Security Vulnerabilities of Git

BEYOND IDENTITY

Repo compromise is costly and irrecoverable

Attackers continue to exploit vulnerabilities in distributed, cloud-based Git environments. Git's great at helping with public, community-developed code, but they're not focused on enterprise-grade security. Recent attacks on Solarwinds, Kaseya, and NotPetya - have revealed that even mature, security-focused companies have enormous supply chain blindspots. They have shown that it's not only costly to remedy a breach of assets and third party tooling, credential theft, and key sprawl - it also erodes fundamental trust with the company and their intellectual property. Many times, that trust is irrecoverable.

NotPetya

Total cost for impacted companies $10B

SolarWinds

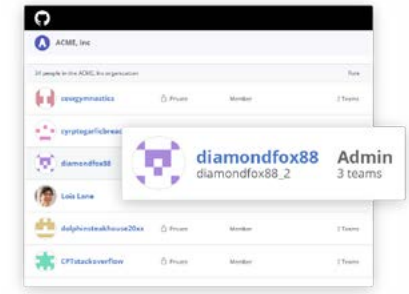
Avg of $12M for impacted companies

Kaseya

Ransomware attackers demanded $70M

Logs and transaction records are unreliable

Logs and transaction records in Git are insufficient for asserting who made that change. It's easy to impersonate someone on Git, contributors are often using their own Git accounts that are not company issued, contributors can write whatever they want in the author field, and to top it all off, security tools often slow down software velocity, so companies avoid using them.



It's easy to impersonate someone on Git

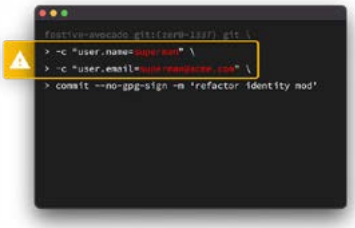
It's impossible to know if contributors are authorized developers when they're using their personal Git accounts, which aren't tied to corporate identity.

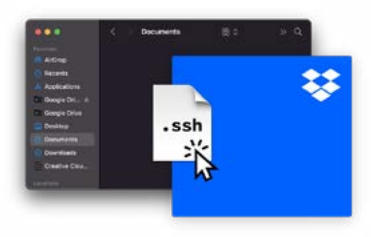
Personal git accounts:

- Username
- Display name
- Display icon

Author field is unreliable

Contributors can sign the author field of a commit with whatever name they'd like, which makes commits untrackable.





Contributors can check in code without the SSO

Contributors can login and check in code to Git without having to login through the SSO, evading corporate authentication controls.

Audit Standards for Code Reviews

It's easy to spoof users in Git, so it's difficult to trace where a vulnerability came from. The only way to achieve code integrity and authenticity is to trust the signature on every commit.

Infrastructure-as-code (IAC)

If your infrastructure is compromised, attackers can open all ports, change the firewall, and your network's wide open. Preventing unauthorized commits is a crucial step in securing your IAC.

Third party development

Third party contributors are checking in code on non-company-issued machines. Verifying author commit signing is the only way to ensure that a malicious actor didn't check in code.

Security tools often compromise software velocity

Speed of development is crucial to your business. Getting things into your developer's hands that can help them do their job without direct support can make or break your ops goals. Running security software in parallel with existing dev processes speeds things up.

About Beyond Identity

Beyond Identity is fundamentally changing the way the world logs in–eliminating passwords and all phishable factors to provide users with the most secure and frictionless authentication on the planet. Our invisible, passwordless MFA platform enables companies to secure access to applications and critical data, stop ransomware and account takeover attacks, meet compliance requirements, and dramatically improve the user experience and conversion rates. Our revolutionary zero-trust approach to authentication cryptographically binds the user's identity to their device, and continuously analyzes hundreds of risk signals for risk-based authentication. For more information on why Snowflake, Unqork, Roblox, and IAG use Beyond Identity, check out www.beyondidentity.com.

Ready to prevent tampering of your software components?

GET A DEMO

beyondidentity.com info@beyondidentity.com

BEYOND IDENTITY