Jael Andre

Professor Bolton

CS 305 Module 8 Journal

08/25/2024

Artemis Financial, a client I had the opportunity to work with, is a financial services company that was looking to enhance the security of its software systems, particularly focusing on encrypting archive files and securing their RESTful API against external threats. They turned to Global Rain for expertise in modernizing their operations and implementing stronger security measures to protect sensitive financial data.

When I began working on the security problems for Artemis Financial, the primary issue they faced was the potential vulnerability in their software, which could expose critical financial information to cyber threats. My role was to dive deep into their systems, identify any weak points, and recommend solutions. One of the things I felt I did well was using tools like the OWASP Dependency-Check to thoroughly review project dependencies, identifying outdated components that needed updates to avoid security risks. Coding securely is vital because it’s the first line of defense against breaches that could lead to devastating consequences, both financially and reputationally, for a company like Artemis. By ensuring robust security, I contributed to protecting the company’s integrity and its clients' trust.

A particularly challenging yet rewarding part of this project was ensuring that all dependencies were up to date without disrupting the system's existing functionality. This process underscored the importance of staying proactive in keeping software components current, as it’s much easier to prevent vulnerabilities than to fix them after they’ve been exploited.

To bolster the security layers, I recommended implementing strong encryption algorithms and made sure to thoroughly review and update all software dependencies. Looking ahead, I plan to continue using tools like OWASP Dependency-Check and regular code reviews to assess and address vulnerabilities, ensuring I always apply the best possible mitigation strategies.

After refactoring the code, I made sure it remained both functional and secure by running comprehensive tests—everything from unit tests to integration tests. This step was crucial to ensure that the changes I made didn’t introduce any new vulnerabilities.

The resources and tools I relied on, especially OWASP Dependency-Check, along with the secure coding practices I applied, will be invaluable in future projects. They’ve equipped me with a solid foundation for maintaining and improving software security, which is an essential skill in today’s tech landscape.

For future employers, I’d be proud to showcase this project as an example of my ability to identify and mitigate security risks while maintaining the functionality of the software. It’s a testament to my commitment to delivering secure, high-quality code that protects a company’s digital assets and strengthens its overall security posture.