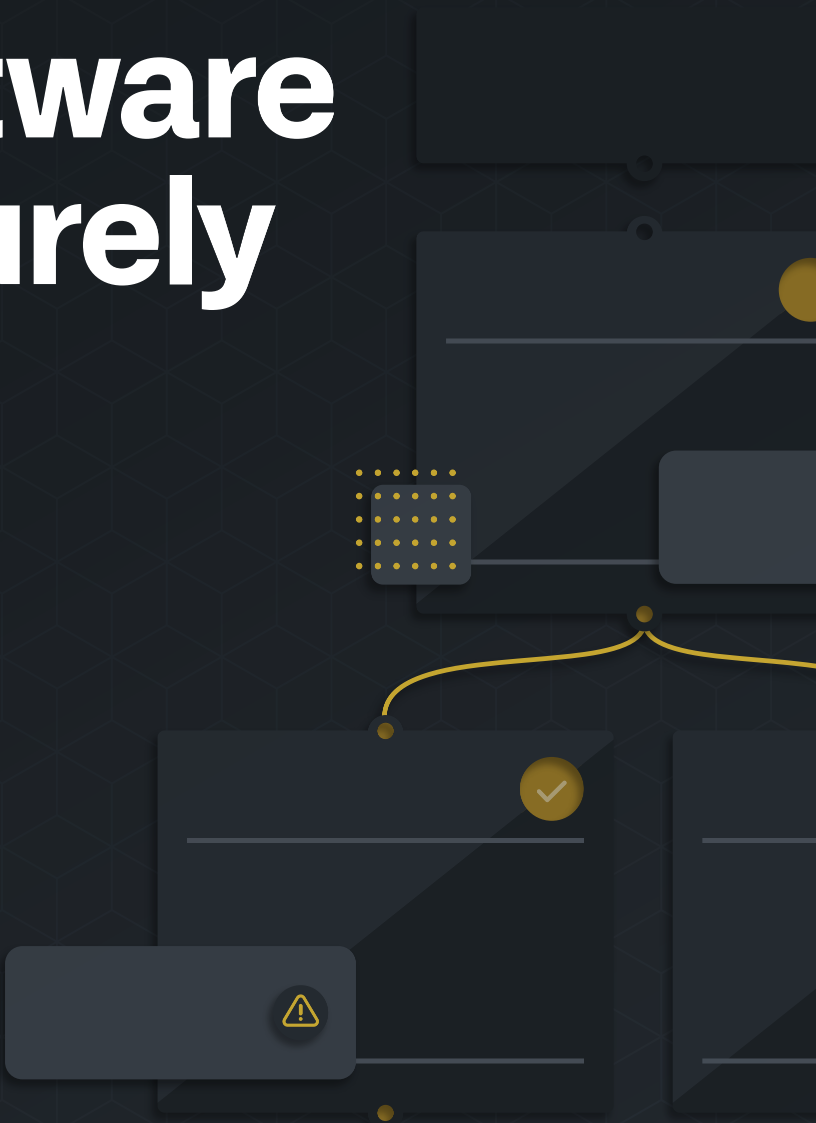
CUSTOMER SPOTLIGHT

How leading software teams build securely on GitHub

FEATURING:

AUTODESK zendesk Pinterest DOW JONES omise twilio

McKESSON NETDATA California Department of Technology stripe Spotify Auth0



WRITTEN BY GITHUB WITH ♥

Contents

Introduction

Staying secure from login to release

Today, every company is a software company. Digital applications are the foundation of great customer experiences, from online banking to healthcare. But while these applications open up businesses to more customers than before, they also expose organizations to more risk. Over the last five years, applications have been reported as the main attack vector causing security breaches.[1]

Secure software development is the only way to protect your code and customers. Still, clunky security tools and team silos can make security an afterthought, or simply add more friction to tight timelines. For developers, it can be frustrating to fix security issues after production. For security teams, it can be a challenge to be included after development starts, or finishes completely.

The good news is secure development doesn't have to be a barrier to collaboration or innovation. Thousands of organizations are using GitHub to free their workflows from insular development and build secure processes that give engineering teams the flexibility to do their best work.

> With consumer expectations higher than ever and increased pressure to lower costs, efficient, collaborative, and secure workflows can help teams shift focus to where it matters most: Building the best, most innovative software.

In this ebook, we'll explore how GitHub customers like Pinterest, Stripe, Dow Jones and others build more secure applications—without disrupting innovation or developer productivity. We'll break down how these organizations use open source, GitHub's built-in automation, Dependabot, and the world's most powerful semantic code analysis engine to ship secure software, faster. Along the way, you'll also learn easy-to-follow security tips your team can put to work today.

[1] 2016, 2017, 2018, 2019, and 2020 Data Breach Investigation Reports, Verizon

PART ONE

Secure access

BRANDS IN THIS CHAPTER:

AUTODESK zendesk Pinterest DOW JONES omise

How do I know that only the right people have access to our code?

Build securely from day one using multi-factor authentication and IdP integrations.

—

Secure software development begins at login. For most organizations, this means setting up secure access by using an identity provider (IdP) like Azure AD, Okta, or Onelogin. By connecting your IdP to your software development platform, you can control and secure access to organization resources like repositories, issues, and pull requests.

Configuring authentication is just step one. Commit signing, access management, SAML single sign-on, audit logs, policy enforcement, and more keep your code safe throughout the entire development lifecycle—from idea to production.

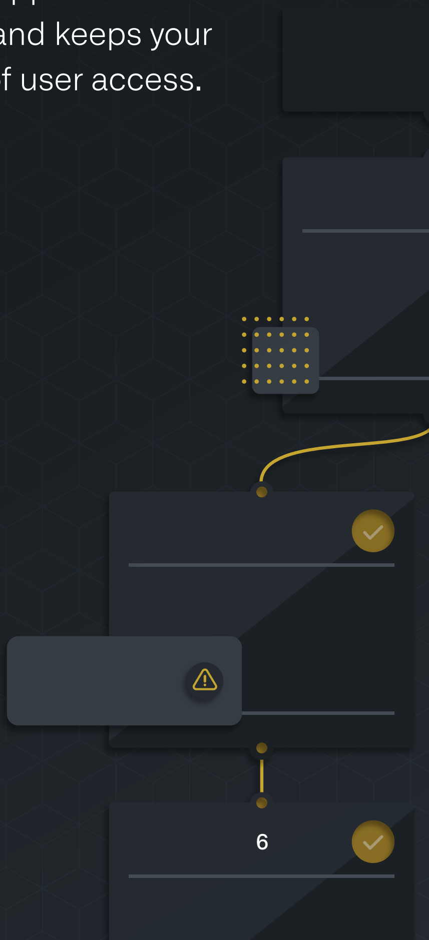
The first line of defense: knowing and controlling who has access to code

With the global breadth and depth of Autodesk software, security is paramount. “GitHub has very high security standards,” said Senior Software Engineer Patrick Lühne. Dedicated security teams use Codacy

GitHub tip

Use a centralized authentication protocol, such as SSO with LDAP or SAML, for all systems within your organization.

Using SSO across your applications minimizes security risk and keeps your organization in control of user access.



and SonarQube integrated with GitHub, which run automatically. They also use SAML single sign-on (SSO) to control access, and two-factor authentication (2FA) to log in to GitHub. “As a software stack, GitHub is very secure and we trust the authentication and authorization.”

Among other features, Zendesk also takes advantage of 2FA along with Okta authentication, to maintain security and protect everything behind a single identity provider.

For Pinterest, which recently switched to GitHub’s SAML-based authentication and uses single sign-on with their internal security tools, “the advantage isn’t just stronger security in terms of how we manage our GitHub organization, it also keeps our roster updated. It’s saved us an enormous amount of time.”

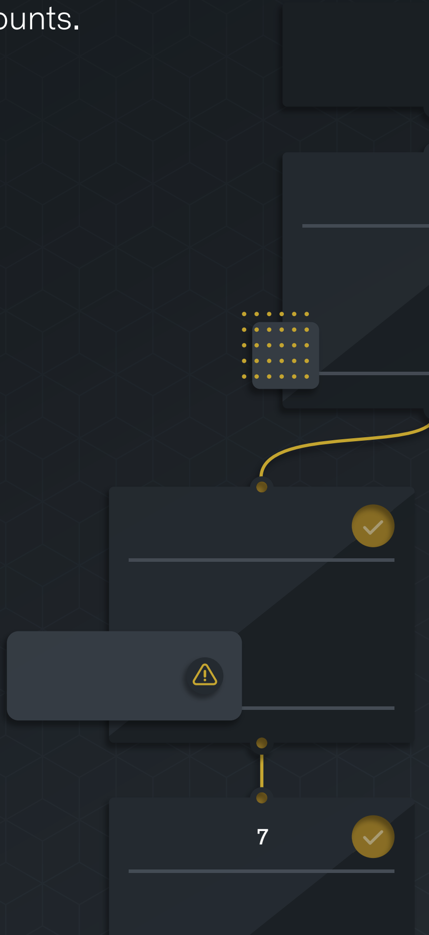
Digital trails support change management, security, and compliance at every step

In Dow Jones’ Lead Cyber Security Engineer Sydney Sweeney’s view, GitHub’s reliable digital trail for security and compliance is paramount. Using their governance bots, Sweeney’s team can check if the right topics are assigned to the right repositories and trace which business units and products each project belongs to.

GitHub tip

Enable MFA–and require when available–on all systems.

Adding multi-factor authentication provides an extra layer of protection and greatly reduces the chance of compromised user accounts.



GitHub also allows individual teams to enforce their own custom security controls.

“Having apps that help automate security within GitHub has been huge for us,” Sweeney confirmed. “The repository secrets have made security much easier to manage. It helps prevent passwords from being pushed into the code by providing a similar developer experience from local development through deployment.” And if a security vulnerability is discovered, product teams can act quickly. GitHub’s built-in security alerts notify developers when a vulnerable dependency is found, and then automatically open pull requests with suggested fixes.

> “Having apps that help automate security within GitHub has been huge for us.”
>
> *— Lead Cyber Security Engineer, Dow Jones*

Meanwhile, online payments platform Omise uses pre-receive hooks, a powerful tool supported by GitHub, to help enforce critical business rules, meet compliance requirements, and prevent undesired changes. When credit card information is at stake, this is particularly important. In the wrong hands, the cards could be exfiltrated.

With GitHub, the Omise teams can follow up-to-date software best practices within strict security requirements. “All of the code needs to be checked in a safe and secure manner, and no one can access it without proper authorization”, explains Chief Technology Officer Robin Clart.

PART TWO

Secure innovation with open source

BRANDS IN THIS CHAPTER:

Pinterest twilio McKESSON

How do I know which open source components we're using and if they are secure?

Take advantage of the benefits of open source—all while keeping your code secure.

—

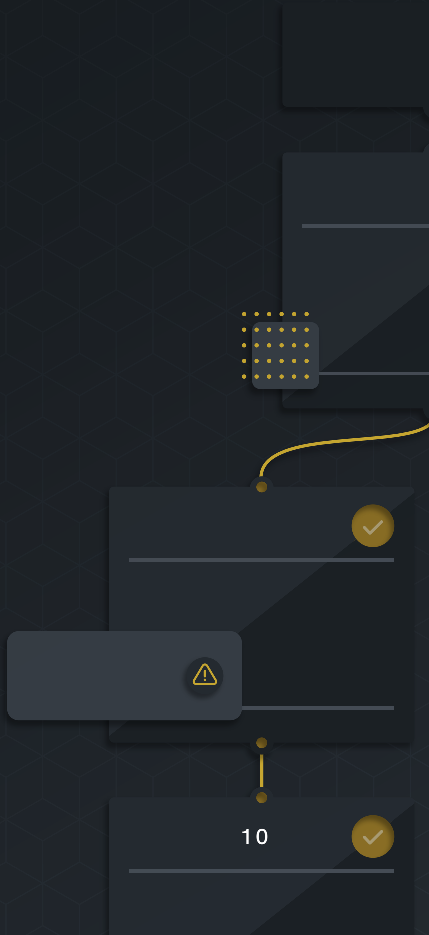
More likely than not, your software supply chain includes open source code. Nearly 99 percent of application code bases today contain open source software, comprising more than 85 percent of the code base for enterprises.[2]

Leading organizations recognize open source as essential, not optional, for modern software development success. That doesn't mean ignoring or downgrading potential risk. Instead, being an informed open source consumer means staying up to date about the open source code you use, and using the best tools and security best practices to keep your team safe along the way.

GitHub tip

Update user tokens and passwords on a regular and automated cadence.

Rotating your tokens and passwords reduces the possibility of any unauthorized access.



[2] 2019 State of the Software Supply Chain Report, Sonatype

Dependabot helps keep open source-related vulnerabilities at bay

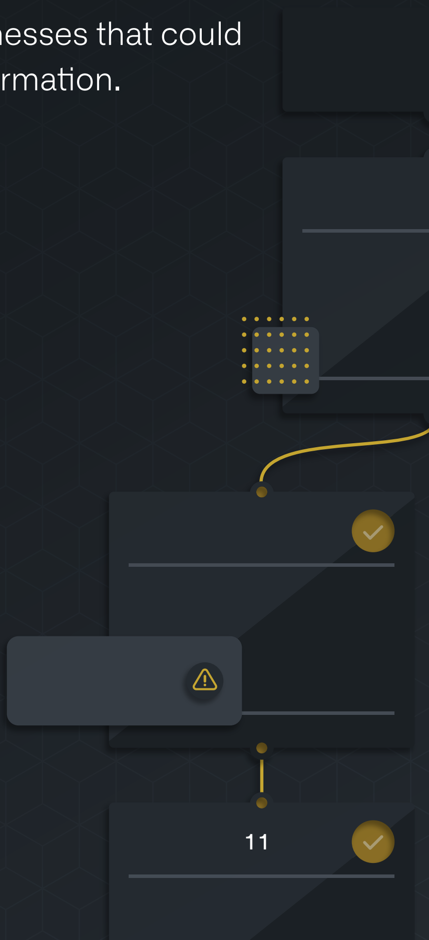
Pinterest places great value in the security GitHub provides, including vulnerability alerts and automated fixes. Thanks to Dependabot, a GitHub feature that automatically checks for out-of-date or insecure libraries and generates pull requests to update them, "we're also presented with the solution, which is very handy," Engineering Architect Jon Parise said. This triggers Pinterest to fix the issue on GitHub. "It's a great reminder to audit our internal software for similar vulnerability."

GitHub has also helped the cloud communications platform Twilio to build more secure code. In addition to code reviews and an automated pull request validation process, they've integrated BlackDuck and Anchore as scanning technologies for the third-party software they leverage. And they integrated SonarQube's static code analysis with GitHub at the pull request level—giving them feedback on insecure dependencies within the code. "GitHub's vulnerability alerts are very useful," Developer Evangelist Dominik Kundel explained.

GitHub tip

Track vulnerable dependencies automatically and create an established process for addressing security alerts.

Scanning source code for vulnerabilities reveals flaws and weaknesses that could be exploited or leak information.



"Especially as we continue to build a variety of different demo applications, having bots like Dependabot catch vulnerabilities is super helpful." Since, they've partnered with GitHub to launch Git Guard, a service that helps protect developers from inadvertently committing account tokens.

Automated security fixes benefit both developers and security teams

Supporting healthcare providers around the globe requires an international team to match. With 78,000 employees worldwide, McKesson works to make better

FEATURE SPOTLIGHT

How Dependabot works

1. **Dependabot checks for updates**

   Dependabot pulls down your dependency files and looks for any outdated or insecure requirements.

2. **Dependabot opens pull requests**

   If any of your dependencies are vulnerable or out-of-date, Dependabot opens individual pull requests to update each one.

3. **You review and merge**

   You check that your tests pass, scan the included changelog and release notes, then hit merge with confidence.

care possible for patients everywhere. But seamless collaboration among a rapidly growing number of employees is a challenge for any company. To bridge the gap between tech and security teams, McKesson turned to GitHub. “Security is just as much the responsibility of the developers as it is of the security team,” explained Hurley. “GitHub allows us to enable security, versus enforcing it. The sooner we can catch vulnerabilities and product issues, the better it is for the company in the long run.”

GitHub makes it easy for developers to address and fix any potential issues in their code. Automated security updates send security alerts whenever vulnerabilities are identified, and automatically opens a pull request with the recommended security updates. With GitHub’s security features, McKesson’s developers can code in confidence knowing their work is secure.

> “We’re less concerned about vulnerabilities in our code because of Dependabot. It’s helped us find and eliminate weak spots and is a big step forward for our software quality.”
>
> *— Lead Software Engineer for Developer Tools, Decathlon*

PART THREE

DevSecOps and shifting security left

BRANDS IN THIS CHAPTER:

NETDATA California Department of Technology Auth0

How can we build more secure applications, faster?

Integrate security directly into the developer workflow.

—

As we've seen, it's easiest and most cost-effective to solve application security problems at the source. While DevSecOps and “shifting security left” are a step in the right direction, reducing vulnerabilities in your production code still takes time.

There’s only one way to truly shift security left and keep up with customer demand: Putting developers front and center for application security. Tools like code scanning and secret scanning ensure every `git push` is scanned for new potential security vulnerabilities—so developers can solve security issues without ever leaving their existing workflows.

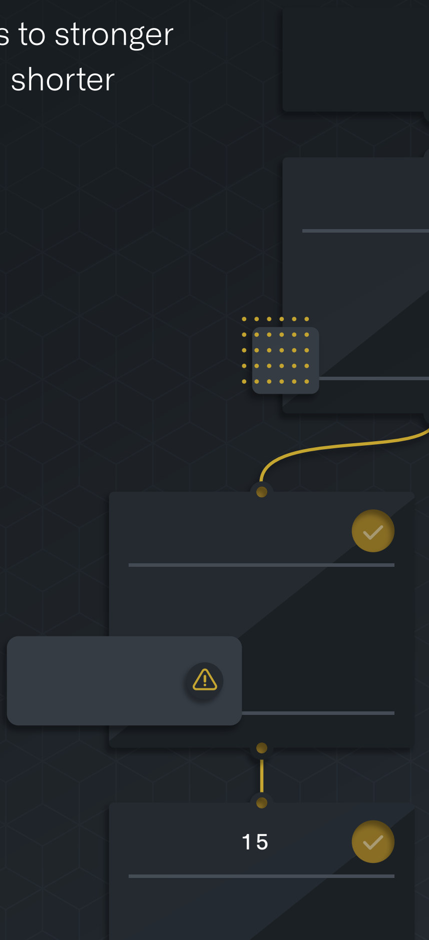
Find and fix vulnerabilities as you build

With a promise to provide real-time insights while growing to scale, the Netdata team needed a way to stay nimble—and tools that could match. To start, they

GitHub tip

Make your security team an integral part of development, and include them in key planning and milestone sessions.

Keeping your security team and developers in sync leads to stronger application security and shorter development cycles.



embraced their roots: open source repositories on GitHub. Their free, open source tool is downloaded 600,000 times a day. “We use GitHub for everything,” explained CEO and Founder Costa Tsaousis. “By actively participating in the open source community, we’re able to tap into millions of users offering immediate feedback and improvements for our software.”

But even with open source feedback and community contributions, maintaining a lean development team puts developers in charge of both building and securing code. As Netdata’s codebase grew, so did the risk of security alerts—when new code was merged, fresh security vulnerabilities were found. Developing features and addressing vulnerability alerts was a difficult and slow balancing act, something that Tsaousis couldn’t rationalize with the company’s own fast-moving innovation: “If a computer can find problems before they reach production, you should take advantage of that.” So the team turned to a new security solution: GitHub Advanced Security with the powerful semantic code analysis engine, CodeQL.

Quickly, their new security toolkit became integral to Netdata’s development process. Thanks to CodeQL, developers were able to find and fix vulnerabilities as they wrote code—long before vulnerabilities could even be identified by their previous security analysis tool. “If Advanced Security reports error issues, the pull request isn’t allowed to be merged,” explained CTO Dimosthenis Kaponis. “If a security issue is found, we’re informed immediately. We go over anything GitHub has highlighted, and we make sure that it’s resolved before releasing a stable release. Without built-in security, you have to go through a number of additional steps when you review the code. For the developer who will press the merge button, it inspires confidence.”

With code and secret scanning, secrets stay in—vulnerabilities out

Being the primary technology entity for the US’ most populated state is no easy task. The California Department of Technology (CDT) keeps state departments up to date on the latest tech—but also ensures over 39 million residents can easily access public health and emergency services information. With GitHub’s support for collaboration and built-in security, the department was able to extend DevOps to DevSecOps. “We wanted to make it stable to increase collaboration and we wanted to document our controls and security. Security is a major part of our work," said Lead DevOps Engineer/Solutions Architect Shamal Siwan.

By combining GitHub and Azure DevOps, security is now a shared responsibility across the development process and built directly into the developer workflow. Before a developer commits code into GitHub, their code is pre-scanned and shown to developers as a "pre-commit." After code is committed, it triggers their CI pipeline, funneling the code through Veracode—CDT's static scan analysis tool—and SCAs. Developers simply point Veracode to one of their GitHub repositories and can immediately assess any reported code vulnerabilities.

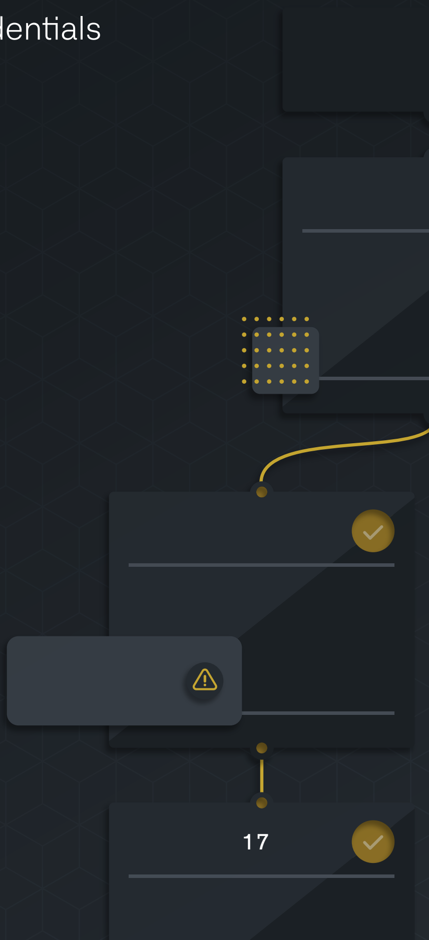
"The goal here is to fail fast," Siwan explained. "You fail fast, you fix it, and you move forward, because guess what? It takes you a minute to solve that security issue in development when it could potentially take you hours or days to fix in production. Today, you only need to lift a finger to commit code."

For Auth0, speed doesn't just give their developers peace of mind. The identity and authentication platform is responsible for securing over 4.5 billion customer logins per month—making time-to-fix a critical factor when a security vulnerability is found. After integrating GitHub Advanced Security, the team

GitHub tip

Scan your team's code for secrets and credentials during code commits.

Scanning credentials automatically prevents leaving hard-coded passwords or other credentials in source code.



quickly saw improvements “We’re already getting some significant value,” said Charlotte Townsley, Director of Security Engineering.

When Auth0’s security engineers discovered a security issue during a pen test, they were able to identify what the problem was and rapidly find where it existed in their codebase by scanning their repositories. “It saved us a lot of time,” Townsley confirmed, whereas prior to implementing Github Advanced Security, teams performed additional pen testing and less targeted scanning, and thus spent more valuable time identifying the bug before pushing an update to customers. “Instead of it taking a full day to find and fix one security issue, we were able to find and fix three issues in the same amount of time.”

> “We chose GitHub Advanced Security for its out-of-the-box functionality and the custom functionality that we can build off of. The two of those together we really liked. Some of the other tools have that a little bit, but maybe not as robust.”
>
> *— Director of Security Engineering, Auth0*

PART FOUR

Secure collaboration

BRANDS IN THIS CHAPTER:

stripe Spotify DOW JONES

How can we collaborate securely?

Increase productivity by bringing open source best practices behind your firewall.

—

According to Harvard researchers, companies that contribute to open source get up to two times more productivity out of open source compared to companies that only consume open source. Put simply, actively participating in open source communities—and learning their best practices—pays off.

Using open source best practices behind your firewall—known as innersourcing—encourages teams to share their expertise and build better software, together. Building with the same tools and systems GitHub has created for the open source community, developers can easily discover and reuse internal code, contribute to open source projects, or get help from your own subject matter experts—all while staying secure and compliant.

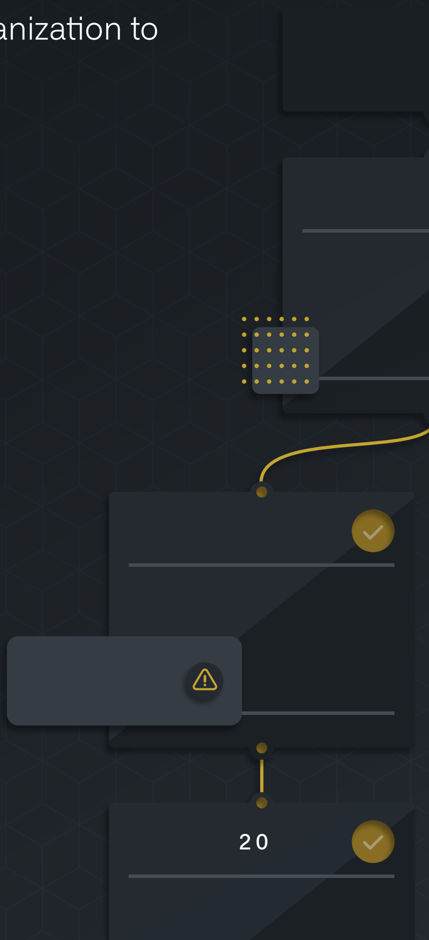
Encourage transparency and break down silos with ‘innersource’

While Stripe locks down its source code by hosting GitHub in a private data center, many of its projects

GitHub tip

Document data use and access policies and ensure they’re easy to find.

Keeping current security policies in one document makes them easier for everyone in your organization to find and follow.



are open for contribution on github.com/stripe. There, the team can collaborate with external developers and partners and take contributions to API libraries, SDKs, and other sample code. Internally, engineers still apply a shared set of principles as they work on critical pieces of their product. One of these principles, innersource, empowers any Stripe developer to get involved in any project and share their ideas and solutions. “This internal community removes friction as we build software, while making all of our projects more open and social,” said Developer Advocate Michael Glukhovsky.

Together, this internal community has not only created a better product; they’ve also contributed to a better developer experience, within the company and beyond. Some of Stripe’s internal tooling just doesn’t exist anywhere else—from custom Slack integrations to unattended deploys that automatically run CI and prepare a pull request for deployment.

Like Stripe, Spotify uses GitHub for innersource projects and collaboration. They also rely on GitHub to securely open up their code, work with external partners, and participate in the open source community. With open source close to the team’s process, they’ve been able to learn from the larger developer community.

Connect with the community and projects behind your code

By combining self-hosting with the cloud, developers at Dow Jones get the best of both worlds: hosting GitHub on-premises for source code and securely engaging in open source. With access to features like GitHub Actions and GitHub Packages, Dow Jones developers can create private package registries to use internally while also exploring and downloading public packages from GitHub.com. “GitHub Packages is the perfect fit for the tools we use like Reapsaw, a Docker image that you build and execute as part of your pipeline,” explained Senior Security Engineer Pranavkumar Patel.

Engaging in the open source community has transformed how Dow Jones builds software, internally and externally. Teams don’t just release new applications to customers faster. According to Chief Information Security Officer Miguel El Lakkis, they also write better code. “The scrutiny that an open source project is exposed to definitely benefits the project itself. When it’s a private project or closed source project, it’s more about the business than about solving the difficult tech challenges that we have.” But El Lakkis also sees this as an opportunity for change. GitHub has helped

Dow Jones' developers create an innersource culture, and bring open source standards like collaboration and transparency into every project. Now developers build on each other's ideas, spend less time on manual tasks, and catch bugs earlier in the development lifecycle.

> "The scrutiny that an open source project is exposed to definitely benefits the project itself. When it's a private project or closed source project, it's more about the business than about solving the difficult tech challenges that we have."
>
> *— Chief Information Security Officer, Dow Jones*

FEATURE SPOTLIGHT

Built-in CI/CD and package management

1. **Get to know GitHub Actions**

   Actions makes it easy to automate all your software workflows, with world-class CI/CD. You can discover, create, and share actions to perform any job you'd like right within your GitHub repository—including CI/CD—and combine actions in a completely customized workflow.

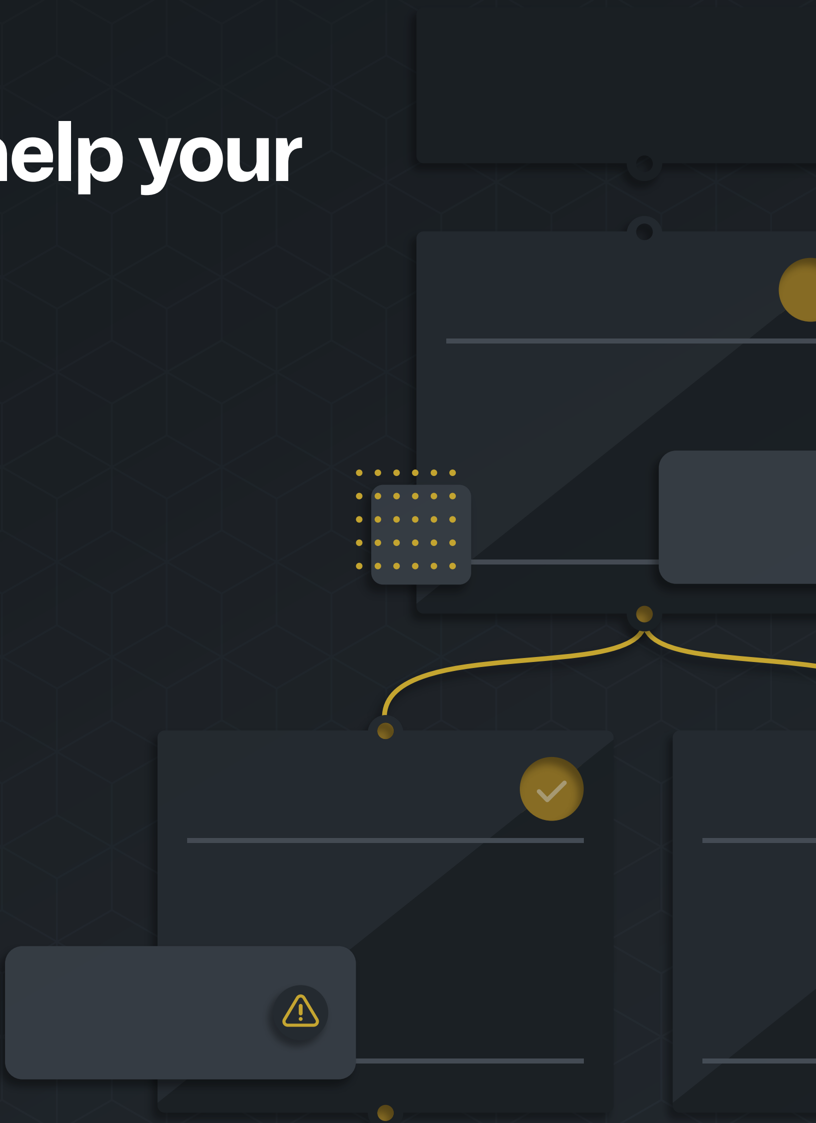
2. **Find out more about GitHub Packages**

   With Packages, you can safely publish and consume packages within your organization or with the entire world. Pair Packages with Actions to simplify package management, including version updates, fast distribution with our global CDN, and dependency resolution, using your existing GITHUB_TOKEN.

GitHub

Ready to see how GitHub can help your organization build securely?

Learn more at github.com/learn/security
or contact our Sales Team



WRITTEN BY GITHUB WITH ♥