Sonatype

WHITEPAPER

Introducing Component Lifecycle Management (CLM):

How to Improve Productivity while Minimizing Risk in Open Source Application Development

Table of Contents

About Sonatype

Sonatype has been on the forefront of creating tools to manage, organize, and better secure components since the inception of the Central Repository and Maven in 2001. Today, over 70,000 companies download over 8 billion components every year from the Central Repository, demonstrating the explosive growth in component-based development. Today's software ecosystem has created a level of complexity that is increasingly hard to manage. Partnering with application developers, security professionals and the open source community, Sonatype has introduced a way to keep pace with modern software development without sacrificing security. We call it Component Lifecycle Management (CLM), the new platform for securing the modern software supply chain.

We believe that to achieve application security, the approach has to be simple to use and integrated in the tools developers already use everyday. With CLM we're improving the visibility, management and security of component-based development across the entire lifecycle. Together with our customers, we're ushering in a new era of application security.

Executive Summary: Modern Applications Call for a New Approach

The last ten years have seen a revolution in the way software is developed and delivered. Organizations have moved away from tedious custom source code programming in favor of quicker component-based approaches.

Most often, a component is open source software that is shared among developers via public repositories, such as the Central Repository, and is downloaded and assembled into applications. More than 400,000+ components are available, such as web frameworks that form the foundation for web applications, logging mechanisms and database access routines to name just a few.

This new development model is agile, enabling organizations to develop faster, reduce cost and improve efficiency; all of which are necessary to meet the demands of the business. Not surprisingly, today more than 80 percent of a typical software application is comprised of components.

However, most application security technologies are designed for source code not components, leaving organizations vulnerable to potential threats. As a result, organizations are vulnerable to threats stemming from shared components, such as security breaches, intellectual property claims, as well as application instability and performance defects.

Furthermore, development teams lack visibility into component usage. It is difficult to know what components were used, identify where they were used and evaluate current security risks. What's more, vulnerabilities often are nested deep within an application and are not easily apparent.

This mismatch between development methods and application security tools has forced a dangerous tradeoff between speed and security. The development and security departments are at odds, with development teams pressured to deliver quickly while the security officer's mandate is to minimize risk. A new approach is needed to secure applications at the pace and scale that business demands. Organizations that fail to address this issue are leaving 80 percent of their application code at risk.

A new category of solution has emerged called Component Lifecycle Management (CLM). The key tenets of this approach include:

- Defining component usage policies and automating enforcement using the tools developers use everyday
- Empowering developers to choose components based on favorable security, licensing, or architectural criteria
- Enabling flexible governance without disrupting the development process
- Providing comprehensive understanding of component usage and risks across the entire software lifecycle
- Continuously monitoring for newly discovered threats, even in older components

This white paper describes the opportunities and challenges presented by component-based software development, and how CLM can amplify the benefits while also reducing the risk.

A Shifting Software Development Landscape

Software Development Trends

IT ONCE WAS...	IT NOW IS...
Waterfall Methodology	Agile Development
Code-based	Component-based
Developed	Assembled
Independent	Collaborative
Proprietary	Open Source

Modern software development is increasingly component-based – In the early years of software development, applications consisted primarily of custom developed code and internally developed components with only a very small fraction of code sourced from outside the organization.

Today's applications are assembled – Developers now assemble applications from existing components often sourced from outside the organization, rather than relying on custom source code.

Agile development has replaced waterfall – In the past, development efforts followed a "waterfall" methodology, a highly structured and sequential process where projects spanned months or even years. Now, the modern development process is rapid, continuous and collaborative. Delivery and iteration have replaced lengthy requirements development processes.

Open source has become an integral part of modern applications – In most cases externally sourced components are from open source communities. Modern applications often rely on hundreds of open source components and frameworks. Use of open source components supports both efficiency and innovation.

The shift in the software development landscape has contributed to vast efficiency gains and cost savings. However, these changes have also introduced new risks and requirements for the modern software development organization.

Modern Benefits, Modern Problems

While component-based software development is widely accepted and growing exponentially, the complexities and risks are just now becoming clear.

Components are Pervasive

Demand for open source components is skyrocketing, with requests from the Central Repository, the industry's primary source for open source components, increasing 800% over the past five years to nearly 8 billion in 2012 and projections of up to 20 billion in 2013. [See Figure 1]

FIGURE 1: Demand for open source components is skyrocketing, with requests from the Central Repository, the industry's primary source for open source components, increasing 800% over the past five years to nearly 8 billion in 2012.

Thousands of Sonatype "Application Health Check" assessments have confirmed that the average application is 80 percent built from components and a recent survey of 3,500 developers confirmed this statistic.

Flawed components introduce significant risk

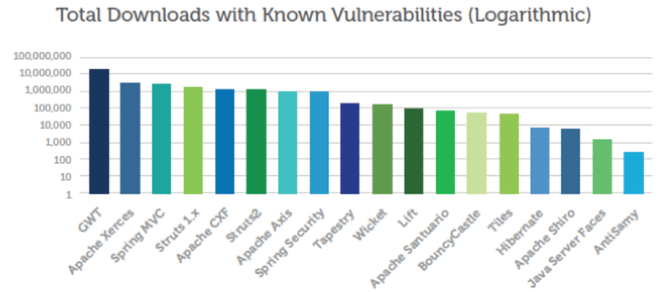
Whether provided by commercial vendors or open source initiatives, components can introduce significant management, security and licensing challenges. Recent analysis of the Central Repository by Aspect Security uncovered widespread security vulnerabilities among the most commonly used open source components. However, few organizations have the proper controls in place to mitigate the risks posed by flawed components. [See Figure 2]

Component complexity exacerbates the problem

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 total class files in the Central Repository, there are fewer than 10 million unique class files being combined in myriad ways. [See Figure 3]

To add to the problem, components may also depend on other components. These relationships, known as transitive dependencies, can be difficult or impossible for developers to understand, track and support without tools designed to manage such a complex supply chain. Component dependencies can introduce security breaches, intellectual property claims, as well as application stability and performance defects. Often, risks are caused by a flawed component nested deep in an application's dependency tree and flaws are not easily apparent.

Security vulnerabilities in open source libraries and frameworks are widespread



Few organizations have a strong process for minimizing component risk

Control of artifacts in development

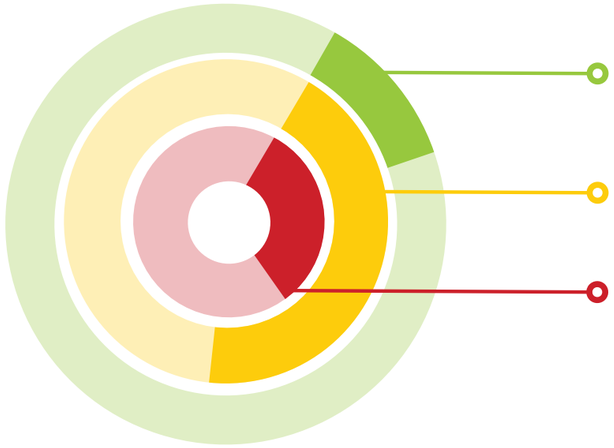


FIGURE 2: Recent analysis of the Central Repository by Aspect Security uncovered widespread security vulnerabilities among the most commonly used open source components. However, few organizations have the proper controls in place to mitigate the risks posed by flawed components.

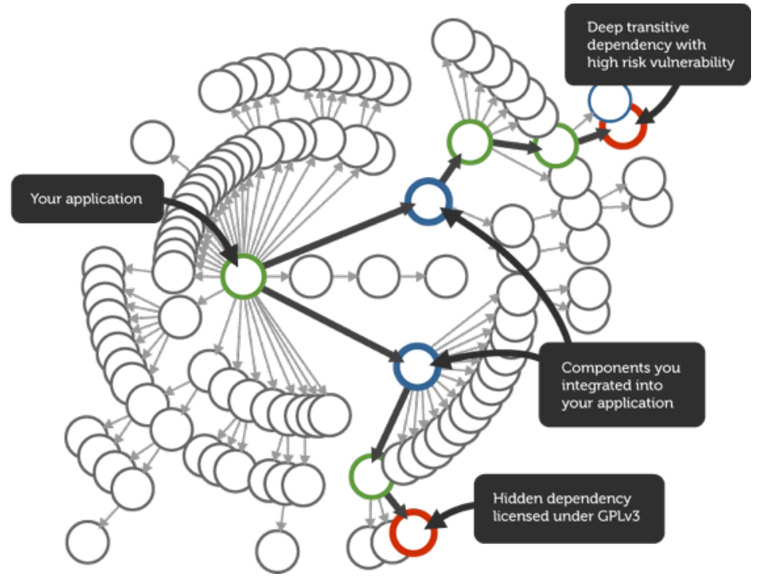


FIGURE 3: Components are enormously complex; each one is made up of hundreds of sub-assemblies (e.g. class files). Class files are commonly shared among components. Of the nearly 200 million total class files in the Central Repository, there are fewer than 10 million unique class files being combined in myriad ways.

Organizations lack actionable security, quality and licensing information.

It is difficult and time consuming for developers to research and determine security, quality and licensing characteristics for all of the components they use to assemble their applications. To do this for direct dependencies is hard enough; to extend this research to all component dependencies is beyond reason. Even if research is conducted, it is difficult to take action because it is not integrated directly in the tools that developers use and problems are found much later in the lifecycle. Given the pressure to deliver applications quickly, developers are forced to take a chance when they select components – exposing the organization to risk.

Organizations regularly consume outdated, flawed, or insecure components

Open source projects innovate rapidly and release frequently. However, since there is no update notification infrastructure for open source components, there is no easy way for component consumers to know when a new version has been released, much less which defects have been identified and fixed. [See Figure 4] This causes organizations to consume outdated, flawed, or insecure components, even years after newer fixed versions are available.

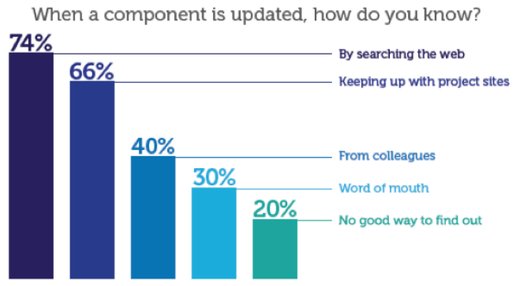


FIGURE 4: There is no easy way for component consumers to know when a new version has been released, much less which defects have been identified and fixed.

Organizations don't have the capacity to manage newly discovered flaws

These days, more and more organizations are aware of and support the use of open source components, however they are unaware of the complexity, unsure of the number of components used and unclear about where they are used. [See Figure 5] When a new defect or security flaw is discovered, many organizations are left exposed, unaware of where or how they are using the affected component. It is a challenge that impacts the full software supply chain.

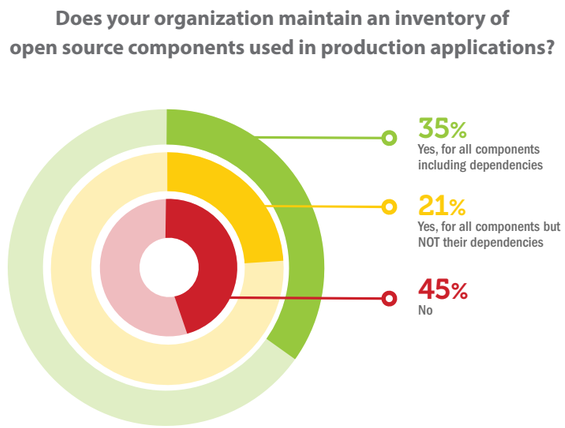


FIGURE 5: These days, more and more organizations are aware of and support the use of open source components, however they are unaware of the complexity, unsure of the number of components used and unclear about where they are used.

Agile development and geographically dispersed development teams adds to the complexity.

Agile software development projects - with their rapid iterations, continuous integration builds and continuous deployment - have all resulted in many more releases over the life of a software project. To support this agile process, developers need real time compliance information with the ability to quickly eliminate any issues. Waiting until the end is counter productive in an agile environment.

Increasingly development teams are geographically dispersed, often including external contractors. Keeping disparate teams in sync and enforcing standards adds yet another layer of complexity to component risk management.

Restrictive, approval-laden policy approaches don't work.

Some organizations attempt to manage component usage by implementing restrictive policies. If a developer wants to use a new component, approvals are needed from the security, legal, and architecture teams. Even if the approval process is "automated" using workflow, it can't keep up with the scale or pace of development. This forces developers to delay their development cycles, work around the policy, or pick from sub-optimal (e.g. out-of-date) components that were previously approved.

Security tools for custom code deliver results late in the development cycle.

Many organizations have turned to application scanner technologies or application lifecycle management to address security concerns. Although these tools play a role in a layered security strategy, they aren't designed for components that make up the majority of applications. Scanning tools are designed to evaluate risk in custom source code – providing results that are delivered after the fact, late in the development lifecycle. Components require a different approach. Known security, licensing and quality intelligence should be integrated throughout the development lifecycle, preventing problems and remediating flaws fast and early in the application lifecycle.

The Case for Component Lifecycle Management (CLM)

A new category of software development products and information services has emerged to help ensure the integrity of the modern software supply chain, amplifying the benefits of modern development while reducing risk.

An effective CLM platform has three critical attributes:

Actionable	Integrated	Continuous
Prevent and remediate flaws by leveraging accurate security, licensing & quality information	Inform and guide developers within existing software development tools	Monitor continuously to identify and remediate new risks in production applications.

Effective CLM solutions offer a platform for sharing components across teams to ensure collaboration and encourage standardization. They also offer governance infrastructure to implement and enforce policies. A CLM platform should work with open source components sourced from the community as well as custom-developed components from inside the enterprise.

With CLM, software development organizations gain the collaborative tools, intelligence and control required to address the reality and risks of agile, component-based development. Organizations that have embraced CLM have seen dramatic improvements in their development efforts. Key benefits include:

- Reduced exposure to security vulnerabilities
- Avoidance of intellectual property and licensing risks
- Compliance with open source policies without disrupting development
- Improved ability to meet regulatory compliance requirements
- Improved software quality
- Improved developer productivity and collaboration

Sonatype's Approach to Component Lifecycle Management

Sonatype Component Lifecycle Management (CLM) is an application security platform custom-designed to secure component-based applications. CLM tracks usage, enforces policy and prevents the use of flawed components throughout the modern software supply chain. By natively integrating into the tools developers already use, risk is removed proactively which drastically reduces downstream "fix" costs. This modern approach to software assurance makes it possible to reduce risk and ensure compliance without impeding development velocity. Sonatype brings practical intelligence to component-based software development. Sonatype pioneered component-based software development with innovations such as Apache Maven, the Central Repository, Nexus, and m2eclipse.

Sonatype CLM provides a comprehensive inventory of components and associated bill-of-materials. Unique binary fingerprint matching identifies component inventory with extreme accuracy. Inventory information is provided at the component download, repository manager and application level. Sonatype's inventory capability provides the foundation for the following capabilities provided by the CLM. [See Figure 6]

SONATYPE CLM

1. Secure Consumption with the use of certified components and integrity checking throughout the lifecycle

2. Govern Development to ensure policy compliance without disrupting developer productivity

3. Profile Exposures to proactively identify and prioritize action

4. Remediate Risk by preventing and quickly fixing security and IP vulnerabilities

5. Monitor Threats in production applications to ensure continuous trust in critical operations

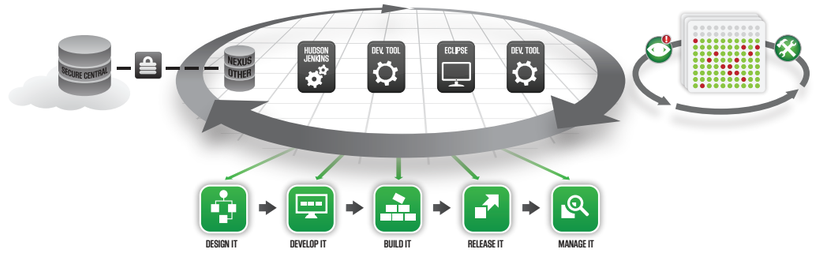


FIGURE 6: CLM tracks usage, enforces policy and prevents the use of flawed components throughout the modern software supply chain.

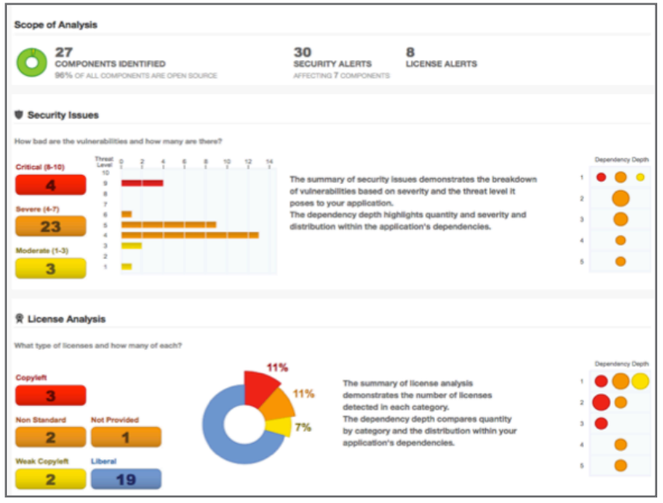


FIGURE 7: Secure Consumption - Sonatype CLM authenticates downloaded components including security, licensing and quality information.

Secure Consumption

Sonatype CLM ensures trust in the software supply chain by authenticating and securely delivering components. [See Figure 7] Sonatype CLM gives you:

- **OSS project validation to ensure high quality, trusted components –** Sonatype provides component intelligence for the components that are checked into the Central Repository. This security, licensing and quality intelligence is used to drive policy and governance throughout the software lifecycle.
- **Authentication throughout the software lifecycle eliminates risk of tampering inside the firewall –** Sonatype uses strongly signed hashes and checks the integrity of the component throughout the lifecycle. This allows you to detect intentional or inadvertent changes to the component.
- **Secure component delivery to eliminate man-in-the-middle attacks –** Sonatype is uniquely positioned to secure the delivery channel between Central and your organization using SSL. This ensures that the component is not manipulated during delivery.

Governed Development

CLM provides developers with security, popularity, and licensing information making it easy to detect and prevent flaws early in the development process. This "zero-latency" approach to remediation reduces the obstacles that usually reduce developer compliance.

- **Rich security, licensing, and popularity metadata drives action in the IDE –** Developers minimize expensive downstream problems by selecting components based on security, licensing and quality intelligence integrated directly in the IDE. [See Figure 8] Component recommendations help developers speed the remediation process for flawed applications.
- **Information and policy enforcement extends across the IDE, repository, and CI server to automate and enforce governance across the entire software lifecycle –** Comprehensive component management requires assistance across the entire software lifecycle. Sonatype provides appropriate guidance in your IDE, repository manager, build and CI environments to ensure policies are enforced. Developers don't have to learn new tools, the information they need is in the tools they use throughout the lifecycle.

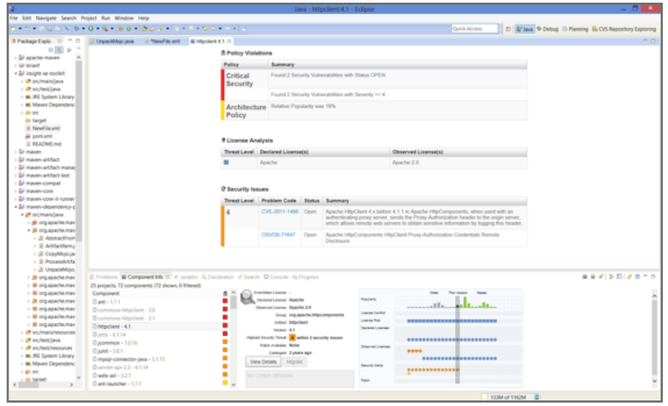


FIGURE 8: Governed Development - Developers minimize expensive downstream problems by selecting components based on security, licensing and quality intelligence integrated directly in the IDE.

Profile Exposure

CLM provides the ability to proactively identify and prioritize your actions. Vulnerabilities are proactively identified and reported in the context of your organizational policies. Developers can prioritize remediation action based on a visual threat summary of security, licensing and architecture factors. Sonatype CLM enables you to:

- Identify at-risk components: Quickly identify and prioritize remediation efforts with a visual threat indicator that summarizes the policy outcome based on security, licensing and architecture factors. [See Figure 9]

FIGURE 9: Profile Exposure - A visual threat indicator summarizes security, licensing and architecture policy considerations allowing developers to easily prioritize and take action.

Remediate Risk

CLM provides the ability to prevent and quickly fix flawed applications. Developers start with the right components and can easily fix applications directly within their IDE. Sonatype CLM enables you to:

- Prevent problems by starting with the right components: Developers can select the best components to use within the IDE based on security, licensing and quality information.
- Push-button migration: Developers can migrate to new component versions with a simple mouse click in their IDE. [See Figure 10]

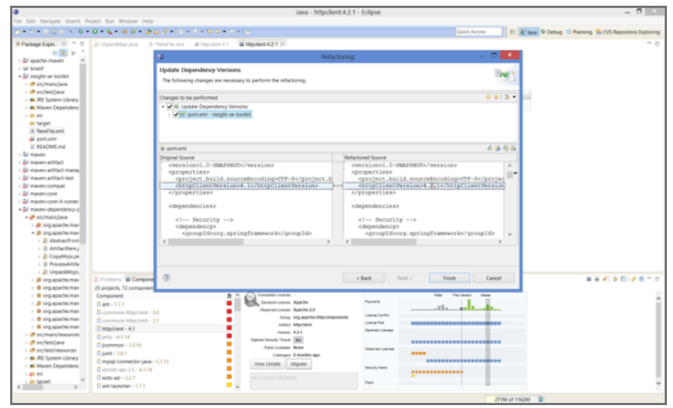


FIGURE 10: Remediate Risk - Developers can assess the versions using a side-by-side comparison and migrate to a new component version with a single mouse click.

Monitor Threats

CLM provides ongoing, continuous monitoring of both development and production applications. Sonatype proactively alerts you to new vulnerabilities that have been discovered. Sonatype CLM enables you to:

- Discover new vulnerabilities: Newly discovered security, licensing, and quality issues are identified and mapped correctly to the application inventory.
- Proactive notification: New violations are proactively reported with contextual information that expedites corrective action.
- Enterprise risk & policy assessment: Executive dashboards provide the ability to assess enterprise risk profile and policy compliance. [See Figure 11]

FIGURE 11: Monitor Threats - Dashboards and reports provide a complete view of global risk with drill-down detail to drive action.

Steps Toward Good Component Best Practices

Organizations interested in minimizing risk, while firmly establishing both control and visibility across today's complex and agile software supply chain, should take the following first steps toward CLM. These steps encompass making proactive improvements in awareness, policy and enforcement:

STEP ONE: Empower developers to choose the right components from the start

- Use security, licensing and quality information to select the right components from the start.
- Make decisions directly in the tools developers use today: IDE, Repository Manager, Build and CI tools.
- Ensure components are delivered securely and remain authenticated throughout the development lifecycle.

STEP TWO: Quickly identify your exposure & remediate flaws

- Identify at risk applications with flawed or suboptimal components.
- Quickly prioritize effort based on summarized security, licensing & architecture factors.
- Easily replace flawed components to meet policy guidelines.

STEP THREE: Precisely identify your components, repository & application inventory

- Track component downloads and usage to understand consumption.
- Assess the health of your repository & determine what is being distributed to development teams.
- Identify what is in your applications and uncover potential security, licensing, and quality problems.

STEP FOUR: Implement flexible policies that speed agile development with guidance for each lifecycle stage

- Establish policies regarding security, the use of viral licenses, and the out-of-date or out-of-version components.
- Guide development efforts with lifecycle appropriate actions: design, development, build, deploy and production monitoring.

STEP FIVE: Proactively monitor & analyze production applications to meet policy compliance goals

- Maintain an inventory of all components and dependencies used in production applications.
- Continuously monitor application bills of materials for updates and newly discovered vulnerabilities.
- Support enterprise risk profile, policy compliance analysis and reporting efforts.

Ways to Assess your Current Component Usage & Risk

Sonatype offers a variety of ways to assess risk across your current software lifecycle. View the components your organization has downloaded, the state of components in your repositories and assess risk of those used in your applications. Get detailed information about your security, licensing and quality risks.

Component Snapshot Report

We'll help you understand which components your organization has downloaded from the Central Repository, and where you might find potential security, licensing, or quality risks. Through the Snapshot Report you'll learn:

- How many components have been downloaded and when.
- Security information including severity threat levels.
- Detailed licensing information such as "copy left," "non standard" or "missing."

Repository Health Check Report

If your organization uses a Nexus Repository (also from Sonatype), you can get detailed information about known security vulnerabilities or unacceptable licenses in your Nexus repository manager.

- Discover which components are in your repository and which ones have known vulnerabilities or license issues.
- Quickly see the breakdown of vulnerabilities based on severity and threat levels.
- Drill down to see detailed information to further assess your risk.

Application Health Check Report

The Application Health Check provides visibility into the components in use within your enterprise applications. With tools provided by Sonatype, in minutes you'll be able to:

- Analyze and understand the composition of any component-based application.
- Uncover potential security, licensing or quality problems.
- Check your applications and code from your suppliers to obtain an accurate view of vulnerabilities introduced to your organization from a third party.

For more information on assessing your risk: visit http://www.sonatype.com/go-fast-be-secure

For a quick online tour of the Sonatype CLM solution visit www.seehow.org.

For general information, visit: www.sonatype.com

Sonatype Inc. · 8161 Maple Lawn Drive, Suite 250 · Fulton, MD 20759 · 1.877.866.2836 · **www.sonatype.com**