PeerSpot

Case Study

Veracode

SumalyaGuha

Security Engineer at a comms service provider with 10,001+ employees

- Review by a Real User
- Verified by PeerSpot

What is our primary use case?

We use Veracode for static code analysis, dynamic code analysis, and software composition analysis. In our organization, we have a bunch of applications that are running on a monorepo or microservice level. We have to do SAST on those applications so that we have a code review done on a bit level.

Going forward through the application pipeline, we do it on the dynamic level, as well, where we are scanning the public URLs of those applications to see what people can see externally. It's a type of out-to-in scanning in which we are analyzing the traffic that is sent out and even the traffic that is coming in, the response and request headers of the URLs, whenever someone is at a single URL.

Finally, for the software composition, Veracode uses a third-party analysis tool in which it has the libraries and the functions that are being used at a source code level. They are open source or dependent files that are used for building that in-house application.

How has it helped my organization?

As a company, we have moved from using contractors and third-party consulting companies to creating our software through more of an in-house model. We are moving more into the DevOps realm with more of our own teams developing our software. Veracode fits that DevSecOps ideology. It is definitely helping us build more secure software than we previously had.

We have a bunch of applications into which we have integrated Veracode and we have seen

that, in the final phase of production delivery, there are fewer vulnerabilities than we used to have.

And because Veracode has remediation and tracking within the platform, it becomes a good single pane of glass where the developers and the security professionals can operate and govern the flaws in the software. And they can take the necessary steps to remediate them.

In the metrics that we generate every month, we have seen the numbers go up with respect to remediation as well as the number of flaws that we catch. The word is spreading, and more and more application teams are using the static code analysis tool inside their pipelines. Overall, we are moving from reactive mode to proactive mode in remediating vulnerabilities through Veracode.

Veracode also helps our developers save time, in the big picture, compared to a situation without Veracode. Let's say there is an application on which no static analysis was done and the audit team says, "Hey, you don't have any static code analysis in your pipelines. You need to do something about that." They could scan the code that is already running in production and find flaws, but those flaws would take a lot more effort, time, and resources to mitigate compared to if they had been detected in a static analysis prior to the code going into production. In that way, it has definitely saved time. But if we are talking about short-term planning for sprints, it takes a little more time than usual because security is coming into the picture, as well. But overall, it helps save time.

Our security posture has gotten better since 2020. It takes time to do the integration of the platform and educate people about how to use Veracode, and then move on to remediating and validating things. But the journey that we had with Veracode has definitely helped us a lot, overall, with respect to bettering our security posture.

What is most valuable?

The static analysis is the most valuable aspect for us.

It also has the ability to block a build. In pipeline scanning, there is a configuration that can be set with respect to the security level of the flaw. If there is a high or a critical issue, there's a way the build can be failed and blocked before going into production. But the best case that I have found for blocking builds is in the staging area. You don't really want any blocking done on the production environment because there are business SLAs that the enterprise has to fulfill. The best case would be blocking the builds in the staging phase, the pre-production environment, so that everything is taken care of before it is pushed to production.

There are three integration points for Veracode. One is the IDE plugin. Whenever a developer is writing code on their IDE platform plugin for Veracode—whether IntelliJ or Visual Studio, et cetera—it tells them if that piece of code has any vulnerabilities and if there is a better way to

write the code.

The next point is the pipeline integration in which, whenever a build is getting pushed from a standalone branch to the main branch, a scan is done on that commit to see if there are any vulnerabilities.

Finally, when the build is published with the whole module, it can do another scan, as well. These three scans have their own pros and cons. The policy scan, which is a build scan, does the scanning on an overall basis with regard to the different standards out there, like OS and Spin5. It scans the first-party and third-party code, which is the most holistic scan that there can be. But the point is that it scans at three different integration points or stages, so it helps developers to remediate their vulnerabilities before they have moved far in the pipeline. Shift-left is definitely possible through Veracode.

What needs improvement?

Veracode's false positive rate is a little toward the higher side. We understand that Veracode doesn't have the business context. I advocate that people look at their code, even though there is a vulnerability, to see exactly what it is. For example, a randomize function is being used to create an ID that is not being hashed. Veracode marks it as a false positive because it doesn't know if the ID is being used for cookie generation or some random ID in the log generator. We, as dev or sec people, have to go in there and analyze what the ID is being used for. But the false positive rate is definitely a little bit on the higher side.

The effect of the false positive rate on developers' confidence in the solution depends on the maturity level of that particular application team with respect to learning Veracode. In the initial stages, obviously, when developers see that, whenever they're writing code or pushing a build, there are a bunch of vulnerabilities, it may affect their confidence. But a couple of months or a couple of quarters down the line, when those same developers have already used Veracode and have raised their maturity level from one to at least three, it doesn't really affect them because they know that they have to go in there and check the vulnerabilities for themselves to determine if it's a false positive or a real vulnerability.

It has definitely taken a little more time to validate the false positives, but I would say there are a lot of true positives, as well, which have been remediated and which have been mitigated for the betterment of the security posture. But it has definitely taken a little more time to mark or validate those positives. Hence, I definitely advocate that people shift a little more to the left. They should do ID and pipeline scanning before they hit policy scanning because, with ID and pipeline scanning, you scan small chunks of code. You remediate that code faster, before it goes to the whole package and there's a bunch that you have to deal with.

Also, container security is slowly becoming a

prevalent part of the development realm. Veracode's SAST, DAST, and SCA are pretty good with respect to industry standards, but with regard to container security, they are in either beta or alpha testing. They need to get that particular feature up and running so that they take care of the container security part.

In addition, there is a new concept out there, the IAST, which is interactive assessment security testing. It is a little more proactive than SAST. So if Veracode can combine that feature with their current technology, they would definitely be a front-runner again for the next five to six years.

For how long have I used the solution?

I've been using Veracode for the last three and a half years.

What do I think about the stability of the solution?

Once or twice a month there is maintenance on the Veracode side because they're updating some signature in their database or something else. I have seen maintenance coming up, but it's not an issue because the pipelines and integrations that we are running keep on running in the background. It's just the GUI that we are not able to access at that particular time.

What do I think about the scalability of the solution?

It's pretty scalable if our enterprise has the licenses for scaling the applications. I haven't faced any issues with regard to scalability, apart from licensing, of course.

How are customer service and support?

We have contacted Veracode's tech support a bunch of times. The only downside is the time needed to schedule a consultation call with the pro services team, keeping in mind that enterprises need to buy pro services licenses before they can use it.

When someone is scheduling a meeting with them, the issue type should be as precise as possible. In that way, they can rope in the exact SME for that particular topic, because in the development realm there are so many languages and so many types of issues out there. There are different personnel for each of those categories. So the more precise the details are for the meeting, the better the SME will be for that particular consultation.

How would you rate customer service and support?

Positive

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

We have only used Veracode, right from the start.

How was the initial setup?

The initial setup was pretty straightforward. They have a SaaS solution and there are a bunch of API integrations that made it pretty straightforward.

As for maintenance, all the upgrades and updates are done on Veracode's side. But there is a wrapper. When we are doing the integration, there is a package that we use to upload the files in Veracode. Sometimes there is a new release for that package and we have to update it in the GitLab repo. That's the only maintenance we need to do.

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

They have made it worth the price with the kind of discount and the kinds of modifications they made for us with regard to licensing. Previously, it was per profile. But they have adjusted according to our requirements because we are a big company and we handle a lot of applications. There's a tiered discount that they have provided us, so the cost is justified.

If someone looking at Veracode is concerned about the price, it depends on their requirements. I wouldn't really recommend Veracode for a small firm, because it might be a little pricey for them. But for a large organization, with more than 1,000 applications in the enterprise, there are tiered levels of pricing. Obviously, there are other cutting-edge solutions that have become available recently, but Veracode is something that a big organization should look at.

What other advice do I have?

When it comes to managing risks, we use the remediation feature that Veracode has. Whenever there is a flaw, we do have tickets open up for it and the application owner or the developer goes through the vulnerabilities. There are times when the vulnerability is a false positive and you can mark it as such within the Veracode platform itself. And we, as security professionals, do the validation for whether the business justification is good or not. And we either have a source code review for the vulnerability or have an exception open up for the remediation step that the application or the owner is asking for. We do risks via the platform, as well as through the ticketing tool that we use.

We are also using SBOM (Software Bill of Materials) for inventing all the different kinds of modules and libraries that we are using for an application. Using the SBOM feature, you would have to leverage the API to get the inventory

from the API calls that Veracode has. But in our organization, we use the GUI report generation more than the SBOM report because there is an executive summary in the GUI report with regard to first-party and third-party flaws. It also has the mitigation steps. SBOM would only give you the list of softwares, libraries, and versions that are being used. It is not as detailed as the GUI report that Veracode provides.

Things to consider when looking at Veracode include the different integration points where you want to integrate Veracode, how big your organization is, and how many applications you want to do security analysis on. If it's a big organization, Veracode is obviously a solution to evaluate, but for a small organization, below 500 apps, it might be a little pricey. Also, you will need a couple of Veracode champions on your team who know it inside out. You will need training provided by Veracode, so make sure that is included during the procurement stage. That will help you implement the tool within your organization faster and much more efficiently.

I would have given Veracode a nine out of 10 a couple of years back, but given the tools that are coming out on the market, and the scope of development, which is increasing, I would place it at eight.

Which deployment model are you using for this solution?

Public Cloud

Read 23 reviews of Veracode

See All Reviews