Security Controls in Shared Source Code Repositories

Cuitlahuac Hernandez
CSD 380
July 19, 2025

Introduction

Topic Overview:
Shared repositories like GitHub, GitLab, and Bitbucket make collaborative development possible, but without proper security controls, they can become vulnerable points of attack.

Why It Matters:
Poorly secured codebases can lead to data leaks, credential exposure, and compromised production systems.

Access Control

- **Use Role-Based Access Control (RBAC)** to limit permissions by role.
- **Avoid giving write access to all collaborators.**
- Use **least privilege principle**: only grant what's necessary.
- Enable **2FA** for all contributors.

Credential and Secret Management

- Never commit API keys, tokens, or passwords to the repo.
- Use secret scanning tools like **GitHub Secret Scanning** or **TruffleHog**.
- Store secrets in a **.env** file and ignore it with **.gitignore**.

Code Review and Pull Requests

- Require **pull request reviews** before merging.
- Set **branch protection rules** to prevent direct pushes to main.
- Implement **automated linting and security checks** during PRs.

Static Code Analysis

- Run tools like **SonarQube, Snyk, or Checkmarx** to scan for vulnerabilities.
- Automate security scanning on each commit/PR.
- Fix vulnerabilities **before** merging into main.

Audit Logs and Monitoring

- Enable audit logs in GitHub/GitLab to monitor:
  - Changes in access rights
  - Repository deletion or visibility changes
- Use webhook notifications for suspicious events.

Dependency Management

- Use dependency scanning tools (like **Dependabot**, **npm audit**, or **Snyk**) to detect outdated or vulnerable libraries.
- Set up alerts for critical vulnerabilities.
- Regularly update dependencies to reduce risk.

Repository Configuration & Hygiene

- Protect default branches.
- Require signed commits if possible.
- Clean up old branches and stale forks regularly.

Conclusion

- Security in shared repositories isn’t optional—it's essential.
- Automate, monitor, and enforce best practices.
- Protecting the codebase protects your product.

Resources

- https://docs.gitlab.com/development/code_review/
- https://cheatsheetseries.owasp.org/cheatsheets/Secrets_Management_Cheat_Sheet.html
- https://docs.github.com/en/organizations/managing-user-access-to-your-organizations-repositories