



# Assignment: Security Controls in Shared Source Code Repositories

CSD 380

MODULE 11.2 ASSIGNMENT

JULIE SAKAI

5/10/25



# Introduction

- Shared source code repositories is important in software development, but it does post security risks if not properly protected.
- Without strong security measures, repositories can become vulnerable to breaches, unauthorized modifications, and supply chain attacks.
- Security is necessary
  - Protect intellectual property and sensitive data.
  - Prevent cyber threats like malware injections.
  - Ensure software integrity and compliances.

---

# Managing Access and Authentication

To secure repositories and access must be carefully controlled.

## Best Practices:

Implement role-based access control (RBAC) to limit permissions

Require multi-factor authentication (MFA) to strengthen login security

Regulate audit and adjust access level to prevent unauthorized entry

By restricting repository access, organization reduce the risk of data leaks and tampering,

---

# Secure Development Practices

Maintaining secure coding practices prevents vulnerabilities from being inserted into a project.



## Best Practices:

Sign and verify code commits to ensure authenticity.

Store credential in secret management tools instead of embedding them in code.

Develop a security policy outlining acceptable repository practices.



These steps help safeguard the integrity of source code and minimizes exposure to threats.

---

# Automated Threat Detection and Monitoring

Detecting security issues early is important for maintaining a protected repository.



## Best Practices:

Use automated vulnerability scanners to find weaknesses in the codebase.

Regular scan dependencies for risks and outdated packages.

Enable security alerts for unusual repository activity.

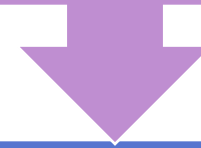


Proactive monitoring ensures threats are identified and addressed before they escalate.

---

# Protecting Open Source and Supply Chain Security

Third-Party libraries and open-source dependencies can introduce security risks if not managed properly.



## Best Practices:

Maintain a Software Bill of Materials (SBOM) to track external components.

Validate dependency integrity using supply chain security techniques.

Securely store container base images and verify their authenticity before deployment.



By securing the software supply chain, organizations can reduce exposure to potential exploits.

---

# Security Tools for Repository Protection

Tools help enforce security measures and detect vulnerabilities.

## Best Practices:

Static analysis tools scan for vulnerabilities in source code.

Secrets detection software prevents exposure of sensitive credentials.

Security testing frameworks identify misconfigurations.

Using security tools provides an additional layer of protection and automates security processes

---

# Continuous Security Practices







# Conclusion

- Securing shared source code repositories is important for software development.
- Control access with RBAC and MFA.
- Use secrets management tools to protect credentials.
- Automate vulnerability detection with security tools.
- Monitor repository activity for any unusual changes and threats.
- Implementing these security measures ensure that repositories remains safe and reliable.



# References

- G. (n.d.). *Quickstart for securing your repository*. GitHub Docs.  
<https://docs.github.com/en/code-security/getting-started/quickstart-for-securing-your-repository>
- H. (2025, May 5). *Best Practices for Securing Code Repositories*. Harness.  
<https://www.harness.io/harness-devops-academy/secure-code-repositories-best-practices>
- Tal, L. (n.d.). *Securing Source Code in Repositories is Essential: How To Get Started*. Snyk.  
<https://snyk.io/articles/securing-source-code-repositories/>