PeerSpot

Case Study

Veracode

Stephen Pack

Software development program leader at Vendavo

Review by a Real User

Verified by PeerSpot

What is our primary use case?

My company produces a SaaS application that is used by very large customers for pricing analytics and sales workflows. The data that our customers put into our software is very sensitive and confidential. This means that they want a high degree of confidence that our solution is secure.

We use Veracode as one of the pillars that we can point to as helping us to deliver on the promise of having a secure product. We have a multi-dimensional security program and Veracode is one important aspect of that.

How has it helped my organization?

Veracode provides guidance for fixing vulnerabilities. It provides guidance to help us understand what it flags, and what we can do about it. It still takes some interpretation and insight on our side, but we aren't generally security experts, so we get good information from Veracode to help inform us.

The developers are able to understand the types of issues Veracode looks for, and then as they see that happen, it helps them to learn. It's good because they consider it the next time and hopefully, we don't need Veracode to flag the issue because there is no issue.

With respect to efficiency when it comes to creating secure software, Veracode is able to help us with very low overhead. There's not a lot of work needed on our side unnecessarily. Once we've wired everything together, it's seamless to get the scan done and get the results back and know what we need to do about them.

We use Veracode for some of our older, more monolithic software, as well as for our newer solutions, which are designed to be cloud-native. We've found Veracode useful in both use cases; first, with our huge monolithic software, as well as with our microservices cloud-native solutions.

In terms of AppSec, there are a lot of benefits that cloud-native design brings in terms of not only cost and scalability, but testability and security. Certainly, the design patterns of cloud-native are well aligned with delivering good security practices. Working with products that support cloud-native solutions is an important part of our evolution.

Using Veracode has helped with developer security training and skill-building. It's definitely a good way to create awareness and to deliver information that's meaningful and in context. It's not abstract or theoretical. It's the code that they've written yesterday that they're getting feedback on, and it is a pretty ideal way to learn and improve.

The static scan capability is very powerful. It's very good in terms of the signal-to-noise ratio. The findings that we get are meaningful, or at least understandable, and there's not a bunch of junk that some other code scanning tools can sometimes produce. Having results like that make it hard to find the valuable bits. Veracode is highly effective at finding meaningful issues.

The speed of the static scan is okay. It meets or exceeds our expectations. For our monolithic application, which is a million lines of code, it takes a while to scan, but that's totally understandable. If it could be done magically in five minutes, I wouldn't say that's bad. Overall, it's very reasonable and appropriate.

Veracode has policy reporting features for ensuring compliance with industry standards and regulations. We have one such policy configured and it's helpful to highlight high-priority areas. We can address and help focus our effects, which ensures that we're spending our time in the best way possible for security movement. The policy is a good structure to guide results over time.

We use Veracode as one metric that we track internally. It gives us information in terms of knowing that we are resolving issues and not introducing issues. I cannot estimate metrics such as, for example, Veracode has made us 10% more secure. I can certainly say it's very important when we talk to our customers about the steps we follow. We do external pen tests, we do web app pen tests, and we also use Veracode. It's certainly very helpful in those conversations, where we can state that it is one of our security practices, but there's no outcome-based quantitative statistic that I can point to.

What is most valuable?

The static scan is the feature that we use the most, as it gives us insight into our source code. We have it integrated with our continuous integration, continuous delivery system, so we

can get insight quickly. We're doing scans daily, so that's the most important feature for us.

The interface is great. It allows us to look at our different applications, understand all of the different types of scans, as well as the results. The types of testing include SAST, DAST, and SCA, and it pulls all of the information together into a single view. It also produces reports that we can give to our customers when requested.

Veracode certainly provides a quick and intuitive way to understand the results, to see the context of them, and to identify what we need to do to address them. In general, it's a pretty quick way to get the information that we need in the most useful way possible. Then, we can turn around an action plan.

We have it integrated with our build pipeline and that works well. It's very important because we don't have to complete a separate, manual step of sending the software up to Veracode to scan it and get the results. It's great. the more things that we can integrate into the build pipeline, the better. It's a very positive thing.

Veracode is very good in terms of not having a lot of false positives. It would be very frustrating if a tool gave you 10 good results but 50 false positives. Even with the issues that we get that we choose not to address, we can still understand why they're being flagged. We have found that the results are meaningful and accurate, which gives us confidence in the solution when fixing vulnerabilities.

We may choose not to address them for different reasons. For example, it could be because it's an issue about input sanitization, but we have another layer on top of that component to handle that task. We can recognize that it's important that Veracode is flagging those things at that lower level, and that they're bringing that additional insight and consideration to the designs that we're choosing. Overwhelmingly, even the issues we choose not to address are still valuable and meaningful, so the actual false positive rate is quite low.

This is a very useful and powerful tool that ensures our code is well-designed and correctly implemented. It is important that it's only one aspect of a security program and not the only insight or the only test. That said, it provides us with some pretty important feedback and insights that we wouldn't have a great way to get otherwise.

What needs improvement?

The ideal situation in terms of putting the results in front of the developers would be with Veracode integration into the developer environment (IDE). They do have a plugin, which we've used in the past, but we were not as positive about it. The pricing model was expensive and the results were not the same as the full solution analysis. It gives a differently scoped "just in time" analysis within the context of the IDE, so it didn't speak to the same problem space.

The best situation would be the one where the

developers don't even need to log into the web portal, and the results from the scans would be delivered into their IDEs. It would be an asynchronous job, but if they could see the results right there, while they're working on the code, then they wouldn't need to go to a separate tool to look at the information to figure out what to do next.

The workflow today on the build side is optimal, so imagine that's still doing the same thing but then in the backend, whenever a developer has that project open in the browser, if they chose to, they could enable a view to see the most recent Veracode results of that module. That scan might be from last night or six hours ago or any other point, and that's fine. It would be the best possible situation to put the results and the actions right in front of the developer, in the tool that they're already using when they're touching the code.

The only other thing that we've found a reasonable workaround with is how to work with microservices in the context of Veracode. This was necessary because Veracode's licensing model and the interaction model are built around an idea of an application. When you're talking about a section of business logic that's being delivered by possibly dozens of microservices, there is some friction with Veracode in terms of how that application gets defined and how the scans occur and get reported on.

When we reached out to Veracode about this, I got a slide deck that provided us with different options of how they recommend proceeding in this context. It was helpful, and clearly a question they've considered and they had answers ready to go on. The ideas helped us and essentially reinforced what we were already thinking. It's getting the job done, but it still feels like a little bit of a square peg in a round hole and it could be a little smoother in terms of that interaction.

The problem boils down to how we fit the microservices architecture into the Veracode notion of an application. We need to be able to get a holistic view across the microservices, which is extremely challenging, especially when those microservices are owned by different teams who have different needs to see and respond to the scans.

For how long have I used the solution?

I have been using Veracode for between five and six years.

What do I think about the stability of the solution?

The stability is great. They've probably had some downtime, but I don't know about them. From our perspective, it's been solid.

I know the web portal has some planned downtimes because I see the splash screens about them. They're good about warning you,

but they're also performed at very weird times, like the middle of the night, so it's never blocked me from getting in when I need to get in.

What do I think about the scalability of the solution?

We use Veracode for all of our software development. We have more than 100 engineers, and our entire engineering team is using it. Obviously, every team has some designated people who look at this more than others, so not everybody's in there every day, but in terms of the software we write, we know that it's all being scanned constantly.

Over the last few years, we've made a couple of acquisitions of other companies and when we've done that, we very quickly brought those solutions in as well. We've seen the value and because of that, it's part of our onboarding process when we integrate other companies into our environment.

If we create another solution or we acquire another company, we will certainly expand our use of Veracode to match within our current solution stack.

How are customer service and support?

The support has been good at understanding issues. There are two aspects of technical support. One concerns issues with the platform in terms of functionality, and the other is that they will provide you with assistance in terms of interpreting your findings.

Our experience from the technical side is that they helped us with figuring out how to best use the platform for microservices applications. They were very helpful in that conversation.

We also have experience with the other layer of technical support that Veracode provides, which is where you can get consultations about the findings. We've done a few of those where you set up an appointment with a Veracode engineer. It helps to understand the results if the platform isn't totally clear on why something is a problem or what we need to do about it. For us, that's been pretty good.

Obviously, the Veracode engineer doesn't have the full understanding of what our application does and in a short call, you can't possibly do an architectural deep dive to understand the context of an issue, but their conversations have been useful when we've had them in terms of understanding issues and context and if we need to do anything.

Which solution did I use previously and why did I switch?

Prior to using Veracode, we used other code quality scanning tools, but not anything at the level of Veracode for security issues.

How was the initial setup?

The initial setup was straightforward. It was pretty easy to get going and we've incrementally gotten better and deeper as we've used it over the years.

The initial setup was manual uploads of applications, and then it was about incorporating it into our build pipelines and using the sandbox to support our microservices architecture. We've gotten more mature over time, but time to initial use and results were very easy.

Only a very short time is required for deployment, as there is very little that has to be done. Ours was completed within a couple of days and that's a matter of coordination in terms of getting our teams to upload a solution and figure it out. It was a learning experience for us but there was no time or delay brought on by the solution.

When we first began with Veracode, the initial strategy was just to get our first solution uploaded and scanned and see what the results looked like. We didn't have a systematic history of doing that, back then.

With approximately 500 employees, we're not a huge company. Deploying it in an enterprise company would be a different situation but for us, it was just a matter of understanding how we needed to configure the platform and how we needed to provide our software and states and get good results.

It probably took a couple of uploads of trial and error and we were running.

What about the implementation team?

We implemented the solution in-house. It is not that complicated.

In terms of maintenance, there is certainly some overhead involved for each team. They have to make sure that the build pipeline integration is still working and essentially, that we're still getting results. Occasionally, for whatever reason, it breaks and somebody has to go in and fix it.

I can't say that there is no staffing required for maintenance but it's rare. In total, a few hours a month across the company is spent keeping it going. More time is spent evaluating and resolving the findings, which is part of our development work. That's not imposed by the solution but rather a positive outcome from using Veracode. As such, I wouldn't count that as maintenance.

What was our ROI?

We have seen a return on our investment with Veracode. I can't point to a dollar figure, but I've been directly involved in customer conversations where we can talk about our security program and how Veracode is an important element. We've distributed report summaries and talked about results with our customers and having this information in those conversations is definitely valuable.

It's also very useful that we can talk about it with

our security auditors. We have SOC 1, SOC 2, and ISO 27001, and they don't specify that you must have a static analysis tool. But when we need to maintain secure engineering practices, having a tool like Veracode is very important for us to demonstrate that to auditors. There's certainly value there as well.

There is also a tremendous value on the marketplace that we get from having those security audits and certificates, which is a second-order of value that Veracode drives.

I can't say with certainty that Veracode reduces the cost of application security, although I would say that it focuses our effort. It gives us guidance and prioritization on where we should spend time. Otherwise, we might not know about particular issues. We might inadvertently spend time on things that aren't that valuable. So, the value is more about focusing on where we need to spend time.

What's my experience with pricing, setup cost, and licensing?

From a cost perspective, it seems okay, although we will probably evaluate alternatives next time it's up for renewal because for us, it's a relatively high cost, and we want to make sure that we are using our resources most appropriately.

I like that the platform provides you with some flexibility. We had to revise our licensing because it did not fit our environment. We wanted to license based on the number of applications, rather than another measure such as the number of lines of code. There was clearly some complexity that led us to be in that situation, although it seems preventable. Ever since our last renewal, the licensing has been smooth and clear. There is a certain amount of flexibility in that regard but also, they allow us some leeway in our current model.

There have been times when for some reason, we spin up a new application on a temporary basis. It may be because we're trying a new configuration. Even though we're licensed for a certain number of applications, the platform lets us exceed that. Consequently, we receive an email stating that we can't do that forever, but it's very useful to have the flexibility for the couple of times that we've used it to briefly exceed the application account.

Which other solutions did I evaluate?

I am not sure what other solutions, if any, the company looked at before choosing Veracode initially. We have renewed it since that time and we pretty quickly decided to stick with Veracode, rather than switching. However, because of the relatively high cost, we will probably evaluate other options next time it's up for renewal.

What other advice do I have?

We see at least quarterly updates about new features or things that have been fixed. It happens without our involvement, which is great.

My advice for anybody who is considering Veracode is to test it. Although I have not compared Veracode against other products as part of an evaluation process, it would be very useful and very easy to actually try it. Top-load your application, get the results and take a look at what Veracode finds. This is the most useful activity somebody could do.

This is a product that lives up to its promise. It's easy to use, and it's predictable. There are some improvement opportunities but on the whole, it's very good at what it does.

I would rate this solution a nine out of ten.

Which deployment model are you using for this solution?

Public Cloud