Sonatype

To Fix It, You Must Find It

Managing Open Source Components Using Advanced Binary Fingerprinting

SONATYPE | www.sonatype.com | sales@sonatype.com
+1 301-684-8080 | 12501 Prosperity Drive, Suite 350, Silver Spring, MD, 20904

BACKGROUND

For the most part, modern software is assembled, not written. More than 80 percent of a typical software application is assembled from open source binary components and frameworks, with custom business logic making up the remaining 20 percent.

This massive reliance on open source components has created new challenges for managing software security, quality and intellectual property. Organizations who rely on custom software are increasingly seeking visibility and control to manage risk and maximize benefit.

But to properly manage open source components, you must know as much as possible about them – starting with precisely identifying them. Security, quality and licensing information is of little use if it's not precisely tied to the component in question.

Component identification is far from straightforward. There are three common mechanisms for identifying binary components:

- **Source Code Scanning**
- **Simple Binary Matching**
- **Advanced Binary Fingerprinting**

SOURCE CODE SCANNING

Source code scanning technologies examine custom code and flag potential matches against fingerprints of known open source code. Source code scanning is a thorough and effective mechanism for surfacing potential 'snippets' of open source code that might have been inserted into otherwise proprietary work.

Source code scanning is, however, ineffective at precisely identifying the binary component with which that code is associated. This is because of the very nature of the component ecosystem. Components are enormously complex; each one is made up of hundreds of sub-assemblies (e.g. class files). Class files are commonly shared among components. Of the nearly 200 million class files in the Central Repository, there are fewer than 10 million unique class files being combined in myriad ways.

Because of this highly commingled nature of the component ecosystem, source code scanners are unable to precisely match a 'code snippet' to a single binary component (and version). The code scanner will often find dozens to hundreds of potential matches, rendering the analysis of any security, quality or licensing information futile.

Users of these technologies frequently report that they "receive too many false positives" or that the scanners require "too much manual research." What they are really observing is that the technology itself is not well suited to the task of binary matching.

Vendors include: Black Duck, Palamida, Coverity and Fortify

SIMPLE BINARY MATCHING

Simple binary identification is achieved by one of two methods. The first simply matches the components by name. This technique is the most simplistic and is the most prone to both false negatives and false positives.

The second technique matches binaries based on the cryptographic hashes of inspected components against a known library of component hashes. This allows precise identification of binary components provided they have not been modified in any way.

Unfortunately, many users modify components in some way that alters the identifying signatures (e.g. re-packaging, rebuilding, combining components, removing unused classes, etc.). When binaries are altered, simple binary matching fails and components are not properly identified.

Users of these technologies often report "false negatives" or "misses."

Vendors include: Black Duck, Palamida, Protecode, OpenLogic

ADVANCED BINARY FINGERPRINTING

To provide precise binary matching without the false positives of source code scanning or the false negatives of simple binary matching, Sonatype has invented a new, patent-pending method called Advanced Binary Fingerprinting. With this method, Sonatype is able to identify the unique combinations of subcomponents that are uniquely identifiable as the specific version of a given component.

Advanced Binary Fingerprinting is able to precisely identify components even when they have been repackaged, rebuilt, or otherwise altered. This method allows proper assignment of security, quality and licensing data to a specific component and version. Advanced binary matching is both fast and precise. A large application can be analyzed and a precise bill of materials delivered in minutes.

Vendors include: Only Sonatype offers Advanced Binary Matching

THE SONATYPE ADVANTAGE

Sonatype brings practical intelligence to component-based software development. Sonatype offers distinct advantages to organizations that need to improve visibility and control over the open source components that they use in software development.

- **Sonatype pioneered component-based software development as the creators of the Apache Maven build system and the Nexus repository manager.**
- **Sonatype is also the creator and exclusive operator of the Central Repository, the industry's primary source for open source components, containing more than 300,000 components and serving more than 4 billion requests every year.**
- **Sonatype is the only vendor to offer actionable component security, quality, and licensing information directly into the tools developers use every day.**
- **Sonatype is the only vendor offering patent-pending Advanced Binary Matching to quickly and precisely identify components, even if they have been altered or repackaged. This unique technology minimizes 'false positives' and eliminates time-consuming manual research.**
- **Sonatype is the only vendor to provide real-time update notifications to alert users when a component they are using has been updated or changed.**
- **Sonatype is the only vendor to analyze all component dependencies enabling quick and precise identification of potential issues, even if they're nested deep within a complex dependency tree.**

For more information about Sonatype, visit www.sonatype.com.