****

# Artemis Financial Vulnerability Assessment Report

Table of Contents

[Document Revision History 3](#_Toc32574607)

[Client 3](#_Toc32574608)

[Instructions 3](#_Toc32574609)

[Developer 4](#_Toc32574610)

[1. Interpreting Client Needs 4](#_Toc32574611)

[2. Areas of Security 4](#_Toc32574612)

[3. Manual Review 4](#_Toc32574613)

[4. Static Testing 4](#_Toc32574614)

[5. Mitigation Plan 4](#_Toc32574615)

## Document Revision History

| **Version** | **Date** | **Author** | **Comments** |
| --- | --- | --- | --- |
| **1.0** | **7/14/2023** | **Steven Anderson** |  |

## Client



## Instructions

Submit this completed vulnerability assessment report. Replace the bracketed text with the relevant information. In the report, identify your findings of security vulnerabilities and provide recommendations for the next steps to remedy the issues you have found.

* Respond to the five steps outlined below and include your findings.
* Respond using your own words. You may also choose to include images or supporting materials. If you include them, make certain to insert them in all the relevant locations in the document.
* Refer to the Project One Guidelines and Rubric for more detailed instructions about each section of the template.

## Developer

Steven Anderson

## Interpreting Client Needs

1. The client Artemis Financial is looking for a modern web software app with the latest software protections and security. They want to make sure they are protected from possible outside attacks.
   1. For a company focusing on banking and finances having a secure communication system for their web app is vital. They need to ensure that the communication with clients is secure from outside access and attack.
   2. Part of the company services include investment and savings so they will be dealing with international clients and travel of their clients. So, the system is protected even when moving overseas.
   3. There are many laws and governmental restrictions that the company will have to take in to account when working in the US and working overseas. In the US if client information is stolen or has unauthorized access that will more than likely violate the CFAA. They also in the US will need a system of access in order to follow ECPA guidelines. Along with many different state laws that need to be followed regarding when a client needs to be notified of a breach and when companies have to tell clients they shared their data. Additionally, they need to as a financial company make sure they follow the rule of the GLBA act which requires them to disclose how they will be sharing any data they do share and the FCRA rules which apply to accessing credit reports. Along with this they need to make sure on international accounts they are following the laws of the countries which could include differing encryption laws.
   4. They will have to deal with hackers as a main threat. Since hackers might try to hack a financial company to access their client information and finances. With ransomware being a threat they might have to face. Along with phishing attacks that will try and get access to their system. They also have to worry about DoS attacks aimed at their system. As far future threats they need to be warry of increased attacks from foreign state-sponsored groups. They also have to worry about advancing technologies that could be used to either take advantage of weaknesses or open new ones.
   5. There are plenty of modernized requirements that modern application will have.
      1. Open-source libraries are a good way of keeping an updating system since these libraries will be updated as new threats are discovered and as time goes on.
      2. By using evolving web app technologies over time, they can not only keep themselves ahead of the trends but also make sure they have enough time to be ready for threats. Such as adaptation of API technologies to make sure they can remain ahead of the threats inherent with the technology.

## Areas of Security

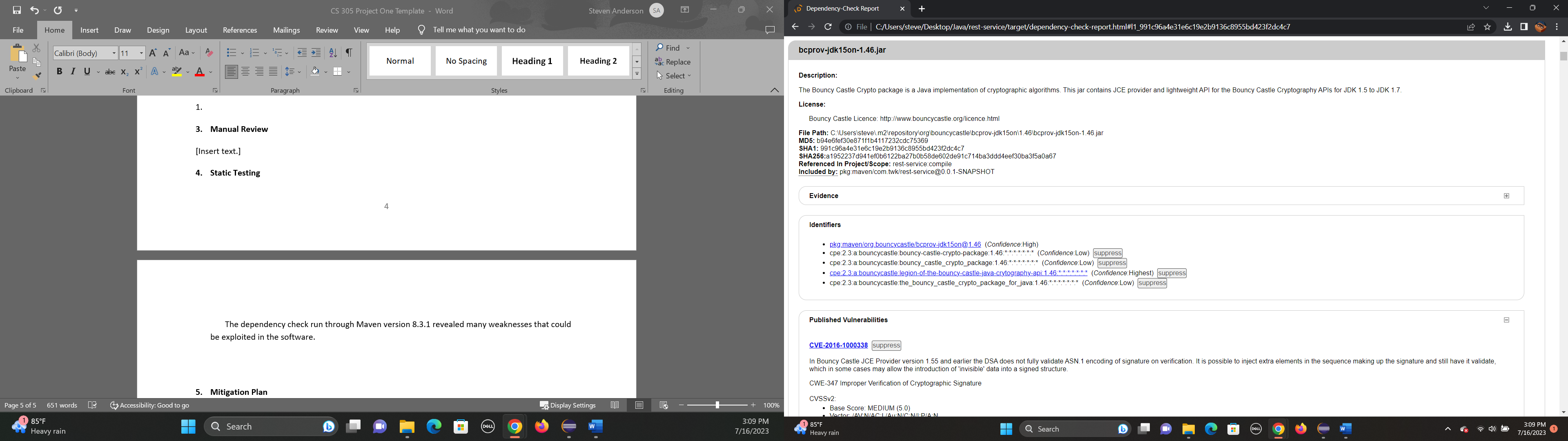
1. For Artemis Financial’s program there are multiple areas of security they need to be focused on for what they are doing.
   1. Securing API’s is important for them since they plan on using this on API devices. They need to ensure the API inputs are secured connections. With some of the vulnerabilities found in the static report this is a very important part of their security focus.
   2. Cryptography was another issue that had found vulnerabilities in the static report and should also be a focus. With banking and finances, they will have plenty of encryption files and connections that need to be secure to prevent client data breaches.
   3. Input Validation: Since users of their program will be making inputs for banking and investment, they will also need attention on input validation. Making sure that the inputs are secure and don’t allow for false inputs being accepted or for authorized inputs being intercepted.
   4. Encapsulation: Since they will have encapsulated classes, they need to focus on this to make sure that data being sent from one part of the structure to the next is secure. They will have values that store the values of accounts and for the updating of accounts so since that data is being used having secure structures will help prevent interception.
   5. Client/Server: Since this will be a web-based application making sure that user log in and data access is secure should also be focused on. They need to ensure that false log ins and data access is blocked by the system. Having a secure system for data distribution would be important for their program.

## Manual Review

* With a manual review of the code, I looked over the .java files I found a few things I can list. In the DocData.java file the Connection con isn’t used and the TODO auto generated block isn’t written. Also, the test isn’t written. And the public DocData is empty and doesn’t have a purpose in the code since it doesn’t do anything. And the java.sql.\* import is a vulnerability for API while the framework could be out of date.
* In the RestServiceApplicationTests.java file the void contextLoads() is empty and doesn’t test anything.
* In the CRUD.java file the fact that there are multiple public CRUD instances should be changed for better organization and readability.
* In the CRUDController.java and the GrettingController.java files the @RestController is harmful while still relying on the outdated Spring MVC dependency.
* In the RestServiceApplication.java file the @SpringBootApplication command is a problem while the program is still using the outdated springboot dependency.
* The code should also have more quotations to label what each portion is doing for future editing and updating. The code also had to correct itself since the package name com.twk.rest-service wasn’t valid and had to be replace with com.twk.restservice.
* Inputs don’t seem to be sanitized before being used which could lead to SQL injection attacks that could leave user data vulnerable.

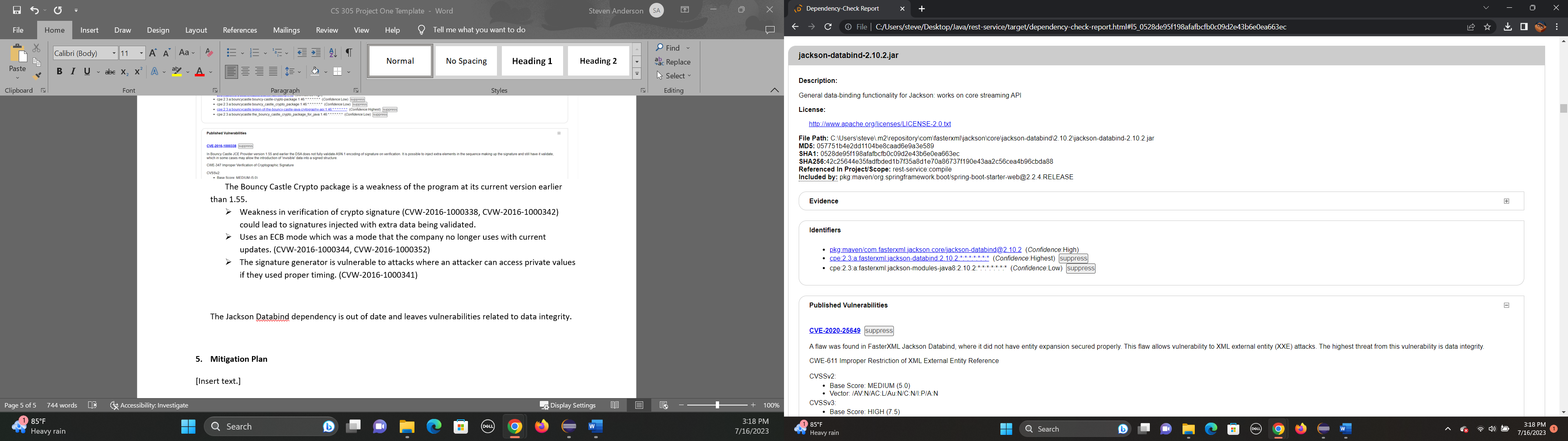
## Static Testing

The dependency check run through Maven version 8.3.1 revealed many weaknesses that could be exploited in the software.



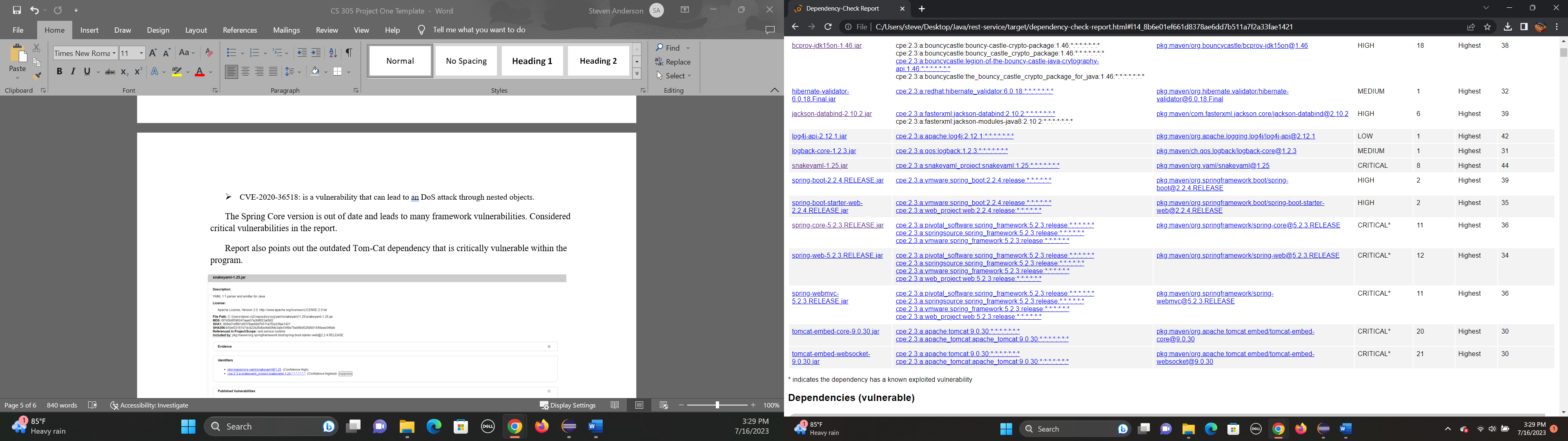
The Bouncy Castle Crypto package is a weakness of the program at its current version.

* Weakness in verification of crypto signature (CVW-2016-1000338, CVW-2016-1000342) could lead to signatures injected with extra data being validated.
* Uses an ECB mode which was a mode that the company no longer uses with current updates. (CVW-2016-1000344, CVW-2016-1000352)
* The signature generator is vulnerable to attacks where an attacker can access private values if they used proper timing. (CVW-2016-1000341)



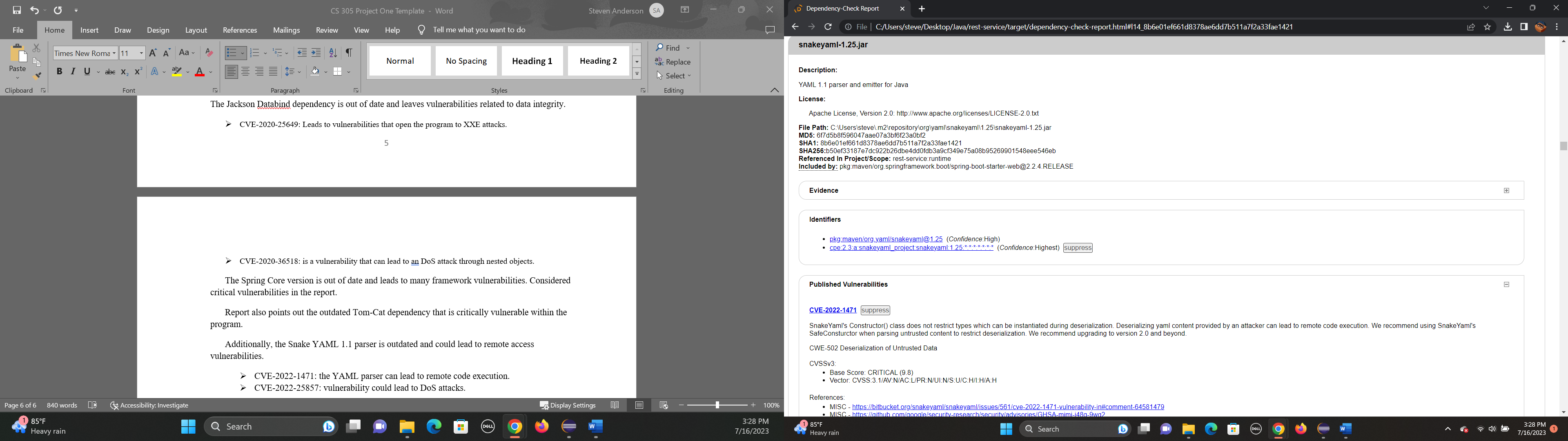
The Jackson Databind dependency is out of date and leaves vulnerabilities related to data integrity.

* CVE-2020-25649: Leads to vulnerabilities that open the program to XXE attacks.
* CVE-2020-36518: is a vulnerability that can lead to an DoS attack through nested objects.



The Spring Core version is out of date and leads to many framework vulnerabilities. Considered critical vulnerabilities in the report.

Report also points out the outdated Tom-Cat dependency that is critically vulnerable within the program.



Additionally, the Snake YAML 1.1 parser is outdated and could lead to remote access vulnerabilities.

* CVE-2022-1471: the YAML parser can lead to remote code execution.
* CVE-2022-25857: vulnerability could lead to DoS attacks.

## Mitigation Plan

To mitigate some of their security concerns and problems I would first recommend updating the dependencies to their latest versions. Updating the Spring Core to version to 4.1.9 would help with the problems the current used spring core has with the framework. Also updating the Snake YAML 1.1 to version 2.0 would help to mitigate the problem with the DoS and remote activation vulnerabilities. Also update the Tom-Cat dependency to version 1.2.48 can help prevent the vulnerabilities it has. Then Jackson DataBind needs to be updated to 2.9.10 or later to prevent the DoS and XXE attack vulnerabilities that can be exploited. The java library Bouncy Castle should be updated to the latest version to reinforce the encryption security. Also, the newer versions of Bouncy Castle don’t use the Crypto part of the later version so updating can help remove this vulnerability.

With my Manual review of the code, I found problems that need mitigated. While some of the problems encountered in the review were problems fixed by updating libraries and dependencies there are others that need attention. Including the multiple parts in the different .java files where classes were not finished or still had TODO parts.

They should also make sure that the input safety has been updated to prevent false inputs, possibly by using the latest Apache Jackson version available. Would also recommend using something to verify inputs aren’t bringing foreign data attached by using something like Apache Commons Text. This can also help to prevent SQL injection attacks.

Would also recommend since they are using API’s and due to the threat of phishing attacks that the employees of the company go through digital security training. This will help the company avoid some of the threats that come from phishing and remote access. They should also have their system stressed tests multiple times a year and run through security test to make sure they don’t fall behind the latest security threats.