GitGuardian Platform

PeerSpot

Verified user

Systems Engineer at a marketing services firm with 11-50 employees

They offer a free tier that provides full functionality for smaller teams

Mar 06, 2024

What is our primary use case?

“We use GitGuardian to detect secrets that have inadvertently been committed to our source code. GitGuardian monitors every Git push and commits we make, and it analyzes the files, looking for things like access tokens, passwords, session ID cookies, etc. If that happens, GitGuardian raises a ticket in our internal ticketing system, and we remedy it. ”

How has it helped my organization?

“When we first deployed GitGuardian, we went back through all of the commits that we did over the course of the last five or six years that the company existed. It immediately found more than a hundred. We detected all sorts of secrets in those repositories. It had a pretty substantial impact from the first day. That was during our trial run, but now it's incorporated into our deployment pipelines. The impact is still there, and it's still tremendous. It's probably not as instantaneous or the same avalanche of detections that we saw on day one. That was impressive, but we

©2025 PeerSpot, All Rights Reserved

don't get that anymore. It has been a constant trickle of tickets.

GitGuardian helps us prioritize remediation. You need to incorporate it into your existing processes, but GitGuardian provides you with the flexibility and the tools. For example, in our environment, we implement ticket creation through webhooks. We have some logic rules stating that our production repositories are a higher priority than our dev or sandbox repositories. Our developers commit all sorts of weird things to those. GitGuardian gives you the tools to do that, but it may not necessarily do that right out of the box when you first deploy it.

To have collaboration between our security and dev teams, you need to have a detection. Previously, we did not have a functional equivalent to GitGuardian in our environment, and it introduced that process, so we could begin having that conversation. The security team is more focused on remediating to ensure that API token or password is invalidated as soon as possible after it was committed. Developers are more focused on why the secret was committed and environment variables to store that particular secret. The collaboration exists in our company largely thanks to GitGuardian.

A webhook creates a ticket in our internal ticketing system, and the ticket goes to the security guys. They look through it. They make sure the secret is invalidated and start that conversation with the developer to say that they committed this, so please don't do that again. That's the end of the story. We don't use 100 percent of GitGuardian's functionality. We are a fairly small company, so we probably don't need all of that. This simple approach works pretty well for a company of our size.

GitGuardian has improved our security team's productivity if we measure it in security incidents per week, hour, etc. Now, we have a separate stream of secret detection tickets going into our system. It's much better to have those during the deployment phase instead of discovering them after a breach or down the road.

It's hard to quantify the time saved. Finding a secret that was accidentally committed to a repo is like searching for a needle in a haystack. And you don't even know if the needle is in that haystack. Now you have something like X-ray vision that lets you see through that haystack and find right where the needle is. It unlocked a new angle on our application security process that did not exist. When a secret was accidentally committed to a repo, it could have been

©2025 PeerSpot, All Rights Reserved

noticed by a security guy or another developer, or maybe not. ”

What is most valuable?

“The most valuable feature of GitGuardian is its core secret detection mechanism. It covers a broad range of technologies. The detection accuracy is extremely good. It correctly detects in about 99 percent of cases. Every false positive we've had wasn't an actual false positive. It was a case where a developer copied a sample code from somewhere, including a dummy password or session ID. GitGuardian may trigger this, but I think that's a good thing because we know it's there, and it is alert.”

What needs improvement?

“GitGuardian had a really nice feature that allowed you to compare all the public GitHub repositories against your code base and see if your code leaked. They discontinued it for some reason about eight months ago, it was in preview and kinda exploratory phase, but for whatever reason, they chose not to move forward with it.

That is unfortunate because it immediately detected a leak of our company code that one of our contractors committed. They leaked our intellectual property into one of their public reports. ”

For how long have I used the solution?

“I have used GitGuardian for 14 or 15 months. ”

©2025 PeerSpot, All Rights Reserved

What do I think about the stability of the solution?

“I have never experienced a single instance of downtime, but I don't sit there 24/7. It's just a useful thing that is sitting in the corner humming and doing its thing. I have never noticed any outages.”

What do I think about the scalability of the solution?

“We are a small company, and it performs beautifully for a company of our size, but I think it will also perform well for a company 20 times our size. If we're talking at the scale of a company the size of Google, then I don't know.”

How are customer service and support?

“I rate GitGuardian support eight out of ten. ”

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

“We didn't have a secret detection solution because it's a fairly new area. However, we also use Snyk to supplement GitGuardian. It does things that GitGuardian doesn't do, like dependency detection and static code analysis. GitGuardian is also doing things that Snyk isn't, so the two complement each other nicely.”

How was the initial setup?

“GitGuardian is a SaaS solution, and the integration process is pretty straightforward. It's similar to other things you integrate with our repository and version control systems. It doesn't require any maintenance. It adds new repositories automatically.”

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

“The purchasing process is convoluted compared to Snyk, the other tool we use. It's like night and day because you only need to punch in your credit card, and you're set. With GitGuardian, getting a quote took two or three weeks. We paid for it in December but have not settled that payment yet.

It's also worth mentioning that GitGuardian is unique because they have a free tier that we've been using for the first twelve months. It provides full functionality for smaller teams. We're a smaller company and have never changed in size, but we got to the point where we felt the service brought us value, and we wanted to pay for it. We also wanted an SLA for technical support and whatnot, so we switched to a paid plan. Without that, they had a super-generous, free tier, and I was immensely impressed with it.”

Which other solutions did I evaluate?

“When we acquired GitGuardian, I compared it to GitHub Advanced Security, an additional premium subscription from GitHub that you can purchase on top of your existing one. It claims to do similar things to what GitGuardian does, but GitGuardian is far superior in terms of the types of secrets it can detect.

I'm not sure if GitHub has caught up since then. I picked GitGuardian over GitHub Security because it had better functionality. Also, not all of our repositories are in GitHub. We also used Azure DevOps. GitHub Advanced Security sort of locks you down within that GitHub sandbox. With GitGuardian, we could scan both GitHub and Azure DevOps repositories and have identical functionality across the two. If

©2025 PeerSpot, All Rights Reserved

we implement a policy in GitGuardian, we would know that it equally applies to secrets committed to both systems.

You also have the option of open-source solutions, but one of our core principles is to lean heavily toward solutions that are not self-hosted, whether it's in the cloud or on-premises. To have an open-source solution, you need to run it somewhere and maintain it. GitGuardian is a software as a service. You sign up and forget about it until your next detection. If a company wants to minimize administrative overhead, GitGuardian is a pretty much no-brainer."

What other advice do I have?

"I rate GitGuardian eight out of 10. Secret detection is critical to application security. You might assume that your developers have a security mindset. Many don't. Sometimes, it isn't even a mistake. They might not realize exactly what they are doing and the amount of damage that could occur because of what they commit to a repo.

When you implement GitGuardian, there will be an influx of detections if you're developing any software that connects to anything with a database, third-party REST API, etc. I recommend looking through the initial list of detections and identifying the most susceptible projects or repositories. Also, look at the developers who produce the most detections. Those are the people who lack a security mindset. Identify the high-risk category of developers."

Which deployment model are you using for this solution?

Public Cloud

©2025 PeerSpot, All Rights Reserved

GitGuardian Platform

For more in-depth insights from real user reviews

Download GitGuardian Platform Buyer's Guide

©2025 PeerSpot, All Rights Reserved

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

©2025 PeerSpot, All Rights Reserved