David Allen

Reflection Paper 2

Date: 10/14/2025

Company: Old Dominion University – Division of Digital Transformation & Technology

Position: Cloud Developer Intern

ODU Fall 2025

Professor Teresa Duvall

Internship Reflection Paper

Second 50 Hours

During my next 50 hours as a cloud developer intern, I have gained knowledge about more vulnerabilities within other code repositories that are managed by the ODU web development team. I've continued to search for many vulnerabilities within the frontend and backend of Student 360 and the Canvas AI Teaching Assistant (CATA) repositories. When conducting my search, I looked for vulnerabilities at each line that was picked up by the SAST scanner within the configuration in the root of the repository. I searched for any variables that were already hardcoded into the repository. This reduces the chances for a user to insert malicious code into the web application to perform other malicious acts. Because I use an open-source scanner, it has the capability to be wrong a lot. We recently had a migration from GitLab Community Edition to Enterprise Edition, but this was not for any additional security scanning tools built-in with GitLab. It was to improve Shibboleth authentication mechanisms only since

Community Edition didn't have the necessary improvements that the development team needed. As of now, there still isn't enough funding or plans to add scanning or monitoring security features that are within GitLab to developer accounts. I still hope they are able to add these features in the future, since it would make things a lot easier because they are already integrated into GitLab. Apart from that, any vulnerabilities that were not true vulnerabilities were excluded from future scans by adding comments on the line they were found.

I did find one vulnerability related to XSS within the CATA repository that could actually be fixed based on the variable it had that required URL input from the user. This variable was known as req.originalURL. I showed the line of code to James, my supervisor. He said to check and see if everything was hardcoded, which it was, but there could've been potential for an attack still with this variable. I was suggested to remove it to reduce thew risk of an attack from the XSS vulnerability. Also, the line of code was unnecessary for the user. It basically outputs a request that the user is already aware of. I removed the req.originalURL variable from the line and changed the output for the user to basically say, "Invalid URL." Once changes were made, I made a merge request in the repository for review. Once approved, the changes are saved into the main repository.

I also learned how to use NPM to check for vulnerabilities within a repository. NPM is a software package that comes with NodeJS. NPM can scan for vulnerabilities related to software dependencies for code, some related to the lines of code found in previous SAST scans. These vulnerabilities were made aware to me by James when he ran the npm audit command in a terminal for both the front end and back end of the cloud portal repository. The scan for the front end picked up 60 vulnerabilities and the scan in the back end picked up 10. Thankfully, npm is highly automated with fixing vulnerabilities. The npm audit fix command and npm update

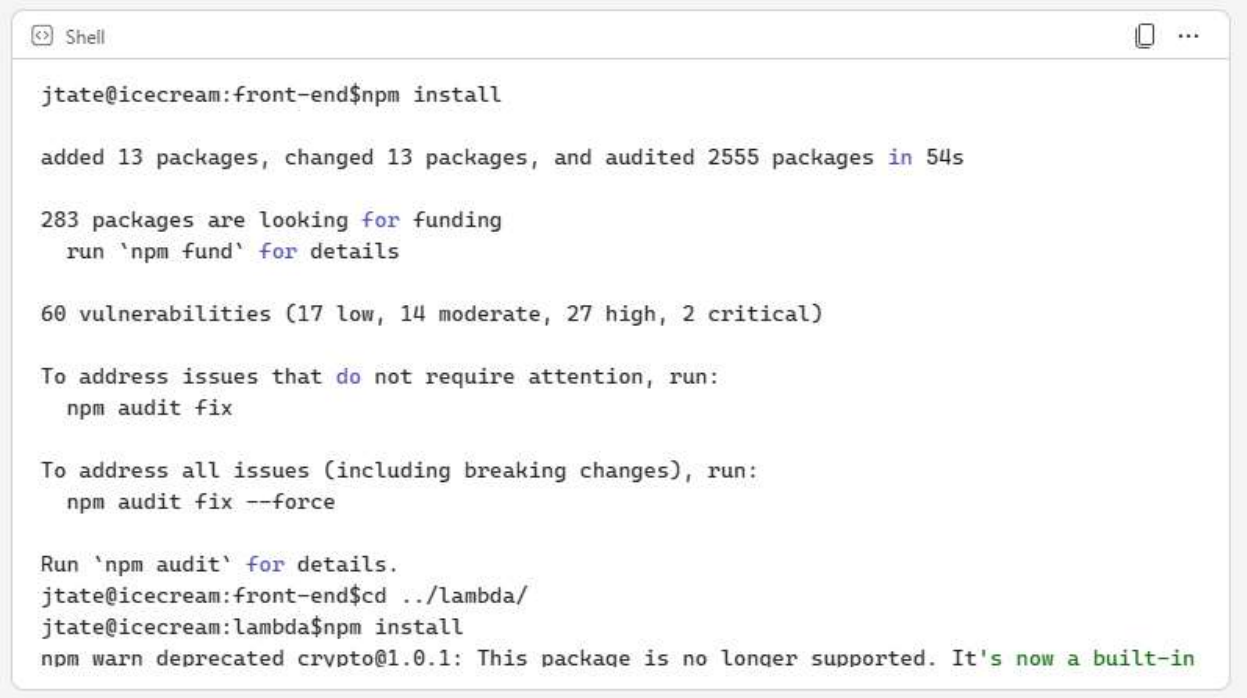
command helped update outdated dependencies in the repository and fix the vulnerabilities. Sometimes npm audit fix doesn't work due to these outdated dependencies, so they must be updated in order to fix vulnerable code. I've only fixed the vulnerabilities within the front end currently. Just by updating multiple outdated dependencies, I've reduced the number of vulnerabilities in the cloud portal repository from 60 to only nine. The remaining vulnerabilities are a part of breakable changes. Applying fixes has the potential to cause bugs and other compatibility issues in a repository. I will wait and see what James says about these breakable changes before I apply them to the repository.

The value of team collaboration and scope of security scans are the most important things I've learned during my second 50 hours of my internship. When making changes to code, others must be aware of these changes. It's important to make changes separately from the original code in a copy of the code repository and have someone review them. When making changes, be as detailed as possible about the changes made so that the reviewer can understand why changes are being made. Showing multiple people changes to code can encourage discussion about the changes and potential issues from different viewpoints. This ensures someone isn't randomly deleting or breaking the application by requiring reviews. Using other software to scan a repository can also determine additional vulnerabilities. SAST scans only static code for potential vulnerabilities, while NPM is used to scan the dependencies for the code. This gives cloud developers a better idea of where vulnerabilities can be found in a web application. Without using different types of scanning tools, vulnerabilities can go unnoticed by the developers, increasing the risk of an exploit to be executed, so it's always important to use different tools to get a broader understanding of vulnerabilities in an application.

Overall, I enjoyed the second 50 hours of my internship. I continued to learn about static analysis and actually participated in a sprint review by demonstrating the XSS vulnerability mentioned earlier for approval. After that, I got to learn about using NPM to scan and fix vulnerabilities in a repository related to software dependencies. James continues to be a helpful supervisor in the process. I am hopeful my next 50 hours will go just as well as the first and second 50 hours.

Appendix

Removing the req.originalURL variable from line 78 in the file



NPM from James about the vulnerabilities

The remaining vulnerabilities after fixing non-breaking changes