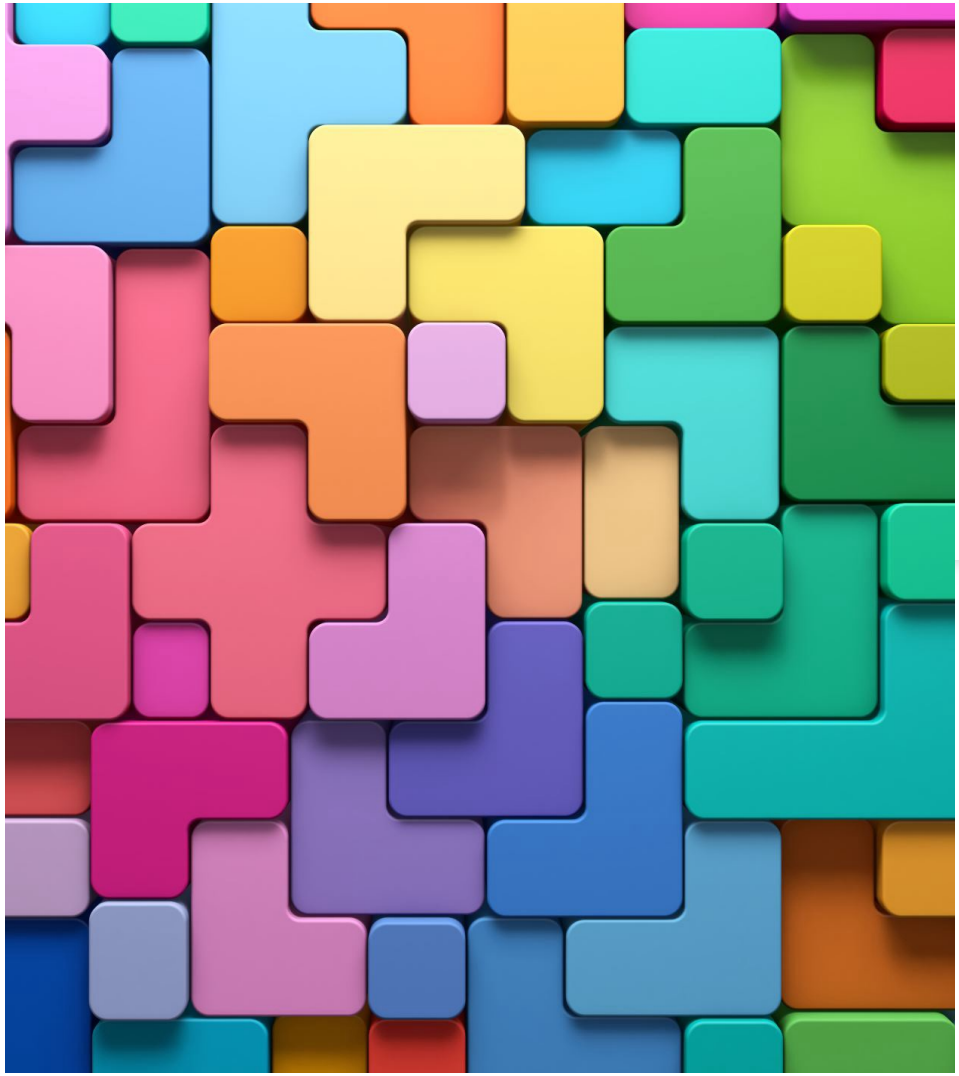
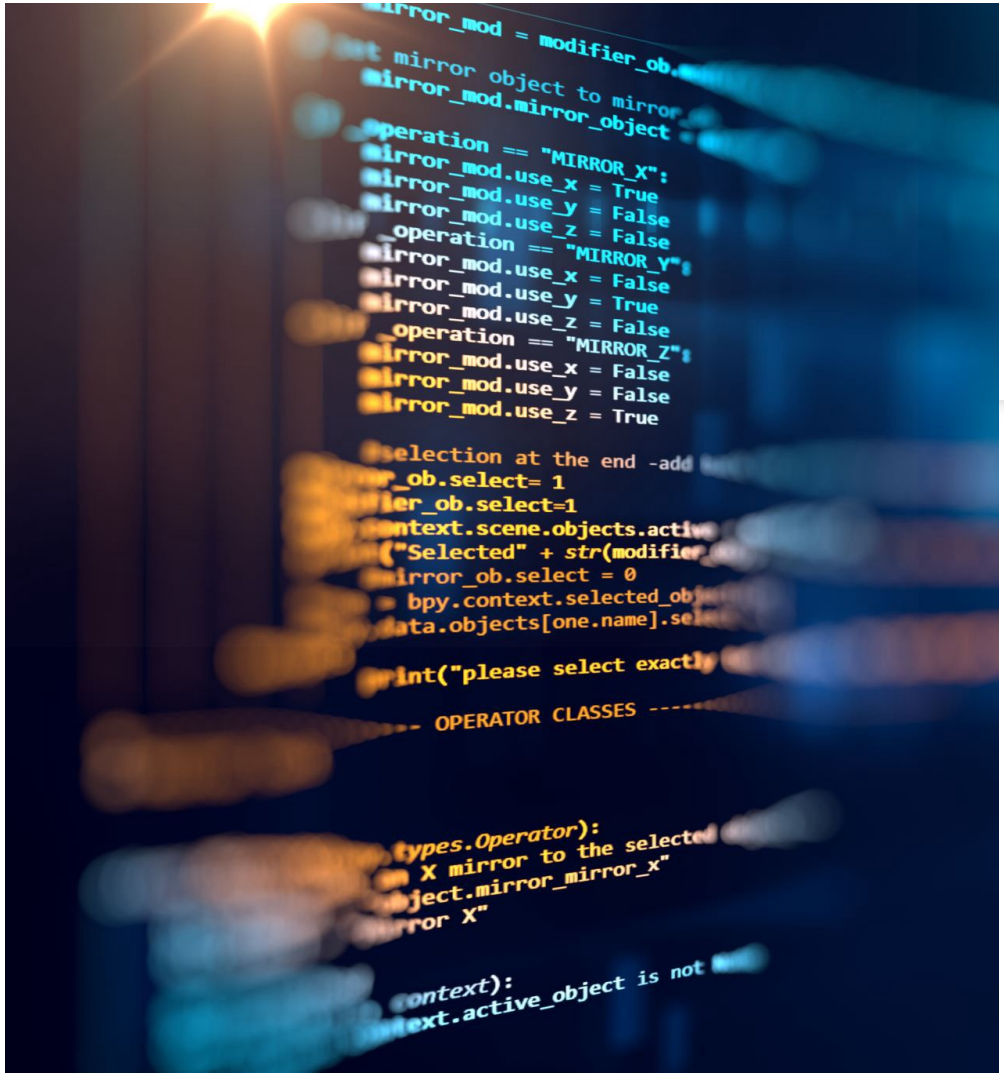


Security Controls in Shared Source Code Repositories

Emmanuel Diaz
CSD380 Module 11.2 Assignment
Prof. Adam Bailey



Securing Code Repositories: Why it matters?

Source code is the backbone of any software project, but this also makes it the primary target for attackers.

A breach can:

- Lead to **data leaks** and compliance failures.
- Introduce **malicious code** into the product.
- Undermine **team productivity** if the integrity of the repository is compromised.

Thoughtful security controls are key to protect work and build trust with stakeholders.

Managing Access: Who needs it?

Not everyone on the team needs full access to everything. To stay secure:

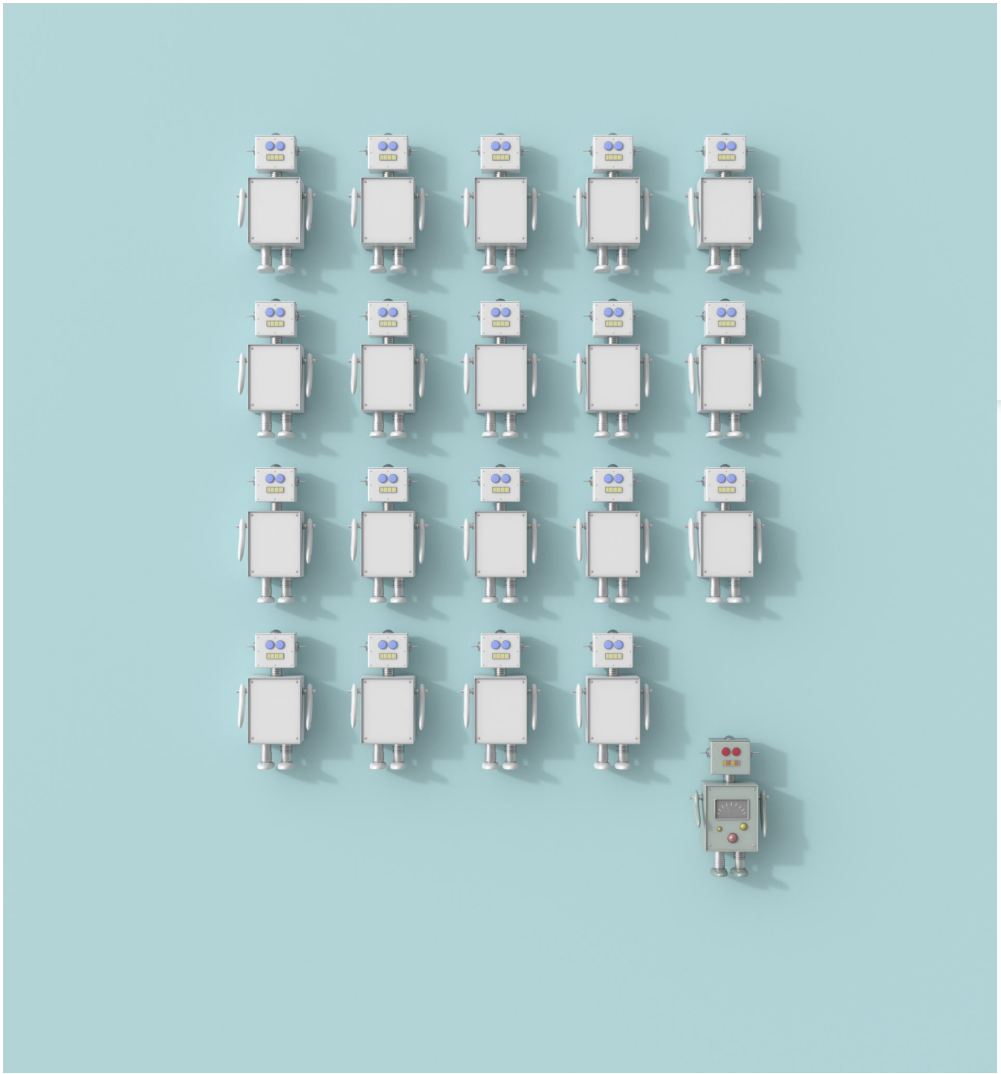
- Use **role-based access control** (RBAC) to give each user the level of access they need without compromising other areas by giving unnecessary access.
- Enable **multi-factor authentication** (MFA) to protect against compromised credentials.
- Regularly audit permissions to spot outdated or excessive access.

It's easier to protect what matters if everyone has the right access.

Setting up Security with Repositories

The way repositories are set can make all the difference.

- Start with **private repositories** for sensitive projects.
- Use **branch protections** to keep your main branches safe.
- Require pull requests and code reviews before merging.
- Mandate **signed commits** to verify authorship.
- Discourage direct commits to main branches, always review first!



Spotting Problems Before its too late

Security is easier when you catch issues early. Tools like GitHub Dependabot and static analysis scanners can:

- Find vulnerabilities in dependencies.
- Highlight risky coding patterns.
- Give you a chance to fix problems before they hit production.

Automation is key, let the tools do the heavy-duty work so the team can focus on building.

Build a Secure Collaboration Culture

Security is not only the tools that we use but also how teams work together towards that goal.

- Educate developers on secure coding practices.
- Make code reviews mandatory—more eyes mean fewer bugs and risks.
- Use encrypted communication channels to share repository details safely.

Monitoring and Auditing

Part of Security is always staying alert. Regularly monitoring can help:

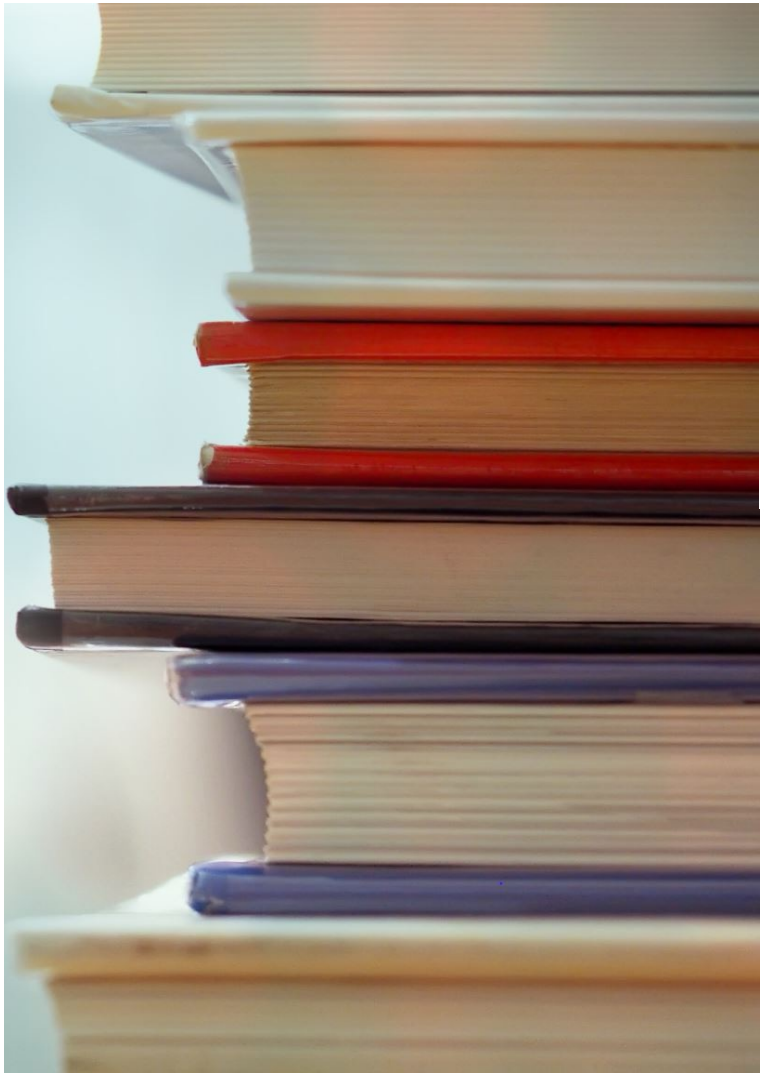
- Identify unusual activity before it becomes a problem.
- Highlight potential vulnerabilities or misconfigurations.
- Provide an audit trail to help investigate issues.

Preparing for the Worst

Murphy’s law says: "Anything that can go wrong will go wrong.”

Even with the best plan things can go wrong, that is why backups are essential.

- Regularly back up your repositories and test your recovery process.
- Keep backups in secure, offsite locations to protect against data loss or ransomware.
- Make recovery easy so downtime is minimal.



Conclusion

Securing shared source code repositories isn't just a technical task—it's a team effort.

- **Restrict access** to only what's necessary.
- **Automate checks** for vulnerabilities and misconfigurations.
- **Monitor activity** to spot issues early.
- **Back up everything** to ensure quick recovery.

Sources:

https://get.assembla.com/blog/source-code-security/

https://docs.github.com/en/code-security/getting-started/quickstart-for-securing-your-repository

https://snyk.io/articles/securing-source-code-repositories/