# CS 305 Project One Template

## Document Revision History

| **Version** | **Date** | **Author** | **Comments** |
| --- | --- | --- | --- |
| **1.0** | **9/21/25** | **Tim Jayson** |  |

## Client



## Instructions

Submit this completed vulnerability assessment report. Replace the bracketed text with the relevant information. In this report, identify your security vulnerability findings and recommend 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 include images or supporting materials. If you include them, make certain to insert them in 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

Tim Jayson

**1. Interpreting Client Needs**

Determine your client’s needs and potential threats and attacks associated with the company’s application and software security requirements. Consider the following questions regarding how companies protect against external threats based on the scenario information:

* What is the value of secure communications to the company?

Global Rain specializes in custom software design and development, and they have a belief that “Security is everyone’s responsibility.” This directive is an important cornerstone for every software project they have for their clients. Working with Artemis Financial it is imperative to focus on security while working with them since they are a financial consulting company. Since it is Artemis Financial’s goal to protect against external threats it is extremely important to use secure communication practices and the latest security measures to keep their data safe. The proper use of security techniques not only will help them protect their client’s data but will also keep them compliant with government regulations on the proper handling of sensitive information. Data such as financial plans, investments, insurance information, and retirement accounts are extremely sensitive and are ideal targets for malicious attackers. Not only would a data breach expose this sensitive information, but it would also erode customer trust, open the company to litigation, and severely damage their reputation.

* Are there any international transactions that the company produces?

While it does not explicitly state that Artemis Financial does business internationally, it is important to consider the governmental requirements around the world and how they impact security requirements. The General Data Protection Regulation (GDPR) imposes restrictions on how personal data is handled and processed in the EU. These regulations include seven protection and accountability principles. These principles are:

* Lawfulness, fairness, and transparency: processing must be lawful, fair, and transparent to the data subject.
* Purpose Limitation: Companies must process data for the legitimate purposes specified explicitly to the data subject when you collected it.
* Data Minimization: Companies should collect and process only as much data as absolutely necessary for the purposes specified.
* Accuracy: Companies must keep personal data accurate and up to date.
* Storage Limitation: Companies may only store personally identifying data for as long as necessary for the specified purpose.
* Integrity and Confidentiality: Processing must be done in such a way as to ensure appropriate security, integrity, and confidentiality.
* Accountability: The data controller is responsible for being able to demonstrate GDPR compliance with all of these principles.

Anther example of international regulations to consider is PIPEDA in Canada. This Canadian law regulates how private-sector organizations handle personal information during commercial activities. It operates on the following 10 principles:

* Accountability: It is the responsibility to comply with all 10 fair information principles, appoint someone to be responsible for your organization’s PIPEDA compliance, protect all personal information help by the organization, and develop and implement personal policies and practices.
* Identifying purposes: It is the responsibility of the organization to identify and document your purposes for collecting personal information, and tell your customers why your organization needs their personal information before or at the time of collection.
* Consent: Obtain consent initially and when making changes to privacy practices including the use of data for new purposes.
* Limiting collection: Collect only the personal information your organization needs to fulfill a legitimate identified purpose.
* Limiting use, disclosure, and retention: Unless required by law or otherwise receive consent, the organization may use or disclose personal information only for the identified purposes for which it was collected. Keep personal information only as long as it is needed to serve those purposes.
* Accuracy: It is the responsibility of the organization to keep personal information as accurate, complete and up to date as necessary, taking into account its use and the interests of the individual.
* Safeguards: It is the responsibility of the organization to develop and implement a security policy to protect personal information.
* Openness: It is the responsibility of the organization to inform your customers and employees that you have policies and practices for managing personal information.
* Individual access: When asked, advise people about the personal information about them your organization holds, explain where the information was obtained, and explain how the information has been used.
* Challenging compliance: It is the responsibility of the organization to provide recourse by developing simple compliant handling and investigation procedures as well as telling complainants about their avenues of recourse.
* Are there governmental restrictions on secure communications to consider?

Financial companies such as Artemis Financial are subject to governmental compliance requirements and must safeguard their customers’ data in accordance with those requirements. In the United States regulations such as the Gramm-Leach-Bliley Act (GLBA) impose strict data protection requirements. These requirements include providing clear privacy requirements to customers, the implementation of security measures to protect consumers nonpublic personal information, and provisions to prevent unauthorized access to customer information. The key requirements of the Gramm-Leach-Bliley Act include the following key requirements:

* Privacy Notice: Financial institutions must provide clear and conspicuous privacy notices to customers, explaining what information is collected, how it is shared, and how it is protected. Customers must be informed of the right to “opt out” of certain information-sharing with nonaffiliated third parties.
* Safeguard Rules: Institutions must develop, implement, and maintain a comprehensive information security program with administrative, technical, and physical safeguards to protect customer data.
* Pretexting Protection: Institutions must take steps to prevent unauthorized access to private information, including by individuals using false pretenses.
* Vendor Management: Institutions are responsible for ensuring third-party service providers that access customer data also comply with GLBA requirements.
* What external threats might be present now and in the immediate future?

With any organization it is important to consider the external security threats. It is especially important to protect against these threats when that organization collects and maintains sensitive information. Some examples of external threats are as follows:

* Man-in-the-Middle (MITM) Attacks: This type of attack occurs when a hacker steals sensitive information by eavesdropping on communications between two online targets such as a web application and the user.
* Session Sniffing: In a session sniffing attack, the attacker uses a sniffer to capture a valid token session known as a “Session ID”, they then use the valid token session to gain unauthorized access to the web server.
* Injection Attacks: These attacks occur when attackers exploit vulnerabilities in application to send malicious code into a system. This can potentially allow them to execute unauthorized commands, access sensitive data, and manipulate system operations.
* Zero-Day Exploits: This is a type of attack where the hacker takes advantage of an unknown/unaddressed flaw in the security of the software. This is known as “Zero-Day” because the software vendor has had zero days to fix the flaw before it was available to be attacked by malicious actors.
* Phishing and Credential Stuffing: Phishing is a type of attack where a malicious actor tricks a user into providing sensitive information such as passwords or other sensitive information by pretending to be a trustworthy entity. Credential Stuff is a type of attack where a malicious actor uses a stolen username and password pair to gain unauthorized access.
* What modernization requirements must be considered, such as the role of open-source libraries and evolving web application technologies?

It is important to stay up to date with current security requirements to keep the system safe from malicious attackers.

* Secure use of Open-Source Libraries: It is important to implement automated dependency checks on the software, routinely check for CVEs in dependencies, and apply patches/version updates promptly.
* Evolving Web Application Technologies: It is important to strong authentication practices and constantly upgrade when vulnerabilities are discovered. It is also important to incorporate security frameworks such as Spring security and OWASP best practices.
* Cloud and DevSecOps Integration: It is important for any organization to integrate security testing in the CI/CD pipeline and ensure compliance with financial industry regulations.
* The use of Artificial Intelligence: As artificial intelligence becomes more prevalent in society it is important to consider the security implications of both using it in the organization and how malicious actors could use it to access sensitive information.

**2. Areas of Security**

Refer to the vulnerability assessment process flow diagram. Identify which areas of security apply to Artemis Financial’s software application. Justify your reasoning for why each area is relevant to the software application.

The following areas of security apply to Artemis Financial and should be taken into consideration when examining their software for vulnerabilities.

* **Input Validation**: Since Artemis Financial’s application will process and interact with clients financial data and personal information it is important to implement strict input validation. Without such restrictions the system could be vulnerable to attacks such as SQL injection, cross-site scripting, and malicious JSON inputs. Proper input validation helps to ensure that only properly formatted and authorized data enters the software which will help reduce the risk of compromise and corruption.
* **APIs**: Since Artemis Financial relies on a RESTful web API it is essential to secure endpoints. APIs are routinely targeted for unauthorized data access and service disruption making it extremely important to ensure strong authentication mechanisms are in place along with secure tokens and rate limiting. In the case of Artemis Financial this is especially important since it is the main gateway between the clients and sensitive financial information.
* **Cryptography**: It is extremely important for Artemis Financial to protect customer data both in transit and at rest. A good way to ensure this protection is the use of robust cryptographic practices. This can be achieved with protocols such as TLS 1.3 to help prevent communication interception. This can be enhanced by using integrity checks and digital signatures to ensure financial data is not altered by malicious attackers. Cryptography is also a requirement of many governmental regulations around the world.
* **Client/Server**: Since Artemis Financial operates a client-server model, it is imperative that both sides enforce security. For the client side it is important to protect against session hijacking and ensure proper token management. On the server side it is important to use strong authentication and role-based user access controls. Properly handing secure client-server communication can help to prevent vulnerabilities such as man-in-the-middle attacks.
* **Code Error**: Since improper error handling can reveal sensitive details to attackers such as database structures, software dependencies, and system architecture it is important for Artemis Financial to properly manage these errors. One way this is achieved is to ensure that error messages are generic for end users and detailed logs are only available for developers and other trusted users. This will help to reduce the risk of attackers exploiting leaked system information.
* **Code Quality**: Since Artemis Financial wants to modernize their software, code quality can help to ensure that their system is more robust against attacks. Focusing on high quality code can help to reduce the number of vulnerabilities attackers are able to exploit, help to prevent performance issues, and make maintaining the system easier. Higher quality coding also makes scaling the business easier in the future and allows future developers to build upon solid coding practices.
* **Encapsulation**: Encapsulation is important to Artemis Financial because it helps to ensure that financial data is accessible only through well-defined secure interfaces. By restricting access to sensitive information, the company can use the prince of least privilege to prevent attackers from accessing and manipulating sensitive information. Encapsulation can also be critical to limiting expose to sensitive system components.

**3. Manual Review**

Continue working through the vulnerability assessment process flow diagram. Identify all vulnerabilities in the code base by manually inspecting the code.

[Include your 7–10 findings here.]

Here are the vulnerabilities that I found in the code base provided.

* + - 1. There is no input validation or limitation on the setMyDateTime function. This is could lead to potential input abuse if not resolved. A screen shot of a computer program

         AI-generated content may be incorrect.
      2. The POM file is attempting to use version 5.3.0 for the Maven dependency check. This is an out of date file and has been updated to 12.1.0 A screen shot of a computer program

         AI-generated content may be incorrect.
      3. The Pom file has a suppression file that prevented the dependency check from running. This had to be removed for the check to run. A black background with white text

         AI-generated content may be incorrect.
      4. The Pom file was also trying to use an old version of Java 1.8 and needed to be changed to 11. A computer code with white text

         AI-generated content may be incorrect.
      5. The Customer class has the account\_balance set to public. This should be set to private to ensure it is secure. A black background with colorful text

         AI-generated content may be incorrect.
      6. The customer class also returns the customers account number with the showInfo() function. This should be considered PII and needs to be protected. A black background with white text

         AI-generated content may be incorrect.
      7. The deposit function is set to an int without limitations. This could lead to depositing negative funds into the account, no way to handle anything less than $1, and the possibility of an overflow input. A black background with white text

         AI-generated content may be incorrect.
      8. There is no encryption on the jdbc URL and that would cause plaintext database traffic A screen shot of a computer program

         AI-generated content may be incorrect.
      9. There are very few comments. This could lead to issues with other developers in the future and can cause confusion.

**4. Static Testing**

Run a dependency check on Artemis Financial’s software application to identify all security vulnerabilities in the code. Record the output from the dependency-check report. Include the following items:

* The names or vulnerability codes of the known vulnerabilities
* A brief description and recommended solutions provided by the dependency-check report
* Any attribution that documents how this vulnerability has been identified or documented previously

A screenshot of a computer

AI-generated content may be incorrect.

The Maven dependency check returned the following vulnerable dependencies.

* Hibernate-validator-6.0.18.final.jar (Medium, CVE Count: 3, Evidence Count: 32)
  + CVE-2025-35035
    - Hibernate Validator before 6.2.0 and 7.0.0, by default and depending how it is used, may interpolate user-supplied input in a constraint violation message with Expression Language. This could allow an attacker to access sensitive information or execute arbitrary Java code
  + CVE-2023-1932
    - A flaw was found in hibernate-validator's 'isValid' method in the org.hibernate.validator.internal.constraintvalidators.hv.SafeHtmlValidator class, which can be bypassed by omitting the tag ending in a less-than character. Browsers may render an invalid html, allowing HTML injection or Cross-Site-Scripting (XSS) attacks.
  + 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-core-2.10.2.jar (High, CVE Count: 2, Evidence Count: 43)
  + CVE-2025-52999
    - jackson-core contains core low-level incremental ("streaming") parser and generator abstractions used by Jackson Data Processor. In versions prior to 2.15.0, if a user parses an input file and it has deeply nested data, Jackson could end up throwing a StackoverflowError if the depth is particularly large. jackson-core 2.15.0 contains a configurable limit for how deep Jackson will traverse in an input document, defaulting to an allowable depth of 1000. jackson-core will throw a StreamConstraintsException if the limit is reached. jackson-databind also benefits from this change because it uses jackson-core to parse JSON inputs. As a workaround, users should avoid parsing input files from untrusted sources.
  + CVE-2025-49128
    - Jackson-core contains core low-level incremental ("streaming") parser and generator abstractions used by Jackson Data Processor. Starting in version 2.0.0 and prior to version 2.13.0, a flaw in jackson-core's `JsonLocation.\_appendSourceDesc` method allows up to 500 bytes of unintended memory content to be included in exception messages. When parsing JSON from a byte array with an offset and length, the exception message incorrectly reads from the beginning of the array instead of the logical payload start. This results in possible information disclosure in systems using pooled or reused buffers, like Netty or Vert.x
* Jackson-databind-2.10.2.jar (High, CVE Count: 6, Evidence Count: 39)
  + 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.
  + CVE-2021-46877
    - jackson-databind 2.10.x through 2.12.x before 2.12.6 and 2.13.x before 2.13.1 allows attackers to cause a denial of service (2 GB transient heap usage per read) in uncommon situations involving JsonNode JDK serialization.
  + CVE-2022-42003
    - In FasterXML jackson-databind before versions 2.13.4.1 and 2.12.17.1, resource exhaustion can occur because of a lack of a check in primitive value deserializers to avoid deep wrapper array nesting, when the UNWRAP\_SINGLE\_VALUE\_ARRAYS feature is enabled.
  + CVE-2022-42004
    - In FasterXML jackson-databind before 2.13.4, resource exhaustion can occur because of a lack of a check in BeanDeserializer.\_deserializeFromArray to prevent use of deeply nested arrays. An application is vulnerable only with certain customized choices for deserialization.
  + CVE-2023-35116
    - jackson-databind through 2.15.2 allows attackers to cause a denial of service or other unspecified impact via a crafted object that uses cyclic dependencies. NOTE: the vendor's perspective is that this is not a valid vulnerability report, because the steps of constructing a cyclic data structure and trying to serialize it cannot be achieved by an external attacker.
* Json-path2.4.0.jar (Medium, CVE Count: 1, Evidence Count: 33)
  + CVE-2023-51074
    - json-path v2.8.0 was discovered to contain a stack overflow via the Criteria.parse() method.
* Json-smart-2.3.jar (High, CVE Count: 3, Evidence Count: 45)
  + CVE-2021-31684
    - A vulnerability was discovered in the indexOf function of JSONParserByteArray in JSON Smart versions 1.3 and 2.4 which causes a denial of service (DOS) via a crafted web request.
  + CVE-2023-1370
    - It was discovered that the code does not have any limit to the nesting of such arrays or objects. Since the parsing of nested arrays and objects is done recursively, nesting too many of them can cause a stack exhaustion (stack overflow) and crash the software.
  + CVE-2021-27568
    - An issue was discovered in netplex json-smart-v1 through 2015-10-23 and json-smart-v2 through 2.4. An exception is thrown from a function, but it is not caught, as demonstrated by NumberFormatException. When it is not caught, it may cause programs using the library to crash or expose sensitive information.
* Log4j-api-2.12.1.jar (Low, CVE Count: 1, Evidence Count: 42)
  + 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-classic-1.2.3.jar (High, CVE Count: 2, Evidence Count: 31)
  + CVE-2023-6378
    - A serialization vulnerability in logback receiver component part of logback version 1.4.11 allows an attacker to mount a Denial-Of-Service attack by sending poisoned data.
  + 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.
* Logback-core-1.2.3.jar (High, CVE Count: 4, Evidence Count: 31)
  + CVE-2023-6378
    - A serialization vulnerability in logback receiver component part of logback version 1.4.11 allows an attacker to mount a Denial-Of-Service attack by sending poisoned data.
  + 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.
  + CVE-2024-12798
    - ACE vulnerability in JaninoEventEvaluator by QOS.CH logback-core upto including version 0.1 to 1.3.14 and 1.4.0 to 1.5.12 in Java applications allows attacker to execute arbitrary code by compromising an existing logback configuration file or by injecting an environment variable before program execution.
  + CVE-2024-12801
    - Server-Side Request Forgery (SSRF) in SaxEventRecorder by QOS.CH logback version 0.1 to 1.3.14 and 1.4.0 to 1.5.12  on the Java platform, allows an attacker to forge requests by compromising logback configuration files in XML.
* Snakeyaml-1.25.jar (Critical, CVE Count: 8, Evidence Count: 44)
  + CVE-2022-1471
    - 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. We recommend upgrading to version 2.0 and beyond.
  + 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.
  + CVE-2022-38749
    - Using snakeYAML to parse untrusted YAML files may be vulnerable to Denial of Service attacks (DOS). If the parser is running on user supplied input, an attacker may supply content that causes the parser to crash by stackoverflow.
  + CVE-2022-38751
    - Using snakeYAML to parse untrusted YAML files may be vulnerable to Denial of Service attacks (DOS). If the parser is running on user supplied input, an attacker may supply content that causes the parser to crash by stackoverflow.
  + CVE-2022-38752
    - Using snakeYAML to parse untrusted YAML files may be vulnerable to Denial of Service attacks (DOS). If the parser is running on user supplied input, an attacker may supply content that causes the parser to crash by stack-overflow.
  + CVE-2022-41854
    - Those using Snakeyaml to parse untrusted YAML files may be vulnerable to Denial of Service attacks (DOS). If the parser is running on user supplied input, an attacker may supply content that causes the parser to crash by stack overflow. This effect may support a denial of service attack.
  + CVE-2022-38750
    - Using snakeYAML to parse untrusted YAML files may be vulnerable to Denial of Service attacks (DOS). If the parser is running on user supplied input, an attacker may supply content that causes the parser to crash by stackoverflow.
* Spring-boot-2.2.4.RELEASE.jar (Critical, CVE Count: 3, Evidence Count: 39)
  + CVE-2023-20873
    - In Spring Boot versions 3.0.0 - 3.0.5, 2.7.0 - 2.7.10, and older unsupported versions, an application that is deployed to Cloud Foundry could be susceptible to a security bypass. Users of affected versions should apply the following mitigation: 3.0.x users should upgrade to 3.0.6+. 2.7.x users should upgrade to 2.7.11+. Users of older, unsupported versions should upgrade to 3.0.6+ or 2.7.11+.
  + CVE-2022-27772
    - 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
  + CVE-2023-20883
    - In Spring Boot versions 3.0.0 - 3.0.6, 2.7.0 - 2.7.11, 2.6.0 - 2.6.14, 2.5.0 - 2.5.14 and older unsupported versions, there is potential for a denial-of-service (DoS) attack if Spring MVC is used together with a reverse proxy cache.
* Spring-boot-starter-web-2.2.4.RELEASE.jar (Critical, CVE Count: 3, Evidence Count: 35)
  + CVE-2023-20873
    - In Spring Boot versions 3.0.0 - 3.0.5, 2.7.0 - 2.7.10, and older unsupported versions, an application that is deployed to Cloud Foundry could be susceptible to a security bypass. Users of affected versions should apply the following mitigation: 3.0.x users should upgrade to 3.0.6+. 2.7.x users should upgrade to 2.7.11+. Users of older, unsupported versions should upgrade to 3.0.6+ or 2.7.11+.
  + CVE-2022-27772
    - 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
  + CVE-2023-20883
    - In Spring Boot versions 3.0.0 - 3.0.6, 2.7.0 - 2.7.11, 2.6.0 - 2.6.14, 2.5.0 - 2.5.14 and older unsupported versions, there is potential for a denial-of-service (DoS) attack if Spring MVC is used together with a reverse proxy cache.
* Spring-context-5.2.3.RELEASE.jar (Critical, CVE Count: 13, Evidence Count: 34)
  + 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-2024-22259
    - Applications that use UriComponentsBuilder in Spring Framework to parse an externally provided URL (e.g. through a query parameter) AND perform validation checks on the host of the parsed URL may be vulnerable to a open redirect https://cwe.mitre.org/data/definitions/601.html  attack or to a SSRF attack if the URL is used after passing validation checks.
  + 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.
  + 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.
  + CVE-2022-22950
    - in Spring Framework versions 5.3.0 - 5.3.16 and older unsupported versions, it is possible for a user to provide a specially crafted SpEL expression that may cause a denial of service condition.
  + CVE-2022-22971
    - In spring framework versions prior to 5.3.20+ , 5.2.22+ and old unsupported versions, application with a STOMP over WebSocket endpoint is vulnerable to a denial of service attack by an authenticated user.
  + CVE-2023-20861
    - In Spring Framework versions 6.0.0 - 6.0.6, 5.3.0 - 5.3.25, 5.2.0.RELEASE - 5.2.22.RELEASE, and older unsupported versions, it is possible for a user to provide a specially crafted SpEL expression that may cause a denial-of-service (DoS) condition.
  + CVE-2023-20863
    - In spring framework versions prior to 5.2.24 release+ ,5.3.27+ and 6.0.8+ , it is possible for a user to provide a specially crafted SpEL expression that may cause a denial-of-service (DoS) condition.
  + CVE-2022-22968
    - In Spring Framework versions 5.3.0 - 5.3.18, 5.2.0 - 5.2.20, and older unsupported versions, the patterns for disallowedFields on a DataBinder are case sensitive which means a field is not effectively protected unless it is listed with both upper and lower case for the first character of the field, including upper and lower case for the first character of all nested fields within the property path.
  + CVE-2022-22970
    - In spring framework versions prior to 5.3.20+ , 5.2.22+ and old unsupported versions, applications that handle file uploads are vulnerable to DoS attack if they rely on data binding to set a MultipartFile or javax.servlet.Part to a field in a model object.
  + CVE-2021-22060
    - In Spring Framework versions 5.3.0 - 5.3.13, 5.2.0 - 5.2.18, and older unsupported versions, it is possible for a user to provide malicious input to cause the insertion of additional log entries. This is a follow-up to CVE-2021-22096 that protects against additional types of input and in more places of the Spring Framework codebase.
  + CVE-2021-22096
    - In Spring Framework versions 5.3.0 - 5.3.10, 5.2.0 - 5.2.17, and older unsupported versions, it is possible for a user to provide malicious input to cause the insertion of additional log entries.
  + CVE-2025-22233
    - CVE-2024-38820 ensured Locale-independent, lowercase conversion for both the configured disallowedFields patterns and for request parameter names. However, there are still cases where it is possible to bypass the disallowedFields checks.
* Spring-core-5.2.3.RELEASE.jar (Critical, CVE Count: 14, Evidence Count: 36)
  + 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-2025-41249
    - The Spring Framework annotation detection mechanism may not correctly resolve annotations on methods within type hierarchies with a parameterized super type with unbounded generics. This can be an issue if such annotations are used for authorization decisions.
  + CVE-2025-41242
    - Spring Framework MVC applications can be vulnerable to a “Path Traversal Vulnerability” when deployed on a non-compliant Servlet container.
  + CVE-2024-22259
    - Applications that use UriComponentsBuilder in Spring Framework to parse an externally provided URL (e.g. through a query parameter) AND perform validation checks on the host of the parsed URL may be vulnerable to a open redirect https://cwe.mitre.org/data/definitions/601.html  attack or to a SSRF attack if the URL is used after passing validation checks.
  + 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.
  + 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.
  + CVE-2022-22950
    - in Spring Framework versions 5.3.0 - 5.3.16 and older unsupported versions, it is possible for a user to provide a specially crafted SpEL expression that may cause a denial of service condition.
  + CVE-2022-22971
    - In spring framework versions prior to 5.3.20+ , 5.2.22+ and old unsupported versions, application with a STOMP over WebSocket endpoint is vulnerable to a denial of service attack by an authenticated user.
  + CVE-2023-20861
    - In Spring Framework versions 6.0.0 - 6.0.6, 5.3.0 - 5.3.25, 5.2.0.RELEASE - 5.2.22.RELEASE, and older unsupported versions, it is possible for a user to provide a specially crafted SpEL expression that may cause a denial-of-service (DoS) condition.
  + CVE-2023-20863
    - In spring framework versions prior to 5.2.24 release+ ,5.3.27+ and 6.0.8+ , it is possible for a user to provide a specially crafted SpEL expression that may cause a denial-of-service (DoS) condition.
  + CVE-2022-22968
    - In Spring Framework versions 5.3.0 - 5.3.18, 5.2.0 - 5.2.20, and older unsupported versions, the patterns for disallowedFields on a DataBinder are case sensitive which means a field is not effectively protected unless it is listed with both upper and lower case for the first character of the field, including upper and lower case for the first character of all nested fields within the property path.
  + CVE-2022-22970
    - In spring framework versions prior to 5.3.20+ , 5.2.22+ and old unsupported versions, applications that handle file uploads are vulnerable to DoS attack if they rely on data binding to set a MultipartFile or javax.servlet.Part to a field in a model object.
  + CVE-2021-22060
    - In Spring Framework versions 5.3.0 - 5.3.13, 5.2.0 - 5.2.18, and older unsupported versions, it is possible for a user to provide malicious input to cause the insertion of additional log entries. This is a follow-up to CVE-2021-22096 that protects against additional types of input and in more places of the Spring Framework codebase.
  + CVE-2021-22096
    - In Spring Framework versions 5.3.0 - 5.3.10, 5.2.0 - 5.2.17, and older unsupported versions, it is possible for a user to provide malicious input to cause the insertion of additional log entries.
* Spring-data-rest-webmvc-3.2.4.RELEASE.jar (Medium, CVE Count: 2, Evidence Count: 27)
  + CVE-2021-22047
    - In Spring Data REST versions 3.4.0 - 3.4.13, 3.5.0 - 3.5.5, and older unsupported versions, HTTP resources implemented by custom controllers using a configured base API path and a controller type-level request mapping are additionally exposed under URIs that can potentially be exposed for unauthorized access depending on the Spring Security configuration.
  + CVE-2022-31679
    - Applications that allow HTTP PATCH access to resources exposed by Spring Data REST in versions 3.6.0 - 3.5.5, 3.7.0 - 3.7.2, and older unsupported versions, if an attacker knows about the structure of the underlying domain model, they can craft HTTP requests that expose hidden entity attributes.
* Spring-expression-5.2.3.RELEASE.jar (Critical, CVE Count: 13, Evidence Count: 36)
  + 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-2024-22259
    - Applications that use UriComponentsBuilder in Spring Framework to parse an externally provided URL (e.g. through a query parameter) AND perform validation checks on the host of the parsed URL may be vulnerable to a open redirect https://cwe.mitre.org/data/definitions/601.html  attack or to a SSRF attack if the URL is used after passing validation checks.
  + 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.
  + 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.
  + CVE-2022-22950
    - in Spring Framework versions 5.3.0 - 5.3.16 and older unsupported versions, it is possible for a user to provide a specially crafted SpEL expression that may cause a denial of service condition.
  + CVE-2022-22971
    - In spring framework versions prior to 5.3.20+ , 5.2.22+ and old unsupported versions, application with a STOMP over WebSocket endpoint is vulnerable to a denial of service attack by an authenticated user.
  + CVE-2023-20861
    - In Spring Framework versions 6.0.0 - 6.0.6, 5.3.0 - 5.3.25, 5.2.0.RELEASE - 5.2.22.RELEASE, and older unsupported versions, it is possible for a user to provide a specially crafted SpEL expression that may cause a denial-of-service (DoS) condition.
  + CVE-2023-20863
    - In spring framework versions prior to 5.2.24 release+ ,5.3.27+ and 6.0.8+ , it is possible for a user to provide a specially crafted SpEL expression that may cause a denial-of-service (DoS) condition.
  + CVE-2024-38808
    - In Spring Framework versions 5.3.0 - 5.3.38 and older unsupported versions, it is possible for a user to provide a specially crafted Spring Expression Language (SpEL) expression that may cause a denial of service (DoS) condition.
  + CVE-2022-22968
    - In Spring Framework versions 5.3.0 - 5.3.18, 5.2.0 - 5.2.20, and older unsupported versions, the patterns for disallowedFields on a DataBinder are case sensitive which means a field is not effectively protected unless it is listed with both upper and lower case for the first character of the field, including upper and lower case for the first character of all nested fields within the property path.
  + CVE-2022-22970
    - In spring framework versions prior to 5.3.20+ , 5.2.22+ and old unsupported versions, applications that handle file uploads are vulnerable to DoS attack if they rely on data binding to set a MultipartFile or javax.servlet.Part to a field in a model object.
  + CVE-2021-22060
    - In Spring Framework versions 5.3.0 - 5.3.13, 5.2.0 - 5.2.18, and older unsupported versions, it is possible for a user to provide malicious input to cause the insertion of additional log entries. This is a follow-up to CVE-2021-22096 that protects against additional types of input and in more places of the Spring Framework codebase.
  + CVE-2021-22096
    - In Spring Framework versions 5.3.0 - 5.3.10, 5.2.0 - 5.2.17, and older unsupported versions, it is possible for a user to provide malicious input to cause the insertion of additional log entries.
* Spring-hateoas-1.0.3.RELEASE.jar (Medium, CVE Count: 1, Evidence Count: 43)
  + CVE-2023-24036
    - Reactive web applications that use Spring HATEOAS to produce hypermedia-based responses might be exposed to malicious forwarded headers if they are not behind a trusted proxy that ensures correctness of such headers, or if they don't have anything else in place to handle (and possibly discard) forwarded headers either in WebFlux or at the level of the underlying HTTP server.
* Spring-tx-5.2.3.RELEASE.jar (Critical, CVE Count: 12, Evidence Count: 34)
  + 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-2024-22259
    - Applications that use UriComponentsBuilder in Spring Framework to parse an externally provided URL (e.g. through a query parameter) AND perform validation checks on the host of the parsed URL may be vulnerable to a open redirect https://cwe.mitre.org/data/definitions/601.html  attack or to a SSRF attack if the URL is used after passing validation checks.
  + 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.
  + 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.
  + CVE-2022-22950
    - in Spring Framework versions 5.3.0 - 5.3.16 and older unsupported versions, it is possible for a user to provide a specially crafted SpEL expression that may cause a denial of service condition.
  + CVE-2022-22971
    - In spring framework versions prior to 5.3.20+ , 5.2.22+ and old unsupported versions, application with a STOMP over WebSocket endpoint is vulnerable to a denial of service attack by an authenticated user.
  + CVE-2023-20861
    - In Spring Framework versions 6.0.0 - 6.0.6, 5.3.0 - 5.3.25, 5.2.0.RELEASE - 5.2.22.RELEASE, and older unsupported versions, it is possible for a user to provide a specially crafted SpEL expression that may cause a denial-of-service (DoS) condition.
  + CVE-2023-20863
    - In spring framework versions prior to 5.2.24 release+ ,5.3.27+ and 6.0.8+ , it is possible for a user to provide a specially crafted SpEL expression that may cause a denial-of-service (DoS) condition.
  + CVE-2022-22968
    - In Spring Framework versions 5.3.0 - 5.3.18, 5.2.0 - 5.2.20, and older unsupported versions, the patterns for disallowedFields on a DataBinder are case sensitive which means a field is not effectively protected unless it is listed with both upper and lower case for the first character of the field, including upper and lower case for the first character of all nested fields within the property path.
  + CVE-2022-22970
    - In spring framework versions prior to 5.3.20+ , 5.2.22+ and old unsupported versions, applications that handle file uploads are vulnerable to DoS attack if they rely on data binding to set a MultipartFile or javax.servlet.Part to a field in a model object.
  + CVE-2021-22060
    - In Spring Framework versions 5.3.0 - 5.3.13, 5.2.0 - 5.2.18, and older unsupported versions, it is possible for a user to provide malicious input to cause the insertion of additional log entries. This is a follow-up to CVE-2021-22096 that protects against additional types of input and in more places of the Spring Framework codebase.
  + CVE-2021-22096
    - In Spring Framework versions 5.3.0 - 5.3.10, 5.2.0 - 5.2.17, and older unsupported versions, it is possible for a user to provide malicious input to cause the insertion of additional log entries.
* Spring-web-5.2.3.RELEASE.jar (Critical, CVE Count: 19, Evidence Count: 34)
  + 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.
  + 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-2024-38809
    - Applications that parse ETags from "If-Match" or "If-None-Match" request headers are vulnerable to DoS attack.
  + CVE-2025-41249
    - The Spring Framework annotation detection mechanism may not correctly resolve annotations on methods within type hierarchies with a parameterized super type with unbounded generics. This can be an issue if such annotations are used for authorization decisions.
  + CVE-2024-22243
    - Applications that use UriComponentsBuilder to parse an externally provided URL (e.g. through a query parameter) AND perform validation checks on the host of the parsed URL may be vulnerable to a open redirect https://cwe.mitre.org/data/definitions/601.html  attack or to a SSRF attack if the URL is used after passing validation checks.
  + CVE-2024-22262
    - Applications that use UriComponentsBuilder to parse an externally provided URL (e.g. through a query parameter) AND perform validation checks on the host of the parsed URL may be vulnerable to a open redirect https://cwe.mitre.org/data/definitions/601.html  attack or to a SSRF attack if the URL is used after passing validation checks.
  + CVE-2024-22259
    - Applications that use UriComponentsBuilder in Spring Framework to parse an externally provided URL (e.g. through a query parameter) AND perform validation checks on the host of the parsed URL may be vulnerable to a open redirect https://cwe.mitre.org/data/definitions/601.html  attack or to a SSRF attack if the URL is used after passing validation checks.
  + 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.
  + CVE-2025-41234
    - In Spring Framework, versions 6.0.x as of 6.0.5, versions 6.1.x and 6.2.x, an application is vulnerable to a reflected file download (RFD) attack when it sets a “Content-Disposition” header with a non-ASCII charset, where the filename attribute is derived from user-supplied input.
  + CVE-2024-38828
    - Spring MVC controller methods with an @RequestBody byte[] method parameter are vulnerable to a DoS attack.
  + 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.
  + CVE-2022-22950
    - in Spring Framework versions 5.3.0 - 5.3.16 and older unsupported versions, it is possible for a user to provide a specially crafted SpEL expression that may cause a denial of service condition.
  + CVE-2022-22971
    - In spring framework versions prior to 5.3.20+ , 5.2.22+ and old unsupported versions, application with a STOMP over WebSocket endpoint is vulnerable to a denial of service attack by an authenticated user.
  + CVE-2023-20861
    - In Spring Framework versions 6.0.0 - 6.0.6, 5.3.0 - 5.3.25, 5.2.0.RELEASE - 5.2.22.RELEASE, and older unsupported versions, it is possible for a user to provide a specially crafted SpEL expression that may cause a denial-of-service (DoS) condition.
  + CVE-2023-20863
    - In spring framework versions prior to 5.2.24 release+ ,5.3.27+ and 6.0.8+ , it is possible for a user to provide a specially crafted SpEL expression that may cause a denial-of-service (DoS) condition.
  + CVE-2022-22968
    - In Spring Framework versions 5.3.0 - 5.3.18, 5.2.0 - 5.2.20, and older unsupported versions, the patterns for disallowedFields on a DataBinder are case sensitive which means a field is not effectively protected unless it is listed with both upper and lower case for the first character of the field, including upper and lower case for the first character of all nested fields within the property path.
  + CVE-2022-22970
    - In spring framework versions prior to 5.3.20+ , 5.2.22+ and old unsupported versions, applications that handle file uploads are vulnerable to DoS attack if they rely on data binding to set a MultipartFile or javax.servlet.Part to a field in a model object.
  + CVE-2021-22060
    - In Spring Framework versions 5.3.0 - 5.3.13, 5.2.0 - 5.2.18, and older unsupported versions, it is possible for a user to provide malicious input to cause the insertion of additional log entries. This is a follow-up to CVE-2021-22096 that protects against additional types of input and in more places of the Spring Framework codebase.
  + CVE-2021-22096
    - In Spring Framework versions 5.3.0 - 5.3.10, 5.2.0 - 5.2.17, and older unsupported versions, it is possible for a user to provide malicious input to cause the insertion of additional log entries.
* Spring-webmvc-5.2.3.RELEASE.jar (Critical, CVE Count: 13, Evidence Count: 36)
  + 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-2024-38816
    - Applications serving static resources through the functional web frameworks WebMvc.fn or WebFlux.fn are vulnerable to path traversal attacks. An attacker can craft malicious HTTP requests and obtain any file on the file system that is also accessible to the process in which the Spring application is running.
  + CVE-2024-22259
    - Applications that use UriComponentsBuilder in Spring Framework to parse an externally provided URL (e.g. through a query parameter) AND perform validation checks on the host of the parsed URL may be vulnerable to a open redirect https://cwe.mitre.org/data/definitions/601.html  attack or to a SSRF attack if the URL is used after passing validation checks.
  + 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.
  + 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.
  + CVE-2022-22950
    - in Spring Framework versions 5.3.0 - 5.3.16 and older unsupported versions, it is possible for a user to provide a specially crafted SpEL expression that may cause a denial of service condition.
  + CVE-2022-22971
    - In spring framework versions prior to 5.3.20+ , 5.2.22+ and old unsupported versions, application with a STOMP over WebSocket endpoint is vulnerable to a denial of service attack by an authenticated user.
  + CVE-2023-20861
    - In Spring Framework versions 6.0.0 - 6.0.6, 5.3.0 - 5.3.25, 5.2.0.RELEASE - 5.2.22.RELEASE, and older unsupported versions, it is possible for a user to provide a specially crafted SpEL expression that may cause a denial-of-service (DoS) condition.
  + CVE-2023-20863
    - In spring framework versions prior to 5.2.24 release+ ,5.3.27+ and 6.0.8+ , it is possible for a user to provide a specially crafted SpEL expression that may cause a denial-of-service (DoS) condition.
  + CVE-2022-22968
    - In Spring Framework versions 5.3.0 - 5.3.18, 5.2.0 - 5.2.20, and older unsupported versions, the patterns for disallowedFields on a DataBinder are case sensitive which means a field is not effectively protected unless it is listed with both upper and lower case for the first character of the field, including upper and lower case for the first character of all nested fields within the property path.
  + CVE-2022-22970
    - In spring framework versions prior to 5.3.20+ , 5.2.22+ and old unsupported versions, applications that handle file uploads are vulnerable to DoS attack if they rely on data binding to set a MultipartFile or javax.servlet.Part to a field in a model object.
  + CVE-2021-22060
    - In Spring Framework versions 5.3.0 - 5.3.13, 5.2.0 - 5.2.18, and older unsupported versions, it is possible for a user to provide malicious input to cause the insertion of additional log entries. This is a follow-up to CVE-2021-22096 that protects against additional types of input and in more places of the Spring Framework codebase.
  + CVE-2021-22096
    - In Spring Framework versions 5.3.0 - 5.3.10, 5.2.0 - 5.2.17, and older unsupported versions, it is possible for a user to provide malicious input to cause the insertion of additional log entries.
* Tomcat-embed-core-9.0.30.jar (Critical, CVE Count: 45, Evidence Count: 30)
  + 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.
  + CVE-2024-50379
    - Time-of-check Time-of-use (TOCTOU) Race Condition vulnerability during JSP compilation in Apache Tomcat permits an RCE on case insensitive file systems when the default servlet is enabled for write (non-default configuration).
  + CVE-2024-52316
    - Unchecked Error Condition vulnerability in Apache Tomcat. If Tomcat is configured to use a custom Jakarta Authentication (formerly JASPIC) ServerAuthContext component which may throw an exception during the authentication process without explicitly setting an HTTP status to indicate failure, the authentication may not fail, allowing the user to bypass the authentication process. There are no known Jakarta Authentication components that behave in this way.
  + CVE-2024-56337
    - Time-of-check Time-of-use (TOCTOU) Race Condition vulnerability in Apache Tomcat.
  + CVE-2025-24813
    - Path Equivalence: 'file.Name' (Internal Dot) leading to Remote Code Execution and/or Information disclosure and/or malicious content added to uploaded files via write enabled Default Servlet in Apache Tomcat.
  + CVE-2025-31651
    - Improper Neutralization of Escape, Meta, or Control Sequences vulnerability in Apache Tomcat. For a subset of unlikely rewrite rule configurations, it was possible for a specially crafted request to bypass some rewrite rules. If those rewrite rules effectively enforced security constraints, those constraints could be bypassed.
  + CVE-2025-49124
    - Untrusted Search Path vulnerability in Apache Tomcat installer for Windows. During installation, the Tomcat installer for Windows used icacls.exe without specifying a full path.
  + CVE-2020-11996
    - A specially crafted sequence of HTTP/2 requests sent to Apache Tomcat 10.0.0-M1 to 10.0.0-M5, 9.0.0.M1 to 9.0.35 and 8.5.0 to 8.5.55 could trigger high CPU usage for several seconds. If a sufficient number of such requests were made on concurrent HTTP/2 connections, the server could become unresponsive.
  + CVE-2020-12934
    - An h2c direct connection to Apache Tomcat 10.0.0-M1 to 10.0.0-M6, 9.0.0.M5 to 9.0.36 and 8.5.1 to 8.5.56 did not release the HTTP/1.1 processor after the upgrade to HTTP/2. If a sufficient number of such requests were made, an OutOfMemoryException could occur leading to a denial of service.
  + CVE-2020-13935
    - The payload length in a WebSocket frame was not correctly validated in Apache Tomcat 10.0.0-M1 to 10.0.0-M6, 9.0.0.M1 to 9.0.36, 8.5.0 to 8.5.56 and 7.0.27 to 7.0.104. Invalid payload lengths could trigger an infinite loop. Multiple requests with invalid payload lengths could lead to a denial of service.
  + CVE-2020-17527
    - While investigating bug 64830 it was discovered that Apache Tomcat 10.0.0-M1 to 10.0.0-M9, 9.0.0-M1 to 9.0.39 and 8.5.0 to 8.5.59 could re-use an HTTP request header value from the previous stream received on an HTTP/2 connection for the request associated with the subsequent stream. While this would most likely lead to an error and the closure of the HTTP/2 connection, it is possible that information could leak between requests.
  + CVE-2021-25122
    - When responding to new h2c connection requests, Apache Tomcat versions 10.0.0-M1 to 10.0.0, 9.0.0.M1 to 9.0.41 and 8.5.0 to 8.5.61 could duplicate request headers and a limited amount of request body from one request to another meaning user A and user B could both see the results of user A's request.
  + CVE-2021-41079
    - Apache Tomcat 8.5.0 to 8.5.63, 9.0.0-M1 to 9.0.43 and 10.0.0-M1 to 10.0.2 did not properly validate incoming TLS packets. When Tomcat was configured to use NIO+OpenSSL or NIO2+OpenSSL for TLS, a specially crafted packet could be used to trigger an infinite loop resulting in a denial of service.
  + CVE-2022-29885
    - The documentation of Apache Tomcat 10.1.0-M1 to 10.1.0-M14, 10.0.0-M1 to 10.0.20, 9.0.13 to 9.0.62 and 8.5.38 to 8.5.78 for the EncryptInterceptor incorrectly stated it enabled Tomcat clustering to run over an untrusted network. This was not correct. While the EncryptInterceptor does provide confidentiality and integrity protection, it does not protect against all risks associated with running over any untrusted network, particularly DoS risks.
  + CVE-2022-42252
    - If Apache Tomcat 8.5.0 to 8.5.82, 9.0.0-M1 to 9.0.67, 10.0.0-M1 to 10.0.26 or 10.1.0-M1 to 10.1.0 was configured to ignore invalid HTTP headers via setting rejectIllegalHeader to false (the default for 8.5.x only), Tomcat did not reject a request containing an invalid Content-Length header making a request smuggling attack possible if Tomcat was located behind a reverse proxy that also failed to reject the request with the invalid header.
  + CVE-2023-44487
    - The HTTP/2 protocol allows a denial of service (server resource consumption) because request cancellation can reset many streams quickly, as exploited in the wild in August through October 2023.
  + CVE-2023-46589
    - Improper Input Validation vulnerability in Apache Tomcat.Tomcat from 11.0.0-M1 through 11.0.0-M10, from 10.1.0-M1 through 10.1.15, from 9.0.0-M1 through 9.0.82 and from 8.5.0 through 8.5.95 did not correctly parse HTTP trailer headers. A trailer header that exceeded the header size limit could cause Tomcat to treat a single request as multiple requests leading to the possibility of request smuggling when behind a reverse proxy.
  + CVE-2024-24549
    - Denial of Service due to improper input validation vulnerability for HTTP/2 requests in Apache Tomcat. When processing an HTTP/2 request, if the request exceeded any of the configured limits for headers, the associated HTTP/2 stream was not reset until after all of the headers had been processed.This issue affects Apache Tomcat: from 11.0.0-M1 through 11.0.0-M16, from 10.1.0-M1 through 10.1.18, from 9.0.0-M1 through 9.0.85, from 8.5.0 through 8.5.98.
  + CVE-2024-34750
    - Improper Handling of Exceptional Conditions, Uncontrolled Resource Consumption vulnerability in Apache Tomcat. When processing an HTTP/2 stream, Tomcat did not handle some cases of excessive HTTP headers correctly. This led to a miscounting of active HTTP/2 streams which in turn led to the use of an incorrect infinite timeout which allowed connections to remain open which should have been closed.
  + CVE-2024-38286
    - Allocation of Resources Without Limits or Throttling vulnerability in Apache Tomcat.
  + CVE-2025-48988
    - Allocation of Resources Without Limits or Throttling vulnerability in Apache Tomcat.
  + CVE-2025-48989
    - Improper Resource Shutdown or Release vulnerability in Apache Tomcat made Tomcat vulnerable to the made you reset attack.
  + CVE-2025-49125
    - Authentication Bypass Using an Alternate Path or Channel vulnerability in Apache Tomcat.  When using PreResources or PostResources mounted other than at the root of the web application, it was possible to access those resources via an unexpected path. That path was likely not to be protected by the same security constraints as the expected path, allowing those security constraints to be bypassed.
  + CVE—2025-52434
    - Concurrent Execution using Shared Resource with Improper Synchronization ('Race Condition') vulnerability in Apache Tomcat when using the APR/Native connector. This was particularly noticeable with client initiated closes of HTTP/2 connections.
  + CVE-2025-5252
    - For some unlikely configurations of multipart upload, an Integer Overflow vulnerability in Apache Tomcat could lead to a DoS via bypassing of size limits.
  + CVE-2025-53506
    - Uncontrolled Resource Consumption vulnerability in Apache Tomcat if an HTTP/2 client did not acknowledge the initial settings frame that reduces the maximum permitted concurrent streams.
  + CVE-2025-46701
    - Improper Handling of Case Sensitivity vulnerability in Apache Tomcat's GCI servlet allows security constraint bypass of security constraints that apply to the pathInfo component of a URI mapped to the CGI servlet.
  + CVE-2020-9484
    - When using Apache Tomcat versions 10.0.0-M1 to 10.0.0-M4, 9.0.0.M1 to 9.0.34, 8.5.0 to 8.5.54 and 7.0.0 to 7.0.103 if a) an attacker is able to control the contents and name of a file on the server; and b) the server is configured to use the PersistenceManager with a FileStore; and c) the PersistenceManager is configured with sessionAttributeValueClassNameFilter="null"
  + CVE-2021-25329
    - The fix for CVE-2020-9484 was incomplete. When using Apache Tomcat 10.0.0-M1 to 10.0.0, 9.0.0.M1 to 9.0.41, 8.5.0 to 8.5.61 or 7.0.0. to 7.0.107 with a configuration edge case that was highly unlikely to be used, the Tomcat instance was still vulnerable to CVE-2020-9494. Note that both the previously published prerequisites for CVE-2020-9484 and the previously published mitigations for CVE-2020-9484 also apply to this issue.
  + CVE-2021-30640
    - A vulnerability in the JNDI Realm of Apache Tomcat allows an attacker to authenticate using variations of a valid user name and/or to bypass some of the protection provided by the LockOut Realm. This issue affects Apache Tomcat 10.0.0-M1 to 10.0.5; 9.0.0.M1 to 9.0.45; 8.5.0 to 8.5.65.
  + CVE-2025-55668
    - Session Fixation vulnerability in Apache Tomcat via rewrite valve.
  + CVE-2024-23672
    - Denial of Service via incomplete cleanup vulnerability in Apache Tomcat. It was possible for WebSocket clients to keep WebSocket connections open leading to increased resource consumption.This issue affects Apache Tomcat: from 11.0.0-M1 through 11.0.0-M16, from 10.1.0-M1 through 10.1.18, from 9.0.0-M1 through 9.0.85, from 8.5.0 through 8.5.98.
  + CVE-2022-34305
    - In Apache Tomcat 10.1.0-M1 to 10.1.0-M16, 10.0.0-M1 to 10.0.22, 9.0.30 to 9.0.64 and 8.5.50 to 8.5.81 the Form authentication example in the examples web application displayed user provided data without filtering, exposing a XSS vulnerability.
  + CVE-2023-41080
    - URL Redirection to Untrusted Site ('Open Redirect') vulnerability in FORM authentication feature Apache Tomcat
  + CVE-2021-24122
    - When serving resources from a network location using the NTFS file system, Apache Tomcat versions 10.0.0-M1 to 10.0.0-M9, 9.0.0.M1 to 9.0.39, 8.5.0 to 8.5.59 and 7.0.0 to 7.0.106 were susceptible to JSP source code disclosure in some configurations. The root cause was the unexpected behaviour of the JRE API File.getCanonicalPath() which in turn was caused by the inconsistent behaviour of the Windows API (FindFirstFileW) in some circumstances.
  + CVE-2021-33037
    - 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 honoured the identify encoding; and - Tomcat did not ensure that, if present, the chunked encoding was the final encoding.
  + CVE-2023-42795
    - Incomplete Cleanup vulnerability in Apache Tomcat.When recycling various internal objects in Apache Tomcat from 11.0.0-M1 through 11.0.0-M11, from 10.1.0-M1 through 10.1.13, from 9.0.0-M1 through 9.0.80 and from 8.5.0 through 8.5.93, an error could cause Tomcat to skip some parts of the recycling process leading to information leaking from the current request/response to the next.Older, EOL versions may also be affected.
  + CVE-2023-45648
    - Improper Input Validation vulnerability in Apache Tomcat.Tomcat from 11.0.0-M1 through 11.0.0-M11, from 10.1.0-M1 through 10.1.13, from 9.0.0-M1 through 9.0.81 and from 8.5.0 through 8.5.93 did not correctly parse HTTP trailer headers. A specially crafted, invalid trailer header could cause Tomcat to treat a single request as multiple requests leading to the possibility of request smuggling when behind a reverse proxy.
  + CVE-2024-21733
    - Generation of Error Message Containing Sensitive Information vulnerability in Apache Tomcat.This issue affects Apache Tomcat: from 8.5.7 through 8.5.63, from 9.0.0-M11 through 9.0.43.
  + CVE-2024-54677
    - Uncontrolled Resource Consumption vulnerability in the examples web application provided with Apache Tomcat leads to denial of service.
  + CVE-2019-17569
    - The refactoring present in Apache Tomcat 9.0.28 to 9.0.30, 8.5.48 to 8.5.50 and 7.0.98 to 7.0.99 introduced a regression. The result of the regression was that invalid Transfer-Encoding headers were incorrectly processed leading to a possibility of HTTP Request Smuggling if Tomcat was located behind a reverse proxy that incorrectly handled the invalid Transfer-Encoding header in a particular manner. Such a reverse proxy is considered unlikely.
  + CVE-2020-1935
    - In Apache Tomcat 9.0.0.M1 to 9.0.30, 8.5.0 to 8.5.50 and 7.0.0 to 7.0.99 the HTTP header parsing code used an approach to end-of-line parsing that allowed some invalid HTTP headers to be parsed as valid. This led to a possibility of HTTP Request Smuggling if Tomcat was located behind a reverse proxy that incorrectly handled the invalid Transfer-Encoding header in a particular manner. Such a reverse proxy is considered unlikely.
  + CVE-2020-12943
    - If an HTTP/2 client connecting to Apache Tomcat 10.0.0-M1 to 10.0.0-M7, 9.0.0.M1 to 9.0.37 or 8.5.0 to 8.5.57 exceeded the agreed maximum number of concurrent streams for a connection (in violation of the HTTP/2 protocol), it was possible that a subsequent request made on that connection could contain HTTP headers - including HTTP/2 pseudo headers - from a previous request rather than the intended headers. This could lead to users seeing responses for unexpected resources.
  + CVE-2023-28708
    - When using the RemoteIpFilter with requests received from a reverse proxy via HTTP that include the X-Forwarded-Proto header set to https, session cookies created by Apache Tomcat 11.0.0-M1 to 11.0.0.-M2, 10.1.0-M1 to 10.1.5, 9.0.0-M1 to 9.0.71 and 8.5.0 to 8.5.85 did not include the secure attribute. This could result in the user agent transmitting the session cookie over an insecure channel.
  + CVE-2021-43980
    - The simplified implementation of blocking reads and writes introduced in Tomcat 10 and back-ported to Tomcat 9.0.47 onwards exposed a long standing (but extremely hard to trigger) concurrency bug in Apache Tomcat 10.1.0 to 10.1.0-M12, 10.0.0-M1 to 10.0.18, 9.0.0-M1 to 9.0.60 and 8.5.0 to 8.5.77 that could cause client connections to share an Http11Processor instance resulting in responses, or part responses, to be received by the wrong client.
* Tomcat-embed-websocket-9.0.30.jar (Critical, CVE Count: 46, Evidence Count: 30)
  + 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.
  + CVE-2024-50379
    - Time-of-check Time-of-use (TOCTOU) Race Condition vulnerability during JSP compilation in Apache Tomcat permits an RCE on case insensitive file systems when the default servlet is enabled for write (non-default configuration).
  + CVE-2024-52316
    - Unchecked Error Condition vulnerability in Apache Tomcat. If Tomcat is configured to use a custom Jakarta Authentication (formerly JASPIC) ServerAuthContext component which may throw an exception during the authentication process without explicitly setting an HTTP status to indicate failure, the authentication may not fail, allowing the user to bypass the authentication process. There are no known Jakarta Authentication components that behave in this way.
  + CVE-2024-56337
    - Time-of-check Time-of-use (TOCTOU) Race Condition vulnerability in Apache Tomcat.
  + CVE-2025-24813
    - Path Equivalence: 'file.Name' (Internal Dot) leading to Remote Code Execution and/or Information disclosure and/or malicious content added to uploaded files via write enabled Default Servlet in Apache Tomcat.
  + CVE-2025-31651
    - Improper Neutralization of Escape, Meta, or Control Sequences vulnerability in Apache Tomcat. For a subset of unlikely rewrite rule configurations, it was possible for a specially crafted request to bypass some rewrite rules. If those rewrite rules effectively enforced security constraints, those constraints could be bypassed.
  + CVE-2025-49124
    - Untrusted Search Path vulnerability in Apache Tomcat installer for Windows. During installation, the Tomcat installer for Windows used icacls.exe without specifying a full path.
  + CVE-2020-11996
    - A specially crafted sequence of HTTP/2 requests sent to Apache Tomcat 10.0.0-M1 to 10.0.0-M5, 9.0.0.M1 to 9.0.35 and 8.5.0 to 8.5.55 could trigger high CPU usage for several seconds. If a sufficient number of such requests were made on concurrent HTTP/2 connections, the server could become unresponsive.
  + CVE-2020-12934
    - An h2c direct connection to Apache Tomcat 10.0.0-M1 to 10.0.0-M6, 9.0.0.M5 to 9.0.36 and 8.5.1 to 8.5.56 did not release the HTTP/1.1 processor after the upgrade to HTTP/2. If a sufficient number of such requests were made, an OutOfMemoryException could occur leading to a denial of service.
  + 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.
  + CVE-2020-11996
    - A specially crafted sequence of HTTP/2 requests sent to Apache Tomcat 10.0.0-M1 to 10.0.0-M5, 9.0.0.M1 to 9.0.35 and 8.5.0 to 8.5.55 could trigger high CPU usage for several seconds. If a sufficient number of such requests were made on concurrent HTTP/2 connections, the server could become unresponsive.
  + CVE-2020-13934
    - An h2c direct connection to Apache Tomcat 10.0.0-M1 to 10.0.0-M6, 9.0.0.M5 to 9.0.36 and 8.5.1 to 8.5.56 did not release the HTTP/1.1 processor after the upgrade to HTTP/2. If a sufficient number of such requests were made, an OutOfMemoryException could occur leading to a denial of service.
  + CVE-2020-17527
    - While investigating bug 64830 it was discovered that Apache Tomcat 10.0.0-M1 to 10.0.0-M9, 9.0.0-M1 to 9.0.39 and 8.5.0 to 8.5.59 could re-use an HTTP request header value from the previous stream received on an HTTP/2 connection for the request associated with the subsequent stream. While this would most likely lead to an error and the closure of the HTTP/2 connection, it is possible that information could leak between requests.
  + CVE-2021-25122
    - When responding to new h2c connection requests, Apache Tomcat versions 10.0.0-M1 to 10.0.0, 9.0.0.M1 to 9.0.41 and 8.5.0 to 8.5.61 could duplicate request headers and a limited amount of request body from one request to another meaning user A and user B could both see the results of user A's request.
  + CVE-2021-41079
    - Apache Tomcat 8.5.0 to 8.5.63, 9.0.0-M1 to 9.0.43 and 10.0.0-M1 to 10.0.2 did not properly validate incoming TLS packets. When Tomcat was configured to use NIO+OpenSSL or NIO2+OpenSSL for TLS, a specially crafted packet could be used to trigger an infinite loop resulting in a denial of service.
  + CVE-2022-29885
    - The documentation of Apache Tomcat 10.1.0-M1 to 10.1.0-M14, 10.0.0-M1 to 10.0.20, 9.0.13 to 9.0.62 and 8.5.38 to 8.5.78 for the EncryptInterceptor incorrectly stated it enabled Tomcat clustering to run over an untrusted network. This was not correct. While the EncryptInterceptor does provide confidentiality and integrity protection, it does not protect against all risks associated with running over any untrusted network, particularly DoS risks.
  + CVE-2022-42252
    - If Apache Tomcat 8.5.0 to 8.5.82, 9.0.0-M1 to 9.0.67, 10.0.0-M1 to 10.0.26 or 10.1.0-M1 to 10.1.0 was configured to ignore invalid HTTP headers via setting rejectIllegalHeader to false (the default for 8.5.x only), Tomcat did not reject a request containing an invalid Content-Length header making a request smuggling attack possible if Tomcat was located behind a reverse proxy that also failed to reject the request with the invalid header.
  + CVE-2023-44487
    - The HTTP/2 protocol allows a denial of service (server resource consumption) because request cancellation can reset many streams quickly, as exploited in the wild in August through October 2023.
  + CVE-2023-46589
    - Improper Input Validation vulnerability in Apache Tomcat.Tomcat from 11.0.0-M1 through 11.0.0-M10, from 10.1.0-M1 through 10.1.15, from 9.0.0-M1 through 9.0.82 and from 8.5.0 through 8.5.95 did not correctly parse HTTP trailer headers. A trailer header that exceeded the header size limit could cause Tomcat to treat a single request as multiple requests leading to the possibility of request smuggling when behind a reverse proxy.
  + CVE-2024-24549
    - Denial of Service due to improper input validation vulnerability for HTTP/2 requests in Apache Tomcat. When processing an HTTP/2 request, if the request exceeded any of the configured limits for headers, the associated HTTP/2 stream was not reset until after all of the headers had been processed.This issue affects Apache Tomcat: from 11.0.0-M1 through 11.0.0-M16, from 10.1.0-M1 through 10.1.18, from 9.0.0-M1 through 9.0.85, from 8.5.0 through 8.5.98.
  + CVE-2024-34750
    - Improper Handling of Exceptional Conditions, Uncontrolled Resource Consumption vulnerability in Apache Tomcat. When processing an HTTP/2 stream, Tomcat did not handle some cases of excessive HTTP headers correctly. This led to a miscounting of active HTTP/2 streams which in turn led to the use of an incorrect infinite timeout which allowed connections to remain open which should have been closed.
  + CVE-2024-38286
    - Allocation of Resources Without Limits or Throttling vulnerability in Apache Tomcat.
  + CVE-2025-48988
    - Allocation of Resources Without Limits or Throttling vulnerability in Apache Tomcat.
  + CVE-2025-48989
    - Improper Resource Shutdown or Release vulnerability in Apache Tomcat made Tomcat vulnerable to the made you reset attack.
  + CVE-2025-49125
    - Authentication Bypass Using an Alternate Path or Channel vulnerability in Apache Tomcat.  When using PreResources or PostResources mounted other than at the root of the web application, it was possible to access those resources via an unexpected path. That path was likely not to be protected by the same security constraints as the expected path, allowing those security constraints to be bypassed.
  + CVE-2025-52434
    - Concurrent Execution using Shared Resource with Improper Synchronization ('Race Condition') vulnerability in Apache Tomcat when using the APR/Native connector. This was particularly noticeable with client initiated closes of HTTP/2 connections.
  + CVE-2025-52520
    - For some unlikely configurations of multipart upload, an Integer Overflow vulnerability in Apache Tomcat could lead to a DoS via bypassing of size limits.
  + CVE-2025-53506
    - Uncontrolled Resource Consumption vulnerability in Apache Tomcat if an HTTP/2 client did not acknowledge the initial settings frame that reduces the maximum permitted concurrent streams.
  + CVE-2025-46701
    - Improper Handling of Case Sensitivity vulnerability in Apache Tomcat's GCI servlet allows security constraint bypass of security constraints that apply to the pathInfo component of a URI mapped to the CGI servlet.
  + CVE-2020-9484
    - When using Apache Tomcat versions 10.0.0-M1 to 10.0.0-M4, 9.0.0.M1 to 9.0.34, 8.5.0 to 8.5.54 and 7.0.0 to 7.0.103 if a) an attacker is able to control the contents and name of a file on the server; and b) the server is configured to use the PersistenceManager with a FileStore; and c) the PersistenceManager is configured with sessionAttributeValueClassNameFilter="null"
  + CVE-2021-25329
    - The fix for CVE-2020-9484 was incomplete. When using Apache Tomcat 10.0.0-M1 to 10.0.0, 9.0.0.M1 to 9.0.41, 8.5.0 to 8.5.61 or 7.0.0. to 7.0.107 with a configuration edge case that was highly unlikely to be used, the Tomcat instance was still vulnerable to CVE-2020-9494. Note that both the previously published prerequisites for CVE-2020-9484 and the previously published mitigations for CVE-2020-9484 also apply to this issue.
  + CVE-2021-30640
    - A vulnerability in the JNDI Realm of Apache Tomcat allows an attacker to authenticate using variations of a valid user name and/or to bypass some of the protection provided by the LockOut Realm. This issue affects Apache Tomcat 10.0.0-M1 to 10.0.5; 9.0.0.M1 to 9.0.45; 8.5.0 to 8.5.65.
  + CVE-2025-55668
    - Session Fixation vulnerability in Apache Tomcat via rewrite valve.
  + CVE-2024-23672
    - Denial of Service via incomplete cleanup vulnerability in Apache Tomcat. It was possible for WebSocket clients to keep WebSocket connections open leading to increased resource consumption.This issue affects Apache Tomcat: from 11.0.0-M1 through 11.0.0-M16, from 10.1.0-M1 through 10.1.18, from 9.0.0-M1 through 9.0.85, from 8.5.0 through 8.5.98.
  + CVE-2022-34305
    - In Apache Tomcat 10.1.0-M1 to 10.1.0-M16, 10.0.0-M1 to 10.0.22, 9.0.30 to 9.0.64 and 8.5.50 to 8.5.81 the Form authentication example in the examples web application displayed user provided data without filtering, exposing a XSS vulnerability.
  + CVE-2023-41080
    - URL Redirection to Untrusted Site ('Open Redirect') vulnerability in FORM authentication feature Apache Tomcat.This issue affects Apache Tomcat: from 11.0.0-M1 through 11.0.0-M10, from 10.1.0-M1 through 10.0.12, from 9.0.0-M1 through 9.0.79 and from 8.5.0 through 8.5.92. Older, EOL versions may also be affected.
  + CVE-2021-24122
    - When serving resources from a network location using the NTFS file system, Apache Tomcat versions 10.0.0-M1 to 10.0.0-M9, 9.0.0.M1 to 9.0.39, 8.5.0 to 8.5.59 and 7.0.0 to 7.0.106 were susceptible to JSP source code disclosure in some configurations. The root cause was the unexpected behaviour of the JRE API File.getCanonicalPath() which in turn was caused by the inconsistent behaviour of the Windows API (FindFirstFileW) in some circumstances.
  + CVE-2021-33037
    - 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 honoured the identify encoding; and - Tomcat did not ensure that, if present, the chunked encoding was the final encoding.
  + CVE-2023-42795
    - incomplete Cleanup vulnerability in Apache Tomcat.When recycling various internal objects in Apache Tomcat from 11.0.0-M1 through 11.0.0-M11, from 10.1.0-M1 through 10.1.13, from 9.0.0-M1 through 9.0.80 and from 8.5.0 through 8.5.93, an error could cause Tomcat to skip some parts of the recycling process leading to information leaking from the current request/response to the next. Older, EOL versions may also be affected.
  + CVE-2023-45648
    - Improper Input Validation vulnerability in Apache Tomcat.Tomcat from 11.0.0-M1 through 11.0.0-M11, from 10.1.0-M1 through 10.1.13, from 9.0.0-M1 through 9.0.81 and from 8.5.0 through 8.5.93 did not correctly parse HTTP trailer headers. A specially crafted, invalid trailer header could cause Tomcat to treat a single request as multiple requests leading to the possibility of request smuggling when behind a reverse proxy.
  + CVE-2024-21733
    - Generation of Error Message Containing Sensitive Information vulnerability in Apache Tomcat.This issue affects Apache Tomcat: from 8.5.7 through 8.5.63, from 9.0.0-M11 through 9.0.43.
  + CVE-2024-54677
    - Uncontrolled Resource Consumption vulnerability in the examples web application provided with Apache Tomcat leads to denial of service.
  + CVE-2019-17569
    - The refactoring present in Apache Tomcat 9.0.28 to 9.0.30, 8.5.48 to 8.5.50 and 7.0.98 to 7.0.99 introduced a regression. The result of the regression was that invalid Transfer-Encoding headers were incorrectly processed leading to a possibility of HTTP Request Smuggling if Tomcat was located behind a reverse proxy that incorrectly handled the invalid Transfer-Encoding header in a particular manner. Such a reverse proxy is considered unlikely.
  + CVE-2020-1935
    - In Apache Tomcat 9.0.0.M1 to 9.0.30, 8.5.0 to 8.5.50 and 7.0.0 to 7.0.99 the HTTP header parsing code used an approach to end-of-line parsing that allowed some invalid HTTP headers to be parsed as valid. This led to a possibility of HTTP Request Smuggling if Tomcat was located behind a reverse proxy that incorrectly handled the invalid Transfer-Encoding header in a particular manner. Such a reverse proxy is considered unlikely.
  + CVE-2020-13943
    - if an HTTP/2 client connecting to Apache Tomcat 10.0.0-M1 to 10.0.0-M7, 9.0.0.M1 to 9.0.37 or 8.5.0 to 8.5.57 exceeded the agreed maximum number of concurrent streams for a connection (in violation of the HTTP/2 protocol), it was possible that a subsequent request made on that connection could contain HTTP headers - including HTTP/2 pseudo headers - from a previous request rather than the intended headers. This could lead to users seeing responses for unexpected resources.
  + CVE-2023-28708
    - When using the RemoteIpFilter with requests received from a reverse proxy via HTTP that include the X-Forwarded-Proto header set to https, session cookies created by Apache Tomcat 11.0.0-M1 to 11.0.0.-M2, 10.1.0-M1 to 10.1.5, 9.0.0-M1 to 9.0.71 and 8.5.0 to 8.5.85 did not include the secure attribute. This could result in the user agent transmitting the session cookie over an insecure channel.
  + CVE-2021-43980
    - The simplified implementation of blocking reads and writes introduced in Tomcat 10 and back-ported to Tomcat 9.0.47 onwards exposed a long standing (but extremely hard to trigger) concurrency bug in Apache Tomcat 10.1.0 to 10.1.0-M12, 10.0.0-M1 to 10.0.18, 9.0.0-M1 to 9.0.60 and 8.5.0 to 8.5.77 that could cause client connections to share an Http11Processor instance resulting in responses, or part responses, to be received by the wrong client.

**5. Mitigation Plan**

Interpret the results from the manual review and static testing report. Then identify the steps to mitigate the identified security vulnerabilities for Artemis Financial’s software application.

These are the issues I found when looking at the code along with the suggested fixes.

1. There is no input validation or limitation on the setMyDateTime function. This is could lead to potential input abuse if not resolved.   
     
   The fix for this would be to properly validate the input and limit the type of data the user is able to input.
2. The POM file is attempting to use version 5.3.0 for the Maven dependency check. This is an out of date file and has been updated to 12.1.0   
     
   This was already fixed as a part of using the POM file to run the dependency check. The fix was updating to 12.1.0.
3. The Pom file has a suppression file that prevented the dependency check from running. This had to be removed for the check to run.   
     
   This has already been fixed by removing the suppression line from the POM file.
4. The Pom file was also trying to use an old version of Java 1.8 and needed to be changed to 11.   
     
   This has already been fixed by referencing Java 11 instead of 1.8
5. The Customer class has the account\_balance set to public. This should be set to private to ensure it is secure.   
     
   The fix for this is to change it from public to private to ensure it remains secure.
6. The customer class also returns the customers account number with the showInfo() function. This should be considered PII and needs to be protected.   
     
   A review needs to be done to ensure that the account number needs to be returned as part of this function. It would be more secure to return the last 4-5 digits of the account number.
7. The deposit function is set to an int without limitations. This could lead to depositing negative funds into the account, no way to handle anything less than $1, and the possibility of an overflow input.   
     
   A good way to fix this would be to change the type of input from int to float, restrict the use to negative numbers, and restrict the total length of the input.
8. There is no encryption on the jdbc URL and that would cause plaintext database traffic   
     
   This should be encrypted with TLS1.3 to ensure that communicaiton with the databse is not in plaintext.
9. There are very few comments. This could lead to issues with other developers in the future and can cause confusion.   
     
   A good fix would be to add lots of detailed descriptions to the code to ensure that the current development team and any future development team knows exactly how everything should function as well any future work planned.

There are very few comments. This could lead to issues with other developers in the future and can cause confusion.

The following static testing report vulnerabilities should be corrected by the following corrective actions.

* Hibernate-validator-6.0.18.final.jar
  + This vulnerability can be resolved by updating to a minimum version of 6.2.x
* Jackson-core-2.10.2.jar
  + This vulnerability can be resolved by updating to a minimum version of 2.15+ but 2.18.x is recommended.
* Jackson-databind-2.10.2.jar
  + This vulnerability can be resolved by updating to a minimum version of 2.15+ but 2.18.x is recommended.
* Json-path2.4.0.jar
  + This vulnerability can be resolved by updating to a minimum version of 2.9.0+
* Json-smart-2.3.jar
  + This vulnerability can be resolved by updating to a minimum version of 2.5.2+
* Log4j-api-2.12.1.jar
  + This vulnerability can be resolved by updating to a minimum version of 2.17.2+
* Logback-classic-1.2.3.jar
  + This vulnerability can be resolved by updating to a minimum version of 1.5.13+
* Logback-core-1.2.3.jar
  + This vulnerability can be resolved by updating to a minimum version of 1.5.13+
* Snakeyaml-1.25.jar
  + This vulnerability can be resolved by updating to a minimum version of 2.0+
* Spring-boot-2.2.4.RELEASE.jar
  + This vulnerability can be resolved by updating to a minimum version of 2.7.x
* Spring-boot-starter-web-2.2.4.RELEASE.jar
  + This vulnerability can be resolved by updating to a minimum version of 2.7.x
* Spring-context-5.2.3.RELEASE.jar
  + This vulnerability can be resolved by updating to a minimum version of 5.2.20+
* Spring-core-5.2.3.RELEASE.jar
  + This vulnerability can be resolved by updating to a minimum version of 5.2.20+
* Spring-data-rest-webmvc-3.2.4.RELEASE.jar
  + This vulnerability can be resolved by updating to a minimum version of 3.6.7+
* Spring-expression-5.2.3.RELEASE.jar
  + This vulnerability can be resolved by updating to a minimum version of 5.2.20+
* Spring-hateoas-1.0.3.RELEASE.jar
  + This vulnerability can be resolved by updating to a minimum version of 1.5.5+
* Spring-tx-5.2.3.RELEASE.jar
  + This vulnerability can be resolved by updating to a minimum version of 5.2.20+
* Spring-web-5.2.3.RELEASE.jar
  + This vulnerability can be resolved by updating to a minimum version of 5.2.20+
* Spring-webmvc-5.2.3.RELEASE.jar
  + This vulnerability can be resolved by updating to a minimum version of 5.2.20+
* Tomcat-embed-core-9.0.30.jar
  + This vulnerability can be resolved by updating to a minimum version of 9.0.99+
* Tomcat-embed-websocket-9.0.30.jar
  + This vulnerability can be resolved by updating to a minimum version of 9.0.99+

Sources:

*Credential stuffing*. OWASP Foundation. (n.d.-a). https://owasp.org/www-community/attacks/Credential\_stuffing

*Gramm-Leach-Bliley Act (GLBA): Comprehensive guide*. GrcTimes. (2025, July 29). https://grctimes.com/gramm-leach-bliley-act-glba/#:~:text=Learn%20about%20the%20Gramm-Leach-Bliley%20Act%20%28GLBA%29%3A%20key%20privacy,compliance%20strategies%2C%20recent%20updates%2C%20and%20practical%20challenges%20.

*Gramm-Leach-Bliley Act*. Federal Trade Commission. (2025, June 17). <https://www.ftc.gov/business-guidance/privacy-security/gramm-leach-bliley-act>

GeeksforGeeks. (2025b, July 23). *What is input validation attack?*. GeeksforGeeks. https://www.geeksforgeeks.org/ethical-hacking/what-is-input-validation-attack/

Lindemulder, G., & Kosinski, M. (2025, September 16). *What is a man-in-the-middle (MITM) attack?*. IBM. https://www.ibm.com/think/topics/man-in-the-middle

*PIPEDA requirements in brief*. Office of the Privacy Commissioner of Canada. (2024, May 1). https://www.priv.gc.ca/en/privacy-topics/privacy-laws-in-canada/the-personal-information-protection-and-electronic-documents-act-pipeda/pipeda\_brief/

*Session hijacking attack*. OWASP Foundation. (n.d.). https://owasp.org/www-community/attacks/Session\_hijacking\_attack

*What is a zero-day exploit?*. IBM. (2025, September 16). https://www.ibm.com/think/topics/zero-day

*What is an API (application programming interface)*. GeeksforGeeks. (2025d, July 21). https://www.geeksforgeeks.org/software-testing/what-is-an-api/

*What is an injection attack?*. CrowdStrike. (n.d.). https://www.crowdstrike.com/en-us/cybersecurity-101/cyberattacks/injection-attack/

Wolford, B. (2025, August 19). *What is GDPR, the EU’s new Data Protection Law?* GDPR.eu. https://gdpr.eu/what-is-gdpr/