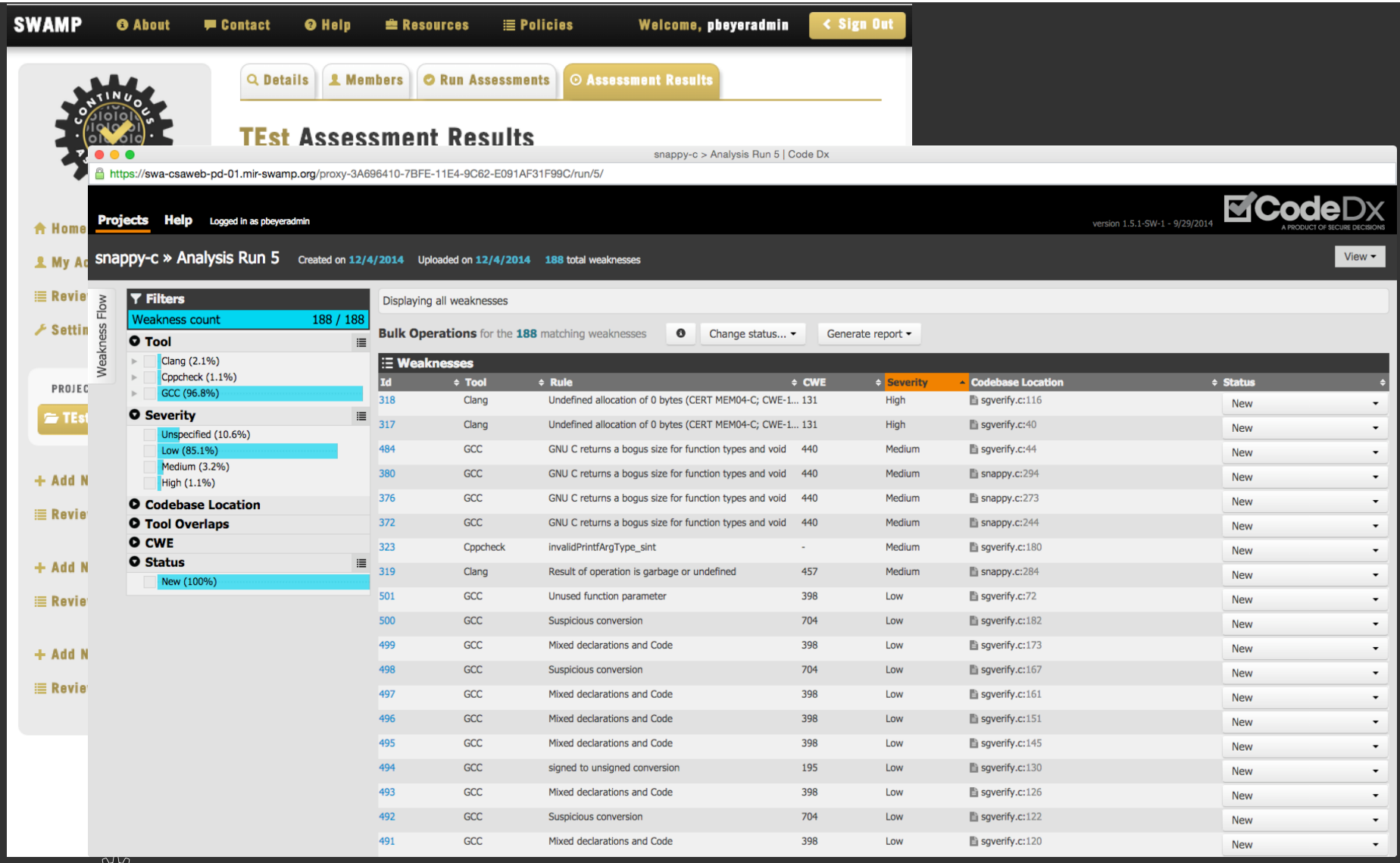

GitHub
Search GitHub
Sign in
Username or Email
Password (forgot password)
Sign in


User Registration Form
Home / Acceptable Use Policy / User Registration Form
First name *
Last name *
Username *
Password *
Confirm password *
Promotional code
Email address *
Confirm email address *
Submit
Cancel

Tour


SWAMP
About
Contact
Resources
Policies
Help
swampdemo
Sign Out
Add New Package
Details
Next
Run New Assessment
Save
Add New camino Run Request Schedule
Home / camino Scheduled Runs / camino Schedules / Add Schedule
Name *
Test
Description *
Test
*Fields are required
Run Requests
Type
Day
Time
Weekly
Thursday
13:30
Add Request
Save
Cancel

Tour


SWAMP
About
Contact
Help
Resources
Policies
Welcome, pbeyeradmin
Sign Out
Details
Members
Run Assessments
Assessment Results
TEst Assessment Results
snappy-c > Analysis Run 5 | Code Dx
https://swa-csaweb-pd-01.mir-swamp.org/proxy-3A696410-7BFE-11E4-9C62-E091AF31F99C/run/5/
Projects
Help
Logged in as pbeyeradmin
version 1.5.1-SW-1 - 9/29/2014
CodeDx
A PRODUCT OF SECURE DECISIONS
snappy-c » Analysis Run 5
Created on 12/4/2014
Uploaded on 12/4/2014
188 total weaknesses
View
Weakness Flow
Filters
Weakness count 188 / 188
Tool
Clang (2.1%)
Cppcheck (1.1%)
GCC (96.8%)
Severity
Unspecified (10.6%)
Low (85.1%)
Medium (3.2%)
High (1.1%)
Codebase Location
Tool Overlaps
CWE
Status
New (100%)
Displaying all weaknesses
Bulk Operations for the 188 matching weaknesses
Change status...
Generate report
Weaknesses
Id
Tool
Rule
CWE
Severity
Codebase Location
Status
318 Clang Undefined allocation of 0 bytes (CERT MEM04-C; CWE-1... 131 High sgverify.c:116 New
317 Clang Undefined allocation of 0 bytes (CERT MEM04-C; CWE-1... 131 High sgverify.c:40 New
484 GCC GNU C returns a bogus size for function types and void 440 Medium sgverify.c:44 New
380 GCC GNU C returns a bogus size for function types and void 440 Medium snappy.c:294 New
376 GCC GNU C returns a bogus size for function types and void 440 Medium snappy.c:273 New
372 GCC GNU C returns a bogus size for function types and void 440 Medium snappy.c:244 New
323 Cppcheck invalidPrintfArgType_sint - Medium sgverify.c:180 New
319 Clang Result of operation is garbage or undefined 457 Medium snappy.c:284 New
501 GCC Unused function parameter 398 Low sgverify.c:72 New
500 GCC Suspicious conversion 704 Low sgverify.c:182 New
499 GCC Mixed declarations and Code 398 Low sgverify.c:173 New
498 GCC Suspicious conversion 704 Low sgverify.c:167 New
497 GCC Mixed declarations and Code 398 Low sgverify.c:161 New
496 GCC Mixed declarations and Code 398 Low sgverify.c:151 New
495 GCC Mixed declarations and Code 398 Low sgverify.c:145 New
494 GCC signed to unsigned conversion 195 Low sgverify.c:130 New
493 GCC Mixed declarations and Code 398 Low sgverify.c:126 New
492 GCC Suspicious conversion 704 Low sgverify.c:122 New
491 GCC Mixed declarations and Code 398 Low sgverify.c:120 New

Contacts

Join the SWAMP at https://www.mir-swamp.org !

- Chief Technology Officer: Miron Livny
  - miron@cs.wisc.edu
- Chief Scientist: Bart Miller
  - bart@cs.wisc.edu
- Operations Officer: Rob Quick
  - rquick@iu.edu
- Project Manager: Irene Landrum
  - ilandrum@continuousassurance.org
- General
  - swamp@continuousassurance.org

SWAMP
SOFTWARE ASSURANCE MARKETPLACE

Continuous Assurance:

Do it Early and Do it Often

SWAMP
SOFTWARE ASSURANCE MARKETPLACE

Questions?

- FAQs

Continuous Assurance:

Do it Early and Do it Often

SWAMP
SOFTWARE ASSURANCE MARKETPLACE