Security Controls in Shared Source Code Repositories

Alisa Steensen

Module 11.2

What is a repository?

Version Control

A space where developers can store and manage their source code files.

Allows multiple people to work on the project and track each developer's changes.

Why you shouldn't blindly trust a code repository?

Real-world example – Github Breach (2023)

- Unauthorized access led to the theft of 3 code signing certificates.
- Github acted quickly and revoked the stolen certificates.
- What if they didn't?
  - Malicious Software could be signed and distributed.
  - Fake certificates could be used to exploit systems globally.

Open-Source Dependencies

- Developers often use packages from public repositories.
- Saves time, but introduces security risks.
- Attackers can publish harmful code disguised as useful packages.

Case Study – Trojanized jQuery Packages

- Infected packages were uploaded to npm, Github, and jsDelivr.
- Caused dependency confusion attacks.
- 68 malicious packages were identified.

Why you shouldn't blindly trust a code repository?

Understanding Dependency Confusion

- Attackers upload modified versions of legitimate packages.
- If developers install them, attackers gain control.
- It's a common software supply chain attack.

Choosing the Right Code Repository

- Many repository platforms exist (e.g., GitHub, Bitbucket, GitLab).
- Your choice impacts your codebase's security.
- Even trusted platforms have faced cyber-attacks.

Final Takeaway – Developer Responsibility

- Platforms offer strong security, but can still be targeted.
- Developers must:
  - Vet all third-party code and packages.
  - Avoid using untrusted or unknown sources.
  - Stay informed and cautious to prevent cyber incidents.

Methods of Affecting a Code Repository

- **Code Repositories can be targeted by attackers.**
  - Most attacks can be prevented by paying attention to small details.
  - Before using the code from a repo, ask yourself:
    - Is the repo active and maintained?
    - Are other developers using it?
    - Was it previously deleted and reinstated?
- **Attack Technique #1 – Typosquatting**
  - Attackers mimic names of popular repositories with slight spelling changes.
  - Developers may accidentally download the wrong package.
  - Frequently used on package registries like npm, PyPI, and others.
- **Attack Technique #2 – Revival Hijack**
  - When a popular repo is deleted, attackers claim the name and reupload infected code.
  - Example from JFrog research:
    - Over **22,000 PyPI packages** could be hijacked this way.
  - Developers may not notice the change and use malicious code.

Methods of Affecting a Code Repository

- **Attack Technique #3 – Credential Theft**
  - Accidentally committing sensitive data (e.g., tokens, API keys) is common.
  - Attackers can use these credentials to:
    - Access private repos.
    - Learn about the user to attack other accounts.
- **Attack Technique #4 – Cloning and Modifying Repositories**
  - Public repos allow pull requests from anyone.
  - If not carefully reviewed, malicious code can be merged.
  - Attackers can plant backdoors or harmful code.

Best Practices

Whether storing files or running CI/CD pipelines, security is critical. Developers must follow strict practices to protect codebases from threats.

- Use — Use Built-in Security Features
- Enforce — Enforce Strong Authentication
- Stay — Stay Updated and Review Regularly
- Manage — Manage Access and Permissions
- Protect — Protect Sensitive Files
- Scan — Scan for Secrets
- Enforce — Enforce Code Signing
- Have — Have an Incident Response Plan

What to do after a source code security breach?

- **Contain the Breach**
- **Isolate compromised systems immediately.**
  - Disconnect from the network to stop the spread.
  - Block repository access for suspicious users.
  - Goal: Stop further unauthorized activity right away.

- **Investigate**
  - Conduct a thorough analysis:
    - Review access logs and audit trails.
    - Identify affected code files and data.
    - Determine what intellectual property or sensitive info is at risk.

- **Notify Relevant Parties**
  - Inform:
    - Dev team
    - Security personnel
    - Management
  - Communication is crucial.

What to do after a source code security breach?

- **Apply Fixes and Patches**
- **Review Access Controls**
- **Audit user permissions.**
- **Enforce principle of least privilege.**
- **Implement strong authentication**
- **Strengthen Monitoring & Detection**
- **Implement or improve:**
  - Automated code scanning
  - Intrusion detection systems (IDS)
  - Security event monitoring tools
- **Review and Improve the Plan**
- **Conduct a post-mortem of the incident.**
- **Evaluate the effectiveness of your response.**
- **Update the incident response plan based on lessons learned.**
- **Regularly test your breach strategy.**

References:

- Open Source Security Foundation. (2023, September 14). *OpenSSF releases Source Code Management Best Practices Guide*. Retrieved July 17, 2025, from https://openssf.org/blog/2023/09/14/openssf-releases-source-code-management-best-practices-guide/
- Encryption Consulting. (2024, November 14). *Are code repositories safe for your source code?* Retrieved July 17, 2025, from https://www.encryptionconsulting.com/are-code-repositories-safe-for-your-source-code/
- Sade, Y. (2025, June 9). *Top 8 Git secrets scanners in 2025: key features and top tools*. Jit.io. Retrieved July 17, 2025, from https://www.jit.io/resources/appsec-tools/git-secrets-scanners-key-features-and-top-tools-
- Assembla. (2025, June [exact day unknown]). *Source Code Security Best Practices: A Complete Guide*. Retrieved July 17, 2025, from https://get.assembla.com/blog/source-code-security/