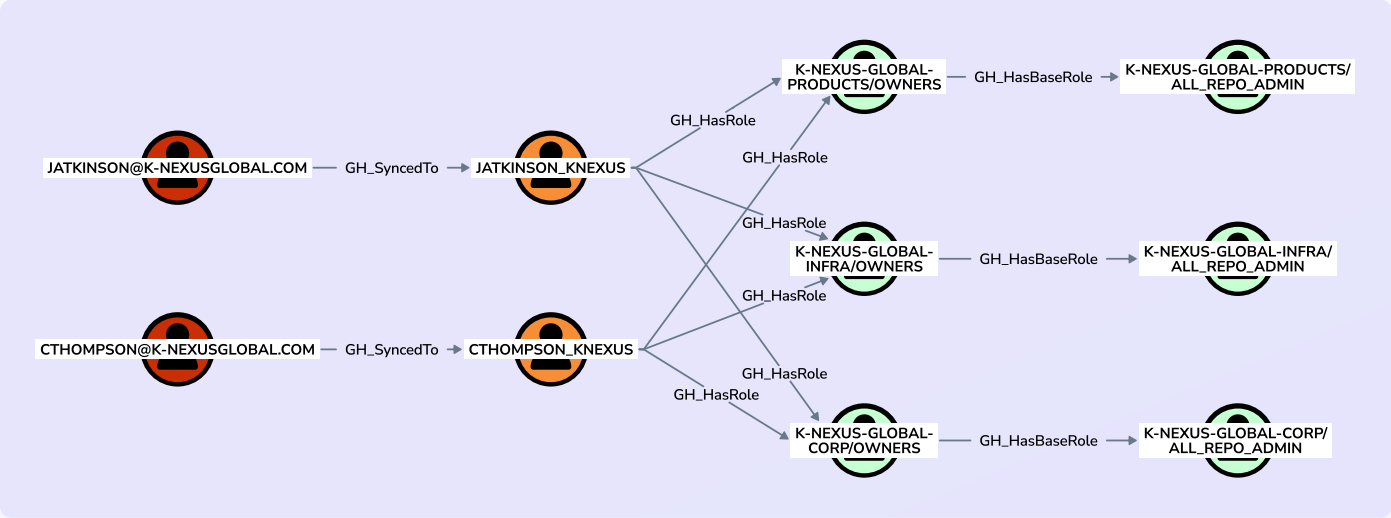
BLOODHOUND ENTERPRISE

OVERVIEW

Protecting GitHub Environments with BloodHound Enterprise



Adversaries don't stop at the identity perimeter. Once inside GitHub, they move through CI/CD pipelines, across OIDC trust relationships, and into cloud production environments.

By the time a security team realizes GitHub was the entry point, the blast radius already extends to AWS, Azure, or even customer-facing infrastructure.

The Problem

GitHub sits downstream of your identity provider and upstream of your cloud infrastructure. That position makes it a natural pivot point for adversaries to compromise a developer identity, a personal access token, or a CI/CD service account. They gain access to whatever GitHub is trusted to execute and wherever those executions are permitted to reach.

The controls governing this exposure are distributed across platforms, teams, and administrative boundaries. A configuration that looks correct in GitHub may interact with an Entra ID group assignment or an AWS role trust policy in ways that create unintended access. **No single team sees the full picture. Adversaries exploit exactly these gaps: the seams between platforms, between teams, and between intended and implemented controls.**

The Solution

BloodHound Enterprise gives defenders a map of what adversaries see—the chained relationships that combine to create exploitable attack paths to critical assets. With OpenGraph, that analysis extends to the platforms that sit outside traditional identity infrastructure but inside real-world attack paths.

The GitHub OpenGraph extension models the identities, permissions, workflows, and external trust relationships that govern effective control in GitHub Enterprise. Rather than inventorying settings and permissions, BloodHound Enterprise shows how those configurations of permission and access combine in practice and where the resulting paths lead.

GitHub OpenGraph Uncovers:

Identity and Access
Maps how user and team permissions accumulate through delegation and nesting, revealing where access compounds beyond what was intended.

Source Code and CI/CD
Models which repositories, branches, and workflows can be influenced, what can execute within them, and what that execution reaches downstream, including the build artifacts and distribution pipelines that connect internal code changes to external impact.

Credentials and Configuration
Surfaces where secrets, tokens, and application credentials are exposed and which identities can reach them—including exposure that static scanning misses.

External Trust Relationships
Maps SSO dependencies and OIDC trust configurations to show where GitHub-controlled execution can pivot into cloud infrastructure.

Taken together, that coverage answers the question no configuration audit can: given everything in place across your environment, where can an adversary go from any particular point in my identity and application network?

What This Looks Like in Practice

In a breach documented by Mandiant in Google Cloud's H1 2026 Cloud Threat Horizons Report, adversaries compromised a developer endpoint via a malicious NPM package, stealing a GitHub personal access token in the process. From there, they used GitHub's OIDC trust relationship to assume an AWS administrator role without ever touching a cloud console. Within 72 hours they had exfiltrated data from production S3 buckets, destroyed infrastructure, and renamed internal repositories public. **GitHub was not incidental to the attack, it was the pivot point.**

SpecterOps has observed the same class of exposure in client environments. During one engagement, BloodHound Enterprise revealed that any employee provisioned through Okta could take over critical GitHub repositories and assume highly privileged AWS roles used for CI/CD. No single platform's configuration review would have surfaced it, because the path didn't live in any single platform.

GitHub Is One Piece of a Larger Attack Surface

No single extension tells the full story. Adversaries don't limit themselves to one platform, and neither does OpenGraph. A path that originates on a Jamf-managed Mac that moves through Okta and lands in GitHub—where it assumes a privileged cloud role—is only visible when all three are in the graph.

BloodHound Enterprise OpenGraph extensions for Okta and Jamf bring the same attack path analysis to identity provider configurations and macOS device management environments. **Together, they give defenders visibility into the cross-platform paths that form between systems, across administrative boundaries, and outside the reach of any single platform review.**

Ready to See Your GitHub Environment the Way Adversaries Do?

Request a demo of BloodHound Enterprise:

Visit specterops.io/get-a-demo ›