GLOBALRAIN

Artemis Financial Vulnerability Assessment Report

Table of Contents

Document Revision History

Version	Date	Author	Comments
1.0	02/04/2024	Gentian Hoxha	

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
Gentian Hoxha

1. Interpreting Client Needs:

Artemis Financial has recognized the importance of updating its operational framework and fortifying its software security defenses to safeguard its web-based software solutions. As a consultancy specializing in financial planning, they manage highly confidential client data pertaining to savings, retirement planning, investment strategies, and insurance policies. Consequently, securing their software application is critically vital. They aspire to insulate their firm from external cyber threats and assure secure data exchange. While the provided details don't encompass international financial transactions or statutory regulations, it is sensible to take these elements into account when formulating security strategies. Moreover, considerations for modernization should incorporate the integration of open-source software and the continuous advancements in web application technologies.

2. Areas of Security:

For Artemis Financial's web application, pertinent security domains include:

1. Access Control: There must be stringent measures for verifying user identities and controlling access to ensure only legitimate users can view sensitive financial information.
2. Data Input Sanitization and Secure Output: Robust data input checks are essential to guard against threats such as SQL injection and cross-site scripting (XSS). Safeguarding against XSS also necessitates secure output practices, especially when presenting content generated by users.
3. User Session Integrity: The management of user sessions must be secure, encompassing the safe initiation, upkeep, and closure of sessions to avert risks like session hijacking or fixation.
4. Privacy of Sensitive Information: Encryption must be applied to sensitive client financial data during both transmission and storage to inhibit unauthorized access and potential data compromises.
5. Encrypted Communications: The application should employ protocols like HTTPS for encrypting all data exchanges between clients and servers to maintain privacy and data integrity.

3. Manual Review:

In the code base examination, two security weaknesses were pinpointed:

1. Vulnerability to Cross-Site Scripting (XSS): The file "userProfile.jsp" is missing adequate output encoding, which could potentially allow attackers to execute harmful scripts on the webpage.
2. Susceptibility to SQL Injection: The "loginDAO.java" file is vulnerable due to the way it forms SQL queries using data provided by users, which could lead to SQL injection attacks.
3. The code quality met the basic standards, but it fell short in terms of error management due to its absence. Regarding the API, it was found to be deficient in several aspects.

4. Static Testing:

Dependency	Vulnerability	Description	Solution
bcprov- jdk15on-1.46.jar	cpe:2.3:a:bouncycastle:bou ncy-castle-crypto-package:1.46:*:*:*:*:*:*:* cpe:2.3:a:bouncycastle:bou ncy_castle_crypto_package :1.46:*:*:*:*:*:*:*	Legion of the Bouncy Castle Legion of the Bouncy Castle Java Cryptography APIs 1.58 up to but not including 1.60 contains a CWE-470: Use of Externally-Controlled Input to Select Classes or Code ('Unsafe Reflection') vulnerability	Version update
	cpe:2.3:a:bouncycastle:legion-of-the-bouncy-castle-java-crytography-api:1.46:*:*:*:*:*:*:* cpe:2.3:a:bouncycastle:the_bouncy_castle_crypto_package_for_java:1.46:*:*:*:*:*:*:*	in XMSS/XMSS^MT private key deserialization that can result in Deserializing an XMSS/XMSS^MT private key can result in the execution of unexpected code. This attack appear to be exploitable via A handcrafted private key can include references to unexpected classes which will be picked up from the class path for the executing application. This vulnerability appears to have been fixed in 1.60 and later.	
spring-aop-5.2.3.RELEASE.jar	cpe:2.3:a:pivotal_software:spring_framework:5.2.3:release:*:*:*:*:*:* cpe:2.3:a:springsource:spring_framework:5.2.3:release:*:*:*:*:*:* cpe:2.3:a:vmware:spring_framework:5.2.3:release:*:*:*:*:*:* cpe:2.3:a:vmware:springsource_spring_framework:5.2.3:release:*:*:*:*:*:*	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.	Upgrade to current version

hibernate-validator-6.0.18.Final.jar	cpe:2.3:a:redhat:hibernate_validator:6.0.18:*:*:*:*:*:*:*	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.	Upgrade to hibernate-validator-
snakeyaml-1.25.jar	cpe:2.3:a:snakeyaml_project:snakeyaml:1.25:*:*:*:*:*:*:*	The Alias feature in SnakeYAML 1.18 allows entity expansion during a load operation, a related issue to CVE-2003-1564.	Migrate to SnakeYAML Engine. It has a configuration option to restrict aliases for collections (the aliases for scalars cannot grow and they are not restricted)
1. log4j-api-2.12.1.jar	cpe:2.3:a:apache:log4j:2.12	the 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.	Upgrade to support this feature. Previous versions can set the system property mail.smtp.ssl.checkserveridentity to true to globally enable hostname verification for SMTPS connections.
tomcat- embed-websocket-9.0.30.jar	cpe:2.3:a:apache:tomcat:9.0.30:*:*:*:*:*:*:* cpe:2.3:a:apache_software_foundation:tomcat:9.0.30:*:*:*:*:*:*:* cpe:2.3:a:apache_tomcat:apache_tomcat:9.0.30:*:*:*:*:*:*:*	Apache Tomcat 10.0.0-M1 to 10.0.6, 9.0.0.M1 to 9.0.46 and 8.5.0 to 8.5.66 did not correctly parse the HTTP transfer-encoding request header in some circumstances leading to the possibility to request smuggling when used with a reverse proxy. Specifically: - Tomcat incorrectly ignored the transfer encoding header if the client declared it would only accept an HTTP/1.0 response; - Tomcat identify encoding; and - Tomcat did not ensure that, if present, the chunked encoding was the final	Upgrade Apache Tomcat

bcprov-jdk15on-1.46.jar	cpe:2.3:a:bouncycastle:bou ncy-castle-crypto-package:1.46:*:*:*:*:*:*:* cpe:2.3:a:bouncycastle:bou ncy_castle_crypto_package :1.46:*:*:*:*:*:*:*	Legion of the Bouncy Castle Legion of the Bouncy Castle Java Cryptography APIs 1.58 up to but not including 1.60 contains a CWE-470: Use of Externally-Controlled Input to Select Classes or Code ('Unsafe Reflection') vulnerability	Version update

Mitigation Plan:

To resolve the detected security vulnerabilities, the immediate course of action involves upgrading all systems to the latest versions. Subsequently, establishing HTTPS is essential to secure web access. Implementing a robust authentication system is also crucial to guarantee that customers have exclusive access to their personal data. After these preliminary measures are set, further steps should be considered to assure the security of sensitive information while maintaining its accessibility to authorized individuals.