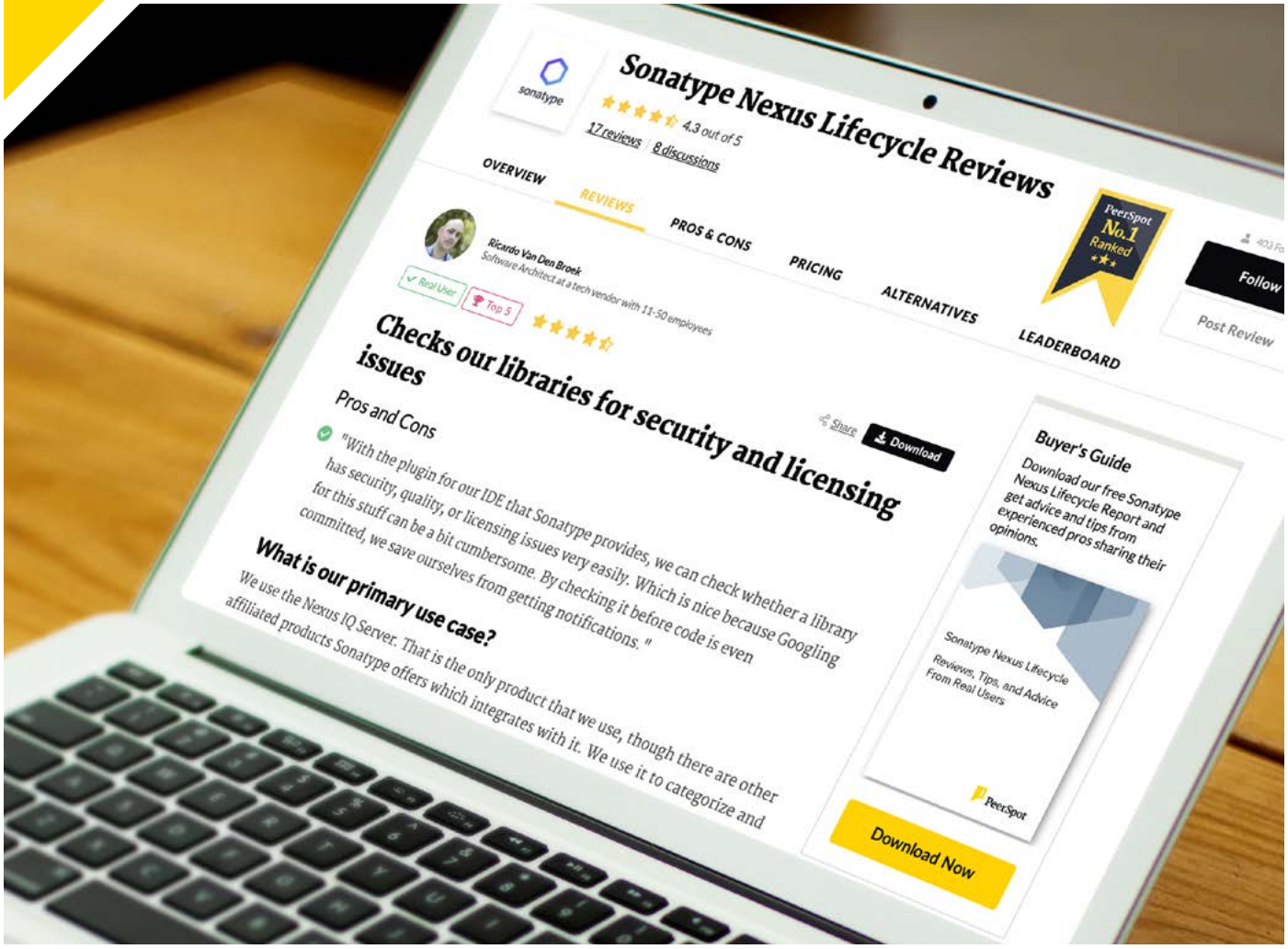
PeerPaper™ Report 2022

Based on real user reviews of Sonatype Nexus Lifecycle

The Top 10 Mistakes when Implementing a Secure Software Supply Chain Solution



PeerSpot

InfoWorld
FROM IDG

Contents

Page 1. **Introduction**

Page 2. **Mistake #1:**
Not Focusing on Developer Productivity

Page 4. **Mistake #2:**
Not Integrating with DevOps Tooling and More

Page 6. **Mistake #3:**
Not Concentrating on Data Accuracy and a Low Rate of False Positives

Page 7. **Mistake #4:**
Not Staying on Top of License Information

Page 8. **Mistake #5:**
Not Blocking Undesirable Components

Page 10. **Mistake #6:**
Not Planning Across Silos

Page 11. **Mistake #7:**
Not Focusing on Customized Policy Enforcements Across the SLDC

Page 12. **Mistake #8:**
Not Having a Strategy and Goals Across DevOps and AppSec Teams

Page 13. **Mistake #9:**
Not Improving Speed-to-Market for Secure Applications

Page 15. **Mistake #10:**
Not Starting as Soon as Possible

Page 16. **Conclusion**

Introduction

The inclusion of open-source code has become ever more common in applications. As the volume of open-source packages continues to rise, insecure components are increasingly finding their way into software supply chains around the world. The need to secure the software supply chain is not surprising. There are many ways in which open-source components can be exploited, leading to major security breaches for organizations using applications that run the compromised code.

In practice, secure software supply chain solutions begin at the open-source management level. Teams need to ensure that components are identified and patched for any vulnerabilities before they enter an organization's supply chain. Third-party libraries being downloaded from open-source ecosystems with both known and unknown vulnerabilities should be retired as soon as possible. Only secure versions should be available to developers. Sonatype's Nexus Lifecycle solution allows teams to secure their software development life cycle at scale, but it is imperative that users are aware of common mistakes made when teams are implementing solutions to manage this risk. Here are the ten most common mistakes.

Mistake #1:

Not Focusing on Developer Productivity

Developer productivity is a common challenge for organizations looking to secure their software supply chains. Developers need to stop using insecure components and begin using alternative versions as soon as possible after they have been identified. This may not affect the initial development of applications, but security needs to be a part of developer workflows up front to minimize risk to applications post-deployment and changes to vulnerable open-source components over time.

According to Sonatype Nexus Lifecycle users, developer productivity improves substantially when the right security supply chain software is deployed. For example, a DevOps Engineer who uses Nexus Lifecycle at a small tech vendor found that both accuracy and productivity increased when they deployed Sonatype's solution within the company. The team did not need to build as many items and as a direct result, "...we were much more easily able to work together, to collaborate, and consume other teams' products."

> **"...we were much more easily able to work together, to collaborate, and consume other teams' products."**
>
> **Read review »**

Increased productivity

A Senior Enterprise Architect from MIB Group, a small insurance company, concurred with this assessment. He believed his developers definitely spent a lot less time looking for components or libraries to download. He elaborated by saying, "There was a very manual process to go through, before Nexus, if they wanted to use a particular open-source library. They had to submit a request and it had to go through a bunch of reviews to make sure that it didn't have vulnerabilities in it, and then they could get a 'yes' or 'no' answer. That took a lot of time."

Now, the developers can download the program and the enterprise security team can look into potential vulnerabilities associated with it. He added, "The security team then says, 'Yeah, we can live with that,' or 'No, you have to mitigate that,' or 'No, you can't use this at all.' We find that out very much earlier in the process now."

Another Computer Architecture Specialist, who works at an energy/utilities company with over 10,000 employees, agreed that Nexus Lifecycle "has helped to increase our productivity a lot, especially with Nexus Repository Manager. It is way more agile. There is no comparison between our productivity before and now."

Mistake #2:

Not Integrating with DevOps Tooling and More

Integration with other developer tooling is an important requirement of organizations. This practice ensures that software supply chain security can be fully integrated into an organization's development environment, rather than just being used as another standalone application. Figure 1 shows some of the ways a secure software supply chain solution can integrate into the software development lifecycle (SDLC).

When a VP/Senior Manager at a financial services firm with over 1,000 employees was in the decision phase of buying an application security solution, this type of integration was of "critical importance" to his team. In fact, he went on to say that they, "built it directly into our continuous integration cycles and that's allowed us to catch things at build time, as well as stop vulnerabilities from moving downstream."

A Security Analyst at a small software company believes that since the integration with Nexus Lifecycle, their company

> "Since integrating this...we have moved to a proactive approach, allowing auditing and decisions on mitigation before any incoming client submissions."
>
> Read review »

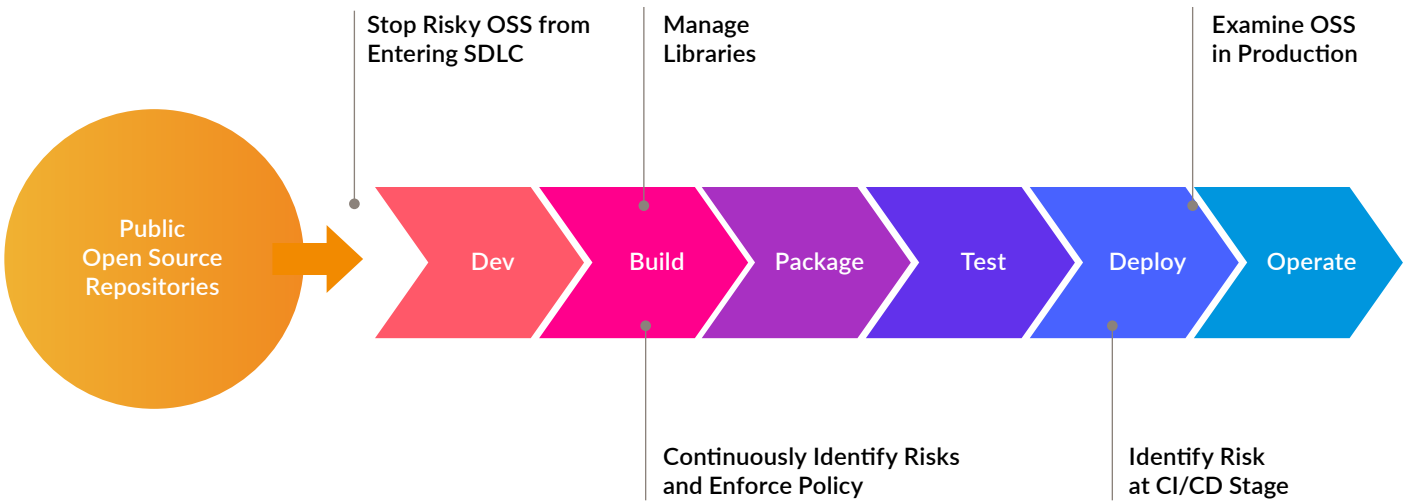


Figure 1 – Integration of a secure software supply chain solution at various stages in the software development lifecycle (SDLC).

has been able to take a more proactive approach. In his own words, "Before we had this in place, we had a much more reactive approach to CVE listings. Since integrating this, and as we've refined our process over the past eight months or a year, we have moved to a proactive approach, allowing auditing and decisions on mitigation before any incoming client submissions."

Meanwhile, the same Security Analyst also noted that this integration was a very simple process. As he said, "We didn't have any problems integrating it. And from what I did see, it looks like a very simple integration, just adding it straight into Jenkins. It integrated quite quickly into the environment."

Mistake #3:

Not Concentrating on Data Accuracy and a Low Rate of False Positives

A low rate of false positives is critical in order to avoid decreasing developer productivity and distrust with security teams. False positives are alerts that indicate vulnerabilities present in components when they are not actually there. Developers and other team members can waste time chasing them down, only to find out that they need not have bothered. This is frustrating and leads to "context switching" that's a further drag on focus.

PeerSpot members want to avoid this scenario. For example, when a Senior DevOps Engineer at Primerica, an insurance company with over 1,000 employees, was looking at different software options, he evaluated Black Duck. However, he "was not impressed by them." In particular, the reviews he read said that "Black Duck came up with more false positives than Sonatype." The software company Security Analyst was able to steer clear of false positives with Sonatype as well. Overall, he found "the data quality does help us solve problems faster."

Helps solve problems faster

Mistake #4:

Not Staying on Top of License Information

License information may also be inaccurate or out of compliance. Teams need to stay up-to-date on any version releases, patches, and license issues for each component. A Product Strategy Group Director at a tech services company with over 1,000 employees spoke to this issue. He liked that Nexus Lifecycle makes sure that his company's products are licensed correctly. He went on to say, "There are also security reasons for making sure that our products aren't introducing vulnerabilities and, if they are, that we can address them."

A Software Architect at a small tech vendor also appreciated the fact that Nexus Lifecycle makes sure his company's products are using the correct licensing. He remarked, "Sometimes with open-source software, the license is a bit more restrictive than might be convenient for what you are doing. Maybe it doesn't allow you to make changes to the library. Or, it's free to use for nonprofits, but if you're using a product which does make a profit, then you might have to purchase a license. Therefore, it protects us from accidentally misusing open-source software and is protection against legal issues."

> **"...it protects us from accidentally misusing open-source software and is protection against legal issues."**
>
> Read review »

Mistake #5:

Not Blocking Undesirable Components

Insecure open-source components should be blocked from a software supply chain. This includes issues such as vulnerabilities, piracy and incompatible licenses. According to MIB Group's Senior Enterprise Architect, his team is able to define what risk level they are willing to take on to block undesirable open-source components from entering their development lifecycle. He shared, "We define what sort of level of risk we're willing to take. For example, for 'security-critical,' we could just fail them across the board; we don't want anything that has a security-critical."

He also makes use of an "architecture blacklist" which does not permit certain components to be used from an architectural standpoint. He went on to say, "For example, we don't want anybody to build anything with Struts 1. We put it on the architecture blacklist. If a component comes in and it has that tag, it fails immediately."

> "It is very powerful, and it brings a lot of security to us. It can block undesirable open-source components from entering our development lifecycle."
>
> Read review »

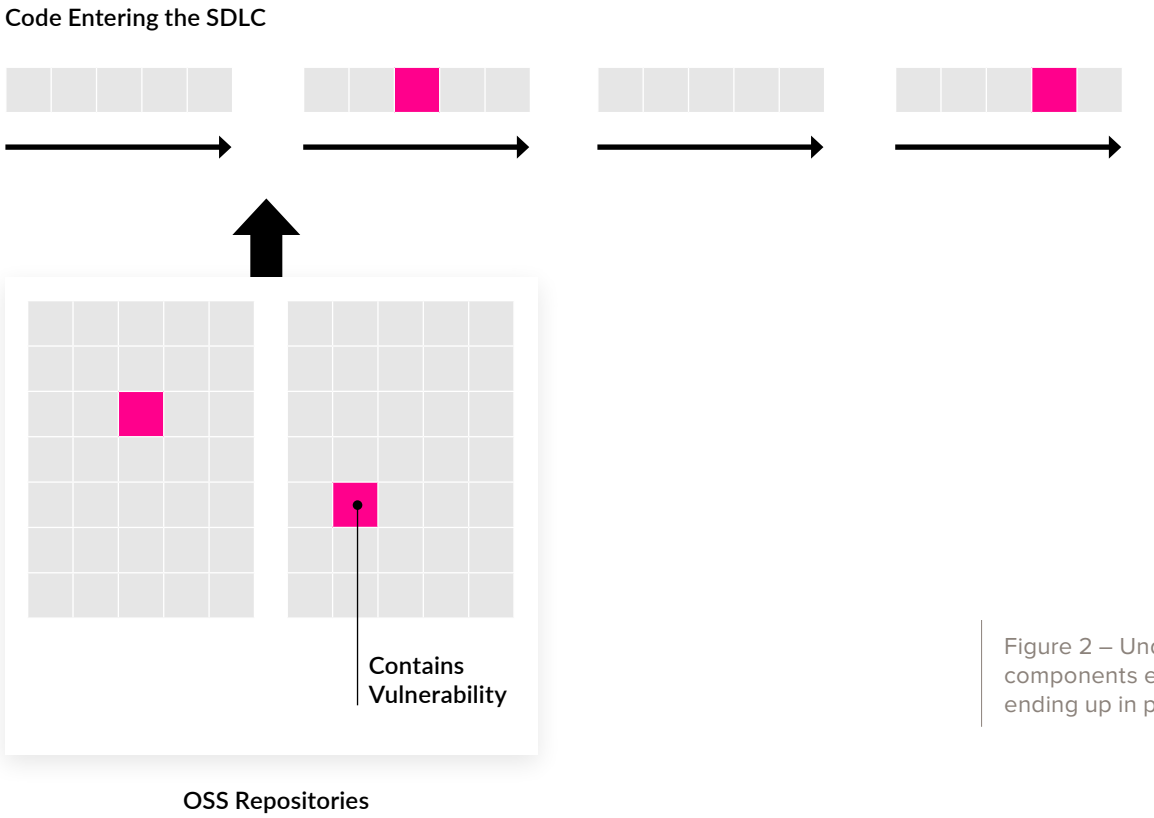


Figure 2 – Undesirable code components entering the SDLC, ending up in production applications.

"50% of the code that we use is open-source. So, it is important to scan it for all kinds of vulnerabilities," said a Product Owner Secure Coding at a financial services firm with over 10,000 employees. He agreed that Nexus Lifecycle brings a lot of value to them when it comes to open-source scanning and third-party dependency scanning. He mentioned, "It is very powerful, and it brings a lot of security to us. It can block undesirable open-source components from entering our development life-cycle." Figure 2 offers a simple illustration of this risk.

Mistake #6:

Not Planning Across Silos

One common mistake made when implementing a secure software supply chain solution is not planning across silos within an organization. Teams need to ensure that everyone – including security, development, and DevOps teams – is aware of the new processes and measures required for open-source security and license management.

A VP and Senior Manager at a financial services firm with over 1,000 employees concurred with this assessment. Specifically, he advised that all stakeholders should be involved in the early stages of planning. He elaborated, saying companies should "have a process and people involved to deal with the findings."

An Information Security Program Preparer / Architect at Alef Education, an educational organization with more than 200 employees, liked the level of awareness that using Nexus Lifecycle brings to a company. He highlighted, "We have 14 development teams, but when we started the program there were 10. The number of development teams continues to increase and they use different tools and techniques in the CI/CD. From my side, in security, the idea is to make them aware. This would be the same whether the product was Sonatype or something else. Making them aware has been a very big challenge for me, to onboard them and make the product effective."

All stakeholders involved in the early stages of planning

Mistake #7:

Not Focusing on Customized Policy Enforcements Across the SLDC

Another common mistake relates to the enforcement of different types of custom policies across a variety of application types. Teams need to decide whether they want to enforce stricter standards on components with licenses that are more permissive, or vice versa.

A VP and Senior Manager at a financial services firm with over 1,000 employees praised Nexus Lifecycle for bringing open-source intelligence and policy enforcement across the SDLC. In his words, "It enforces the SDLC contributors to only use the proper and allowed libraries at the proper and allowed time in the life cycle of development."

The Product Strategy Group Director also appreciated the open-source intelligence and policy enforcement across the SDLC. He explained, "As the teams are setting up their development environments, we have now gotten them to build Sonatype into their development pipeline. They scan their codebase so they actually catch things at the point that they introduce new, open-source software into the products, to make sure they're not actually introducing vulnerabilities or licensing-policy breaches."

> **"It enforces the SDLC contributors to only use the proper and allowed libraries at the proper and allowed time in the life cycle of development."**
>
> Read review »

Catches vulnerabilities or breaches

Mistake #8:

Not Having a Strategy and Goals Across DevOps and AppSec Teams

Organizations need to be aware of the potential impact of not implementing a specific software supply chain strategy. This includes addressing the security concerns of software development teams, security teams and other stakeholders. It should also be clear to organizations that not having goals for implementing a secure software supply chain solution can lead to higher costs over time.

For a VP and Senior Manager at a financial services firm with over 1,000 employees, his deployment strategy was to involve the legal, development, info security, and DevOps leaders. He wanted the leadership to understand the tool's capabilities, the defaults and "also to come up with a strategy to manage the outcomes, the findings." Along with that, he wanted "to implement a process that ensured that the findings – the breaks and the vulnerabilities that are found – would be visible. Notifications had to be made so that someone can triage and deal with them."

The software company Security Analyst also successfully embedded strategy into his process. He stated, "The requirements were that it would easily integrate into our pipeline, so that it was very automated and hands-off. Part of the implementation strategy was that we expected to use Jenkins, which is our main build-management tool."

Very automated and hands-off

Mistake #9:

Not Improving Speed-to-Market for Secure Applications

Improvements in secure software supply chains allow teams to produce secure applications faster as they speed up development rework and common security gates prior to production. This can speed up time-to-market for new development projects and services. In addition to security benefits, this improvement helps organizations avoid significant liabilities from insecure component violations and vulnerabilities that could be introduced into applications.

With respect to Nexus Lifecycle, the tech vendor's DevOps Engineer was pleased that the solution improves the time it takes to release the company's secure apps to market. He related, "We have been able to replace homemade scripts, which took a few hours to create, by very much simpler workflows provided natively by Nexus, which are working in a few minutes, or tens of minutes. It has saved us about 40 percent, in terms of time. But more than the duration, it's helpful that we don't have to maintain or make scripts. That's the most important thing."

> "It has saved us about 40 percent, in terms of time. But more than the duration, it's helpful that we don't have to maintain or make scripts. That's the most important thing."
>
> Read review »

40% of time saved

> "It's helping us avoid reactive costs and maintenance to the cycles after the fact..."
>
> Read review »

Improves time to market

Meanwhile, an Application Security professional at a comms service provider with over 1,000 employees also agreed that the solution has "improved the time that it takes to release secure apps to market" although he hasn't put an exact number on that. Further, a VP and Senior Manager at a financial services firm with over 1,000 employees felt that the Nexus Lifecycle solution improves "the time it takes to release secure apps to market."

That said, he also liked that Nexus Lifecycle software helps his company save money. He highlighted the fact that "It's helping us avoid reactive costs and maintenance to the cycles after the fact," adding, "If an industry vulnerability is found, we get that notification really early."

Mistake #10:

Not Starting as Soon as Possible

It is important to begin the process of implementing a secure software supply chain sooner rather than later. Components need to be identified and blocked from development environments if they are found to be unsafe, so it's important not to wait too long before beginning the process.

Primerica's Senior DevOps Engineer expressed the view that organizations should implement Nexus Lifecycle as soon as possible. He advised companies to "get it implemented into your environment as quickly as you can because it's going to help." He then said, "Once you get it, get your devs on it because they're going to thank you for it."

The tech vendor's Software Architect also encouraged businesses to implement Nexus Lifecycle right away. He stated, "You will have to clean up sooner or later. I remember when we fired it up it and immediately found things that the last solution didn't find. This made sense after we realized that IQ Server gets continued updates, and our last solution was just getting updates whenever we were able to get new hard drives sent to us. Our first scan popped up with several high vulnerability and security issues. At that time, the Sonatype people were on a call with us to help us out setting it up. We asked them if seeing this many alerts was pretty average, and they told us it was pretty normal in their experience. So that's when the cleanup started."

> **"I remember when we fired it up it and immediately found things that the last solution didn't find. Our first scan popped up with several high vulnerability and security issues."**
>
> Read review »

Conclusion

Organizations are at significant risk of security, licensing, and performance issues from commercial open-source components. If not addressed, these issues can cause more work for development teams while also increasing costs. A secure software supply chain solution allows organizations to block high-risk or vulnerable components before they are introduced into applications. As noted by PeerSpot reviews, this approach delivers benefits that include reduced security and licensing risks, as well as improved time-to-market for new applications, resulting in increased developer productivity. Avoiding these most common mistakes is an important first step toward having a successful experience overall with a secure software supply chain.

About PeerSpot

User reviews, candid discussions, and more for enterprise technology professionals.

The Internet has completely changed the way we make buying decisions. We now use ratings and review sites to see what other real users think before we buy electronics, book a hotel, visit a doctor or choose a restaurant. But in the world of enterprise technology, most of the information online and in your inbox comes from vendors. What you really want is objective information from other users. PeerSpot provides technology professionals with a community platform to share information about enterprise solutions.

PeerSpot is committed to offering user-contributed information that is valuable, objective, and relevant. We validate all reviewers with a triple authentication process, and protect your privacy by providing an environment where you can post anonymously and freely express your views. As a result, the community becomes a valuable resource, ensuring you get access to the right information and connect to the right people, whenever you need it.

www.peerspot.com

About Sonatype

Sonatype is the leader in developer-friendly, full-spectrum software supply chain automation providing organizations total control of their cloud-native development lifecycles, including third-party open-source code, first-party source code, infrastructure as code, and containerized code. The company supports 70% of the Fortune 100 and its commercial and open-source tools are trusted by 15 million developers around the world. With a vision to transform the way the world innovates, Sonatype helps organizations of all sizes build higher quality software that's more aligned with business needs, more maintainable and more secure.

Sonatype has been recognized by Fast Company as one of the Best Workplaces for Innovators in the world, two years in a row and has been named to the Deloitte Technology Fast 500 and Inc.