David Allen

Reflection Paper 1

Date: 09/23/2025

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

Position: Cloud Developer Intern

ODU Fall 2025

Professor Teresa Duvall

Internship Reflection Paper

First 50 Hours

During my first 50 hours as a cloud developer intern, I have gained knowledge about what the typical schedule of a cloud developer would be like working at Old Dominion University. To start my shift, I participated in a daily stand-up meeting. My supervisor, James, said this is part of the agile method of project management. It's a scrum meeting where everyone goes around in order and answers 3 questions: "What was accomplished yesterday?", "What will be accomplished today?", and "What are my blockers or impediments, if any, and who can I reach out to overcome them?" I believe this has been a great way to get organized and quickly summarize what was accomplished from the day before and what I have to do now each day. I know I must ask my supervisor for any help when necessary. We communicate together through Microsoft Teams in case I have any questions about the tasks I am given.

When starting, I was told that software development or IT has a lot of on-the-job self-training. I was given some simple tasks to get myself familiar with what a cloud developer would be looking at. Most of the code I'd be dealing with was JavaScript, or some other form of it such as NodeJS or React. I was tasked with reading up on basic JavaScript syntax. I had to know exactly what I was reading in order to analyze code for security issues. In one instance I found a weak hash algorithm that needed to be updated from MD5 to SHA-256, since I'm familiar with the most up-to-date cryptographic standards and decided to let James know about this vulnerability. I used some websites, such as W3Schools and Playcode to get familiar with writing some JavaScript. The programs were very basic, such as performing simple addition and multiplication operations and outputting the result. While getting familiar with JavaScript, I also had to get familiar with how GitLab works. GitLab is used as a code repository for programs or web applications. Essentially, it's hosting the cloud portal known as myODU and many other applications within the ODU system, such as Student360.

Although I have not written any code to improve any web applications yet, I have been performing a lot of security testing as a Cybersecurity major. I am needed for my field of expertise in certain situations that haven't had much attention with the codebase. The biggest thing I have been currently tasked with is performing Static Applications Security Testing (SAST). I had to configure a scanner that would be integrated with the GitLab code repository. James told me that he wanted an open-source, or free, solution that can scan the code in GitLab. He wants to be able to upgrade the GitLab subscriptions, but it's currently not in the school's budget. It took some time to set the scanner up since I was new with GitLab, but I overall enjoyed the process. GitLab has a free solution for SAST that involves copying and pasting a template into a master configuration file. After that, I just added an extra module for the security

scanning job. Now, when I update a file, a security scan is included. The total time it takes to update and scan is about two or three minutes. I have to note that I worked with clones of the actual codebases. In GitLab, there's an option to fork a repository, which basically clones a code repository into your own personal workspace. I don't want to be responsible for breaking the myODU portal, so I take the safest measures possible when dealing with large code repositories. Once my scan is complete, it outputs the results into a text file, since we do not have access to higher paid subscription to view vulnerabilities in a neatly organized HTML file. After I scanned for vulnerabilities, I was tasked with finding false positives that were picked up by the scan and exclude them from future scanning. It just involved writing some comments along the line of code to exclude a specific line from the scan. It takes me a while to look through the code and see where everything is to sort our false positives, but I'm getting used to it.

Overall, I enjoyed my first 50 hours at ODU as a Cloud Developer Intern. I feel like I've learned things that haven't been shown in a classroom. I've never been taught how to use GitLab before, which I know Git is very popular for accessing or storing code. So far, the experience has gone really well, especially with the rest of the team involved. They seem highly organized with multiple short meetings and a Sprint meeting for about an hour every two weeks. They each get to show off what has been accomplished, and it seems like a lot has to be done in order to make everything work properly for ODU online. It's made me realize what is needed for a web application to be secure for the user. A lot of students and staff require a safe and easy-to-use application to access information online. Without developers, no one would be able to access their information on their own. I look forward to the next 50 hours of this internship.

Appendix

```
18
19  include:
20    - template: Jobs/SAST.gitlab-ci.yml
21
```

The security template for GitLab

```
22  stages:          # List of stages for jobs, and their order of execution
23    - build
24    - test
25    - security
26    - deploy
```

The added security stage for the template

```
47
48  sast:
49    stage: security
50    artifacts:
51      paths:
52        - gl-sast-report.json
53      untracked: false
54      when: on_success
55      access: all
56      expire_in: 30 days
57
```

The SAST scan that outputs vulnerabilities in a file

An example output file of a SAST scan in GitLab