Veza for GitHub

Your source code is probably some of the most sensitive data your organization holds. It's not only the cornerstone of your intellectual property, but also a potential launching pad for supply chain attacks, especially with the rise of Infrastructure-as-Code.

Veza allows you to definitively answer the question of:

Who can take **what action** on **what source code?**

Identity security challenges in GitHub

Complexity of access controls

There are over 90 distinct permissions a user can have on any given repository. Standard roles can help to aggregate permissions, but roles vary by repository. This makes managing access for a high number of contributors to a high number of repositories difficult to achieve.

Private and public repositories

It's common for companies to use private and public repositories in the same organization for different tasks. For example, key source code in private repositories, and open-source projects or sample apps in public repositories. At scale it's hard to identify where external collaborators should be, and where they shouldn't.

Company vs. personal identities

GitHub handles often follow developers from job to job throughout their career and exist in a global namespace. This makes it hard to distinguish internal from external users. Who exactly is CodeNinja666, anyway? Should they be able to push changes to source?

Four key source code risk factors

In addition to being valuable IP in itself, source code in GitHub or other version control systems is a uniquely valuable tool for attackers and a compromised repository can be a springboard for further attacks.

01 **Exposure of auth secrets**

Source code frequently contains authentication secrets used by machine identities that can be leveraged by a hacker to access other systems.

02 **Attack path analysis**

If an attacker can download and analyze your source code they can find and exploit vulnerabilities and back doors in your system.

03 **Supply chain attacks**

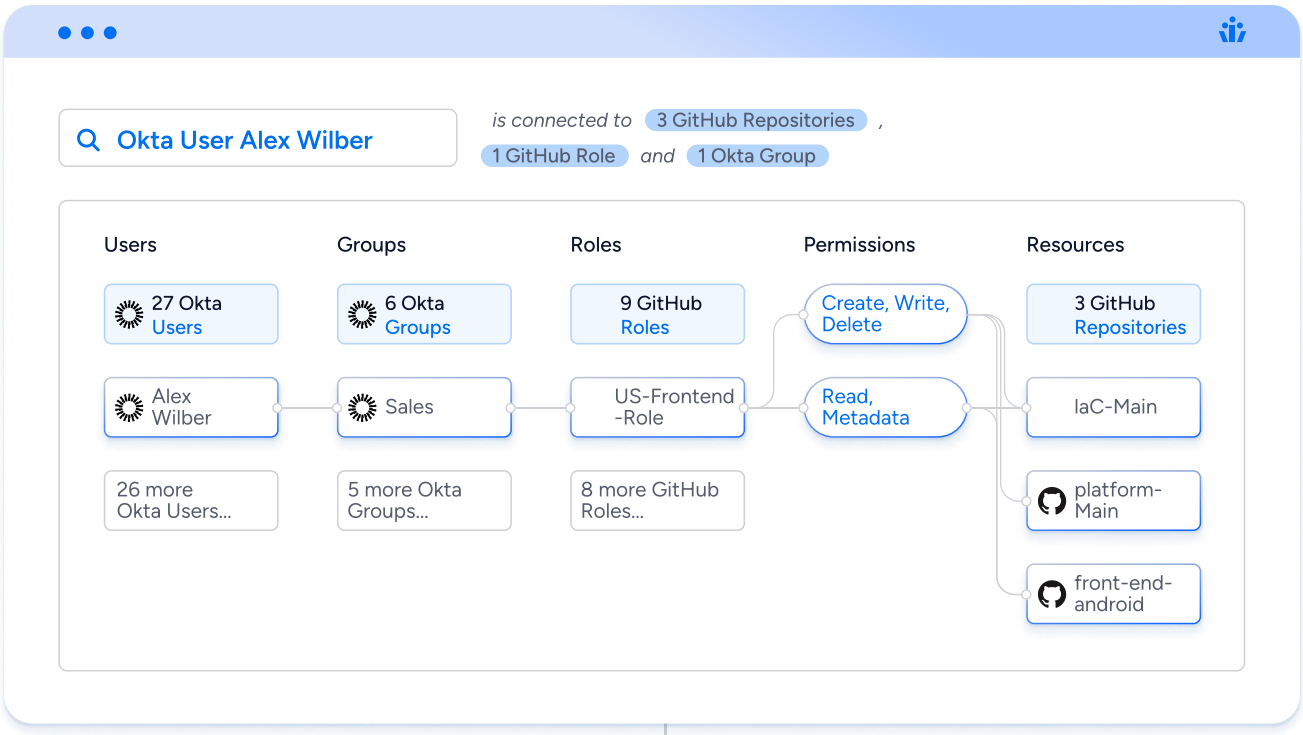
Storage of packages in GitHub repositories, combined with wide adoption of open-source gives attackers a way to inject malicious code into your source.

04 **Infrastructure-as-Code (IaC)**

The blueprint of your production environment itself may be stored in a GitHub repository, making GitHub central to your operational security.

How Veza can help

Veza is powered by its Access Graph, which gives organizations the ability to visualize access relationships between all identities and systems by connecting users, groups, roles, and permissions.



Key benefits

Reduced risk

Surface and prioritize identities with the greatest levels of access to source code in GitHub repositories.

Monitor external users

Compare GitHub users back to your identity provider to easily identify and manage external contributors and make sure access is limited to public repositories.

Least privilege

Automatically detect inappropriate access, including new privileged users on critical repositories like your production codebase or IaC repo.

Team efficiency

Use Veza to delegate access decisions and reviews to business managers who best understand each repository.

Key features

Effective permissions

Translate GitHub RBAC permissions into simple, human-readable language—"create, read, update, delete".

Automated monitoring

Watch continuously for policy violations and new privileged accounts, so you can comply with internal controls and external regulations without burdening security and governance teams.

Activity monitoring

Identify roles and users in GitHub with unused privileges to remove dormant identities, right-size access, and cut unused permissions.

Agentless read-only connections

Reading from both **GitHub** and your **identity providers**, gives a complete picture of the access granted to federated identities, revealing governance blindspots, like local and external users.