GitGuardian Platform

PeerSpot

Verified user

Senior Application Security Engineer at Bazaarvoice

Impressive detection capabilities, fantastic UI, and incredibly knowledgeable support team

May 15, 2024

What is our primary use case?

“We brought in GitGuardian Internal Monitoring to review all of our code within GitHub so that we can identify and fix any exposed secrets.”

How has it helped my organization?

“I have been very impressed with the breadth of its detection capabilities. I did a proof of concept with a couple of other common tools for the same kind of thing, and I found GitGuardian to be the best. It finds everything that I would expect it to find. It found more than I thought we would find, so I am very happy with the detection.

I am very happy with the number of specific detectors and keys that it can find for Google, AWS, Twitter, Facebook, etc. It has a lot of specific detectors for different categories, but it also has quite a lot of automatic validity checking, so it can tell

whether your Twitter keys or AWS keys are active and or have been revoked. If they are revoked, it is not a problem. Validity checking is fantastic.

GitGuardian Platform's accuracy is incredibly high. There are a couple of categories of generic secrets that I can find. When you turn those on, you end up with quite a few false positives. With the specific detection categories, the false positive rate is incredibly low, but when you turn on the generic categories, it goes up a bit. I am very happy with the number of things that it does find, instead of focusing on true positives versus false positives.

It has not helped to decrease false positives because we did not have a tool in the first place, so we did not have any false positives. We have not decreased our rate at all.

GitGuardian Platform has absolutely helped to quickly prioritize remediation. The severity or criticality that the tool automatically assigns has been very helpful. The built-in validity checking has also been helpful. Whenever you have keys that are marked as valid within the tool, you know that they are high priority and need to be resolved sooner than the ones that are not marked as valid.

We have significantly reduced our potential risk exposure of secrets. In the past nine months, by using GitGuardian, we have been able to identify and resolve a large number of secrets within our code, which reduces the risk if our code were to become public. It has greatly reduced our security risk. It has reduced the potential risk of exposure of secrets by about 75%. We have not only been able to resolve existing issues; we are also more likely to prevent these issues from occurring with improved security culture and the features within the tool.

GitGuardian Platform efficiently supports our shift-left strategy. It provides a command line interface, which can link with the shift left with your standard development processes. Whenever developers are writing code and trying to commit and push that up to GitHub, the command line interface can be integrated into that to prevent secrets from getting into GitHub. It can help go almost as far left as possible.

GitGuardian Platform has greatly improved awareness, and it has reduced the number of secrets that end up in our code. There has been a two-layer impact where it has helped people think about this as an issue and it has also helped them

stop doing it even if they are not thinking about it.

GitGuardian Platform has improved our ability to collaborate. We work closely with the development teams to identify the issues, investigate the issues, and troubleshoot and resolve those issues. Due to the way the tool works, it has helped us gather people into teams and work with them so that we can help resolve the findings.

We have one or two of the playbooks automatically enabled. So far, I find them very helpful. The main one so far is that the secrets will automatically resolve when they are revoked, which is incredibly useful. For example, whenever someone goes into the AWS platform to revoke AWS keys, they do not have to go into GitGuardian. It automatically detects that they have been revoked and closes the issue. It is a lot less work than having to go into two different tools and more. There are a couple of playbooks we have for the CLI, so whenever we ignore issues via CLI because something might be a false positive or something might only be a testing key, it will auto-resolve within the UI. The playbooks make it easier to avoid using the UI, but you do not have to. It is one of the catch-22 situations. There is UI, but they are enabling you to not even have to check it in the first place. Playbooks have reduced 50% to 60% of manual work. If developers accidentally commit active keys, they do not have to go in. They do what they need to do to resolve the keys. They do not need to think about it again.

GitGuardian Platform has increased our security team's productivity by about 50%. When we previously had noticed keys, it would have been manual. It would only have been occasionally when I was looking through the code and found the keys. I would have had to reach out to developers and discuss that. It has definitely greatly increased our productivity because we can now automate sending out tickets and assigning them to the right teams. A couple of clicks can send out the information for someone to look into rather than having to message them and try to discuss it with their team. It is a lot more automated.

It is hard to measure the increase in secrets detection rate because previously, we did not have any solution, so we were not detecting anything. After implementing GitGuardian, we can now see what we have got.

Similarly, it is hard to measure the reduction in the mean time to remediation as

we did not have something before. It was more manual before. There is probably a 70% reduction because, previously, if I found an issue, I would reach out to our team and spend a while discussing it with them, whereas now, we can just send out a Jira ticket. They can log in and have a look. There is a lot less discussion back and forth."

What is most valuable?

"There is quite a lot to like. Its user interface is fantastic, and being able to sort the incidents by whether they are valid or for a certain repository or a certain user has been very beneficial in helping investigate what has been found.

The CLI provided by the tool is fantastic for preventing the secrets from getting into GitHub in the first place. The more you use the CLI, the less you use the user interface."

What needs improvement?

"Automated Jira tickets would be fantastic. At the moment, I believe we have to go in and click to create a Jira ticket. It would be nice to automate.

I believe there is a feature on the road map for better handling of issues that have over one occurrence. It is difficult to investigate when there are a large number of secrets. It is hard to know where they are and what to do. These two things would be nice."

For how long have I used the solution?

"I have been using GitGuardian Internal Monitoring for a_"

What do I think about the stability of the solution?

"It is a stable solution. I have not noticed any issues with performance, downtime, or anything like that. I would rate it a ten out of ten for stability."

What do I think about the scalability of the solution?

"It is scalable. All it requires is someone with GitHub admin permissions. We can integrate as many repos and sources as we want. I would rate it a ten out of ten for scalability.

We have 316 users using this solution. We plan to increase its usage. There are a couple of features in GitGuardian. There is a feature where CLI integrates with your development process for pre-commits. We plan on testing and rolling out that feature so that every developer has pre-committed automatically enabled on their machine. The idea is that it will basically prevent any secrets from getting into GitHub. Another, a lot more minor, feature is GitHub pull requests. Every time there is a pull request, the GitGuardian bot will comment on it if there are secrets. There is an option to block the pull request when secrets are found. We plan to implement that as well."

How are customer service and support?

"Their technical support so far has been fantastic. Anytime I raise a ticket, it is resolved and answered very quickly. I am very impressed. Their support is incredibly knowledgeable. Whenever I have questions about detection or remediation, they are very detailed in their answers, and they clearly know a lot about the tool.

Our experience has been fantastic with the purchase, the onboarding, and the customer success team. Everything has been straightforward. Everyone so far has

been nice, friendly, and helpful. When there were any hurdles, they helped me resolve them straight away. I would rate their support a ten out of ten."

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

"We did not use any similar solution before. It was a manual process. We did not monitor anything. We just occasionally noticed things to be resolved. It was a manual process."

How was the initial setup?

"To implement it, the only thing required from our side was having someone with admin permissions to enable the installation. It was minimal from our side.

It does not require any maintenance from our side."

What about the implementation team?

"It required just one person with GitHub admin privileges. I clicked a few buttons, and then he went in and approved it, and that was it."

What was our ROI?

"It has definitely saved us a lot of time. To be able to view everything important and narrow our focus to resolve issues has sped up our development process and decreased our security risk."

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

“I am only aware of the base price. I do not know what happened with our purchasing team in discussions with GitGuardian. I was not privy to the overall contract, but in terms of the base MSRP price, I found it reasonable.”

Which other solutions did I evaluate?

“We reviewed three or four main secret detection products available. We reviewed GitHub Advanced Security and BluBracket.

We chose GitGuardian for a number of reasons. Its user interface is absolutely fantastic. Being able to filter instances has been the main thing. It helps us to focus and narrow down our remediation efforts.

There is also the ability to create teams and assign developers and teams to only see what they are responsible for. There are a number of other products, but they are missing that feature of narrowing visibility. We wanted a tool that the security team could set up and the developers could log in to use. A lot of the other tools on the market are only for the security team. It would have been more manual on our side to reach out to teams to get them to resolve things. This way, we can add users to teams, assign them the repositories that they maintain, and they can work away by themselves.”

What other advice do I have?

“To a security colleague at another company who is using an open-source secrets detection solution, I would be happy to recommend GitGuardian. I have been setting up and using the tool. I can happily, personally, and professionally recommend this tool to others.

In my opinion, secret detection is incredibly important to a security program for

application development. It is critical to our company's obligation and security process. Without it, you do not know what secrets could be leaked, so once you implement it, you know where you stand and you know what you need to do. You can resolve as well as prevent these things.

I would definitely recommend doing a proof of concept to make sure it fits your use case. I would be more than happy to recommend it. There would not be any caveats. Go ahead and test it out. If it fits exactly what you need, go for it.

Overall, I would rate GitGuardian Platform a ten out of ten. I am very happy with what we are able to do with it and how it works."

GitGuardian Platform

For more in-depth insights from real user reviews

Download GitGuardian Platform Buyer's Guide

About PeerSpot

PeerSpot is the leading review site for software running on AWS and other platforms. We created PeerSpot to provide a trusted platform to share information about software, applications, and services. Since 2012, over 22 million people have used PeerSpot to choose the right software for their business.

PeerSpot helps tech professionals by providing:

- A list of products recommended by real users
- In-depth reviews, including pros and cons
- Specific information to help you choose the best vendor for your needs

Use PeerSpot to:

- Read and post reviews of products
- Access over 30,000 buyer's guides and comparison reports
- Request or share information about functionality, quality, and pricing

Join PeerSpot to connect with peers to help you:

- Get immediate answers to questions
- Validate vendor claims
- Exchange tips for getting the best deals with vendor

Visit PeerSpot: www.peerspot.com

PeerSpot

244 5th Avenue, Suite R-230 • New York, NY 10001

reports@peerspot.com

+1 646.328.1944