



Security Controls in Shared Source Code Repositories

Shared code repositories enable teams to collaborate and track code changes effectively. However, they also pose significant security risks. This talk outlines simple methods for protecting your code and safeguarding private information.

Presented by: **Steve Stylin**

Date: 7/18/2025

Introduction

Repository Role

Highlight the role of repositories in team-based development.

Security Risks

Summarize the primary security risks associated with shared access.

Best Practices

Outline key security best practices for repositories.

Understanding Security Risks

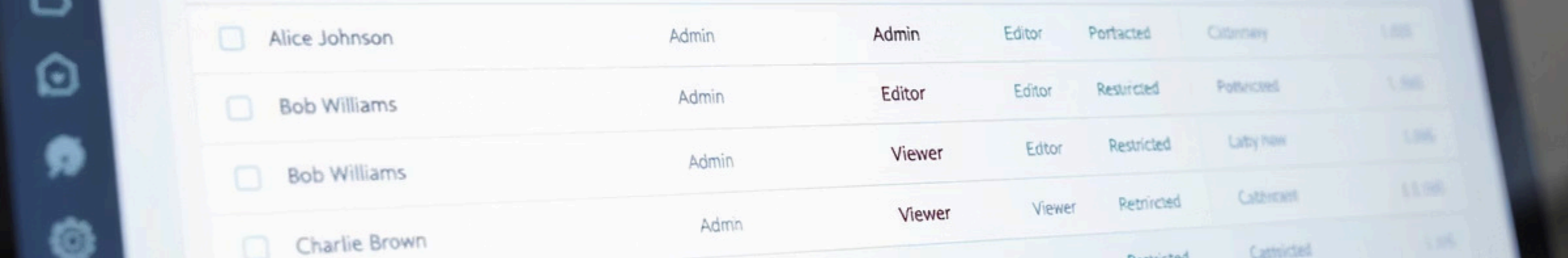
Primary Risks

- Unauthorized code access, data exposure, malware injection
- Leaking passwords and private information
- Hackers are putting malicious code into your project

Business Impacts

- ❌ Financial loss, reputational damage, loss of trust





<input type="checkbox"/>	Alice Johnson	Admin	Admin	Editor	Portacted	Editor	1,000
<input type="checkbox"/>	Bob Williams	Admin	Editor	Editor	Restricted	Restricted	1,000
<input type="checkbox"/>	Bob Williams	Admin	Viewer	Editor	Restricted	Editor	1,000
<input type="checkbox"/>	Charlie Brown	Admin	Viewer	Viewer	Restricted	Editor	1,000

Access Control

Principle of Least Privilege

Give each person only the access they really need.

Apply the principle of least privilege to minimize access rights and privileges.

Regular Permission Reviews

Regularly check and update access permissions.

Authentication Mechanisms



Strong Passwords

Implement robust password policies to prevent unauthorized access.



Multi-Factor Authentication

Require a second step to log in, like a code sent to a phone.



SSH Keys

Use secure login methods, such as SSH keys.

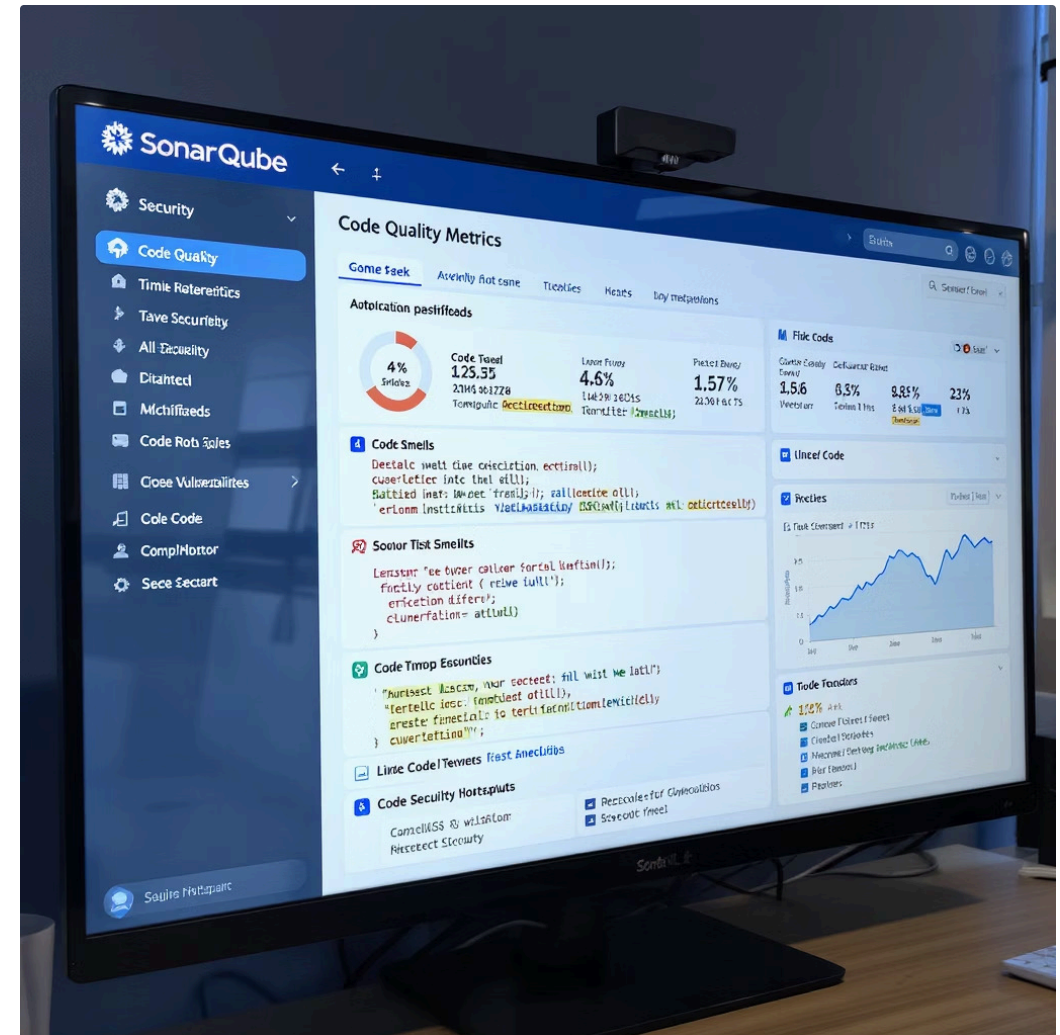
Citations: NIST Special Publication 800-63B on Authentication and Lifecycle Management.

Code Review and Monitoring

Best Practices

- Mandatory reviews, static analysis, and activity monitoring
- Utilize automated tools to scan code for errors
- Watch for unusual or suspicious activity in the repository

Example Tool



SonarQube for continuous code quality inspection.

Secrets Management



No Hardcoded Secrets

Never include passwords or API keys directly in your code.



Use Secret Vaults

Utilize specialized tools to safeguard secrets, such as Vault or AWS Secrets Manager.



Rotate Credentials

Change passwords and secrets often.

Citations: OWASP Secrets Management Cheat Sheet.



Incident Response Plan

1

Create Clear Plan

Create a clear plan to follow in the event of a security problem.

2

Practice Response

Practice what to do in a security crisis with your team.

3

Document Lessons

Write down what you learned from past problems to get better next time.

Continuous Education and Training



Best Practices

- Ongoing training, updated threats, and fostering awareness
- Stay informed about the latest security threats and effective mitigation strategies
- Promote a team culture where everyone cares about security



Conclusion

Importance

Strong security controls for shared repositories are very important.

Risk Reduction

Adopting best practices can significantly reduce your security risks.

Vigilance

Continue to improve and stay vigilant to maintain security.



References

- [GitGuardian \(2024\). State of Secrets Sprawl Report.](#)
- [SonarQube Documentation](#)
- [GitHub Security Best Practices](#)
- [NIST Special Publication 800-63B](#)
- [OWASP Secrets Management Cheat Sheet](#)