****

# 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** | **01.22.2022** | **Mariam Haji** |  |

## 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

Mariam Haji

## Interpreting Client Needs

Artemis Financial is a financial planning company that develops individual financial plans for their customer such as savings, retirement, investments, and insurance plans, this makes their data and the data they collect from clients to be very sensitive data. The value of secure communication is invaluable for them.

Because the company deals with a vast number of clients who can be in any demographical areas there are chances that they may make international transaction for when a client is overseas or retires and resides overseas.

There are some government restrictions for financial institutions such as the Sarbanes-Oxley Act which requires as destruction and falsification of records, retention period for storing records and specific business records that companies need to store, which includes electronic communications. (Kenton, 2022).

Some of the threats that are present now and in the immediate future are:

* Password attacks used by hackers to maliciously authenticate, enter and steal data from password protected accounts.
* Network vulnerabilities such as SQL injections that seek to take advantage of weaknesses in code that can allow hackers to gain access to a network in order to plant malware, exfiltrate data or damage systems.
* Finally, data loss or data manipulation.

Some of the modernization requirements that can be used to protect data and making the application more secure can be use of open-source libraries, as well as use of best security practices including encrypting sensitive data to make sure that information does not show up in exceptions.

## Areas of Security

Areas of security that apply to Artemis Financials’ web application are:

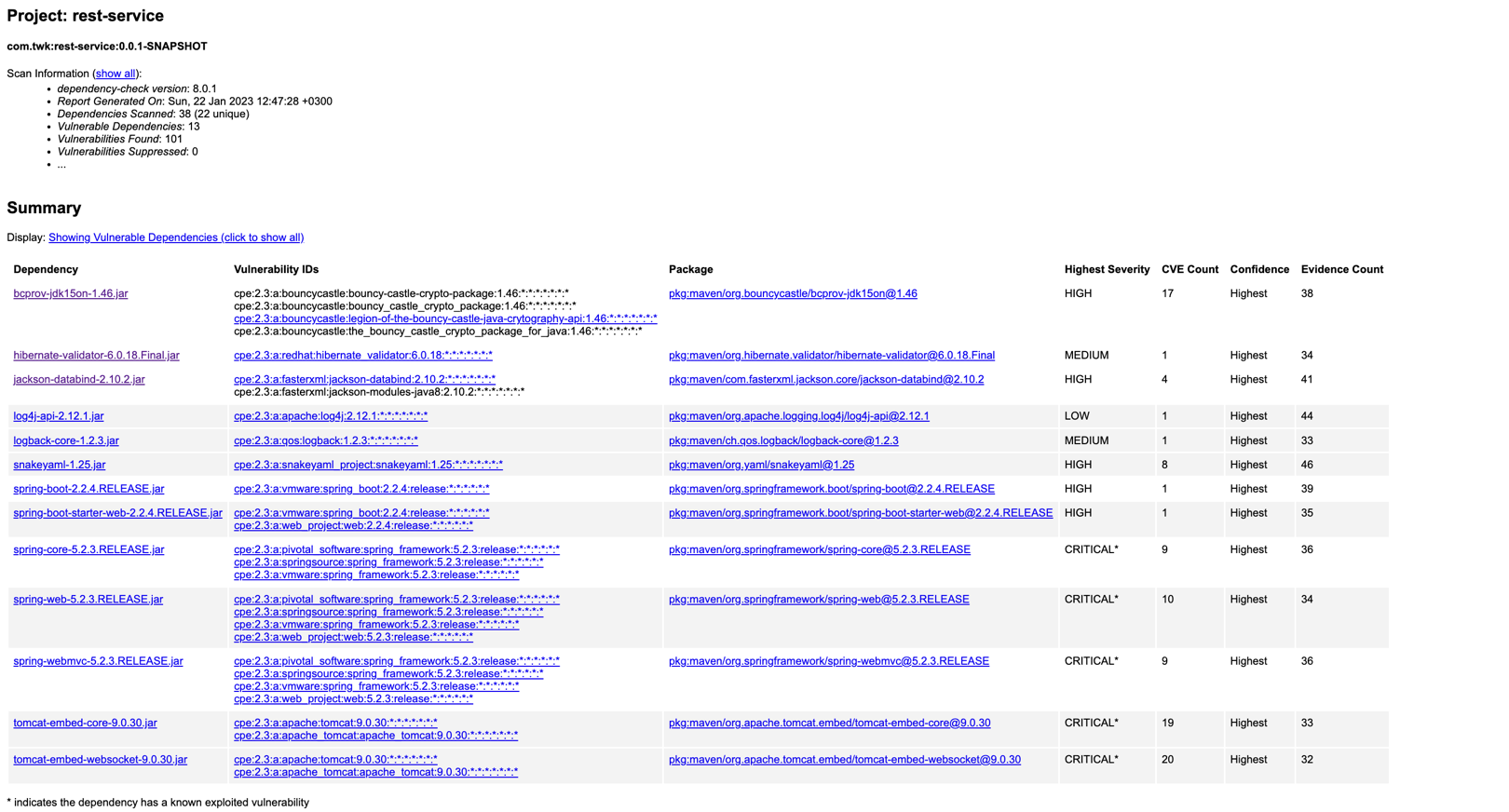
* **Input Validation:** This is the best way to ensures only properly formatted data enters the applications system component.
* **APIs:** APIs such as RESTful API will establish a secure connection between the application and external systems.
* **Cryptograph:** Cryptography needs to be used to ensure that only the intended recipient can decrypt the message and read its contents.
* **Client/Server:** Since the data and information will be traveling across different locations the client/server environment needs to be secure to make sure that only valid users and programs to information.
* **Code Error:** Code errors can be a major risk as detailed internal error message and error codes may reveal implementation details that should never be revealed to external users making the system vulnerable to hackers on potential flaws with the application.

## Manual Review

Some of the major concerns and vulnerabilities with the code is:

* In the GreetingController there is mapping to a page that shows the id of a user and a hello world message.
* The CRUDController file has access to the DocData file making this a concern as well since the page that the CRUDController maps to currently doesn’t work. In addition to that the CRUD Controller file requests HTML form data to bind to the request parameter and doesn’t validate the HTML form data input.
* The customer.java file retrieves and displays private data.

## Static Testing



**bcprov-jdk15on-1.46.jar:**

CVE-2016-1000338

In Bouncy Castle JCE Provider version 1.55 and earlier the DSA does not fully validate ASN.1 encoding of signature on verification. It is possible to inject extra elements in the sequence making up the signature and still have it validate, which in some cases may allow the introduction of 'invisible' data into a signed structure.

CVE-2016-1000341

In the Bouncy Castle JCE Provider version 1.55 and earlier DSA signature generation is vulnerable to timing attack. Where timings can be closely observed for the generation of signatures, the lack of blinding in 1.55, or earlier, may allow an attacker to gain information about the signature's k value and ultimately the private value as well.

CVE-2020-0187 (OSSINDEX)

In engineSetMode of BaseBlockCipher.java, there is a possible incorrect cryptographic algorithm chosen due to an incomplete comparison. This could lead to local information disclosure with no additional execution privileges needed. User interaction is not needed for exploitation.Product: AndroidVersions: Android-10Android ID: A-148517383

**hibernate-validator-6.0.18.Final.jar:**

CVE-2020-10693

A flaw was found in Hibernate Validator version 6.1.2.Final. A bug in the message interpolation processor enables invalid EL expressions to be evaluated as if they were valid. This flaw allows attackers to bypass input sanitation (escaping, stripping) controls that developers may have put in place when handling user-controlled data in error messages.

**jackson-databind-2.10.2.jar**

CVE-2020-25649

A flaw was found in FasterXML Jackson Databind, where it did not have entity expansion secured properly. This flaw allows vulnerability to XML external entity (XXE) attacks. The highest threat from this vulnerability is data integrity.

CVE-2020-36518

jackson-databind before 2.13.0 allows a Java StackOverflow exception and denial of service via a large depth of nested objects.

**log4j-api-2.12.1.jar:**

CVE-2020-9488

Improper validation of certificate with host mismatch in Apache Log4j SMTP appender. This could allow an SMTPS connection to be intercepted by a man-in-the-middle attack which could leak any log messages sent through that appender. Fixed in Apache Log4j 2.12.3 and 2.13.1

**logback-core-1.2.3.jar**

CVE-2021-42550

In logback version 1.2.7 and prior versions, an attacker with the required privileges to edit configurations files could craft a malicious configuration allowing to execute arbitrary code loaded from LDAP servers.

**snakeyaml-1.25.jar**

CVE-2022-1471 (OSSINDEX)

SnakeYaml's Constructor() class does not restrict types which can be instantiated during deserialization. Deserializing yaml content provided by an attacker can lead to remote code execution. We recommend using SnakeYaml's SafeConsturctor when parsing untrusted content to restrict deserialization.

CVE-2017-18640

The Alias feature in SnakeYAML before 1.26 allows entity expansion during a load operation, a related issue to CVE-2003-1564.

CVE-2022-25857

The package org.yaml:snakeyaml from 0 and before 1.31 are vulnerable to Denial of Service (DoS) due missing to nested depth limitation for collections.

**spring-boot-2.2.4.RELEASE.jar**

CVE-2022-27772

\*\* UNSUPPORTED WHEN ASSIGNED \*\* spring-boot versions prior to version v2.2.11.RELEASE was vulnerable to temporary directory hijacking. This vulnerability impacted the org.springframework.boot.web.server.AbstractConfigurableWebServerFactory.createTempDir method. NOTE: This vulnerability only affects products and/or versions that are no longer supported by the maintainer.

**spring-boot-starter-web-2.2.4.RELEASE.jar:**

CVE-2022-27772

\*\* UNSUPPORTED WHEN ASSIGNED \*\* spring-boot versions prior to version v2.2.11.RELEASE was vulnerable to temporary directory hijacking. This vulnerability impacted the org.springframework.boot.web.server.AbstractConfigurableWebServerFactory.createTempDir method. NOTE: This vulnerability only affects products and/or versions that are no longer supported by the maintainer.

**spring-core-5.2.3.RELEASE.jar:**

CVE-2022-22965

A Spring MVC or Spring WebFlux application running on JDK 9+ may be vulnerable to remote code execution (RCE) via data binding. The specific exploit requires the application to run on Tomcat as a WAR deployment. If the application is deployed as a Spring Boot executable jar, i.e. the default, it is not vulnerable to the exploit. However, the nature of the vulnerability is more general, and there may be other ways to exploit it.

CVE-2021-22118

In Spring Framework, versions 5.2.x prior to 5.2.15 and versions 5.3.x prior to 5.3.7, a WebFlux application is vulnerable to a privilege escalation: by (re)creating the temporary storage directory, a locally authenticated malicious user can read or modify files that have been uploaded to the WebFlux application, or overwrite arbitrary files with multipart request data.

**spring-web-5.2.3.RELEASE.jar:**

CVE-2016-1000027

Pivotal Spring Framework through 5.3.16 suffers from a potential remote code execution (RCE) issue if used for Java deserialization of untrusted data. Depending on how the library is implemented within a product, this issue may or not occur, and authentication may be required. NOTE: the vendor's position is that untrusted data is not an intended use case. The product's behavior will not be changed because some users rely on deserialization of trusted data.

**spring-webmvc-5.2.3.RELEASE.jar:**

CVE-2020-5421

In Spring Framework versions 5.2.0 - 5.2.8, 5.1.0 - 5.1.17, 5.0.0 - 5.0.18, 4.3.0 - 4.3.28, and older unsupported versions, the protections against RFD attacks from CVE-2015-5211 may be bypassed depending on the browser used through the use of a jsessionid path parameter.

**tomcat-embed-core-9.0.30.jar**

CVE-2020-1938

When using the Apache JServ Protocol (AJP), care must be taken when trusting incoming connections to Apache Tomcat. Tomcat treats AJP connections as having higher trust than, for example, a similar HTTP connection. If such connections are available to an attacker, they can be exploited in ways that may be surprising. In Apache Tomcat 9.0.0.M1 to 9.0.0.30, 8.5.0 to 8.5.50 and 7.0.0 to 7.0.99, Tomcat shipped with an AJP Connector enabled by default that listened on all configured IP addresses. It was expected (and recommended in the security guide) that this Connector would be disabled if not required. This vulnerability report identified a mechanism that allowed: - returning arbitrary files from anywhere in the web application - processing any file in the web application as a JSP Further, if the web application allowed file upload and stored those files within the web application (or the attacker was able to control the content of the web application by some other means) then this, along with the ability to process a file as a JSP, made remote code execution possible. It is important to note that mitigation is only required if an AJP port is accessible to untrusted users. Users wishing to take a defence-in-depth approach and block the vector that permits returning arbitrary files and execution as JSP may upgrade to Apache Tomcat 9.0.31, 8.5.51 or 7.0.100 or later. A number of changes were made to the default AJP Connector configuration in 9.0.31 to harden the default configuration. It is likely that users upgrading to 9.0.31, 8.5.51 or 7.0.100 or later will need to make small changes to their configurations.

**tomcat-embed-websocket-9.0.30.jar:**

CVE-2020-8022

A Incorrect Default Permissions vulnerability in the packaging of tomcat on SUSE Enterprise Storage 5, SUSE Linux Enterprise Server 12-SP2-BCL, SUSE Linux Enterprise Server 12-SP2-LTSS, SUSE Linux Enterprise Server 12-SP3-BCL, SUSE Linux Enterprise Server 12-SP3-LTSS, SUSE Linux Enterprise Server 12-SP4, SUSE Linux Enterprise Server 12-SP5, SUSE Linux Enterprise Server 15-LTSS, SUSE Linux Enterprise Server for SAP 12-SP2, SUSE Linux Enterprise Server for SAP 12-SP3, SUSE Linux Enterprise Server for SAP 15, SUSE OpenStack Cloud 7, SUSE OpenStack Cloud 8, SUSE OpenStack Cloud Crowbar 8 allows local attackers to escalate from group tomcat to root. This issue affects: SUSE Enterprise Storage 5 tomcat versions prior to 8.0.53-29.32.1. SUSE Linux Enterprise Server 12-SP2-BCL tomcat versions prior to 8.0.53-29.32.1. SUSE Linux Enterprise Server 12-SP2-LTSS tomcat versions prior to 8.0.53-29.32.1. SUSE Linux Enterprise Server 12-SP3-BCL tomcat versions prior to 8.0.53-29.32.1. SUSE Linux Enterprise Server 12-SP3-LTSS tomcat versions prior to 8.0.53-29.32.1. SUSE Linux Enterprise Server 12-SP4 tomcat versions prior to 9.0.35-3.39.1. SUSE Linux Enterprise Server 12-SP5 tomcat versions prior to 9.0.35-3.39.1. SUSE Linux Enterprise Server 15-LTSS tomcat versions prior to 9.0.35-3.57.3. SUSE Linux Enterprise Server for SAP 12-SP2 tomcat versions prior to 8.0.53-29.32.1. SUSE Linux Enterprise Server for SAP 12-SP3 tomcat versions prior to 8.0.53-29.32.1. SUSE Linux Enterprise Server for SAP 15 tomcat versions prior to 9.0.35-3.57.3. SUSE OpenStack Cloud 7 tomcat versions prior to 8.0.53-29.32.1. SUSE OpenStack Cloud 8 tomcat versions prior to 8.0.53-29.32.1. SUSE OpenStack Cloud Crowbar 8 tomcat versions prior to 8.0.53-29.32.1.

## Mitigation Plan

Some of the mitigation plans that can be used in the Artemis Financial application is to update all versions libraries and software they are using since as per the dependency checks most of the versions used are either unsupported or out of date.

In addition to that all files connected to the database should have proper security mitigation to access and store data in the databases. All user input should also be validated before being stored in the system and all sensitive data should be encrypted. Finally, all errors should be documented and handled correctly.

# References

Kenton, W. (2022, May 08). *Sarbanes-Oxley Act: What It Does to Protect Investors*. Retrieved from Investopedia: https://www.investopedia.com/terms/s/sarbanesoxleyact.asp