Source Code Repository Security Controls

Kendall Taylor | DevOps | Module 11.2 Assignment

Source Code Security

- The protection of the confidentiality and integrity of an applications source code
- A breach could lead to software vulnerabilities, access issues, or even data theft.
- Providing strong security builds client trust and protects against data breaches and unauthorized access

Why Source Code Security?

- Vulnerability Mitigation
  - Securing the source code can help to detect possible vulnerabilities before they can be found and exploited.
- Protection from Hackers and Malicious Actors
  - Provides safety from stolen data and code tampering
- Company and Customer Privacy
  - Helps to secure company information including business logic, algorithms, etc.
  - Maintains confidentiality of user's sensitive information.

Source Code Security: Best Practices

- Identities and Perimeters
  - Implement authentication methods to ensure the correct people are accessing the repository.
  - Least privilege should be used to ensure only those who need access are authorized to view the source code
- Sign and Authenticate Code Commits
  - Code signing for commits to ensure integrity and authenticity
  - Utilizing cryptographic signatures to verify the authenticity of a commit.

Source Code Security: Best Practices (cont.)

- Secrets Management
  - Do not hard code sensitive data and information into the source code
  - Store, manage, and access this data separately from the code.
- Protection Policies
  - Define protection policies and source code security measures and ensure that all authorized members are familiar with them.
  - Train and inform team members about best practices and ways to implement code protection.

Source Code Security: Best Practices (cont.)

- Monitor and Manage Source Code
  - Use automated testing and scanning tools to continuously monitor and detect vulnerabilities within the code.
  - Be aware of current security patches for the frameworks that are used in the code.
- Prevent Vulnerabilities of Code during Production
  - Use code review to detect security flaws before the code is committed and sent to production.
  - Use proper change management to ensure that changes are authorized and approved.

What to do if there is a Breach

- Restrict and isolate the systems that have been affected by the breach
- Identify the files, data, and property that have been compromised
- Notify parties that the breach has or will affect. (Stakeholders, development team, security, management, etc.)
- Patch and fix any known vulnerabilities to ensure they won't be exploited again.
- Review and manage access controls.
- Utilize automatic code scanning, security monitoring, and detection tools to enhance security.
- Review the response plan to assess the outcome and effectiveness and ensure that necessary changes are made.

Resources

- Berecki, B. (2022, June 10). *Best practices for source code security*. Endpoint Protector Blog. https://www.endpointprotector.com/blog/your-ultimate-guide-to-source-code-protection/
- *Source code security best practices: A complete guide: Blog*. Assembla. (2023, August 7). https://get.assembla.com/blog/source-code-security/
- Tal, L. (2023, November 16). *Securing source code in repositories is essential: How to get started*. Snyk. https://snyk.io/articles/securing-source-code-repositories/