Security Controls in Shared Source Code Repositories

Ian Lewis

05/11/25

Introduction

- Shared repositories (like GitHub, GitLab, Bitbucket) are essential for collaborative development.
- They can introduce security risks if not properly controlled.
- Ultimate goal is to present best practices to secure source code repositories from breaches, leaks, and malicious activity.

Threats to Source Code Repositories

- Credential leaks (ex. API keys in commits).
- Unauthorized access to codebases.
- Malicious commits or insider threats.
- Dependency vulnerabilities.
- Insecure CI/CD pipelines.

Access Control Best Practices

- Use role-based access control (RBAC) to limit users to necessary permissions.
- Enforce multi-factor authentication (MFA).
- Revoke access immediately when users leave the team or project.
- Use SSH keys instead HTTPS passwords.

Code Review and Approval Workflows

- Require pull requests and mandatory code reviews.
- Implement branch protection rules (ex. require review, disallow direct commits to main).
- Use signed commits to verify authorship.

Secrets and Sensitive Data Protection

- Use git-secrets or similar tools to scan for keys/tokens.
- Store secrets in environment variables or vault services (ex. HashiCorp Vault, AWS Secrets Manager).
- Add .gitignore rules to prevent accidental commits of sensitive files.

Secure CI/CD

- Restrict CI/CD tokens to minimum privileges.
- Ensure build systems use isolated environments.
- Validate third-party dependencies for vulnerabilities.
- Scan artifacts before deployment (ex. Snyk, SonarQube).

Monitoring and Incident Response

- Enable audit logs to track activity.
- Set up automated alerts for suspicious behavior.
- Regularly review logs for unauthorized access or privilege escalations.
- Prepare an incident response plan specific to repository breaches.

Training and Organizational Policies

- Educate developers on secure coding practices.
- Create and enforce security policies for repositories.
- Conduct periodic security audits and risk assessments.
- Promote a culture of security from the ground up.

Key Takeaways

- Shared code repositories must be secure by design.
- Use layered controls: access, review, monitoring, automation.
- Combine tools, policies, and culture for strong security posture.
- Security is not a one-time fix, but an ongoing process.

Thank You

- **Sources:**

- Assembla. (07.08.23). *Source code security: Are you doing enough to protect your code?* Retrieved May 11, 2025, from https://get.assembla.com/blog/source-code-security/
- Digital Guardian. (n.d.). *Code protection: How to protect your source code*. Retrieved May 11, 2025, from https://www.digitalguardian.com/blog/code-protection-how-protect-your-source-code
- Encryption Consulting. (04.29.25). *Are code repositories safe for your source code?* Retrieved May 11, 2025, from https://www.encryptionconsulting.com/are-code-repositories-safe-for-your-source-code/
- Endpoint Protector. (04.08.22). *Your ultimate guide to source code protection*. Retrieved May 11, 2025, from https://www.endpointprotector.com/blog/your-ultimate-guide-to-source-code-protection/
- National Cyber Security Centre (NCSC). (n.d.). *Protect your code repository*. Retrieved May 11, 2025, from https://www.ncsc.gov.uk/collection/developers-collection/principles/protect-your-code-repository
- Security Patterns. (2024). *Code management security pattern*. Retrieved May 11, 2025, from https://securitypatterns.io/docs/01-code-mgmt-security-pattern/
- Snyk. (01.09.23). *Securing source code repositories*. Retrieved May 11, 2025, from https://snyk.io/articles/securing-source-code-repositories/
- TechTarget. (05.26.22. *Top 4 source code security best practices*. Retrieved May 11, 2025, from https://www.techtarget.com/searchsecurity/tip/Top-4-source-code-security-best-practices
- Wiz. (10.31.24). *Source code security: Best practices and techniques*. Retrieved May 11, 2025, from https://www.wiz.io/academy/source-code-security