Sonatype Nexus Lifecycle Review From A Customer

IT Central Station
Unbiased reviews from the tech community

Sonatype Nexus Lifecycle

Read 10 reviews of Sonatype Nexus Lifecycle

Review by a Real User

Verified by IT Central Station

Information Security Specialist at a financial services firm with 1,001-5,000 employees

Devin Duffy

WHAT IS OUR PRIMARY USE CASE?

We use it to check if a software package has a vulnerability for enforcement of quarantines on the proxy level for housing all of our artifacts that we build in-house.

HOW HAS IT HELPED MY ORGANIZATION?

We're no longer building blindly with vulnerable components. We have awareness, we're pushing that awareness to developers, and we feel we have a better idea of what the threat landscape looks like. Things that we weren't even aware of that were bugs or vulnerabilities, we are now aware of them and we can remediate really quickly. It has absolutely helped bring open-source intelligence and policy enforcement across our SDLC. In partnership with the developers who have helped get the word out there, it has given developers the tools they need to figure out what to build with. We implemented a Slack bot using their data and engineers can query it to find good components. It's been working out very well for us. We use it to automate open-source governance and minimize risks. That's my job. We tear apart the Jenkins build logs, we find artifacts, and we use it to scan those artifacts and notify the teams that there are vulnerabilities in their builds. We also have the automated lookup as well, so that's how we use it in our enterprise at the moment. We have enterprise-level blocking for about 95 percent of crappy components. Finally, it has increased developer safety.

WHAT IS MOST VALUABLE?

The most valuable feature is the aggregation of threat details. In addition, it's their customer service. They've got really great customer service. I encourage developers to challenge whenever they see a security vulnerability that may not actually be a vulnerability, or that may be a false positive. When I bring that up with Sonatype - whereas a lot of vendors try to excuse their product or excuse their thinking - if it is, in fact, an issue or mistake, they'll own up to it and they'll fix it. The data integrity of the feeds that we get from them is a solid eight or nine out of ten. There have been some discrepancies but when we have brought them they have fixed them immediately. Their data is good enough to run a lot of orchestrated frameworks off of. It's been good.

Sonatype Nexus Lifecycle

Read 10 reviews of Sonatype Nexus Lifecycle

WHAT NEEDS IMPROVEMENT?

Application onboarding is a little bit clunky. But I use their API for that, and their API is alright. Their documentation is pretty good but there was a little bit of a learning curve with it. Onboarding an application through the GUI is intuitive but it's time-consuming. By time-consuming I mean for huge, enterprise-level companies, I don't mean for a small organization. For a small organization, it's not going to be prohibitive, but it's for large organizations with many enrollments that the GUI becomes unfeasible. I would also like to see a separate repo for components that have been un-quarantined for specific teams to use.

FOR HOW LONG HAVE I USED THE SOLUTION?

One to three years.

WHAT DO I THINK ABOUT THE STABILITY OF THE SOLUTION?

It's had really good uptime. I've been running a lot of scans on it and it has been producing a lot of reports and we haven't seen it having issues. Any issues have usually been in regard to what we're running it on, if what we're running it on is not beefy enough or if there are too many connections at once. There has been nothing from them stability-wise, just on our end from the architecture.

WHAT DO I THINK ABOUT THE SCALABILITY OF THE SOLUTION?

The scalability has been very good.

HOW ARE CUSTOMER SERVICE AND TECHNICAL SUPPORT?

Their support is good and their supplementary support, in a weekly call where they talk about how their product is doing for us, is very helpful. If we need to open an issue, they usually respond within ten minutes and it's by email. I wouldn't say their answers are in-depth, but brevity is important. I appreciate the brevity of the responses. Things that we don't necessarily need a big response for, they won't give a big response for. But I feel like they cover all the angles.

WHAT WAS OUR ROI?

Security tools are difficult to pin down in terms of ROI. I can't honestly say that I have seen an ROI but, at the same time, it's an invisible cost. You don't know what it could cost you. You don't know what an attacker could find and exploit on your system. So all security products have a negative ROI, unless you can show attacks that have failed. If the attacks don't exist in your package because you removed the vulnerable component, you won't know. It's like a vaccine. It's valuable, but I don't think we can quantify the savings, just because of the way security works. We had a version of Apache Solr in a production application that we found vulnerabilities in and it hadn't gone through a Nexus scan. If it had gone through a Nexus scan they would have known that if they just bumped the version they wouldn't have had SQL injection in production.

WHAT OTHER ADVICE DO I HAVE?

Have an idea of where you're going to put it in the SDLC. Have an idea of where it's going to catch builds. Know what it does and how it works, to understand how the proxy and the firewall work. Understand how to scan components. Be ready to have an "orange team" - that's a new term - to have "success engineers," people to work with developers. People who can say, "Hey, this is how you use this, this is how you check your builds," people who will be able to help the developers transition to coding more securely. Also, learning to scope and prioritize remediating stuff, learning to scope down your issues, is really important, to figure out which vulnerability in your existing code base you're going to tackle first. That's something I learned while using it. You can look at the entire threat landscape and just be overwhelmed and not get anything done. It's like a messy house. You've got to start somewhere, in a corner, and make that corner clean. Same thing here: You've got to pick your biggest pain point, you've got to pick the easiest way to fix it, and you've got to move forward from there. It blocks undesirable open-source components from entering your development lifecycle. Sometimes we have to let the components in. The design of it says that if we let one

component in for one team then we let it in for everybody else. That's a bit of a pain, but they are taking care of it and figuring out how to deal with that sort of issue. It's an unintended consequence. It's doing a great job at keeping stuff out but when we have to let something in for one team, we let it in for all the teams because that's the way the proxy works. I feel that it has increased the time it takes for apps to go to market but that's only because it's a new mitigating control. Before, nobody really cared about what they were throwing into their builds. All of a sudden that became a security concern and we needed a vendor like Sonatype. Once we started this, it was a necessary evil to make people examine what they are throwing into their builds and explore other components that are not vulnerable to things, right out of the gate. It is a new mitigating control to find a new class of vulnerability. It helps enforce secure coding practices and that can have a time cost when you're first rolling it out but, after a while, it may not have as much of a cost because more developers are familiar with it. Regarding the Success Metrics feature, I sort of use it, but I built a framework using their scanning engine, so I actually keep track of metrics on my own. It's okay, but it's not really my thing and we don't really have much of a use for it, to really invest in integrating the Success Metrics into what we've got going on, pipeline-wise. In terms of its integration with our existing DevOps tools, they've addressed these issues since then, but I ended up having to build a whole pipeline off of what they had, an undocumented API. They made it into a regular, documented API at a later point in time and that was just pulling vulnerability details for a specific component. Other than that, it works well. In terms of the number of users we have on the solution, "using it" is a very loose term. My team is automating it, other teams ingest the results, and between 900 and 1,500 people see the messages or have been impacted by it somehow. There are two or three people who work to maintain it who are Linux administrators. We're planning on rolling it out two more pipelines. In about a month we're going to triple our audience. Overall I would rate this solution at nine out of ten. Most everything that I've tried to do with it has been possible. I've had very few complaints that they didn't immediately address, or didn't explain that remediation would be unfeasible. They never talk down to me like other vendors do. They're a good vendor and they provide what I feel is a solid product.

Read 10 reviews of Sonatype Nexus Lifecycle