A Guide for Software Assurance for SWIP

August 30, 2019

Randy Heiland (heiland@iu.edu), Mats Rynge (rynge@isi.edu), Karan Vahi (vahi@isi.edu),
Ewa Deelman (deelman@isi.edu), Von Welch (vwelch@iu.edu)

Table of Contents

Abstract

The Scientific Workflow Integrity with Pegasus (SWIP) project adds data integrity checking to the Pegasus workflow management system (https://pegasus.isi.edu/). As part of SWIP, we performed software assurance (SwA) on the Pegasus software using the Software Assurance Marketplace (SWAMP, https://www.mir-swamp.org/). Initially, we planned to perform SwA only on the parts of the code base related to SWIP, i.e., only the code related to the data integrity checks. However, during the course of the SWIP project, a decision was made to perform SwA on the entire Pegasus code base. In addition, the project took on a research effort of trying to quantify differences in SwA results between Pegasus versions. We summarize our SwA process and results here. SwA results provide insight, but they are still subjective; developers of the software being assessed (Pegasus in this project) need to determine how those results need to be addressed.

Introduction

The Pegasus workflow management system (https://pegasus.isi.edu/) is software that supports scientific workflows. It lets users configure and execute scientific workflows over a wide range of computational and storage resources. During this project, Pegasus (https://github.com/pegasus-isi/pegasus) incorporated continuous integration (via Bamboo), but did not perform any software assurance (SwA) on the code. The primary goal of the SWIP project is to ensure data integrity in Pegasus workflows. However, a secondary goal is to perform SwA on the Pegasus code, as the data integrity functionality was being added. The Software Assurance Marketplace (SWAMP, https://www.mir-swamp.org/) has been used to perform SwA on Pegasus and will be described in this report. A User Guide for SWAMP itself can be found at https://www.swampinabox.org/doc/SWAMP-User-Manual.pdf (version 2016-3-15 for this report).

Pegasus consists of software in multiple programming languages, e.g., Java, Python, C, C++, Bash, etc.. SWAMP assesses software “packages” in just one programming language at a time. Therefore, for this project, we manually separated Pegasus code into chunks of common languages and had SWAMP analyze each separately.

SWAMP is a Web application, i.e., one uploads software, uses its Web interface to perform SwA, and either views the results in the browser (using *Code Dx* or a *Native* viewer, described in the SWAMP User Guide) or downloads results for further processing. Assessment results from SWAMP can be downloaded as SCARF (SWAMP Common Assessment Result Format) files, which are XML-formatted files. SWAMP provides multiple assessment tools for each programming language and each tool can produce (dramatically) different results for the same code.

One of our research goals with using SWAMP for SWIP was to track changes in software assurance results across versions of Pegasus. At the start of the SWIP project (around Oct 2016), Pegasus was at version 4.7.0, therefore we treated this as a baseline version for our software assurance analysis. Pegasus was at version 4.9.2 at the end of the SWIP project.

A partial log of activity related to this SwA project can be found at https://github.com/IU-CACR/SWIP/issues/8 and some analysis scripts and example assessment files from SWAMP can be found at https://github.com/IU-CACR/SWIP/tree/master/static_analysis.

Pegasus

Pegasus's releases on GitHub can be found here: https://github.com/pegasus-isi/pegasus/releases. In order to use SWAMP effectively, it is necessary to separate out the entire code base into chunks of code in common languages. The reason for this is due to the fact that SWAMP performs static analysis on code in a single programming language at a time. Pegasus contains Java, Python, C, C++, Bash shell, and others. To get a breakdown in the files and languages used, we execute a script that counts lines of code:

```
    2196 text files.
    1993 unique files.
     607 files ignored.

http://cloc.sourceforge.net v 1.62  T=18.10 s (84.6 files/s, 36999.3 lines/s)
-------------------------------------------------------------------------------
Language                     files          blank        comment           code
-------------------------------------------------------------------------------
XML                             42           5710             83         317893
Java                           800          31051          97520         101807
Python                         155           7936          10846          27581
C                               32           1660           1932           7563
Bourne Again Shell             157           2075            977           7329
HTML                            63            552             94           7256
XSD                             21              4             13           6294
C++                             26           1097            660           5772
Perl                            36           2187           3032           5038
Bourne Shell                    64            746            638           2810
R                               40            393            954           2264
C/C++ Header                    44            357            518           1941
Javascript                       4            117             37           1052
CSS                              4             71             31            901
PHP                              7             85             21            683
make                            12            173             57            584
Ant                              1             68             21            502
JSON                            10              1              0            318
XSLT                             3             36             11            210
ASP.Net                         10             48              0            127
Windows Resource File            1              0              0              2
-------------------------------------------------------------------------------
SUM:                          1532          54367         117445         497927
-------------------------------------------------------------------------------
```

```
    2369 text files.
    2119 unique files.
     679 files ignored.

http://cloc.sourceforge.net v 1.62  T=19.79 s (81.5 files/s, 35155.4 lines/s)
-------------------------------------------------------------------------------
Language                     files          blank        comment           code
-------------------------------------------------------------------------------
XML                             49           6277            105         320134
Java                           810          32391         100718         106977
Python                         182           8767          11952          32272
Bourne Again Shell             177           2306           1074           7883
C                               33           1678           1964           7624
HTML                            63            552             94           7268
XSD                             22              4             15           6576
C++                             26           1097            661           5774
Perl                            36           2188           3032           5042
YAML                             1             90             29           4114
Bourne Shell                    74            940            764           3468
R                               40            393            954           2265
C/C++ Header                    45            362            533           1949
Javascript                       4            134             37           1137
CSS                              4             71             31            901
PHP                              7             85             21            683
make                            12            174             58            599
Ant                              1             74             23            545
JSON                            13              3              0            380
XSLT                             3             36             11            210
ASP.Net                         10             48              0            127
Windows Resource File            1              0              0              2
-------------------------------------------------------------------------------
SUM:                          1613          57670         122076         515930
-------------------------------------------------------------------------------
```

Figure 1. Breakdown of Pegasus code: version 4.7.0 (left) and 4.9.2 (right)

We focus our SwA on three of the most relevant programming languages: Java, Python, and C. In spite of the fact that the number of Java files is far greater than those in other languages, we point out that they had a small role in the SWIP project. Most of the data integrity functionality was handled by the Python scripts. Because of this, we focused on the Pegasus Python code for much of the SwA project.

SWAMP

The Software Assurance Marketplace (SWAMP) is an online, free service to perform software assurance. It allows one to:

- upload [compressed] archives of software (or provide a link to a public Git repository) to SWAMP "Packages",
- select one or more of the available SwA tools to use for the analysis - depending on the programming language,
- either download the results of the analysis in a SCARF (SWAMP Common Assessment Result Format) format or view the results in the Web browser (using the *Code Dx* or *Native* viewer).

It is worth highlighting that each SwA tool returns SCARF results in a common XML format, however, the *results* of each SwA tool are not converted to a common format. In other words, the unique severity codes from each SwA tool are retained. For example, comparing just two tools (to assess Python code), Flake8 and Bandit, we see the equivalent severity represented with different keywords:

```
<BugSeverity>Fatal</BugSeverity>
<BugSeverity>HIGH</BugSeverity>
```

Moreover, the XML elements themselves can vary between results from different programming languages being assessed. For example, a simple test on some C code shows that instead of a <BugSeverity> element, the assessment tools used <BugGroup>:

```
$ grep -i buggroup scarf*.xml
scarf-clang-sa.xml:    <BugGroup>Logic error</BugGroup>
scarf-cppcheck.xml:    <BugGroup>style</BugGroup>
scarf-cppcheck.xml:    <BugGroup>error</BugGroup>
scarf-gcc.xml:    <BugGroup>warning</BugGroup>
scarf-gcc.xml:    <BugGroup>warning</BugGroup>
scarf-gcc.xml:    <BugGroup>warning</BugGroup>
scarf-gcc.xml:    <BugGroup>warning</BugGroup>
```

To try to provide common output of the SCARF results, we wrote a script (`parseSCARF.py`) that maps the unique codes from each tool into High, Medium, and Low categories, and also prints the actual lines (and line #s) in the code that are being flagged, along with the description of the vulnerability. It is possible to specify which categories are desired when one runs the script. We show results from the `parseSCARF` script below. (We also wrote a separate script, `parseSCARF_c.py`, that is specific to C code. These can be found in the SWIP static_analysis github repository).

A User Manual and other useful information for learning how to use the SWAMP can be found at https://www.mir-swamp.org/#help. We provide screenshots related to this project here and, in much more detail, in the Appendix.

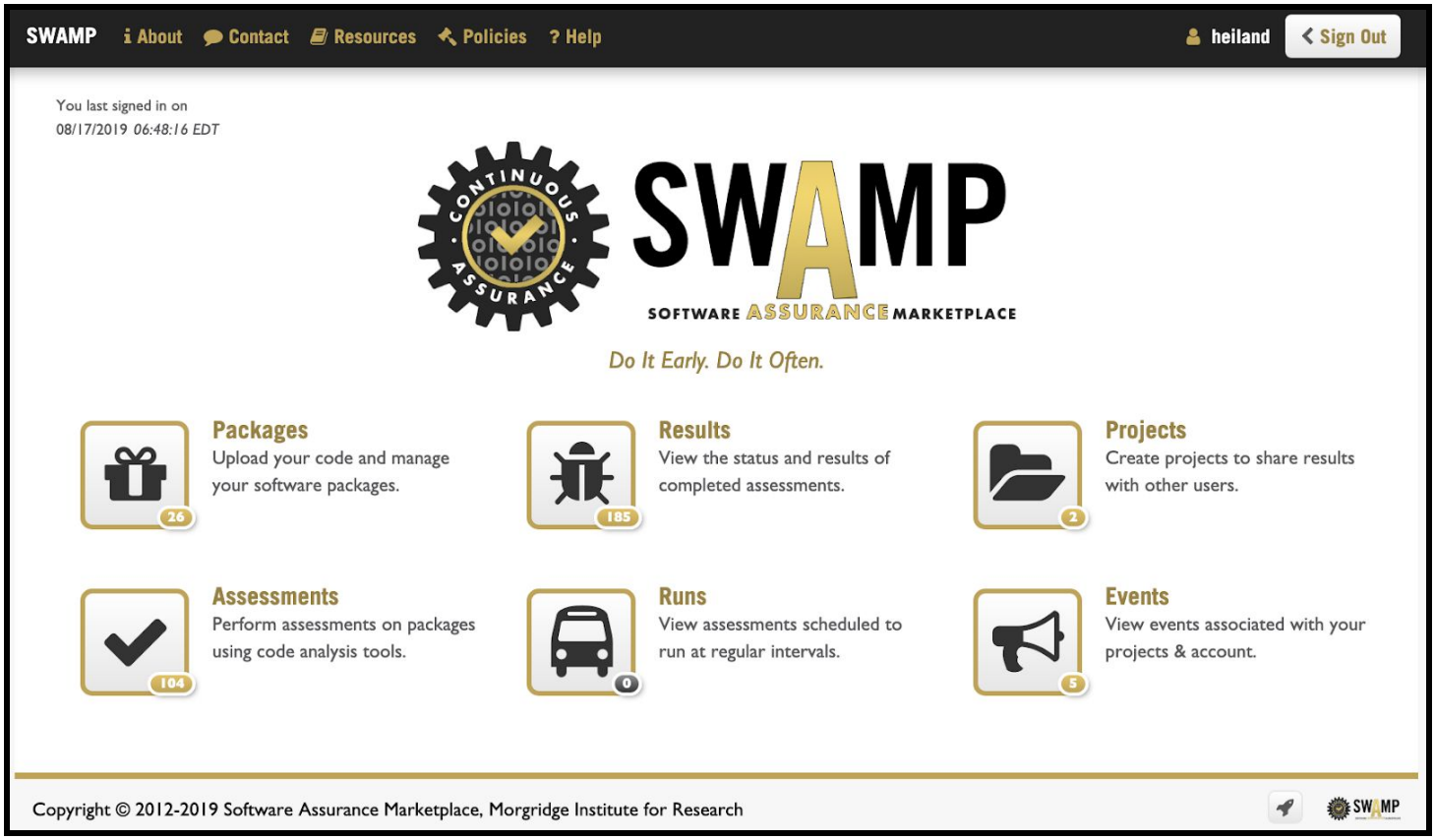


Figure 2. A summary screen of a SWAMP account is shown after signing in.

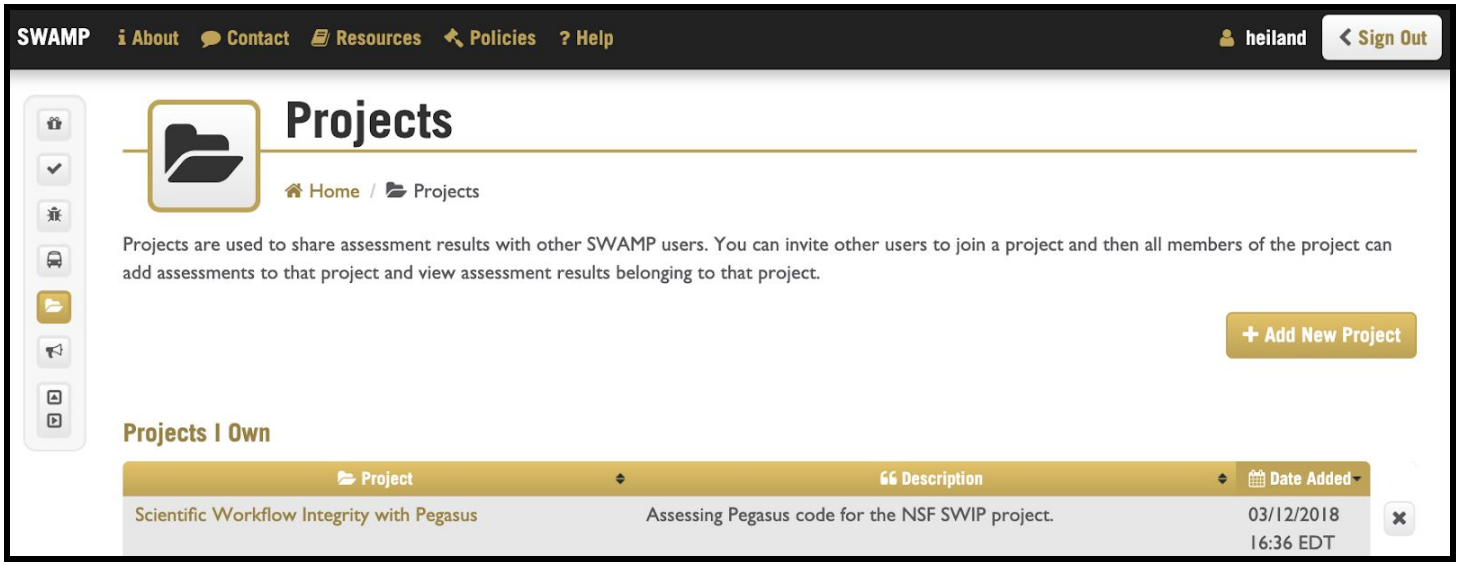


Figure 3. Assessments can be organized into Projects.

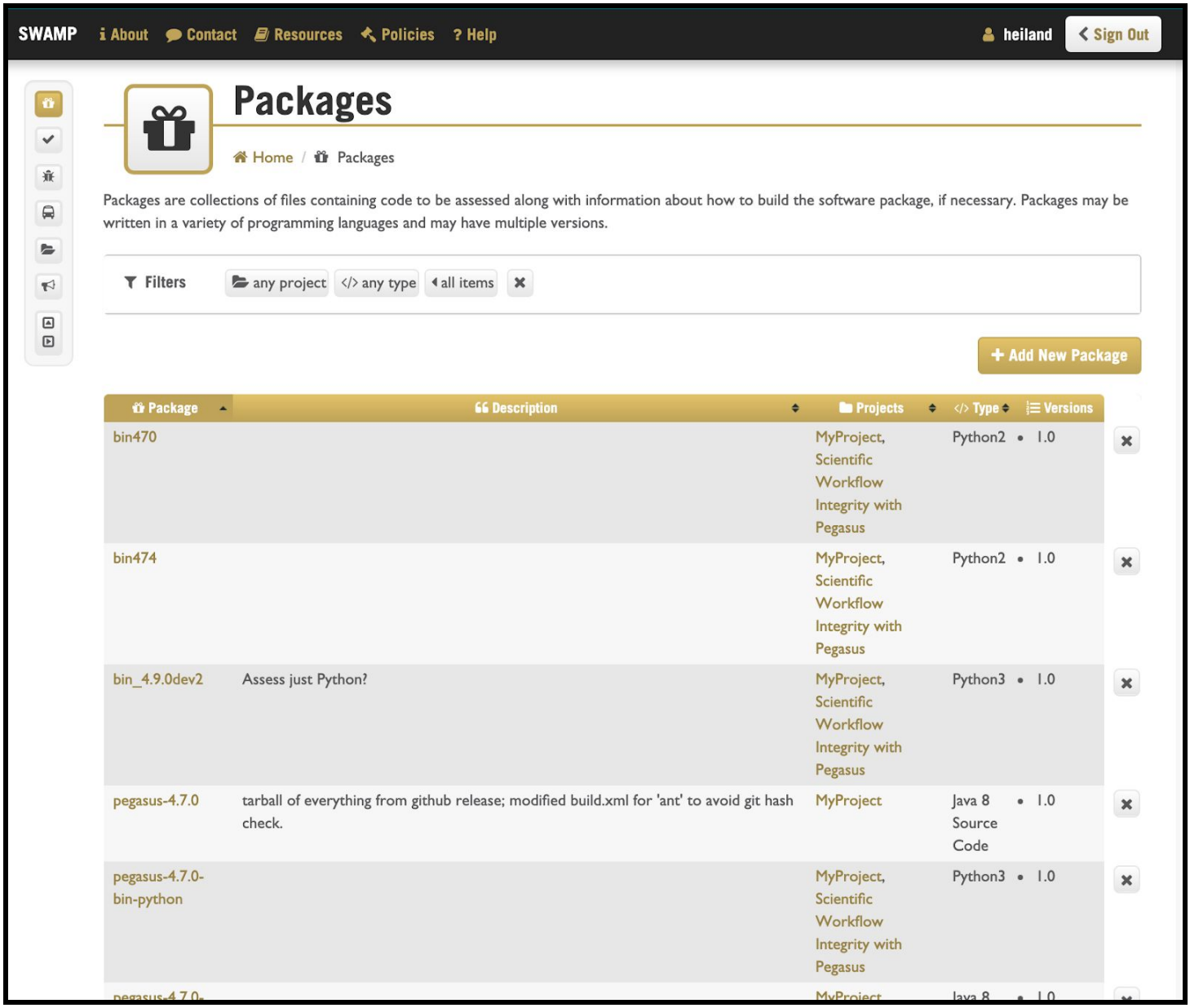


Figure 4. A user uploads code to be assessed into Packages.

The following screenshot shows SWAMP's Web UI after assessing (with 3 different tools) Pegasus's core Python scripts in a package called "pegasus4.9.0-python". Note that we had to pre-process (rename) the scripts to append the ".py" suffixes so that SWAMP would recognize them as being Python code.

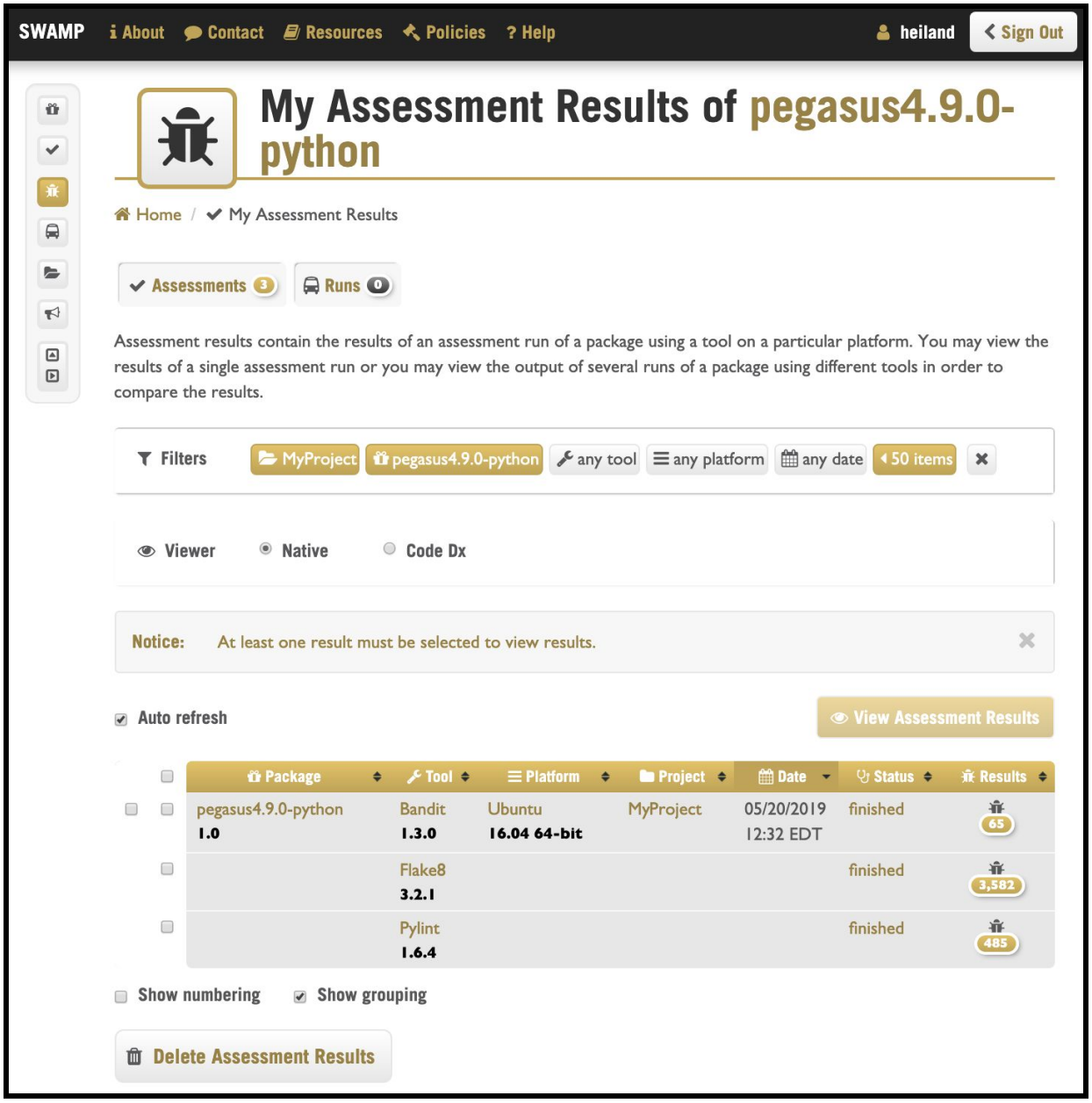


Figure 5. SWAMP Web UI after an assessment has completed.

An example email that a user can, optionally, request be sent after a SWAMP assessment:

> Dear Randy Heiland,
>
> Your assessment of pegasus4.9.0-python version 1.0 using Bandit version 1.3.0 on Ubuntu version 16.04 64-bit completed at 2019-05-20 16:35:44 with a status of 'Finished'.
>
> If you have any questions please contact the SWAMP staff at: support@continuousassurance.org .
>
> -The Software Assurance Marketplace (SWAMP)

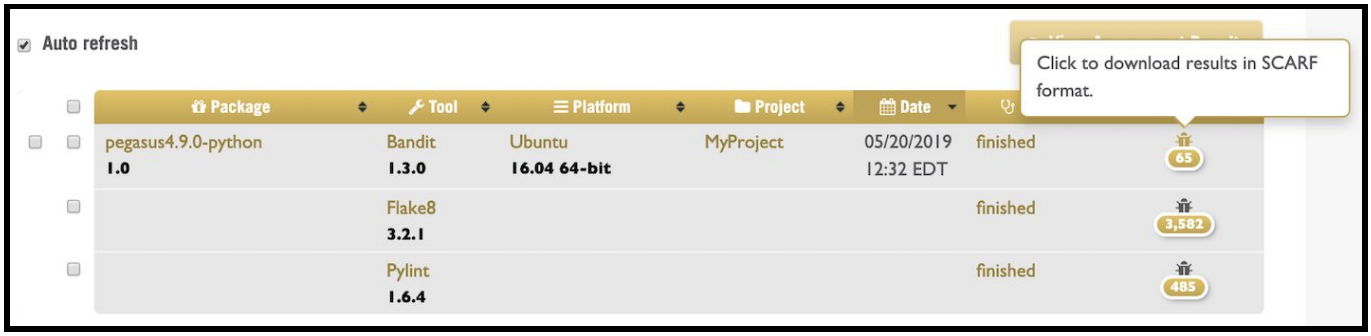


Figure 6. Showing how a user can download results in a SCARF file from the Web UI.

SWAMP results

As we introduced above, we have written `parseSCARF.py`, that parses SWAMP analysis results in a SCARF file and sorts results (vulnerabilities) into high/medium/low priority. It take as arguments:

```
<name of SCARF file> <flag for high-priority vulns> <flag for medium-priority vulns> <flag for low-priority vulns>
```

The goal was to have the script process *any* SCARF file, i.e., a SCARF file generated by any tool, for any language. While it has improved considerably during this project, we cannot claim that it can process any SCARF file; in fact, we wrote a separate script `parseSCARF_c.py` to parse SCARF files from assessment tools for C code. We show results for some of the SCARF files of interest to us for SWIP.

Python code

Bandit is one of multiple tools provided by SWAMP for assessing Python code. Recall that much of the SWIP-specific code changes to Pegasus took place in the Python scripts. Here, we show how one would use parseSCARF (on Pegasus 4.9.0) to obtain results:

```
$ python parseSCARF.py scarf_bandit.xml 1 0 0
flags =  1 1 1

---- scarf_bandit.xml:
AnalyzerReport
{'uuid': '4860ca25-b51a-4dd4-9b4d-6ac8769845ee', 'tool_version': '1.3.0', 'parser_fw': 'resultparser',
'build_root_dir': '/home/builder/build', 'package_version': '1.0', 'tool_name': 'bandit',
'assessment_start_ts': '1558370062.5608888', 'platform_name': 'ubuntu-16.04-64', 'package_name':
'pegasus4.9.0-python', 'parser_fw_version': '3.1.8', 'package_root_dir': 'pkg1'}

================== High priority
Line 48   in  pkg1/pegasus-analyzer.py  ==>  subprocess call with shell=True identified, security issue.
Line 50   in  pkg1/pegasus-analyzer.py  ==>  subprocess call with shell=True identified, security issue.
Line 684  in  pkg1/pegasus-analyzer.py  ==>  Starting a process with a shell, possible injection detected,
security issue.
...
```

Two other Python-related assessments tools in SWAMP are Flake8 and Pylint. We summarize the results obtained from parseSCARF from each of these tools for different versions of Pegasus. You will see that different assessment tools (for the very same code) generate very different outcomes, including wildly different numbers of potential vulnerabilities. For example, Bandit seems to be quite conservative about reporting vulnerabilities; Flake8 seems to play it safe and, perhaps, over-report. But a developer would need to examine an assessment report in more detail to determine what, in their opinion, needs attention.

Pegasus 4.7.0

Bandit assessment results for Pegasus 4.7.0:
----- Got 32 high priority vulnerabilities.
----- Got 9 medium priority vulnerabilities.
----- Got 44 low priority vulnerabilities.

Flake8 assessment results for Pegasus 4.7.0:
----- Got 92 high priority vulnerabilities.
----- Got 2296 medium priority vulnerabilities.
----- Got 998 low priority vulnerabilities.

Pylint assessment results for Pegasus 4.7.0:
----- Got 0 high priority vulnerabilities.
----- Got 86 medium priority vulnerabilities.
----- Got 1038 low priority vulnerabilities.

Pegasus 4.8.0

Bandit assessment results for Pegasus 4.8.0:
----- Got 9 high priority vulnerabilities.
----- Got 0 medium priority vulnerabilities.
----- Got 8 low priority vulnerabilities.

Flake8 assessment results for Pegasus 4.8.0:
----- Got 0 high priority vulnerabilities.
----- Got 2337 medium priority vulnerabilities.
----- Got 1055 low priority vulnerabilities.

Pylint assessment results for Pegasus 4.8.0:
----- Got 0 high priority vulnerabilities.
----- Got 17 medium priority vulnerabilities.
----- Got 0 low priority vulnerabilities.

Pegasus 4.9.0

Bandit assessment results for Pegasus 4.9.0:
----- Got 28 high priority vulnerabilities.
----- Got 6 medium priority vulnerabilities.

----- Got 31 low priority vulnerabilities.

Flake8 assessment results for Pegasus 4.9.0:
----- Got 41 high priority vulnerabilities.
----- Got 2446 medium priority vulnerabilities.
----- Got 1095 low priority vulnerabilities.

Pylint assessment results for Pegasus 4.9.0:
----- Got 0 high priority vulnerabilities.
----- Got 77 medium priority vulnerabilities.
----- Got 408 low priority vulnerabilities.

Pegasus 4.9.2

Bandit assessment results for Pegasus 4.9.2:
----- Got 28 high priority vulnerabilities.
----- Got 6 medium priority vulnerabilities.
----- Got 31 low priority vulnerabilities.

Flake8 assessment results for Pegasus 4.9.2:
----- Got 41 high priority vulnerabilities.
----- Got 2496 medium priority vulnerabilities.
----- Got 1097 low priority vulnerabilities.

Pylint assessment results for Pegasus 4.9.2:
----- Got 0 high priority vulnerabilities.
----- Got 77 medium priority vulnerabilities.
----- Got 408 low priority vulnerabilities.

By using parseSCARF across different versions of Pegasus, we can provide comparative assessment data. The following histogram plot captures some of the above data, with the different Pegasus versions on the x-axis. We show the number of high priority counts for both Bandit and Flake8, but the medium priority counts for Pylint (since there were no high priority ones).

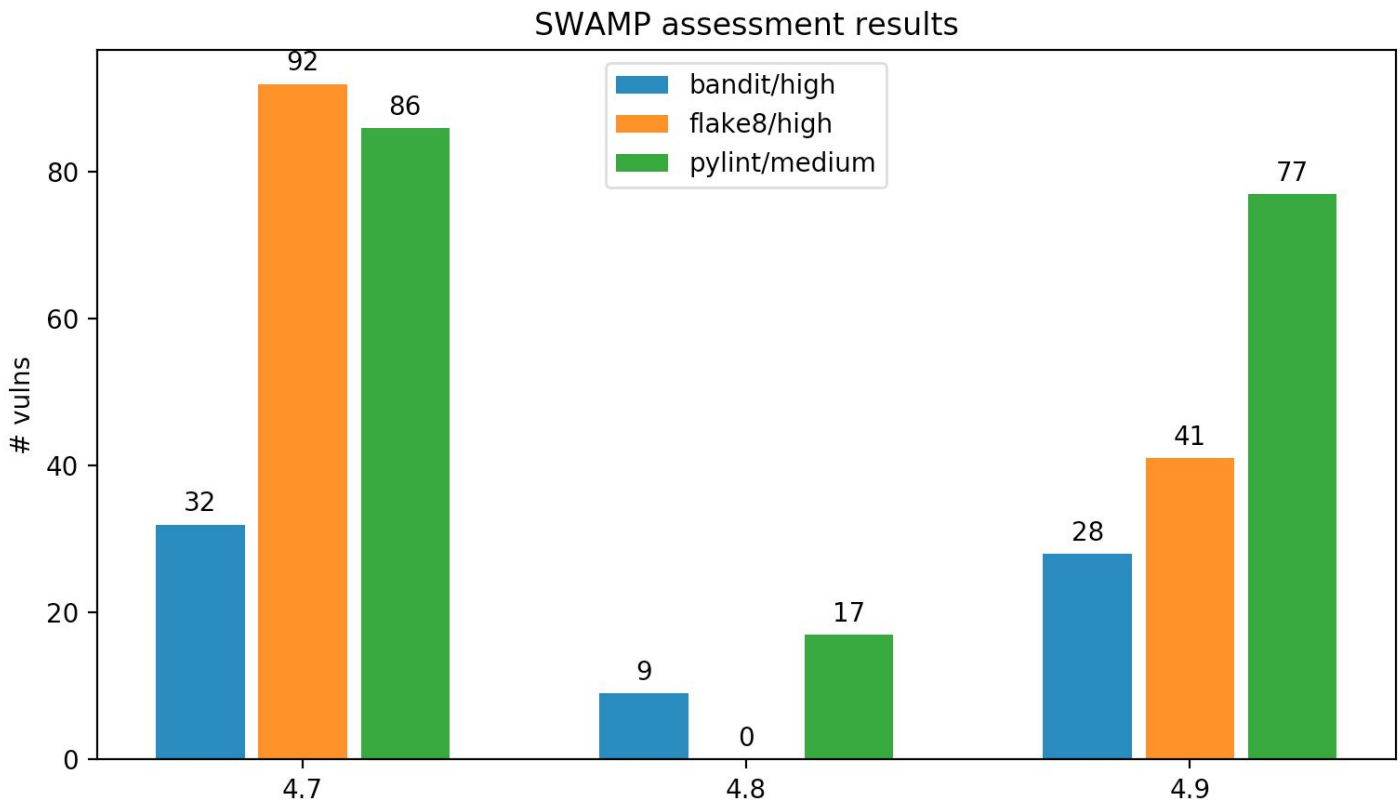


Figure 7. Assessment results for the Python scripts in different Pegasus versions.

One reason the number of potential vulnerabilities increases from 4.8 to 4.9 is that 4.9 contains more Python scripts. Beyond that, a developer would need to take a closer look at the details of the reported vulnerabilities.

Java code

Using SWAMP to assess Pegasus' Java code was more challenging that Python code. The assessment tools for Java packages required that the code be compiled and, specifically for Pegasus, Apache Ant was required. This led to complications which were eventually resolved by consulting with both the SWAMP and Pegasus developers. In the end, we needed to manually edit the build.xml file before a build in SWAMP could be successful. We documented this at https://github.com/IU-CACR/SWIP/issues/8#issuecomment-445864077.

There were five assessment tools available for Java code: OWASP Dependency Check, checkstyle, SpotBugs, PMD, and error-prone. The results from each of these were, like the Python assessment tools, quite different. Our parseSCARF script had problems parsing the SCARF file from the OWASP Dependency Check tool, but we summarize outputs from the other four for multiple versions of Pegasus. We show how one would use parseSCARF (on Pegasus 4.9.2) to obtain sample results:

```
$ python parseSCARF.py scarf-pmd.xml 1 0 0|head
flags =  1 0 0

---- scarf-pmd.xml:
AnalyzerReport
```

```
{'tool_version': '5.8.1', 'parser_fw_version': '3.1.8', 'package_root_dir': 'pkg1', 'uuid':
'a6d33969-dfb2-4ed1-b71a-d85ff05a22a7', 'assess_fw_version': '2.6.12', 'build_root_dir':
'/home/builder/build', 'tool_name': 'pmd', 'parser_fw': 'resultparser', 'package_version': '1.0',
'package_name': 'pegasus-4.9.2-git-release-buildxml-edited', 'platform_name': 'ubuntu-16.04-64', 'assess_fw':
'java-assess', 'assessment_start_ts': '1566950831.1220715'}

================= High priority
Line 257  in  pkg1/src/edu/isi/pegasus/common/util/ProfileParser.java  ==>  Avoid reassigning parameters such
as 'args'
Line 83   in  pkg1/src/edu/isi/pegasus/common/util/Separator.java  ==>  Avoid throwing null pointer
exceptions.
Line 119  in  pkg1/src/edu/isi/pegasus/common/util/Separator.java  ==>  Avoid throwing null pointer
exceptions.
...
```

Pegasus 4.7.0

Checkstyle assessment results for Pegasus 4.7.0:
----- Got 0 high priority vulnerabilities.
----- Got 16184 medium priority vulnerabilities.
----- Got 0 low priority vulnerabilities.

PMD assessment results for Pegasus 4.7.0:
----- Got 884 high priority vulnerabilities.
----- Got 12251 medium priority vulnerabilities.
----- Got 5201 low priority vulnerabilities.

SpotBugs assessment results for Pegasus 4.7.0:
----- Got 4174 high priority vulnerabilities.
----- Got 0 medium priority vulnerabilities.
----- Got 0 low priority vulnerabilities.

Error-prone (2.0.21) assessment results for Pegasus 4.7.0:
----- Got 0 high priority vulnerabilities.
----- Got 7 medium priority vulnerabilities.
----- Got 96 low priority vulnerabilities.

Pegasus 4.8.0

Checkstyle assessment results for Pegasus 4.8.0:
----- Got 0 high priority vulnerabilities.
----- Got 16148 medium priority vulnerabilities.
----- Got 0 low priority vulnerabilities.

PMD assessment results for Pegasus 4.8.0:
----- Got 891 high priority vulnerabilities.
----- Got 12514 medium priority vulnerabilities.
----- Got 5213 low priority vulnerabilities.

SpotBugs assessment results for Pegasus 4.8.0:
----- Got 4290 high priority vulnerabilities.
----- Got 0 medium priority vulnerabilities.
----- Got 0 low priority vulnerabilities.

Error-prone (2.0.21) assessment results for Pegasus 4.8.0:
----- Got 0 high priority vulnerabilities.
----- Got 8 medium priority vulnerabilities.
----- Got 30 low priority vulnerabilities.

Pegasus 4.9.2

Checkstyle assessment results for Pegasus 4.9.2:
----- Got 0 high priority vulnerabilities.
----- Got 17639 medium priority vulnerabilities.
----- Got 0 low priority vulnerabilities.

PMD assessment results for Pegasus 4.9.2:
----- Got 1010 high priority vulnerabilities.
----- Got 13131 medium priority vulnerabilities.
----- Got 5422 low priority vulnerabilities.

SpotBugs assessment results for Pegasus 4.9.2:
----- Got 4419 high priority vulnerabilities.
----- Got 0 medium priority vulnerabilities.
----- Got 0 low priority vulnerabilities.

Error-prone (2.3.1) assessment results for Pegasus 4.9.2:
----- Got 0 high priority vulnerabilities.
----- Got 0 medium priority vulnerabilities.
----- Got 2400 low priority vulnerabilities

While re-running assessments of the Java code on the same version of Pegasus, but after new versions of the assessment tools were installed in SWAMP, we discovered that results from the same assessment tool can also produce very different results, as captured in Figure 8. While this shouldn't come as a surprise, the differences in some of the tools are surprising. Notice, for example, the number of vulnerabilities in different versions of the error-prone tool and, to a lesser extent, the OWASP Dependency Check tool. To be honest, it wasn't until we saw these differences that we realized we should have also been reporting the versions of the tools in this report. The versions are captured in the SCARF files and can be retrieved upon request. However, the primary purpose of this report is to 1) describe the use of SWAMP and 2) share snapshots of assessment results for different versions of Pegasus.

Package	Tool	Platform	Project	Date	Status	Results
pegasus-4.8.0-git-release-buildxml-edited **1.0**	checkstyle **8.20**	Ubuntu **16.04 64-bit**	MyProject	08/26/2019 21:27 EDT	finished	16,902
	error-prone **2.3.1**				finished	2,330
	OWASP Dependency Check **2.1.1**				finished	364
	PMD **5.8.1**				finished	18,618
	SpotBugs **3.1.12**				finished	4,172
pegasus-4.8.0-git-release-buildxml-edited **1.0**	checkstyle **8.0**	Ubuntu **16.04 64-bit**	MyProject	10/29/2018 14:19 EDT	finished	16,148
	error-prone **2.0.21**				finished	38
	OWASP Dependency Check **2.1.1**				finished	465
	PMD **5.8.1**				finished	18,618
	SpotBugs **3.1.0**				finished	4,290

Figure 8. Note that different versions of assessment tools can produce very different results.

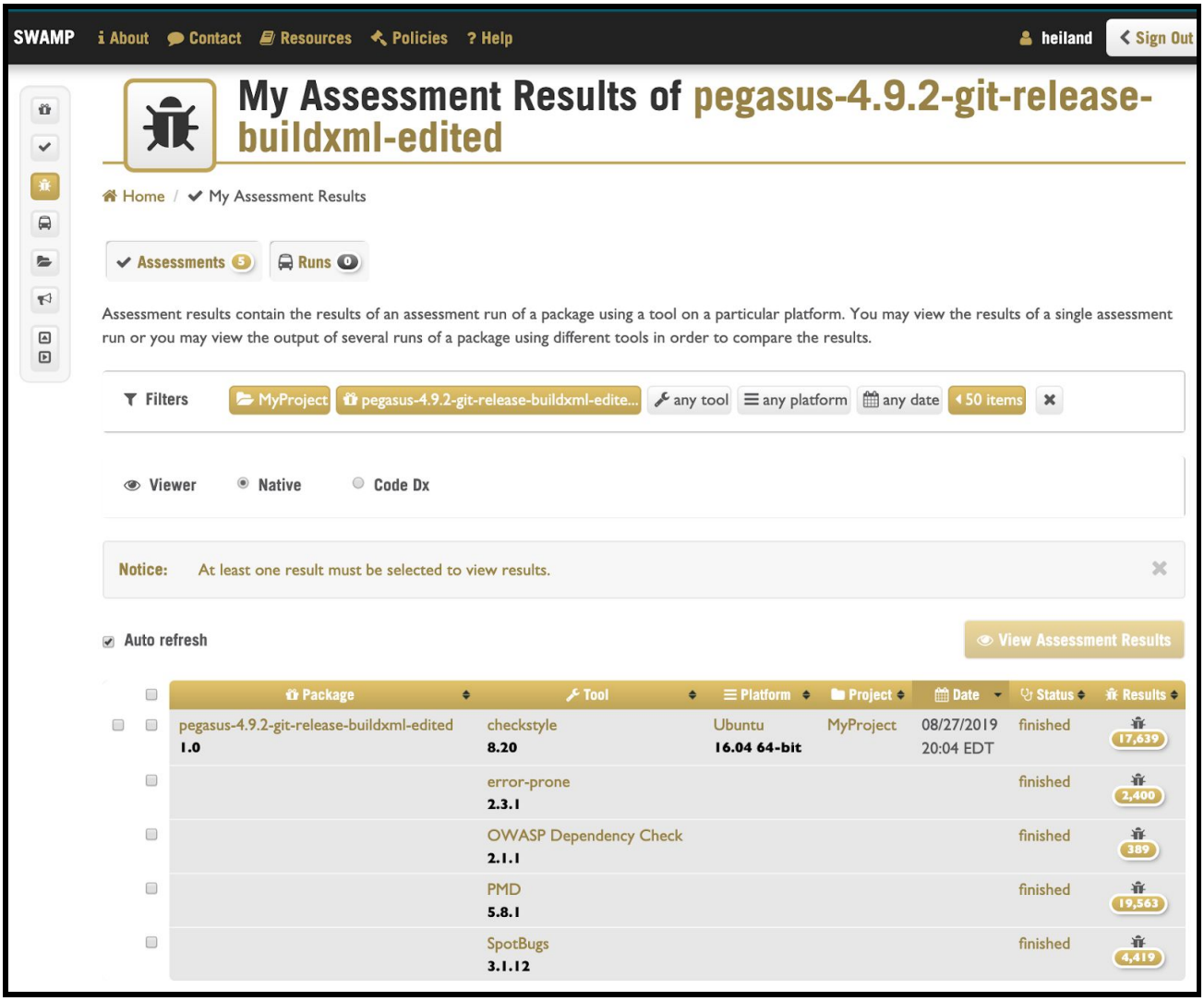
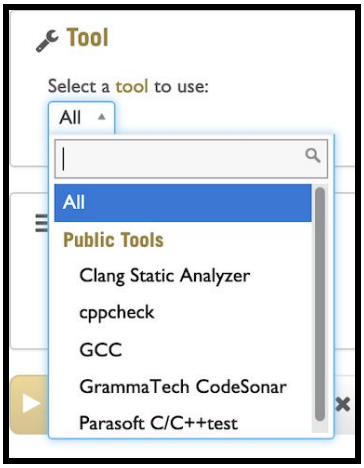


Figure 9. Using newer versions of three Java assessment tools for 4.9.2 code.

C code

To assess C code in Pegasus, we wrote a script to extract all C files and bundled them together to create a Package for SWAMP.

While there seemed to be five assessment tools available for C code:



we only obtained results from three when we performed our assessment, suggesting that perhaps the other two had licensing issues.

Package	Tool	Platform	Project	Date	Status	Results
c_files_4.9.2 1.0	Clang Static Analyzer 3.8	Ubuntu 16.04 64-bit	MyProject	08/27/2019 05:42 EDT	finished	1
	cppcheck 1.75				finished	7
	GCC current				finished	223

Figure 10. Assessment results for C code in Pegasus 4.9.2.

We summarize results from just *cppcheck* for the start (4.7.0) and end (4.9.2) versions of Pegasus for the SWIP project and note that there appear to be no differences.

Pegasus 4.7.0

cppcheck assessment results for Pegasus 4.7.0:
---- Got 3 high priority vulnerabilities.
----- Got 1 medium priority vulnerabilities.
----- Got 3 low priority vulnerabilities.

========== High priority
Line 2090 in pkg1/src_tools_pegasus-kickstart_interpose.c ==> Memory leak: argv
Line 2124 in pkg1/src_tools_pegasus-kickstart_interpose.c ==> Memory leak: argv
Line 2159 in pkg1/src_tools_pegasus-kickstart_interpose.c ==> Memory leak: argv

========== Medium priority
Line 253 in pkg1/src_tools_pegasus-kickstart_interpose.c ==> Non reentrant function 'readdir' called. For threadsafe applications it is recommended to use the reentrant replacement function 'readdir_r'.

Pegasus 4.9.2

cppcheck assessment results for Pegasus 4.9.2:
----- Got 3 high priority vulnerabilities.
----- Got 1 medium priority vulnerabilities.
----- Got 3 low priority vulnerabilities.

Details from cppcheck include:
========== High priority
Line 2090 in pkg1/src_tools_pegasus-kickstart_interpose.c ==> Memory leak: argv
Line 2124 in pkg1/src_tools_pegasus-kickstart_interpose.c ==> Memory leak: argv
Line 2159 in pkg1/src_tools_pegasus-kickstart_interpose.c ==> Memory leak: argv

========== Medium priority
Line 253 in pkg1/src_tools_pegasus-kickstart_interpose.c ==> Non reentrant function 'readdir' called. For threadsafe applications it is recommended to use the reentrant replacement function 'readdir_r'.

SWAMP customized ‘diff’ script

We had discussions with the SWAMP development team about our desire to do comparative assessments across versions of Pegasus and learned of a customized “diff” tool that they had developed to operate on SCARF files. It is a Perl script that attempts to extract meaningful (not just syntax) differences in two different SCARF files (from the same assessment tool operating on different versions of a code base). Upon trying it initially, we discovered that the script listed all entries of the SCARF files if the associated SWAMP packages had different directly paths associated with the files (see the “Naive” use in the Appendix). After consulting with the SWAMP team, we learned of the “`--no-source_file`” argument for the script which led to much better results (see the “Proper use in the Appendix). We only used this script on the Python assessment SCARF files, in part because these were considered more relevant for SWIP, and in part because those SCARF files were not as unwieldy as the Java SCARF files.. The following is a portion of its output when using it on Pegaus 4.7 and 4.8 Python scripts.

```
~/git/SWIP/static_analysis$ diff2 --no-source_file scarf_bandit_pegasus4.7.xml scarf_bandit_pegasus4.8.xml
--- scarf_bandit_pegasus4.7.xml
+++ scarf_bandit_pegasus4.8.xml
- 1 BugInstances categorized by BugGroup: undefined, BugCode: subprocess_popen_with_shell_equals_true,
SourceFile: undefined, StartLine: 898
- 1 BugInstances categorized by BugGroup: undefined, BugCode: subprocess_popen_with_shell_equals_true,
SourceFile: undefined, StartLine: 36
…
- 1 BugInstances categorized by BugGroup: undefined, BugCode: start_process_with_a_shell, SourceFile:
undefined, StartLine: 158
- 1 BugInstances categorized by BugGroup: undefined, BugCode: hardcoded_tmp_directory, SourceFile: undefined,
StartLine: 177
```

SWAMP APIs

In addition to using SWAMP from a browser, as we have mostly done in this report, it is also possible to access SWAMP's functionality from two APIs: a REST API (https://www.mir-swamp.org/api/#api) and a Java-CLI API (https://github.com/mirswamp/java-cli). We did explore the use of these and documented part of our experience at https://github.com/IU-CACR/SWIP/issues/8. However, we got accustomed to using SWAMP from the Web interface early on and that seemed to work well enough for our needs.

Summary

SWAMP provides a free, fairly easy to use service to perform software assurance (SwA), using several (optionally selectable) static analysis tools. We used SWAMP to perform SwA on different versions of Pegasus that were relevant to the SWIP project (starting with Pegasus 4.7.0 and ending with 4.9.2). SWAMP performs SwA on code in only one programming language at a time, therefore we manually extracted Python and C code from Pegasus in order to assess those files. Pegasus itself is primarily Java code and, when uploaded in its entirety to SWAMP, that is what was assessed. This report 1) describes how one would upload code and perform assessments using SWAMP (primarily in the Appendix), and 2) provides some results from the Pegasus SwA. We also attempted to compare SwA results across multiple versions of Pegasus. While SwA tools do indeed provide insight into potential vulnerabilities in code, their results are subjective and need to be scrutinized by the developers of the code being assessed. Finally, as challenging as it is to interpret and compare results from multiple assessment tools (on the same package of code), attempts to compare those results over multiple versions of the code being assessed is an even greater challenge, when the code may be undergoing considerable changes.

Acknowledgements

We thank the National Science Foundation (1642070, 1642090 and 1642053) for funding this research. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation. We also thank the staff at the Software Assurance Marketplace (SWAMP) for their helpful assistance.
Disclosure: Welch is a Co-PI with the SWAMP project.

Appendix: Example SWAMP Assessments

SWAMP assessment of Pegasus Python code

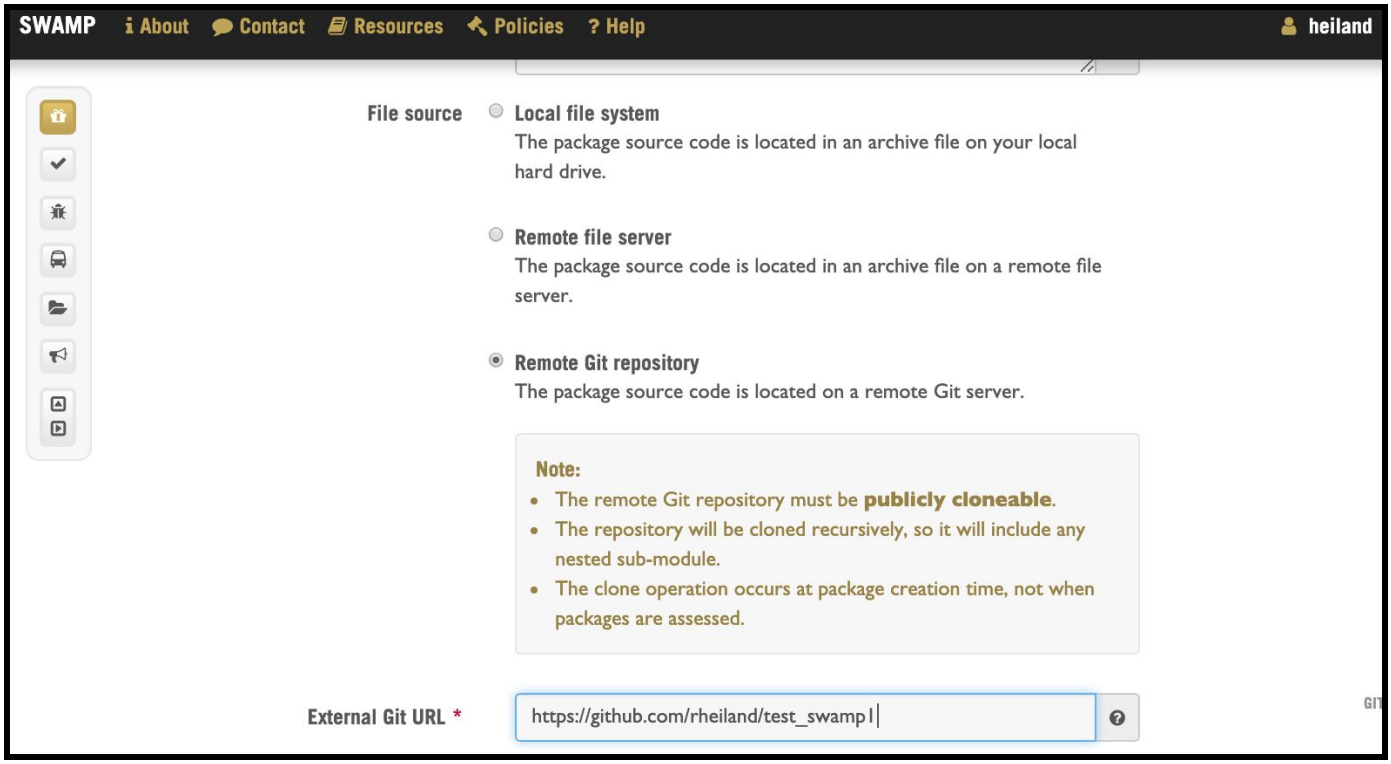


Figure A1. A user has options for uploading code into a Package. Typically, we uploaded from the “local file system”, however, for performing simple tests, we sometimes uploaded from GitHub.

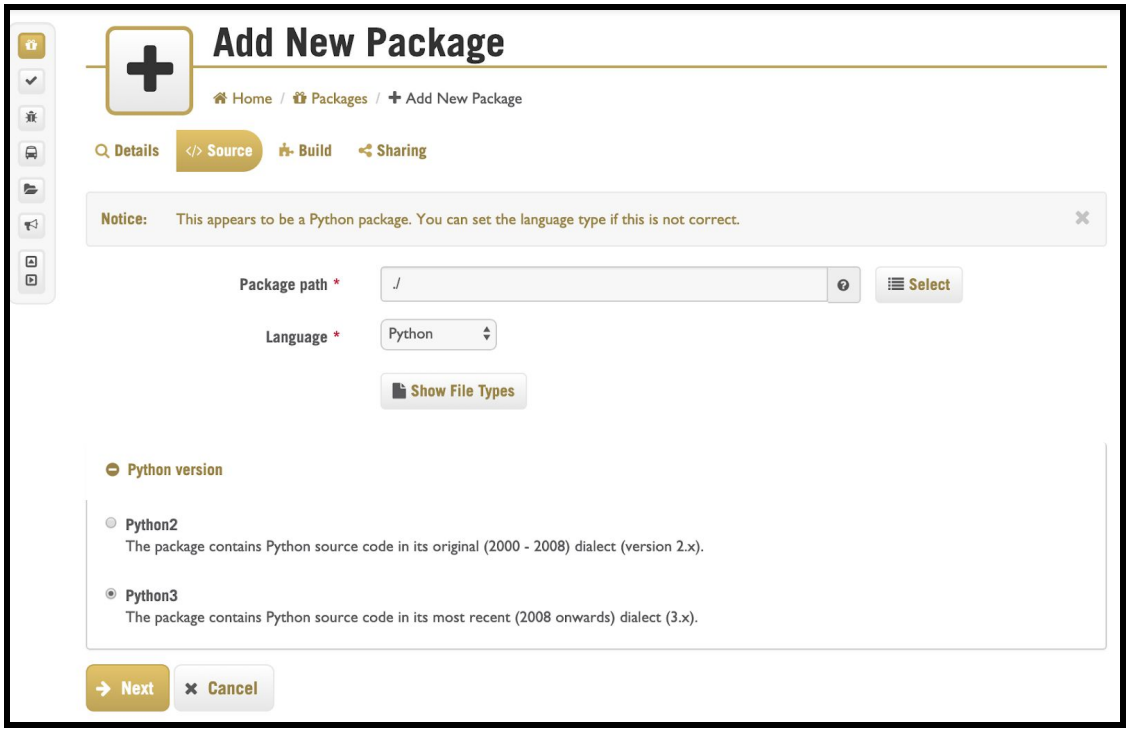


Figure A2. For a Python package, you need to specify if Python 2 or 3.

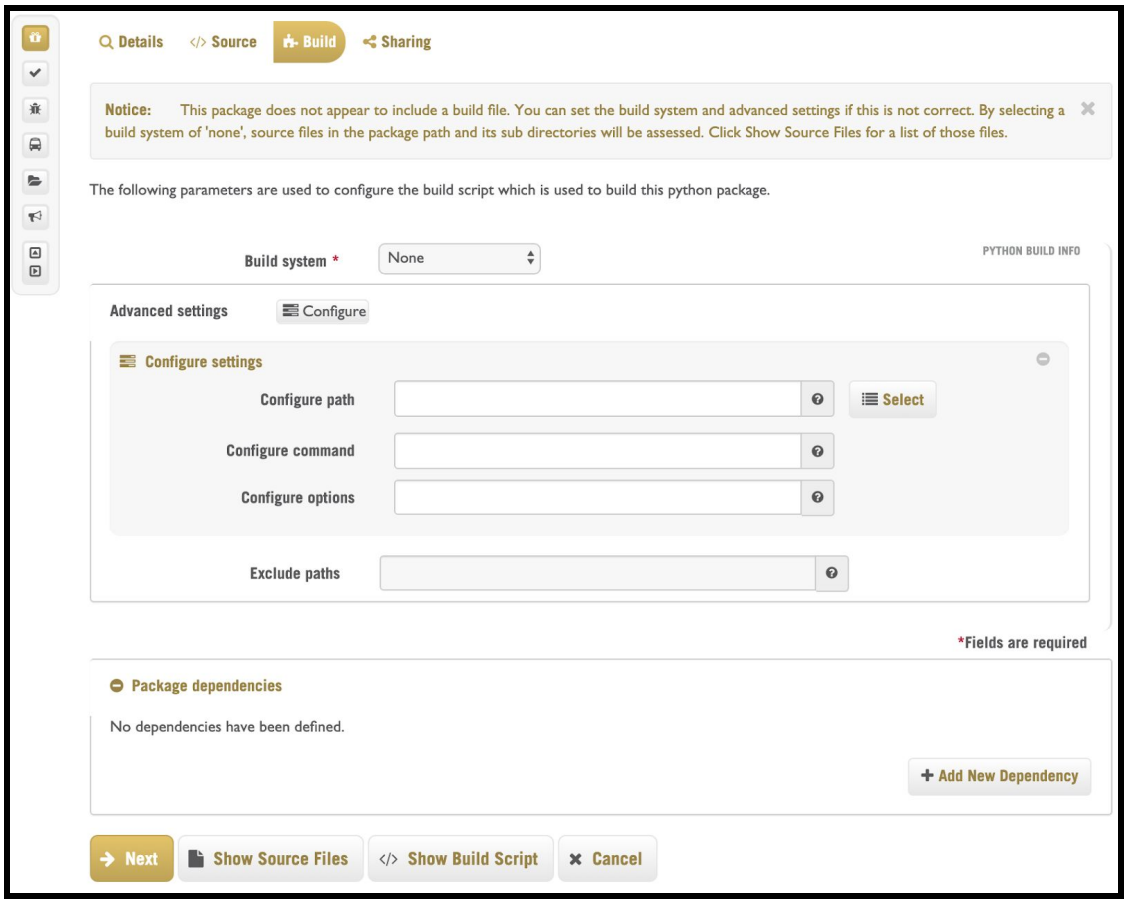


Figure A3. In case your package needs to “build” (compile) for the assessment (Python doesn’t).

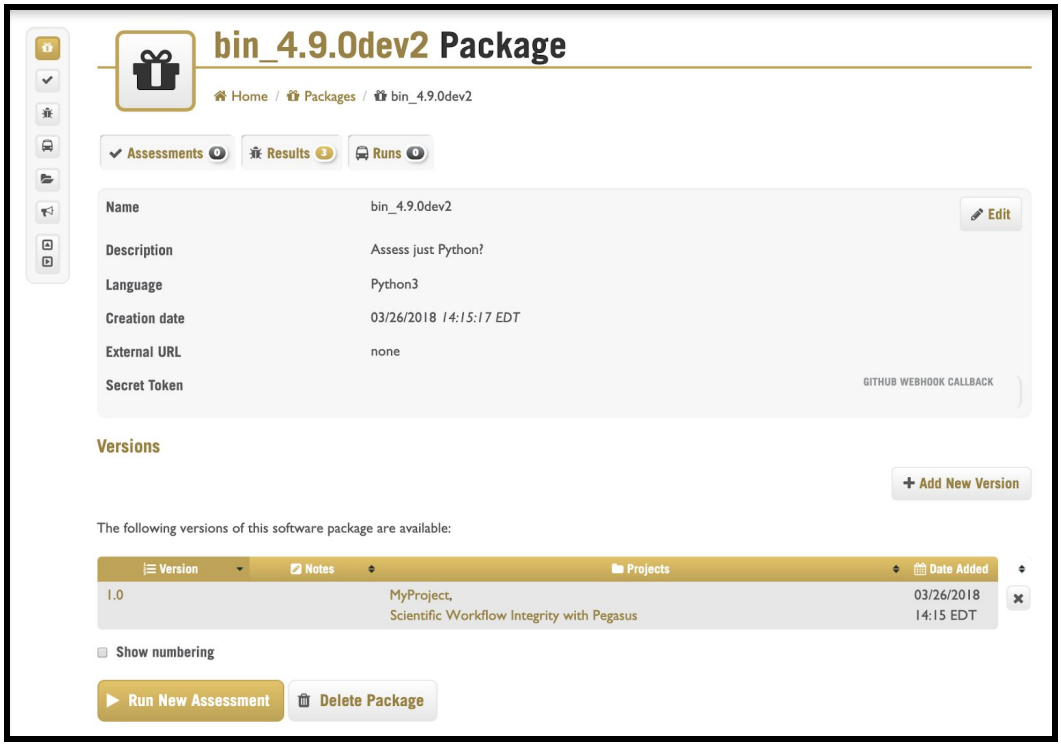


Figure A4. Select the assessment Tools (depends on the package's language)

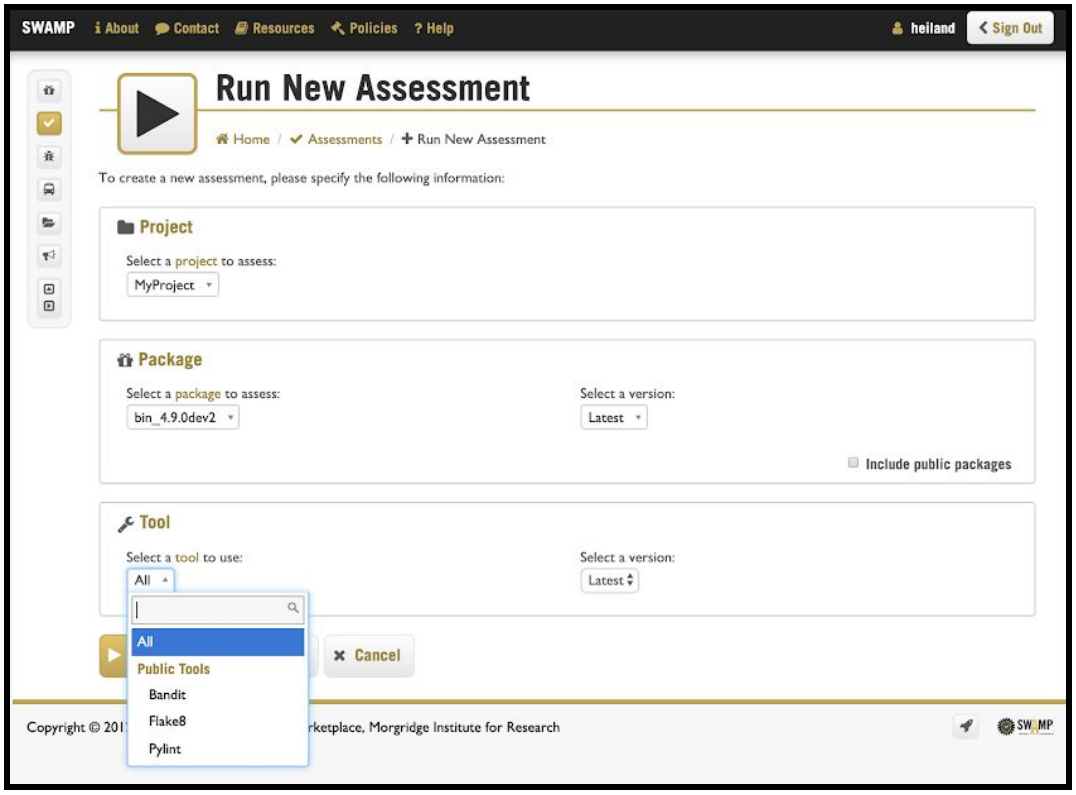


Figure A5. Select the assessment Tools (depends on the package's language)

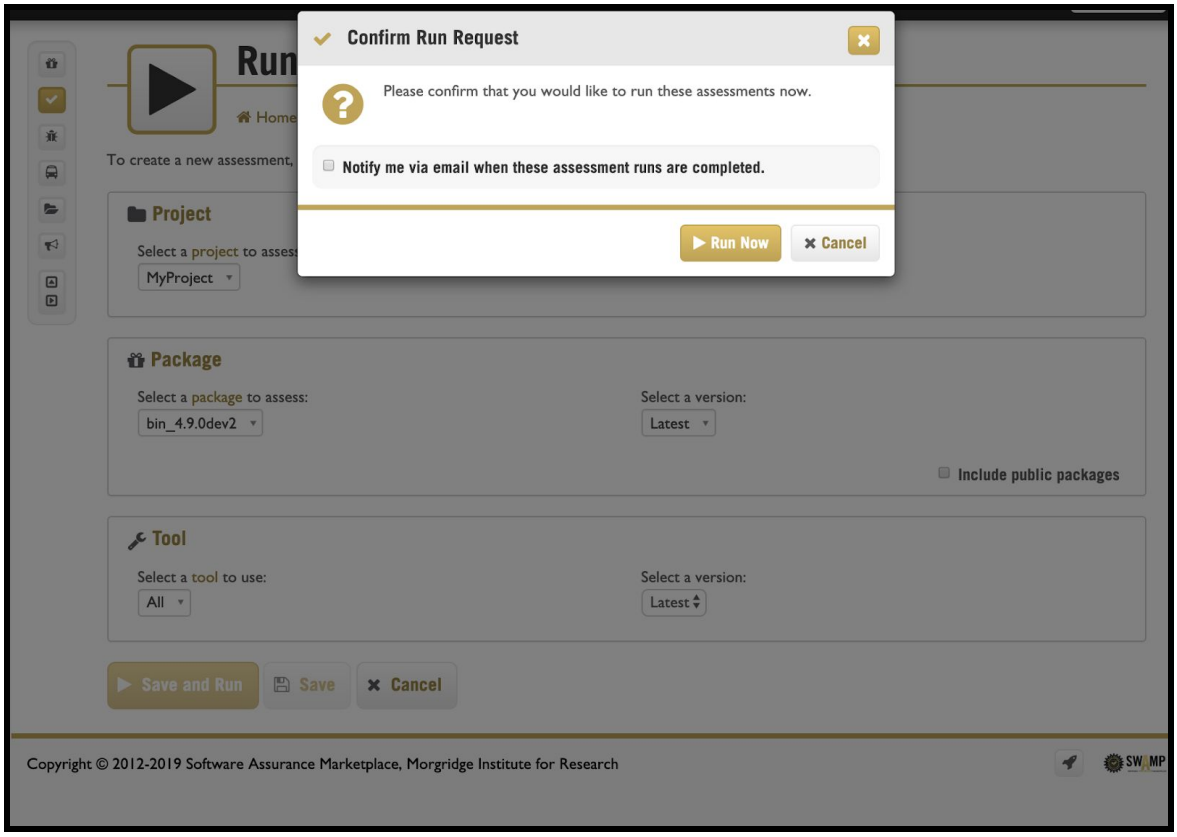


Figure A6. Option to have results emailed to you.

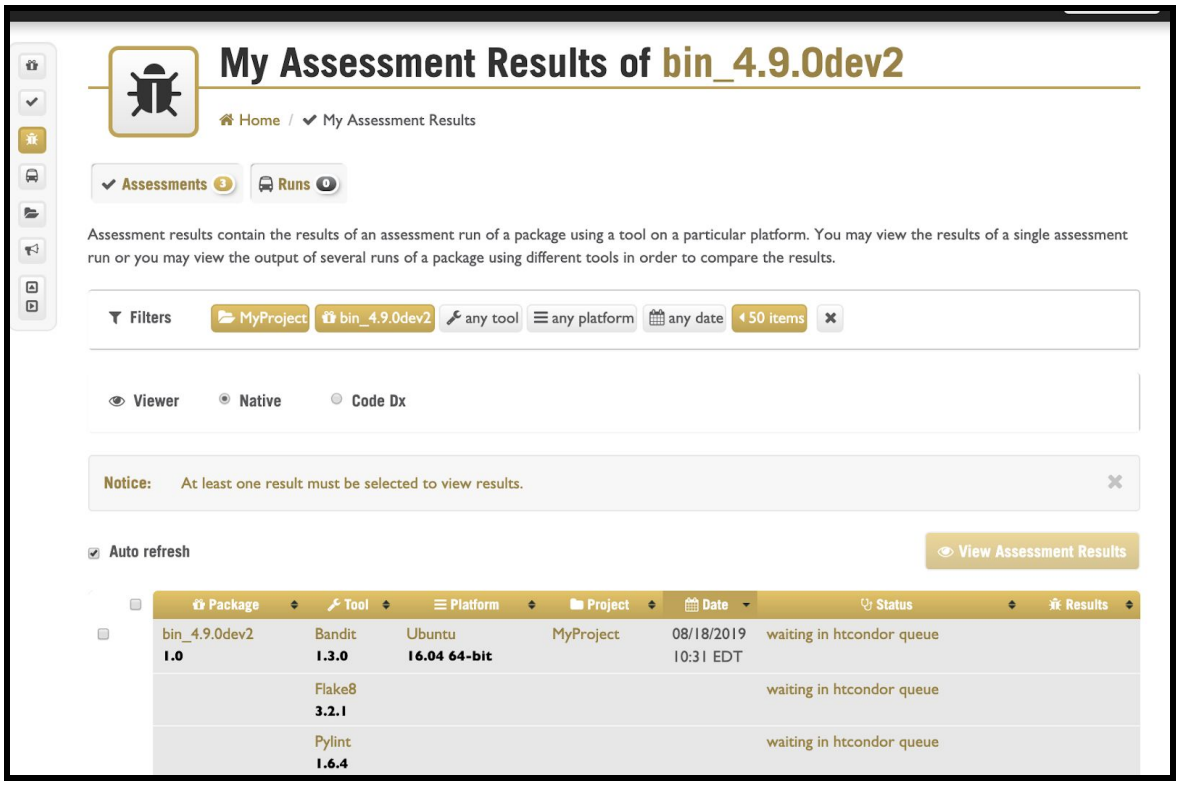


Figure A7. Assessment Status is dynamically updated - “waiting in htcondor queue”

Package	Tool	Platform	Project	Date	Status	Results
bin_4.9.0dev2 1.0	Bandit 1.3.0	Ubuntu 16.04 64-bit	MyProject	08/18/2019 10:31 EDT	starting virtual machine	
	Flake8 3.2.1				starting virtual machine	
	Pylint 1.6.4				starting virtual machine	

Figure A8. Assessment Status “starting virtual machine”

Package	Tool	Platform	Project	Date	Status	Results
bin_4.9.0dev2 1.0	Bandit 1.3.0	Ubuntu 16.04 64-bit	MyProject	08/18/2019 10:31 EDT	obtaining vm ip address	
	Flake8 3.2.1				starting virtual machine	
	Pylint 1.6.4				obtaining vm ip address	

Figure A9. Assessment Status “obtaining vm ip address”

Package	Tool	Platform	Project	Date	Status	Results
bin_4.9.0dev2 1.0	Bandit 1.3.0	Ubuntu 16.04 64-bit	MyProject	08/18/2019 10:31 EDT	performing assessment	
	Flake8 3.2.1				obtaining vm ip address	
	Pylint 1.6.4				performing assessment	

Figure A10. Assessment Status “performing assessment”

Package	Tool	Platform	Project	Date	Status	Results
bin_4.9.0dev2 1.0	Bandit 1.3.0	Ubuntu 16.04 64-bit	MyProject	08/18/2019 10:31 EDT	shutting down the vm	
	Flake8 3.2.1				checking for connected users	
	Pylint 1.6.4				shutting down the vm	

Figure A11. Assessment Status “shutting down the vm”

Package	Tool	Platform	Project	Date	Status	Results
bin_4.9.0dev2 1.0	Bandit 1.3.0	Ubuntu 16.04 64-bit	MyProject	08/18/2019 10:31 EDT	post-processing	
	Flake8 3.2.1				shutting down the vm	
	Pylint 1.6.4				extracting assessment results	

Figure A12. Assessment Status “post-processing”, “extracting assessment results”

Package	Tool	Platform	Project	Date	Status	Results
bin_4.9.0dev2 1.0	Bandit 1.3.0	Ubuntu 16.04 64-bit	MyProject	08/18/2019 10:31 EDT	finished	20
	Flake8 3.2.1				saving results	
	Pylint 1.6.4				finished	20

Figure A13. Assessment Status “saving results”

Package	Tool	Platform	Project	Date	Status	Results
bin_4.9.0dev2 1.0	Bandit 1.3.0	Ubuntu 16.04 64-bit	MyProject	08/18/2019 10:31 EDT	finished	20
	Flake8 3.2.1				finished	3,458
	Pylint 1.6.4				finished	20

Figure A14. Assessment Status “finished”

To view the assessment results, select the row(s) of results and then: 1) use the CodeDx viewer in the SWAMP web interface, 2) use the Native viewer in SWAMP, or 3) click the Result’s SCARF (.xml) files to download them (the small oval icon with a number inside, below the “bug” icon, under Results).

Select row(s) to view then select a web Viewer (Native or Code Dx) or download SCARF files.

Native Viewer Report

Home / Native Viewer Report

Summary

Package	bin_4.9.0dev2 **version 1.0**
Tool	Bandit **version 1.3.0**
Platform	Ubuntu **version 16.04 64-bit**
Number of weaknesses found	20
Create date	08/18/2019 *10:34:20 EDT*

Results

List | Tree | Filter

File	Row	Col	Code
pegasus-db-admin.py	21		blacklist
pegasus-db-admin.py	27		subprocess_popen_with_shell_equals_true
pegasus-db-admin.py	29		subprocess_popen_with_shell_equals_true
pegasus-em.py	4 - 6		blacklist
pegasus-em.py	9		subprocess_popen_with_shell_equals_true
pegasus-em.py	11		subprocess_popen_with_shell_equals_true
pegasus-exitcode.py	27 - 29		blacklist
pegasus-exitcode.py	32		subprocess_popen_with_shell_equals_true
pegasus-globus-online-init.py	4		blacklist

Figure A15. Using the Native viewer to see assessment results.

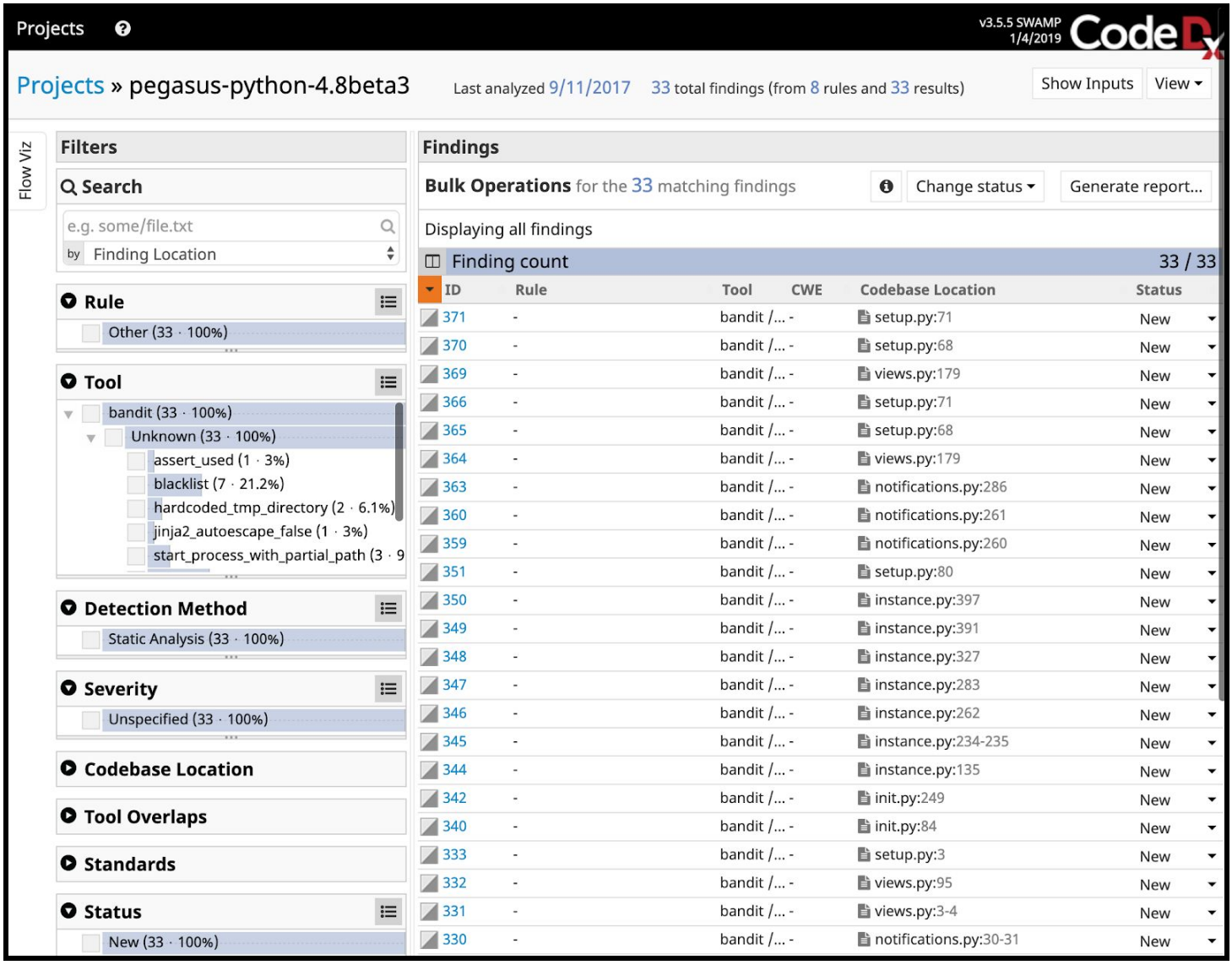


Figure A16. Using the Code Dx web viewer to see assessment results.

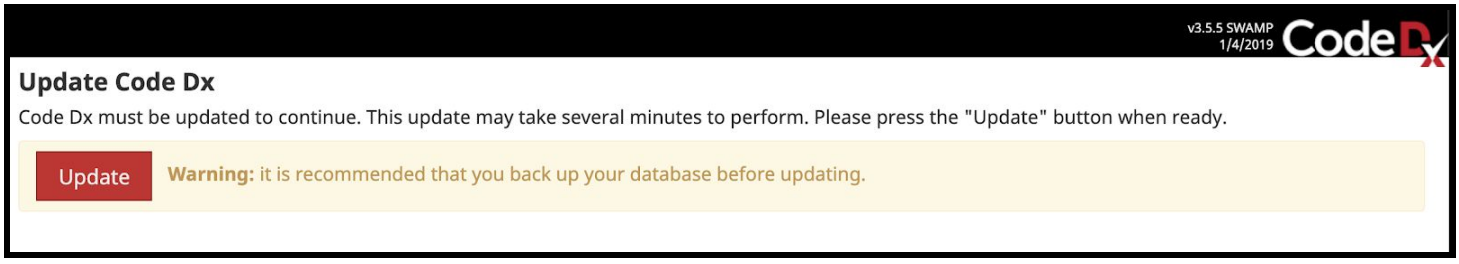


Figure A17. Sometimes Code Dx needs to be updated when you attempt to use it.

And another option for viewing results of SCARF files is to use our `parseSCARF.py` script as described above. It organizes potential vulnerabilities into High, Medium, and Low categories, gives counts of each, and provides minimal descriptions of the vulnerabilities and line numbers of their occurrences.

SWAMP assessment of Pegasus Java code

Assessing the Pegasus Java code using SWAMP requires a few careful steps, as discussed above. We capture some of those steps at https://github.com/IU-CACR/SWIP/issues/8#issuecomment-445864077 and repeat the critical steps here, both in text and SWAMP screenshots.

- Download the desired release (.zip) from https://github.com/pegasus-isi/pegasus/releases/
- Unzip it and edit its build.xml to comment out problem-causing sections for SWAMP:

```
 <!-- Get Git Hash if it isn't already set -->
    <echo>Setting pegasus.build.git.hash</echo>
<!-- comment out:
    <exec executable="/bin/bash" outputproperty="pegasus.build.git.hash"
failonerror="true">
        <arg value="-c"/>
        <arg value="git rev-parse HEAD"/>
    </exec>
-->
  </target>
...
  <target name="compile-r" description="Compile R DAX API">
<!--  comment out:
      <mkdir dir="${dist.share}/r"/>
      <exec executable="./setup.sh" dir="lib/pegasus/r/Pegasus" failonerror="true">
        <arg line="${dist.share}/r" />
      </exec>
-->
  </target>

  <target name="dist-r" depends="dist-clean,compile-r" description="Copy R DAX tarball to
dist folder">
<!-- comment out:
      <copy preservelastmodified="true" todir="dist">
          <fileset dir="${dist.share}/r" includes="*.tar.gz"/>
          <mapper>
            <mapper type="regexp" from="^(.*)\.tar\.gz" to="pegasus-r-\1.tar.gz"/>
          </mapper>
      </copy>
-->
  </target>
```

- re-zip, with the edited build.xml, and upload to SWAMP, e.g.:
  $ zip -r pegasus-4.9.2-git-release-buildxml-edited.zip *
- add dependency for build: "python-setuptools"
- run assessment (using all available Java tools)
- download SCARF files (renaming them with tool suffixes appended)

Details | Source | Build | Sharing

Notice: This appears to be a Java source, Python, Web scripting, C/C++ or Java bytecode package. You can set the language type if this is not correct.

Package path * ./ Select

Language * Java

Show File Types

Java type

- **Java source**
  The package contains uncompiled Java code in its original source code format (.java files).
- **Java bytecode**
  The package contains Java code which has been compiled (.class, .jar, or .apk files).
- **Android**
  The package contains uncompiled Java code for the Android platform.

Java version

- **Java7**
  The package contains Java code for the Java7 platform.
- **Java8**
  The package contains Java code for the Java8 platform.

Add New Package

Home / Packages / Add New Package

Details | Source | Build | Sharing

Notice: This package appears to use the 'Ant' build system. You can set the build system if this is not correct.

The following parameters are used to configure the build script which is used to build this Java package.

JAVA SOURCE BUILD INFO

Build system * Ant

Advanced settings Configure | Build

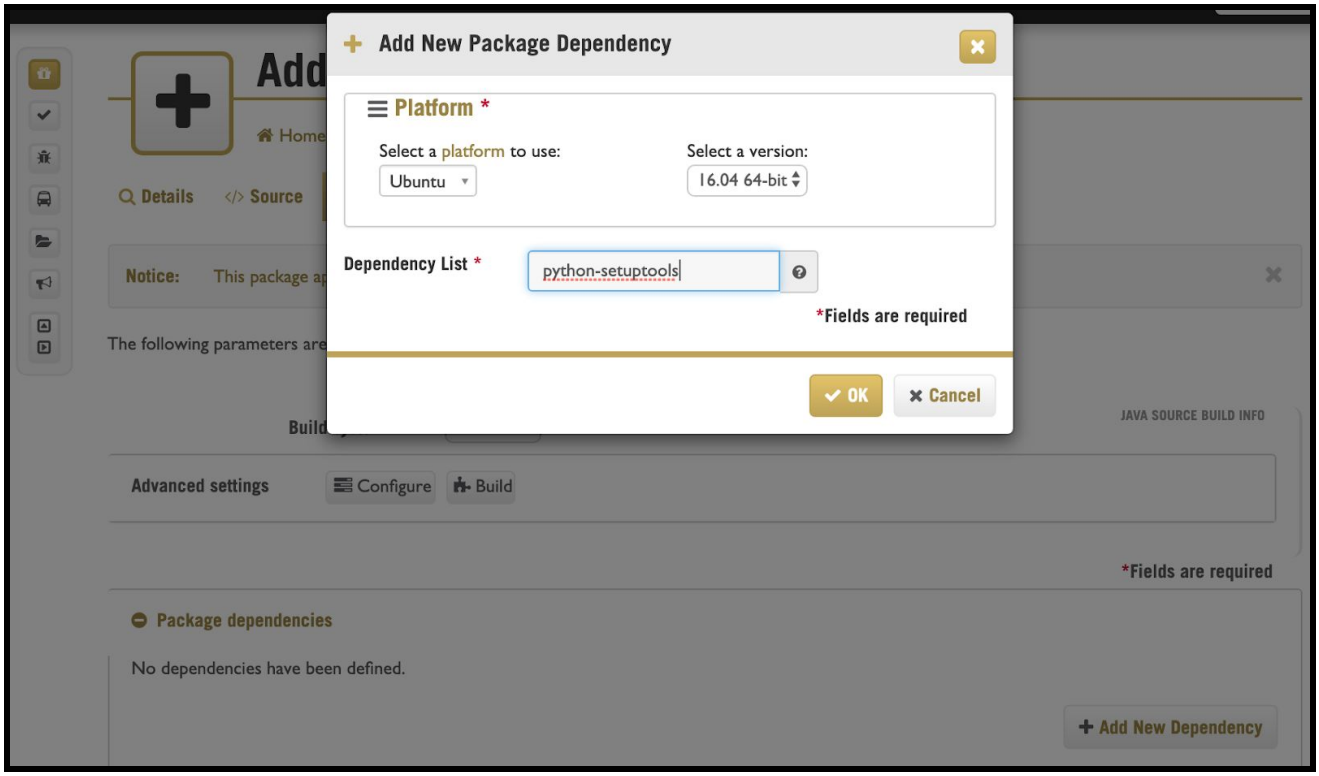
*Fields are required

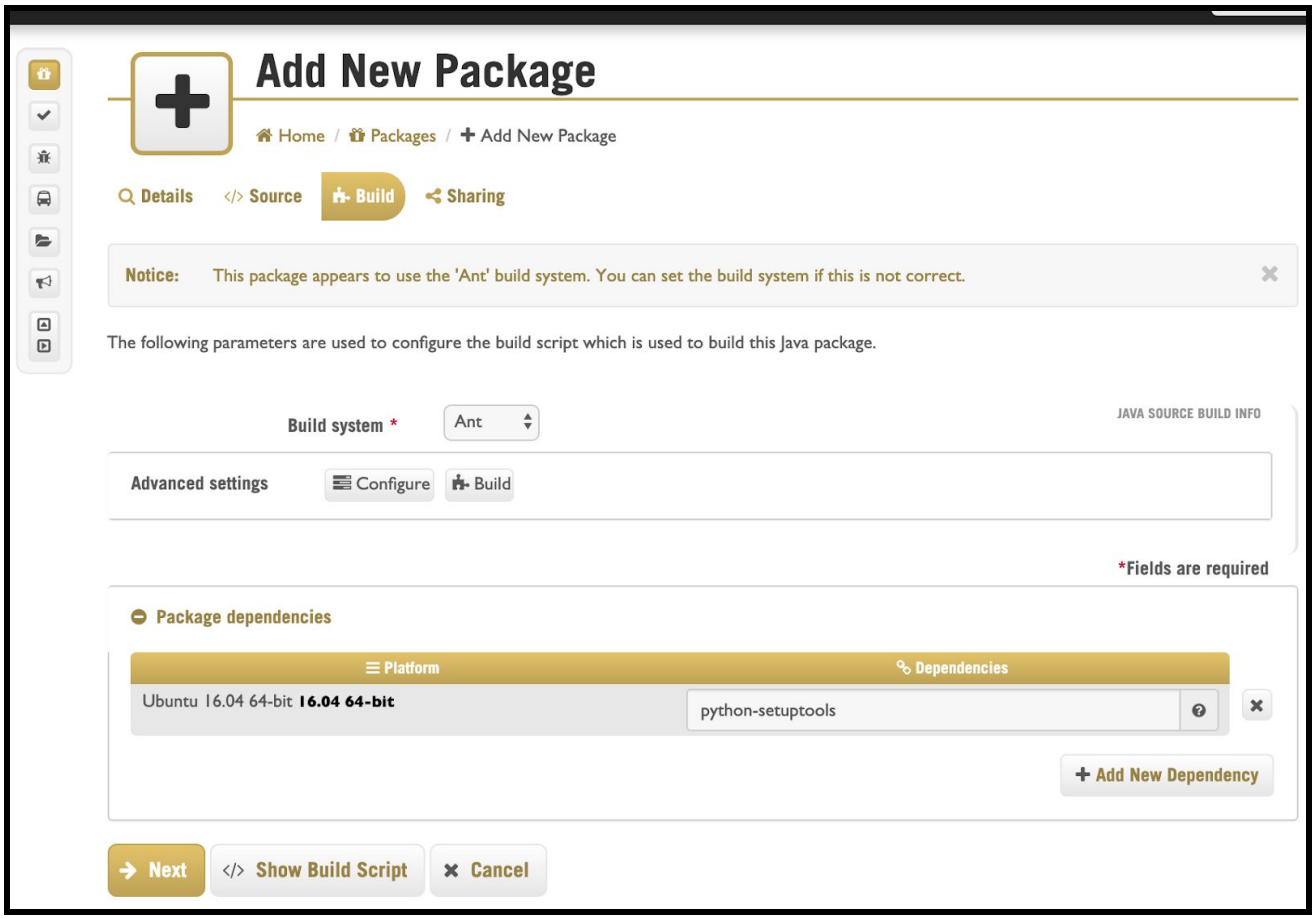
Package dependencies

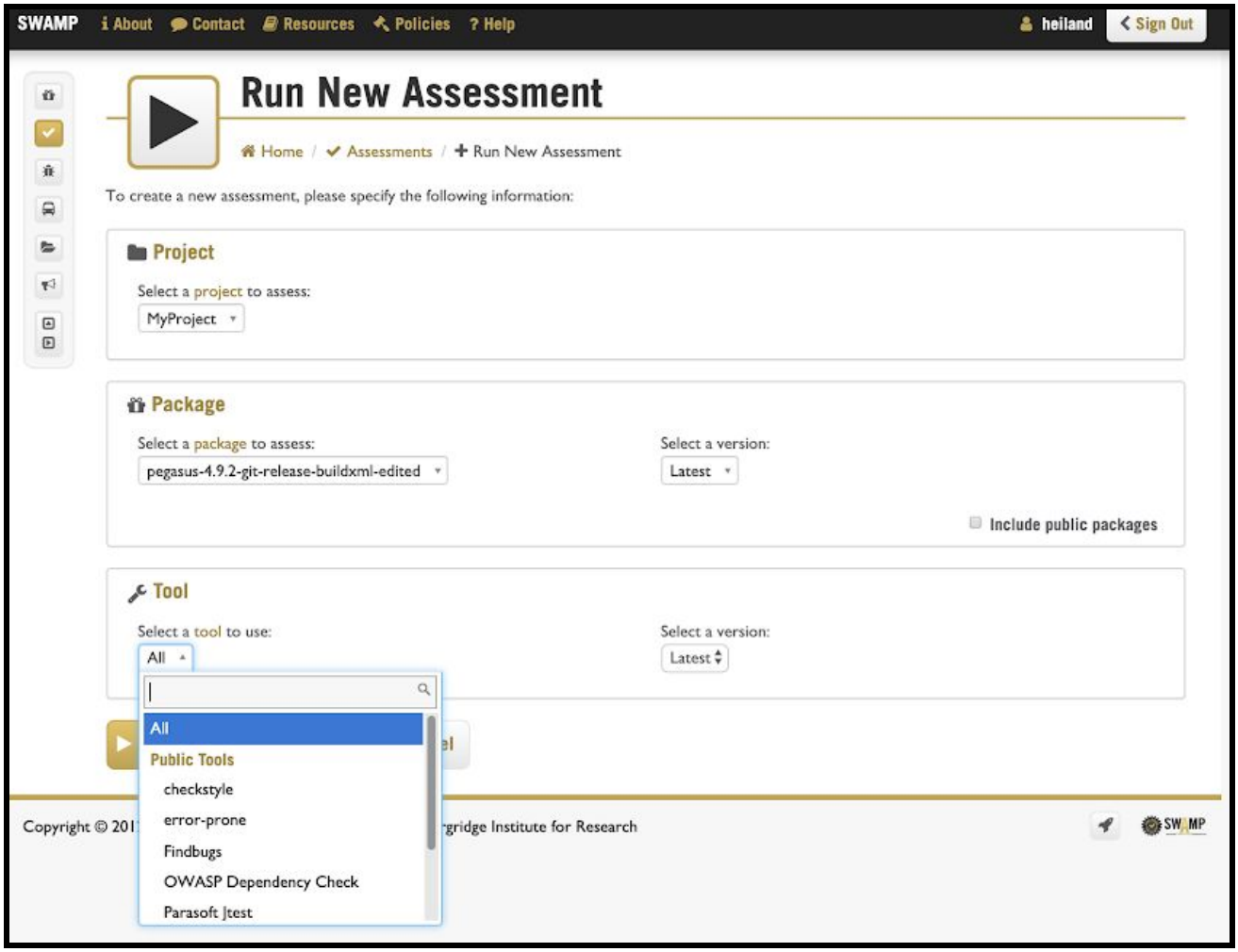
No dependencies have been defined.

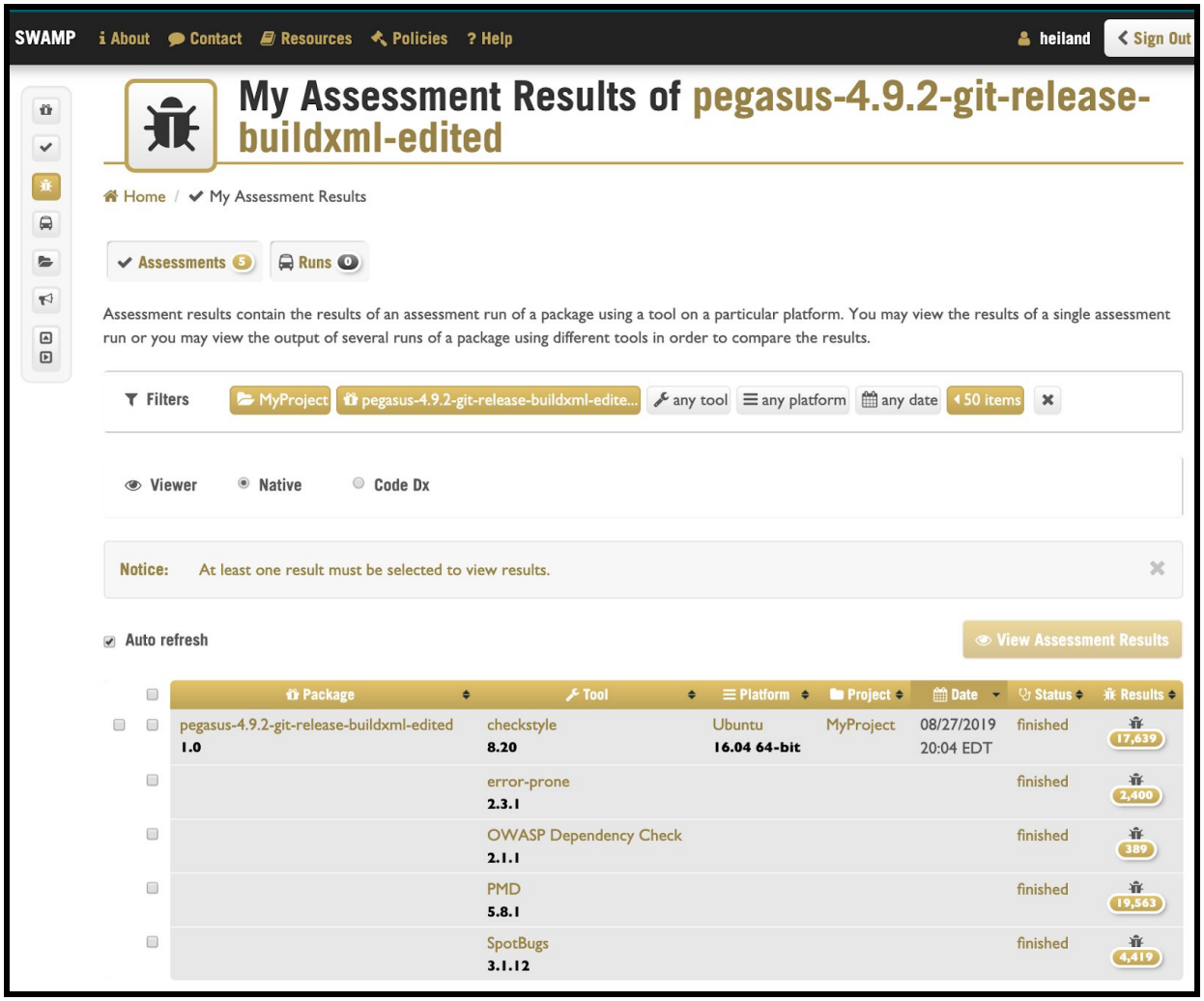
+ Add New Dependency

Next | Show Build Script | Cancel


Add New Package Dependency
Platform *
Select a platform to use:
Ubuntu
Select a version:
16.04 64-bit
Dependency List *
python-setuptools
*Fields are required
OK
Cancel
Details
Source
Notice:
The following parameters are
JAVA SOURCE BUILD INFO
Advanced settings
Configure
Build
*Fields are required
Package dependencies
No dependencies have been defined.
Add New Dependency


Add New Package
Home / Packages / Add New Package
Details
Source
Build
Sharing
Notice:
This package appears to use the 'Ant' build system. You can set the build system if this is not correct.
The following parameters are used to configure the build script which is used to build this Java package.
Build system *
Ant
JAVA SOURCE BUILD INFO
Advanced settings
Configure
Build
*Fields are required
Package dependencies
Platform
Dependencies
Ubuntu 16.04 64-bit 16.04 64-bit
python-setuptools
Add New Dependency
Next
Show Build Script
Cancel


SWAMP
About
Contact
Resources
Policies
Help
heiland
Sign Out
Run New Assessment
Home / Assessments / Run New Assessment
To create a new assessment, please specify the following information:
Project
Select a project to assess:
MyProject
Package
Select a package to assess:
pegasus-4.9.2-git-release-buildxml-edited
Select a version:
Latest
Include public packages
Tool
Select a tool to use:
All
Select a version:
Latest
All
Public Tools
checkstyle
error-prone
Findbugs
OWASP Dependency Check
Parasoft Jtest
Copyright © 201
gridge Institute for Research
SWAMP



Finally, click/download the SCARF files under 'Results'.

Naive use of SWAMP diff script

```
~/git/SWIP/static_analysis$ diff2 scarf_bandit_pegasus4.7.xml scarf_bandit_pegasus4.8.xml
--- scarf_bandit_pegasus4.7.xml
+++ scarf_bandit_pegasus4.8.xml
+ 1 BugInstances categorized by BugGroup: undefined, BugCode: blacklist, SourceFile: pkg1/pegasus-service.py, StartLine: 4
+ 1 BugInstances categorized by BugGroup: undefined, BugCode: blacklist, SourceFile: pkg1/pegasus-db-admin.py, StartLine:
21
+ 1 BugInstances categorized by BugGroup: undefined, BugCode: blacklist, SourceFile: pkg1/pegasus-exitcode.py, StartLine:
27
+ 1 BugInstances categorized by BugGroup: undefined, BugCode: blacklist, SourceFile: pkg1/pegasus-s3.py, StartLine: 4
+ 1 BugInstances categorized by BugGroup: undefined, BugCode: blacklist, SourceFile: pkg1/pegasus-em.py, StartLine: 4
+ 1 BugInstances categorized by BugGroup: undefined, BugCode: blacklist, SourceFile: pkg1/pegasus-submitdir.py, StartLine:
4
+ 1 BugInstances categorized by BugGroup: undefined, BugCode: subprocess_without_shell_equals_true, SourceFile:
pkg1/pegasus-s3.py, StartLine: 8
+ 1 BugInstances categorized by BugGroup: undefined, BugCode: subprocess_without_shell_equals_true, SourceFile:
pkg1/pegasus-s3.py, StartLine: 9
+ 1 BugInstances categorized by BugGroup: undefined, BugCode: subprocess_popen_with_shell_equals_true, SourceFile:
pkg1/pegasus-db-admin.py, StartLine: 27
```

```
+ 1 BugInstances categorized by BugGroup: undefined, BugCode: subprocess_popen_with_shell_equals_true, SourceFile:
pkg1/pegasus-db-admin.py, StartLine: 29
+ 1 BugInstances categorized by BugGroup: undefined, BugCode: subprocess_popen_with_shell_equals_true, SourceFile:
pkg1/pegasus-service.py, StartLine: 9
+ 1 BugInstances categorized by BugGroup: undefined, BugCode: subprocess_popen_with_shell_equals_true, SourceFile:
pkg1/pegasus-service.py, StartLine: 11
+ 1 BugInstances categorized by BugGroup: undefined, BugCode: subprocess_popen_with_shell_equals_true, SourceFile:
pkg1/pegasus-exitcode.py, StartLine: 32
+ 1 BugInstances categorized by BugGroup: undefined, BugCode: subprocess_popen_with_shell_equals_true, SourceFile:
pkg1/pegasus-em.py, StartLine: 11
+ 1 BugInstances categorized by BugGroup: undefined, BugCode: subprocess_popen_with_shell_equals_true, SourceFile:
pkg1/pegasus-em.py, StartLine: 9
+ 1 BugInstances categorized by BugGroup: undefined, BugCode: subprocess_popen_with_shell_equals_true, SourceFile:
pkg1/pegasus-submitdir.py, StartLine: 9
+ 1 BugInstances categorized by BugGroup: undefined, BugCode: subprocess_popen_with_shell_equals_true, SourceFile:
pkg1/pegasus-submitdir.py, StartLine: 11
- 1 BugInstances categorized by BugGroup: undefined, BugCode: exec_used, SourceFile: pkg1/bin/pegasus-init.py, StartLine:
9
- 1 BugInstances categorized by BugGroup: undefined, BugCode: exec_used, SourceFile: pkg1/bin/pegasus-monitord.py,
StartLine: 47
- 1 BugInstances categorized by BugGroup: undefined, BugCode: subprocess_popen_with_shell_equals_true, SourceFile:
pkg1/bin/pegasus-submitdir.py, StartLine: 11
- 1 BugInstances categorized by BugGroup: undefined, BugCode: subprocess_popen_with_shell_equals_true, SourceFile:
pkg1/bin/pegasus-submitdir.py, StartLine: 9
- 1 BugInstances categorized by BugGroup: undefined, BugCode: subprocess_popen_with_shell_equals_true, SourceFile:
pkg1/bin/pegasus-globus-online.py, StartLine: 44
- 1 BugInstances categorized by BugGroup: undefined, BugCode: subprocess_popen_with_shell_equals_true, SourceFile:
pkg1/bin/pegasus-globus-online.py, StartLine: 42
- 1 BugInstances categorized by BugGroup: undefined, BugCode: subprocess_popen_with_shell_equals_true, SourceFile:
pkg1/bin/pegasus-db-admin.py, StartLine: 27
- 1 BugInstances categorized by BugGroup: undefined, BugCode: subprocess_popen_with_shell_equals_true, SourceFile:
pkg1/bin/pegasus-db-admin.py, StartLine: 29
- 1 BugInstances categorized by BugGroup: undefined, BugCode: subprocess_popen_with_shell_equals_true, SourceFile:
pkg1/bin/pegasus-analyzer.py, StartLine: 47
- 1 BugInstances categorized by BugGroup: undefined, BugCode: subprocess_popen_with_shell_equals_true, SourceFile:
pkg1/bin/pegasus-analyzer.py, StartLine: 898
- 1 BugInstances categorized by BugGroup: undefined, BugCode: subprocess_popen_with_shell_equals_true, SourceFile:
pkg1/bin/pegasus-analyzer.py, StartLine: 1284
- 1 BugInstances categorized by BugGroup: undefined, BugCode: subprocess_popen_with_shell_equals_true, SourceFile:
pkg1/bin/pegasus-analyzer.py, StartLine: 847
- 1 BugInstances categorized by BugGroup: undefined, BugCode: subprocess_popen_with_shell_equals_true, SourceFile:
pkg1/bin/pegasus-analyzer.py, StartLine: 49
- 1 BugInstances categorized by BugGroup: undefined, BugCode: subprocess_popen_with_shell_equals_true, SourceFile:
pkg1/bin/pegasus-plots.py, StartLine: 20
- 1 BugInstances categorized by BugGroup: undefined, BugCode: subprocess_popen_with_shell_equals_true, SourceFile:
pkg1/bin/pegasus-plots.py, StartLine: 22
- 1 BugInstances categorized by BugGroup: undefined, BugCode: subprocess_popen_with_shell_equals_true, SourceFile:
pkg1/bin/pegasus-init.py, StartLine: 9
- 1 BugInstances categorized by BugGroup: undefined, BugCode: subprocess_popen_with_shell_equals_true, SourceFile:
pkg1/bin/pegasus-exitcode.py, StartLine: 32
- 1 BugInstances categorized by BugGroup: undefined, BugCode: subprocess_popen_with_shell_equals_true, SourceFile:
pkg1/bin/pegasus-monitord.py, StartLine: 47
- 1 BugInstances categorized by BugGroup: undefined, BugCode: subprocess_popen_with_shell_equals_true, SourceFile:
pkg1/bin/pegasus-statistics.py, StartLine: 16
- 1 BugInstances categorized by BugGroup: undefined, BugCode: subprocess_popen_with_shell_equals_true, SourceFile:
pkg1/bin/pegasus-statistics.py, StartLine: 18
- 1 BugInstances categorized by BugGroup: undefined, BugCode: subprocess_popen_with_shell_equals_true, SourceFile:
pkg1/bin/pegasus-transfer.py, StartLine: 532
- 1 BugInstances categorized by BugGroup: undefined, BugCode: subprocess_popen_with_shell_equals_true, SourceFile:
pkg1/bin/pegasus-transfer.py, StartLine: 557
```

- 1 BugInstances categorized by BugGroup: undefined, BugCode: subprocess_popen_with_shell_equals_true, SourceFile: pkg1/bin/pegasus-transfer.py, StartLine: 3302
- 1 BugInstances categorized by BugGroup: undefined, BugCode: subprocess_popen_with_shell_equals_true, SourceFile: pkg1/bin/pegasus-service.py, StartLine: 9
- 1 BugInstances categorized by BugGroup: undefined, BugCode: subprocess_popen_with_shell_equals_true, SourceFile: pkg1/bin/pegasus-service.py, StartLine: 11
- 1 BugInstances categorized by BugGroup: undefined, BugCode: subprocess_popen_with_shell_equals_true, SourceFile: pkg1/bin/pegasus-metadata.py, StartLine: 38
- 1 BugInstances categorized by BugGroup: undefined, BugCode: subprocess_popen_with_shell_equals_true, SourceFile: pkg1/bin/pegasus-metadata.py, StartLine: 36
- 1 BugInstances categorized by BugGroup: undefined, BugCode: subprocess_popen_with_shell_equals_true, SourceFile: pkg1/bin/pegasus-em.py, StartLine: 9
- 1 BugInstances categorized by BugGroup: undefined, BugCode: subprocess_popen_with_shell_equals_true, SourceFile: pkg1/bin/pegasus-em.py, StartLine: 11
- 1 BugInstances categorized by BugGroup: undefined, BugCode: hardcoded_tmp_directory, SourceFile: pkg1/bin/pegasus-analyzer.py, StartLine: 177
- 1 BugInstances categorized by BugGroup: undefined, BugCode: set_bad_file_permissions, SourceFile: pkg1/bin/pegasus-analyzer.py, StartLine: 1465
- 1 BugInstances categorized by BugGroup: undefined, BugCode: set_bad_file_permissions, SourceFile: pkg1/bin/pegasus-analyzer.py, StartLine: 338
- 1 BugInstances categorized by BugGroup: undefined, BugCode: start_process_with_a_shell, SourceFile: pkg1/bin/pegasus-analyzer.py, StartLine: 681
- 1 BugInstances categorized by BugGroup: undefined, BugCode: start_process_with_a_shell, SourceFile: pkg1/bin/pegasus-plots.py, StartLine: 158
- 1 BugInstances categorized by BugGroup: undefined, BugCode: start_process_with_a_shell, SourceFile: pkg1/bin/pegasus-plots.py, StartLine: 129
- 1 BugInstances categorized by BugGroup: undefined, BugCode: start_process_with_a_shell, SourceFile: pkg1/bin/pegasus-plots.py, StartLine: 174
- 1 BugInstances categorized by BugGroup: undefined, BugCode: start_process_with_a_shell, SourceFile: pkg1/bin/pegasus-plots.py, StartLine: 114
- 1 BugInstances categorized by BugGroup: undefined, BugCode: assert_used, SourceFile: pkg1/bin/pegasus-transfer.py, StartLine: 3073
- 1 BugInstances categorized by BugGroup: undefined, BugCode: try_except_pass, SourceFile: pkg1/bin/pegasus-dagman.py, StartLine: 176
- 1 BugInstances categorized by BugGroup: undefined, BugCode: try_except_pass, SourceFile: pkg1/bin/pegasus-transfer.py, StartLine: 1321
- 1 BugInstances categorized by BugGroup: undefined, BugCode: try_except_pass, SourceFile: pkg1/bin/pegasus-transfer.py, StartLine: 3718
- 1 BugInstances categorized by BugGroup: undefined, BugCode: try_except_pass, SourceFile: pkg1/bin/pegasus-transfer.py, StartLine: 3172
- 1 BugInstances categorized by BugGroup: undefined, BugCode: try_except_pass, SourceFile: pkg1/bin/pegasus-transfer.py, StartLine: 2891
- 1 BugInstances categorized by BugGroup: undefined, BugCode: try_except_pass, SourceFile: pkg1/bin/pegasus-transfer.py, StartLine: 1413
- 1 BugInstances categorized by BugGroup: undefined, BugCode: try_except_pass, SourceFile: pkg1/bin/pegasus-transfer.py, StartLine: 3116
- 1 BugInstances categorized by BugGroup: undefined, BugCode: try_except_pass, SourceFile: pkg1/bin/pegasus-transfer.py, StartLine: 3431
- 1 BugInstances categorized by BugGroup: undefined, BugCode: try_except_pass, SourceFile: pkg1/bin/pegasus-transfer.py, StartLine: 560
- 1 BugInstances categorized by BugGroup: undefined, BugCode: try_except_pass, SourceFile: pkg1/bin/pegasus-transfer.py, StartLine: 2671
- 1 BugInstances categorized by BugGroup: undefined, BugCode: try_except_pass, SourceFile: pkg1/bin/pegasus-transfer.py, StartLine: 2455
- 1 BugInstances categorized by BugGroup: undefined, BugCode: try_except_pass, SourceFile: pkg1/bin/pegasus-transfer.py, StartLine: 1717
- 1 BugInstances categorized by BugGroup: undefined, BugCode: try_except_pass, SourceFile: pkg1/bin/pegasus-transfer.py, StartLine: 2122
- 1 BugInstances categorized by BugGroup: undefined, BugCode: try_except_pass, SourceFile: pkg1/bin/pegasus-globus-online.py, StartLine: 171

- 1 BugInstances categorized by BugGroup: undefined, BugCode: subprocess_without_shell_equals_true, SourceFile: pkg1/bin/pegasus-s3.py, StartLine: 9
- 1 BugInstances categorized by BugGroup: undefined, BugCode: subprocess_without_shell_equals_true, SourceFile: pkg1/bin/pegasus-s3.py, StartLine: 8
- 1 BugInstances categorized by BugGroup: undefined, BugCode: subprocess_without_shell_equals_true, SourceFile: pkg1/bin/pegasus-dagman.py, StartLine: 59
- 1 BugInstances categorized by BugGroup: undefined, BugCode: subprocess_without_shell_equals_true, SourceFile: pkg1/bin/pegasus-dagman.py, StartLine: 136
- 1 BugInstances categorized by BugGroup: undefined, BugCode: subprocess_without_shell_equals_true, SourceFile: pkg1/bin/pegasus-dagman.py, StartLine: 60
- 1 BugInstances categorized by BugGroup: undefined, BugCode: subprocess_without_shell_equals_true, SourceFile: pkg1/bin/pegasus-dagman.py, StartLine: 93
- 1 BugInstances categorized by BugGroup: undefined, BugCode: subprocess_without_shell_equals_true, SourceFile: pkg1/bin/pegasus-dagman.py, StartLine: 117
- 1 BugInstances categorized by BugGroup: undefined, BugCode: blacklist, SourceFile: pkg1/bin/parseSCARF.py, StartLine: 26
- 1 BugInstances categorized by BugGroup: undefined, BugCode: blacklist, SourceFile: pkg1/bin/parseSCARF.py, StartLine: 6
- 1 BugInstances categorized by BugGroup: undefined, BugCode: blacklist, SourceFile: pkg1/bin/pegasus-monitord.py, StartLine: 39
- 1 BugInstances categorized by BugGroup: undefined, BugCode: blacklist, SourceFile: pkg1/bin/pegasus-exitcode.py, StartLine: 27
- 1 BugInstances categorized by BugGroup: undefined, BugCode: blacklist, SourceFile: pkg1/bin/pegasus-transfer.py, StartLine: 279
- 1 BugInstances categorized by BugGroup: undefined, BugCode: blacklist, SourceFile: pkg1/bin/pegasus-transfer.py, StartLine: 2974
- 1 BugInstances categorized by BugGroup: undefined, BugCode: blacklist, SourceFile: pkg1/bin/pegasus-transfer.py, StartLine: 46
- 1 BugInstances categorized by BugGroup: undefined, BugCode: blacklist, SourceFile: pkg1/bin/pegasus-transfer.py, StartLine: 40
- 1 BugInstances categorized by BugGroup: undefined, BugCode: blacklist, SourceFile: pkg1/bin/pegasus-transfer.py, StartLine: 2979
- 1 BugInstances categorized by BugGroup: undefined, BugCode: blacklist, SourceFile: pkg1/bin/pegasus-transfer.py, StartLine: 289
- 1 BugInstances categorized by BugGroup: undefined, BugCode: blacklist, SourceFile: pkg1/bin/pegasus-transfer.py, StartLine: 3771
- 1 BugInstances categorized by BugGroup: undefined, BugCode: blacklist, SourceFile: pkg1/bin/pegasus-service.py, StartLine: 4
- 1 BugInstances categorized by BugGroup: undefined, BugCode: blacklist, SourceFile: pkg1/bin/pegasus-graphviz.py, StartLine: 4
- 1 BugInstances categorized by BugGroup: undefined, BugCode: blacklist, SourceFile: pkg1/bin/pegasus-graphviz.py, StartLine: 5
- 1 BugInstances categorized by BugGroup: undefined, BugCode: blacklist, SourceFile: pkg1/bin/pegasus-graphviz.py, StartLine: 223
- 1 BugInstances categorized by BugGroup: undefined, BugCode: blacklist, SourceFile: pkg1/bin/pegasus-globus-online.py, StartLine: 30
- 1 BugInstances categorized by BugGroup: undefined, BugCode: blacklist, SourceFile: pkg1/bin/pegasus-dagman.py, StartLine: 33
- 1 BugInstances categorized by BugGroup: undefined, BugCode: blacklist, SourceFile: pkg1/bin/pegasus-em.py, StartLine: 4
- 1 BugInstances categorized by BugGroup: undefined, BugCode: blacklist, SourceFile: pkg1/bin/pegasus-metadata.py, StartLine: 25
- 1 BugInstances categorized by BugGroup: undefined, BugCode: blacklist, SourceFile: pkg1/bin/pegasus-statistics.py, StartLine: 7
- 1 BugInstances categorized by BugGroup: undefined, BugCode: blacklist, SourceFile: pkg1/bin/pegasus-db-admin.py, StartLine: 21
- 1 BugInstances categorized by BugGroup: undefined, BugCode: blacklist, SourceFile: pkg1/bin/pegasus-analyzer.py, StartLine: 38
- 1 BugInstances categorized by BugGroup: undefined, BugCode: blacklist, SourceFile: pkg1/bin/pegasus-plots.py, StartLine: 9
- 1 BugInstances categorized by BugGroup: undefined, BugCode: blacklist, SourceFile: pkg1/bin/pegasus-init.py, StartLine: 4

```
- 1 BugInstances categorized by BugGroup: undefined, BugCode: blacklist, SourceFile: pkg1/bin/pegasus-submitdir.py,
StartLine: 4
- 1 BugInstances categorized by BugGroup: undefined, BugCode: blacklist, SourceFile: pkg1/bin/pegasus-s3.py, StartLine: 4
~/git/SWIP/static_analysis$
```

Proper use of SWAMP diff script

```
~/git/SWIP/static_analysis$ diff2 --no-source_file scarf_bandit_pegasus4.7.xml scarf_bandit_pegasus4.8.xml
--- scarf_bandit_pegasus4.7.xml
+++ scarf_bandit_pegasus4.8.xml
- 1 BugInstances categorized by BugGroup: undefined, BugCode: subprocess_popen_with_shell_equals_true, SourceFile:
undefined, StartLine: 898
- 1 BugInstances categorized by BugGroup: undefined, BugCode: subprocess_popen_with_shell_equals_true, SourceFile:
undefined, StartLine: 36
- 1 BugInstances categorized by BugGroup: undefined, BugCode: subprocess_popen_with_shell_equals_true, SourceFile:
undefined, StartLine: 38
- 1 BugInstances categorized by BugGroup: undefined, BugCode: subprocess_popen_with_shell_equals_true, SourceFile:
undefined, StartLine: 847
- 1 BugInstances categorized by BugGroup: undefined, BugCode: subprocess_popen_with_shell_equals_true, SourceFile:
undefined, StartLine: 9
- 1 BugInstances categorized by BugGroup: undefined, BugCode: subprocess_popen_with_shell_equals_true, SourceFile:
undefined, StartLine: 49
- 1 BugInstances categorized by BugGroup: undefined, BugCode: subprocess_popen_with_shell_equals_true, SourceFile:
undefined, StartLine: 44
- 1 BugInstances categorized by BugGroup: undefined, BugCode: subprocess_popen_with_shell_equals_true, SourceFile:
undefined, StartLine: 22
- 1 BugInstances categorized by BugGroup: undefined, BugCode: subprocess_popen_with_shell_equals_true, SourceFile:
undefined, StartLine: 20
- 1 BugInstances categorized by BugGroup: undefined, BugCode: subprocess_popen_with_shell_equals_true, SourceFile:
undefined, StartLine: 532
- 1 BugInstances categorized by BugGroup: undefined, BugCode: subprocess_popen_with_shell_equals_true, SourceFile:
undefined, StartLine: 557
- 1 BugInstances categorized by BugGroup: undefined, BugCode: subprocess_popen_with_shell_equals_true, SourceFile:
undefined, StartLine: 1284
- 1 BugInstances categorized by BugGroup: undefined, BugCode: subprocess_popen_with_shell_equals_true, SourceFile:
undefined, StartLine: 16
- 1 BugInstances categorized by BugGroup: undefined, BugCode: subprocess_popen_with_shell_equals_true, SourceFile:
undefined, StartLine: 42
- 1 BugInstances categorized by BugGroup: undefined, BugCode: subprocess_popen_with_shell_equals_true, SourceFile:
undefined, StartLine: 18
- 2 BugInstances categorized by BugGroup: undefined, BugCode: subprocess_popen_with_shell_equals_true, SourceFile:
undefined, StartLine: 47
- 1 BugInstances categorized by BugGroup: undefined, BugCode: subprocess_popen_with_shell_equals_true, SourceFile:
undefined, StartLine: 3302
- 1 BugInstances categorized by BugGroup: undefined, BugCode: blacklist, SourceFile: undefined, StartLine: 289
- 2 BugInstances categorized by BugGroup: undefined, BugCode: blacklist, SourceFile: undefined, StartLine: 4
- 1 BugInstances categorized by BugGroup: undefined, BugCode: blacklist, SourceFile: undefined, StartLine: 223
- 1 BugInstances categorized by BugGroup: undefined, BugCode: blacklist, SourceFile: undefined, StartLine: 30
- 1 BugInstances categorized by BugGroup: undefined, BugCode: blacklist, SourceFile: undefined, StartLine: 26
- 1 BugInstances categorized by BugGroup: undefined, BugCode: blacklist, SourceFile: undefined, StartLine: 7
- 1 BugInstances categorized by BugGroup: undefined, BugCode: blacklist, SourceFile: undefined, StartLine: 33
- 1 BugInstances categorized by BugGroup: undefined, BugCode: blacklist, SourceFile: undefined, StartLine: 25
- 1 BugInstances categorized by BugGroup: undefined, BugCode: blacklist, SourceFile: undefined, StartLine: 2979
- 1 BugInstances categorized by BugGroup: undefined, BugCode: blacklist, SourceFile: undefined, StartLine: 6
- 1 BugInstances categorized by BugGroup: undefined, BugCode: blacklist, SourceFile: undefined, StartLine: 38
- 1 BugInstances categorized by BugGroup: undefined, BugCode: blacklist, SourceFile: undefined, StartLine: 3771
- 1 BugInstances categorized by BugGroup: undefined, BugCode: blacklist, SourceFile: undefined, StartLine: 40
```

```
- 1 BugInstances categorized by BugGroup: undefined, BugCode: blacklist, SourceFile: undefined, StartLine: 9
- 1 BugInstances categorized by BugGroup: undefined, BugCode: blacklist, SourceFile: undefined, StartLine: 5
- 1 BugInstances categorized by BugGroup: undefined, BugCode: blacklist, SourceFile: undefined, StartLine: 2974
- 1 BugInstances categorized by BugGroup: undefined, BugCode: blacklist, SourceFile: undefined, StartLine: 46
- 1 BugInstances categorized by BugGroup: undefined, BugCode: blacklist, SourceFile: undefined, StartLine: 279
- 1 BugInstances categorized by BugGroup: undefined, BugCode: blacklist, SourceFile: undefined, StartLine: 39
- 1 BugInstances categorized by BugGroup: undefined, BugCode: set_bad_file_permissions, SourceFile: undefined, StartLine:
338
- 1 BugInstances categorized by BugGroup: undefined, BugCode: set_bad_file_permissions, SourceFile: undefined, StartLine:
1465
- 1 BugInstances categorized by BugGroup: undefined, BugCode: assert_used, SourceFile: undefined, StartLine: 3073
- 1 BugInstances categorized by BugGroup: undefined, BugCode: try_except_pass, SourceFile: undefined, StartLine: 3116
- 1 BugInstances categorized by BugGroup: undefined, BugCode: try_except_pass, SourceFile: undefined, StartLine: 2891
- 1 BugInstances categorized by BugGroup: undefined, BugCode: try_except_pass, SourceFile: undefined, StartLine: 3172
- 1 BugInstances categorized by BugGroup: undefined, BugCode: try_except_pass, SourceFile: undefined, StartLine: 3718
- 1 BugInstances categorized by BugGroup: undefined, BugCode: try_except_pass, SourceFile: undefined, StartLine: 2122
- 1 BugInstances categorized by BugGroup: undefined, BugCode: try_except_pass, SourceFile: undefined, StartLine: 1413
- 1 BugInstances categorized by BugGroup: undefined, BugCode: try_except_pass, SourceFile: undefined, StartLine: 1321
- 1 BugInstances categorized by BugGroup: undefined, BugCode: try_except_pass, SourceFile: undefined, StartLine: 176
- 1 BugInstances categorized by BugGroup: undefined, BugCode: try_except_pass, SourceFile: undefined, StartLine: 560
- 1 BugInstances categorized by BugGroup: undefined, BugCode: try_except_pass, SourceFile: undefined, StartLine: 2455
- 1 BugInstances categorized by BugGroup: undefined, BugCode: try_except_pass, SourceFile: undefined, StartLine: 1717
- 1 BugInstances categorized by BugGroup: undefined, BugCode: try_except_pass, SourceFile: undefined, StartLine: 2671
- 1 BugInstances categorized by BugGroup: undefined, BugCode: try_except_pass, SourceFile: undefined, StartLine: 171
- 1 BugInstances categorized by BugGroup: undefined, BugCode: try_except_pass, SourceFile: undefined, StartLine: 3431
- 1 BugInstances categorized by BugGroup: undefined, BugCode: subprocess_without_shell_equals_true, SourceFile: undefined,
StartLine: 117
- 1 BugInstances categorized by BugGroup: undefined, BugCode: subprocess_without_shell_equals_true, SourceFile: undefined,
StartLine: 60
- 1 BugInstances categorized by BugGroup: undefined, BugCode: subprocess_without_shell_equals_true, SourceFile: undefined,
StartLine: 93
- 1 BugInstances categorized by BugGroup: undefined, BugCode: subprocess_without_shell_equals_true, SourceFile: undefined,
StartLine: 136
- 1 BugInstances categorized by BugGroup: undefined, BugCode: subprocess_without_shell_equals_true, SourceFile: undefined,
StartLine: 59
- 1 BugInstances categorized by BugGroup: undefined, BugCode: exec_used, SourceFile: undefined, StartLine: 47
- 1 BugInstances categorized by BugGroup: undefined, BugCode: exec_used, SourceFile: undefined, StartLine: 9
- 1 BugInstances categorized by BugGroup: undefined, BugCode: start_process_with_a_shell, SourceFile: undefined,
StartLine: 129
- 1 BugInstances categorized by BugGroup: undefined, BugCode: start_process_with_a_shell, SourceFile: undefined,
StartLine: 114
- 1 BugInstances categorized by BugGroup: undefined, BugCode: start_process_with_a_shell, SourceFile: undefined,
StartLine: 681
- 1 BugInstances categorized by BugGroup: undefined, BugCode: start_process_with_a_shell, SourceFile: undefined,
StartLine: 174
- 1 BugInstances categorized by BugGroup: undefined, BugCode: start_process_with_a_shell, SourceFile: undefined,
StartLine: 158
- 1 BugInstances categorized by BugGroup: undefined, BugCode: hardcoded_tmp_directory, SourceFile: undefined, StartLine:
177
~/git/SWIP/static_analysis$
```