Security Controls in Shared Source Code Repositories

—

Brittni Ray
Module 11.2
CSD 380

Introdution

What is a shared source code repository?

- A shared source code repository (such as GitHub or BitBucket) is a platform in which developers can keep, manage and trace varying versions of code. Such repositories help maintain version control and encourage efficient collaboration between team members as they work on the same code base. Although a fantastic option, these repositories can sometimes expose code to threats if it is not being properly secured—such as data breaches or code injection.

Security Risks in Shared Repositories

Some of these resulting risks are as follows:

1. Unauthorized Access
    a. Attackers or disgruntled employees could gain access to a private repository
2. Code Injection
    a. In an either intentional or unintentional way, malicious code could be injected into a repository
3. Sensitive Data Leak
    a. A team member could accidentally commit code that includes sensitive data, such as credentials or API keys.
4. Denial of Service (DoS)
    a. These would include any not good-hearted attempts to overload the repository with irrelevant commits.

Best Practice 1 - Control Access and Permissions

- Follow the principle of least privilege, which dictates that only the most basic and necessary of rights should be granted to each user
- Use multi-factor authentication in order to enhance security, ensuring further that no one will be able to gain easy access to an account that is not their own
- Regularly review and audit the logs, looking for any sort of activity or commits that may seem unusual.

Best Practice 2 - Review Code

- Mandatory code reviews, especially peer reviews that can be conducted more often and more quickly, can help ensure that both quality of code and security of the code are good before it is merged into main.
- Automatic systems can also conduct a static code analysis, which will automatically review this code for any security vulnerabilities.

Best Practice 3 - Manage Confidential Information

- Ensure that any information that is sensitive or confidential is stored directly within the source code. Information like passwords and API keys should never be in a place where many users can access it
- Similarly, secure personal passwords and information to access a repository in a place where no one can get to it, such as a password manager

Best Practice 4- Ensure Shared Repository is Up-to-Date

- Make it a priority to check the application that you are using to work on a shared repository regularly. The latest version of software tends to be on trend with stopping any trending vulnerabilities that may be happening at the time, leaving older versions weaker and more vulnerable to attacks

Sources

Marcel.L. (2024, August 17). *GitHub Repository Best Practices*. DEV Community. https://dev.to/pwd9000/github-repository-best-practices-23ck

Pandey, P. (2024, August 16). *Git security: Best practices for keeping your code safe*. DEV Community. https://dev.to/prankurpandeyy/git-security-best-practices-for-keeping-your-code-safe-1nep

Protect your code repository. NCSC. (n.d.). https://www.ncsc.gov.uk/collection/developers-collection/principles/protect-your-code-repository

Static code analysis. Static Code Analysis | OWASP Foundation. (n.d.-a). https://owasp.org/www-community/controls/Static_Code_Analysis