GitGuardian Platform

PeerSpot

Reviewer700988

Director Cloud DevOps SRE at a tech company
with 201-500 employees

Helps us to quickly prioritize remediation and has improved the coordination between developers and security personnel

Mar 08, 2023

What is our primary use case?

“We use GitGuardian to check standard configurations and scan for possible leaked secrets. Developers and software engineers sometimes commit to AWS keys, login credentials, SMTP databases, and other secrets.”

How has it helped my organization?

“Given the size of our operation, there's a lot of work to do on the security side in GitHub alone. GitGuardian enables us to avoid leaks in the source code on the GitHub side and helps devise a plan to fix them. Sometimes it doesn't find the leak, but it identifies the type of leak. The solution typically does an excellent job on that part. We can locate the crucial leaks and try to remediate those first. GitGuardian makes the job easier and faster.

It improved the coordination between developers and security personnel. Having a top-down mindset is not so great in terms of security. We have some roadblocks that get in the way of security best practices. GitGuardian's features help us to improve that. People need to improve their mindsets as well.

We don't have a security team. The company doesn't have this in the core. We began implementing security in our code with GitGuardian, so we don't have a baseline to compare it to. We had nothing, and now we have GitGuardian for GitHub. It works pretty well and helped us to improve for sure. The time-to-remediation depends on the software engineers. We do not do the remediation; they prioritize as they want, so that's the mindset issue again.

GitGuardian helps us to quickly prioritize remediation. At the same time, we need to work on internal policies regarding what engineers should do. They do not prioritize remediation as much as we think they should. This is a company problem. We didn't have as much emphasis on IT security, cybersecurity, or DevSecOps before we started doing this. We are trying to change their mindset and show how dangerous it could be if secrets are leaked.

We didn't require much preparation to use GitGuardian except for a one-hour training session with GitGuardian. The tool is pretty easy to use and has nice consoles. In one or two hours, we are ready to utilize the tool. The rest was checking configurations and reading documentation. We had to read up on features like single sign-on and how to note a secret leak as a comment in the pull request."

What is most valuable?

"The entire GitGuardian solution is valuable. The product is doing its job and showing us many things. We get many false positives, but the ability to automatically display potential leaks when developers commit is valuable. The dashboards show us recent and historical commits, and we have a full scan that shows historical leaked secrets.

I would rate the accuracy an eight out of ten. We get false positives, but it's

not because the tool is working incorrectly. Our software engineers commit things like the API key because they know they're unimportant. We consider them false positives because they are not real leaks. The false positive rate is low and will probably improve with time.

The AWS secrets tool and ggshield have the same functionalities, but I'm not sure how they do everything behind the scenes. GitGuardian has good tech knowledge, but we still see too many false positives. We don't have a granular way to tell GitGuardian on the SaaS side to ignore specific secrets. We have to filter everything after it's done.

GitGuardian has single sign-on integration, which we implemented to make tasks easier for everyone. With SSO, we can send a link to GitGuardian instead of creating a ticket for that. People couldn't engage correctly with GitGuardian before we implemented SSO.”

What needs improvement?

“GitGuardian could have more detailed information on what software engineers can do. It only provides some highly generic feedback when a secret is detected. They should have outside documentation. We send this to our software engineers, who are still doing the commits. It's the wrong way to work, but they are accustomed to doing it this way. When they go into that ticket, they see a few instructions that might be confusing. If I see a leaked secret committed two years ago, it's not enough to undo that commit. I need to go in there, change all my code to utilize GitHub secrets, and go on AWS to validate my key.

It would be helpful to have small instructions to show developers how to deal with an issue. They ask us what they need to do each time, but it's always more or less the same. GitGuardian could send them clear steps, so they can engage without needing help every time. ”

For how long have I used the solution?

“I have used GitGuardian for around six months.”

What do I think about the stability of the solution?

“GitGuardian is stable for our use case.”

What do I think about the scalability of the solution?

“We have almost a thousand report stores, and it scans correctly, so we don't face any scaling issues.”

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

“I don't remember the specifics of the contract, but we have a one-year license for a set number of developers. It's reasonably priced. ”

What other advice do I have?

“I rate GitGuardian a ten out of ten. It's a user-friendly product that's ready to go. You don't need anything besides the initial onboarding training to use this tool. If you are concerned about your security and want something ready to go, GitGuardian is an excellent option for a fair price. I recommend it. GitGuardian is a better choice than an open source solution if you are serious about preventing leaks on GitHub and your developers lack security awareness.

Secret detection is one of the essential aspects of application development. Leaked secrets are the main reasons for getting hacked. Often, secrets are leaked by an employee searching and finding secrets they should not, or someone makes a private post public because they don't know the secrets were there. Many bad situations happen because developers don't know what they are doing or don't care. The company mindset needs to change, but we still have a long way to go. ”

Which deployment model are you using for this solution?

Public Cloud

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