

# Access Governance for GitHub




*Enforce access policies on source code repositories to achieve least privilege and meet compliance requirements*



GitHub is the de facto software collaboration platform for millions of developers worldwide. Like many other apps, GitHub offers its own permissions management system to help organizations implement role-based access control (RBAC). However, dependencies across an organization’s source code repositories cause development teams to struggle to create and maintain access rules and permissions from one repository to another. Repositories are often created on the fly and users are given access without consideration of SaaS security and least privilege. Moreover, the RBAC model of GitHub is complicated, utilizing organization roles (owners, members, moderators, billing managers), repository roles, actions, and account types. To secure access to sensitive data in GitHub, organizations need to understand the relationships between identities and the actions they can perform.

Gaining unauthorized access to source code is only the first step in a series of exploits that can be carried out by a threat actor. With just one-time access, a threat actor can easily download code for offline viewing, giving them ample time to harvest user credentials and API keys.

## Veza secures your GitHub repos

- 
**Visualize**  
 Enable identity, security, and IT compliance teams to understand the full scope of access permissions for all users and roles in GitHub.
- 
**Remediate**  
 Achieve continuous compliance with automated workflows for user access reviews and entitlement certifications. Reject access requests and manage temporal permissions all in one place.
- 
**Control**  
 Ensure that only active accounts in your identity provider can access repositories and manage deployment permissions for collaborators before they can push changes to a branch in GitHub.

## Challenges

- 
 Lengthy and expensive processes for meeting compliance as a result of manual efforts or utilizing third-party consulting services
- 
 Proliferation of local users, local roles, privileged users, and external users
- 
 Understanding and manage the digital blast radius of each user
- 
 “Security theater” - rubber stamping of entitlement certifications
- 
 Understanding Complex GitHub RBAC structures (role modeling, role analysis)

## Benefits

- 
**Reduced risk of IP theft or contamination from source code repos**
- 
**90% faster access reviews and certifications**
- 
**Meet regulatory compliance (SOX, ISO27001, SOC)**

## Use Cases

- ★ Privilege Access Violations
- ★ SaaS Access Governance
- ★ User Access Reviews

## RBAC Entities Supported

Organizational Accounts

Users

Roles

GitHub Apps

Teams

Code Repositories

*“To secure our customers’ data and stay compliant with global regulations, it’s critical to maintain the integrity and confidentiality of our source code. Veza enables us to monitor and enforce our access policies across GitHub and other data systems, allowing us to manage role-based access control at scale. With Veza, we can understand the combined effect of our access control layers to maintain least privilege.”*

Frank Dellé  
Head of Global Compliance



## About Veza

Veza is the identity security company. Identity and security teams use Veza to secure identity access across SaaS apps, on-prem apps, data systems, and cloud infrastructure. Veza solves the blind spots of traditional identity tools with its unique ability to ingest and organize permissions metadata in the Veza Authorization Graph. Global enterprises like Blackstone, Wynn Resorts, and Expedia trust Veza to visualize access permissions, monitor permissions activity, automate access reviews, and remediate privilege violations. Founded in 2020, Veza is headquartered in Los Gatos, California, and is funded by Accel, Bain Capital, Ballistic Ventures, GV, Norwest Venture Partners, and True Ventures. Visit us at [veza.com](https://veza.com) and follow us on [LinkedIn](#), [Twitter](#), and [YouTube](#).

## Who uses Veza for GitHub

### IT and IAM teams

- ✓ Architect least privilege principles for the organization’s entire codebase
- ✓ Identify orphaned local Github Enterprise accounts to close access control gaps and free up license seats
- ✓ Verify that GitHub admin accounts are in a group requiring multi-factor authentication

### Risk, Compliance and Audit teams

- ✓ Perform end-to-end GitHub access reviews from access certification to renewal, to remediation in a shared interface
- ✓ Enable internal auditors to verify that access control policies adhere to organizational security standards

### Security Engineering teams

- ✓ Apply more stringent access controls on user accounts that have admin permissions on GitHub
- ✓ Determine the level of risk associated with third-party app access on internal repositories
- ✓ Get alerted to inappropriate access of repositories by external collaborators

The screenshot shows the Veza interface with two filter panels. The first filter panel is for 'User Type' with a dropdown set to 'Outside Collaborator' and a 'True' button. The second filter panel is for 'User Status' with a dropdown set to 'Is Active' and a 'False' button. Below the filters, there are two cards: '1 GitHub User' and '1 GitHub Resource'. A card for 'Mr. X' is highlighted, showing a table of violations.

Name	Mr. X
Type	Custom User
Application Type	GitHub
Violation	GitHub outside collaborator
Violation	GitHub local users with no IdP