Solar appScreener 3.5

Release notes

2020

3 key release features

Solar appScreener now supports Rust

Popular language Rust became 33rd supported programming language. It can be used for:

- high performance servers;
- command line tools;
- operating system modules;
- web browsers;
- etc.

Solar appScreener allows companies using Rust to check their safety with source or binary code analysis.

Integration with Subversion

Code to be analyzed is downloaded directly from a repository, so there is no need to import source code files each time. Integration with the Subversion repository helps to secure software development lifecycle (SDLC), allowing the programmer to establish quality control, to automate new software build verification and to reduce time spent.

Improved usability

No more manual choice between Python 2 and Python 3 is required. Solar appScreener determines a version automatically. Also, analysis module optimization helps to scan apps on JAVA, Scala, Kotlin and Android faster. Besides, to create new users and groups faster multiple selection from users and projects drop-down lists was implemented.

What`s new

- New programming languages supported:
  - Rust.
- New interface functionality:
  - added Jira task ID in Detailed Results;
  - implemented creation of a new export template as a copy of an existing one;
- New report functionality:
  - added number of vulnerabilities in Vulnerability Table and Comparison Table: the total number and the number of included vulnerabilities;
  - added Jira information. Now you can include brief Jira tasks information in the report: ID, parent task, task type, priority, and assignee.
- New integration functionality:
  - implemented integration with Subversion. Now you can start analysis using a link to your Subversion repository;
  - added credentials input fields (Username and Password) for private repositories code analysis (Git, Subversion);
  - updated report settings in plugins: Jenkins, Azure DevOps Server, TeamCity. All export settings implemented in UI 3.4.0 version are now available in plugins. The same updates were made for CLT export settings (Command Line Tool).
- New distributions capability:
  - implemented experimental support for the PostgreSQL database.

What has been improved

- Analysis modules improvement:
  - Configuration files: implemented a full-fledged analysis module:
    - analysis settings: instead of the Analyze configuration files option, use the Config files checkbox in the list of languages;
    - overview: added analysis module statistics: progress, time, number of analyzed code lines, number of vulnerabilities (previously analysis statistics for configuration files were implicitly included into the analysis results);
    - detailed results: you can show/hide vulnerabilities found in configuration files using the corresponding filter.
  - Java, Scala, Kotlin, Android:
    - supplemented standard libraries list. Use the filter in Detailed results to hide vulnerabilities found in standard libraries. Full list of supported libraries is in the user guide;
    - optimized analysis module. Now, on average, applications are analyzed faster.
  - Python: implemented automatic version determination. Choosing a version of Python when starting the analysis is no longer required;
  - C/C ++: implemented Makefile support. Now you can analyze codebases that are built using Makefile.
  - vulnerability traces (data flow graphs) have been improved. Added trace elements for
    - method declarations;
    - method parameters.
- User interface improvement:
  - added multiple selection from users and projects drop-down lists: creating new users and user groups has become faster;
  - improved quick action buttons in the Projects section: implemented the ability to open pages in new tabs.
- Vulnerability scanning rule base improvement:
  - improved vulnerability search algorithms;
  - added new vulnerability search rules for supported programming languages:
    - Apex: 1;
    - C/C++: 10;
    - C#: 3;
    - COBOL: 1;
    - Config: 8;

    - Delphi: 1;
    - Go: 1;
    - Java/Scala/Kotlin/Android: 1;
    - Javascript: 20;
    - Objective C: 32;
    - Perl: 7;
    - PHP: 11;
    - PL/SQL: 1;
    - Ruby: 1;
    - Rust: 17;
    - T-SQL: 1;
    - TypeScript: 1;
    - VB.NET: 30;
    - VB6: 1;
    - VBA: 1;
    - VBScript: 1;
    - 1C: 1.
  - supplemented vulnerability descriptions.

What has been fixed

- User interface features:
  - Breadcrumbs wrapping;
  - Exclude from analysis tooltip position on the Home page;
  - Example tab editing in Rule Management: now you can save your changes;
  - Make as default template option in the Export Report settings;
  - Sorting user groups by user number;
  - Work with project groups for all users role.
- Report features:
  - Missing axes on Scan History graphs;
  - Report downloading without selecting scans: you can export project summary without details of specific scan;
  - Option to include/exclude Table of Contents and Scan Information.
- Integrations features:
  - LDAP: user registration date.