ActiveState

ActiveState Software Supply Chain Security

Ensuring the security and integrity of the components and processes in use across your software development lifecycle is key to securing your software supply chain. In practice, this means being able to do:

- **Observability** - verify the provenance (i.e., origin) of all imported third-party tools, libraries, code snippets, packages etc, as well as the integrity and security of any software artifact produced or consumed during the software development process.
- **Scalability** - securely compile, build and/or package code using a hardened build system that's resistant to tampering.
- **Rapid Remediation** - discover and remediate vulnerabilities and other security issues quickly, as they arise.

ActiveState provides a number of features that can help you achieve all of these goals.

Software Attestations

Software attestations are key to establishing trust for the third-party software you use by offering a way to independently validate its security and integrity. Simply put, provenance is metadata (information) about how a software artifact was produced, including the origin or source for all components used in/by the build process, such as:

- The origin of the proprietary source code (e.g., a git repository)
- The origin of all dependencies (both prebuilt dependencies and source code)

The ActiveState Platform vendors all of the Python, Perl, Ruby, Tcl & C source code required to build each dependency, and generates a signed attestation for each of them as part of the build process, enabling users to verify the security and integrity of all the components they work with.

Secure Build Service

The ActiveState Platform vendors all source code, and implements a number of controls to ensure the integrity and security of our build process, including:

- **Hardened Build Service** - a dedicated service that runs on a minimal set of predefined, locked-down resources.
- **Scripted Builds** - build scripts cannot be accessed and modified within the build service, preventing exploits.
- **Ephemeral, Isolated Build Steps** - every step in a build process executes in its own container, which is discarded at the completion of each step to avoid cross-pollution.
- **Hermetic Environments** - containers have no internet access, preventing (for example) dynamic packages from including remote resources.

The output is a reproducible build in which the same inputs produce the same outputs every time.

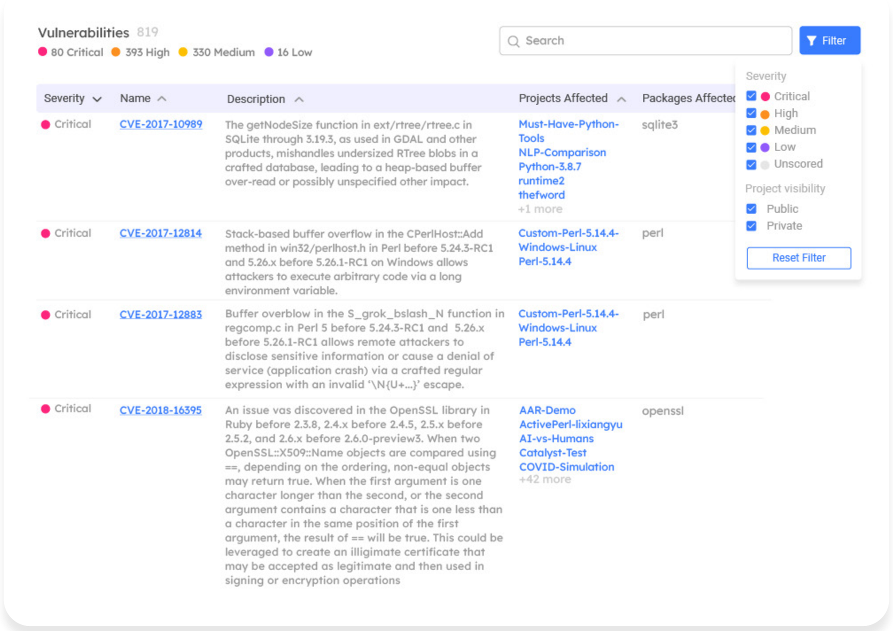
SBOMs

A Software Bill of Materials (SBOM) is a machine-readable list of all the ingredients in an application, service, API or runtime environment. SBOMs also contain detailed information about each software component, allowing organizations to more easily track software usage across multiple seemingly unrelated apps that contain common components. As a result, organizations can (for example) identify vulnerabilities more quickly and efficiently.

The ActiveState Platform programmatically generates both JSON and SPDX SBOMs for Python, Perl, Ruby and Tcl runtime environments. Unlike other solutions, ActiveState maintains a history of all changes made to the environment, and can regenerate the SBOM for each, providing organizations with a forensic audit trail.

CVE Dashboard

The ActiveState Common Vulnerabilities and Exposures (CVE) Dashboard provides a single, comprehensive view of all Python, Perl, Ruby and Tcl vulnerabilities found in packages, dependencies, transitive dependencies and OS-native binaries located in projects across the entire enterprise, enabling more effective risk assessment and remediation efforts:



Vulnerabilities 819
● 80 Critical ● 393 High ● 330 Medium ● 16 Low

Severity	Name	Description	Projects Affected	Packages Affected
● Critical	CVE-2017-10989	The getNodeSize function in ext/rtree/rtree.c in SQLite through 3.19.3, as used in GDAL and other products, mishandles undersized RTree blobs in a crafted database, leading to a heap-based buffer over-read or possibly unspecified other impact.	Must-Have-Python-Tools NLP-Comparison Python-3.8.7 runtime2 thefword +1 more	sqlite3
● Critical	CVE-2017-12814	Stack-based buffer overflow in the CPerlHost::Add method in win32/perlhost.h in Perl before 5.24.3-RC1 and 5.26.x before 5.26.1-RC1 on Windows allows attackers to execute arbitrary code via a long environment variable.	Custom-Perl-5.14.4-Windows-Linux Perl-5.14.4	perl
● Critical	CVE-2017-12883	Buffer overblow in the S_grok_bslash_N function in regcomp.c in Perl 5 before 5.24.3-RC1 and 5.26.x before 5.26.1-RC1 allows remote attackers to disclose sensitive information or cause a denial of service (application crash) via a crafted regular expression with an invalid '\N{U+...}' escape.	Custom-Perl-5.14.4-Windows-Linux Perl-5.14.4	perl
● Critical	CVE-2018-16395	An issue vas discovered in the OpenSSL library in Ruby before 2.3.8, 2.4.x before 2.4.5, 2.5.x before 2.5.2, and 2.6.x before 2.6.0-preview3. When two OpenSSL::X509::Name objects are compared using ==, depending on the ordering, non-equal objects may return true. When the first argument is one character longer than the second, or the second argument contains a character that is one less than a character in the same position of the first argument, the result of == will be true. This could be leveraged to create an illigimate certificate that may be accepted as legitimate and then used in signing or encryption operations	AAR-Demo ActivePerl-lixiangyu AI-vs-Humans Catalyst-Test COVID-Simulation +42 more	openssl

Uniquely, security teams can Immediately update projects with newer (fixed) versions of vulnerable packages, automatically rebuild the runtime environment ready for testing, and thereby decrease Mean Time To Remediation (MTTR).

ActiveState
Secure open source integration

www.activestate.com
Toll-free in NA: 1-866.631.4581
solutions@activestate.com