SECURITY CONTROLS IN SHARED SOURCE CODE REPOSITORIES

Sarah Ewing
12/14/2024
Module 11.2 Assignment

WHAT IS SOURCE CODE PROTECTION?

ACCESS CONTROL

Implementing strict authentication and authorization measures to ensure only authorized personnel can access the source code

ENCRYPTION

Utilizing encryption techniques to protect the source code both while it's stored and when it's being transmitted.

CODE INTEGRITY

Using tools like version control systems and code signing to maintain and verify the integrity and authenticity of the source code.

WHY IS SOURCE CODE PROTECTION NEEDED?

PROTECT INTELLECTUAL PROPERTY

Loss of intellectual property can lead to financial losses and reduced market share

PREVENT SOFTWARE PIRACY

Piracy results in unauthorized access, use, or modification, depriving developers of credit and compensation

AVOID MALWARE EXPLOITS

Vulnerabilities can be exploited by hackers to create malware or launch cyber attacks

MAINTAIN APPLICATION STABILITY

Unauthorized modifications can cause bugs, crashes, and poor user experiences

ENSURE DATA PRIVACY

Theft of sensitive information in source code could compromise user data protection and privacy

COMPLIANCE REQUIREMENTS

Non-compliance can lead to legal penalties, reputation damage, or financial loss

SECURITY CONTROL RECOMMENDATIONS

Implement Automated Code Scanning

- Proactive approach to identifying vulnerabilities and security weaknesses in source code
- Enables early detection of potential security flaws during the development process
- Tools:
  - Kiuwan - cloud-based platform for detecting and fixing security vulnerabilities
  - SonarQube - open-source platform for continuous code inspection and analysis
  - Checkmarx - provides static application security testing to find and fix vulnerabilities

SECURITY CONTROL RECOMMENDATIONS

Use Two-Factor Authentication

- Adds additional security to source code repositories
- Requires two forms of identification to access the codebase
- Guarantees the integrity of the codebase
- Safeguards sensitive information
- Contributes to a more resilient software development ecosystem

SECURITY CONTROL RECOMMENDATIONS

Limit User Access

- Prevents unauthorized modifications and leaks
- Implementing role-based access controls ensures only authorized personnel access specific parts of the codebase
- Version control systems like Perforce offer robust access control features
- Administrators can define and manage user permissions with systems that offer this feature

SECURITY CONTROL RECOMMENDATIONS

Deploy Encryption Tools

- Encryption tools help secure sensitive information in source code
- Techniques safeguard sensitive configurations, passwords, and API keys stored in the codebase
- Protects source code from unauthorized access
- Tools like HashiCorp Vault or Azure Key Vault can manages secrets securely

SECURITY CONTROL RECOMMENDATIONS

Create Clear Source Code Security Policies

- Establish clear and comprehensive source code security policies to guide developers on secure coding practices
- Policies should cover:
  - Code review processes
  - Vulnerability handling
  - Guidelines for integrating third-party libraries
- Well-defined security policies help maintain consistency across the development team
- Reduces the risk of security breaches

RESOURCES

- https://www.digitalguardian.com/blog/code-protection-how-protect-your-source-code
- https://research.aimultiple.com/source-code-security/
- https://get.assembla.com/blog/source-code-security/
- https://snyk.io/articles/securing-source-code-repositories/