****

# Artemis Financial Vulnerability Assessment Report

Table of Contents

[Document Revision History 3](#_Toc32574607)

[Client 3](#_Toc32574608)

[Instructions 3](#_Toc32574609)

[Developer 4](#_Toc32574610)

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

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

[3. Manual Review 4](#_Toc32574613)

[4. Static Testing 4](#_Toc32574614)

[5. Mitigation Plan 4](#_Toc32574615)

**Document Revision History**

| **Version** | **Date** | **Author** | **Comments** |
| --- | --- | --- | --- |
| **1.0** | **09/13/2023** | **Frank Bolstad** | **GlobalRain Artemis Financial Vulnerability Assessment Report** |

## Client



## Developer

Frank Bolstad

## Interpreting Client Needs

We are Global Rain, a software engineering company that specializes in custom software design and development. We believe “Security is everyone’s responsibility.”

Artemis Financial is a client of ours. They are a consulting company that develops custom financial plans for their customers based on each customer’s specific wants and needs. These financial plans include savings, retirement, investment, and insurance plans.

Artemis Financial want their software to have the most up-to-date and effective software security. They would like our, Global Rain’s, expertise on how to protect their company and their software from external threats.

We will review Artemis Financials’ web-based software application to identify any security vulnerabilities. We will, then, document them and their remediations in a vulnerability assessment report.

Customers’ private information as well as their financial data is used and stored in the application and their databases, so secure communications are absolutely essential. With this being a web-based application, customers should be able to access their accounts from anywhere in the world, so the necessary regulations must be in place to comply with all policies that involve international transactions.

There are also many government restrictions Artemis must follow when it comes to handling customer’s money and data. Artemis’ application uses the RESTful web API, so all application requests and responses must be secure to protect the data from external threats.

One reason to modernize Java applications is to take full advantage of the cloud deployment model. A way to modernize the application is to refactor it which would gradually move functionality of the old application architecture to the new one. Red Hat OpenShift is a trusted platform that can assist with the entire duration of the application lifecycle. This platform allows developers to build, deploy, and operate applications with security as a main focus within a hybrid cloud.

## Areas of SecurityA computer screen shot of a diagram Description automatically generated

There are many potential areas of security weakness in applications. A vulnerability is a flaw or weakness in the system. Threats attempt to exploit vulnerabilities to access protected data.

Above is the Vulnerability Assessment Process Flow Diagram. I think all of the security processes on the diagram are critical depending on the situation. If these processes are followed, the application and/or system are protected from attacker’s malicious intent.

The areas of security I think are most relevant to this specific scenario are Input Validation, APIs, and Cryptography. Input Validation secures input and representations from a user, APIs secures application programming interfaces interactions, and Cryptography which provides encryption and identifies vulnerabilities.

Input Validation must be secure since the application will be accepting user input and that input must be validated.

The application must have secure API interactions since it uses RESTful API, which stands for representational state transfer, and it is an interface that two systems use to exchange information securely using the internet.

The third process I chose was Cryptography which encrypts data use and identifies vulnerabilities. Artemis is a financial organization that is in possession of user’s most private information, so encryption to keep that data secure as well as being able to identify and mitigate vulnerabilities will keep all of that data safe and secure.

## Manual Review

Refer to the seven security areas outlined in the Vulnerability Assessment Process Flow Diagram. Use what you’ve learned in steps 1 and 2 to guide your manual review. Identify all vulnerabilities in the Project One Code Base, linked in Supporting Materials, by manually inspecting the code. Document your findings in your vulnerability assessment report. Be sure to include a description that identifies where the vulnerabilities are found (specific class file, if applicable).

The next step is to manually review the code. The OWASP secure coding practices guide and the Java secure coding guidelines were used as references to examine and verify the code provided.

Many secure coding practices were overlooked in the development of this application’s code. They are as follows:

UTF-8 character set is not encoded and is not validated.

Client data is not validated.

‘business\_name’ parameter in CRUDController.java has input validation errors.

There is no character “white” list.

There is also no input sanitization.

No validation of data type, length, or range.

No HTML entity encoding.

No check for null bytes.

No check for new line.

No authentication.

Default authentication used for the test database.

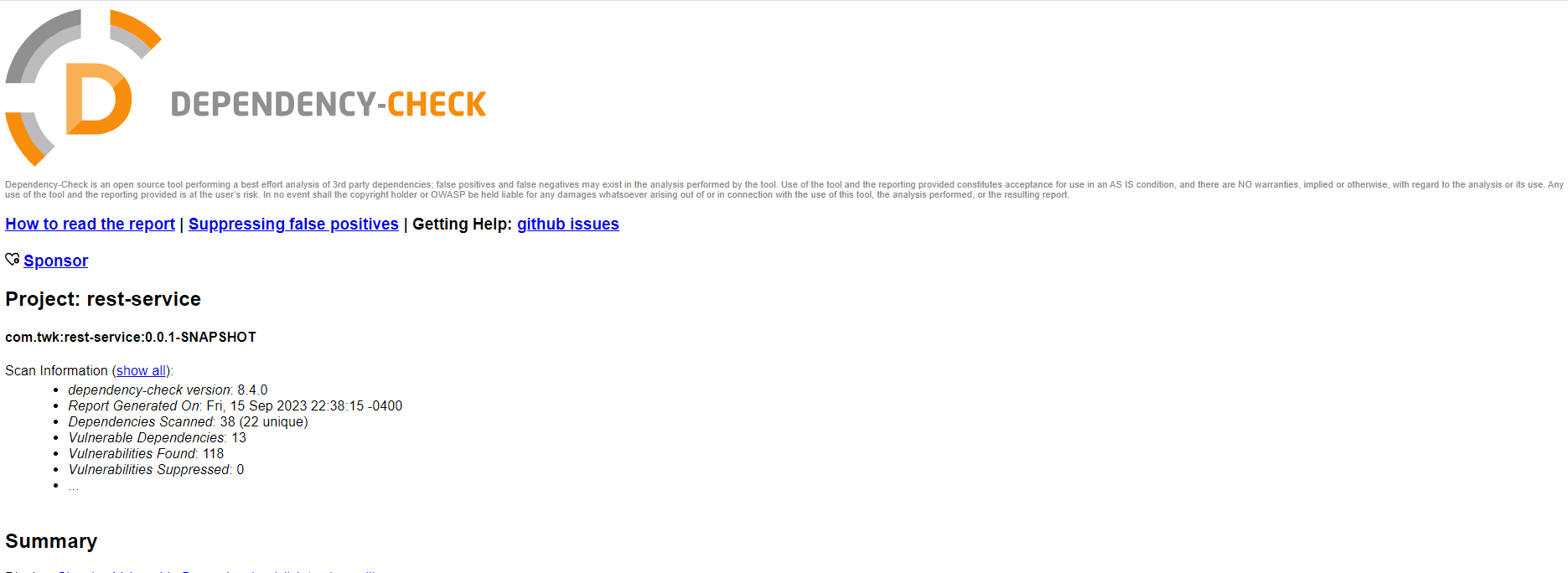
No authorization, for example: permission management.

No session management.

Sensitive information in exception messages (on line 21 of DocData.java).

Some classes could be final and could prohibit them being extended (CRUDController.java).

## Static Testing



To keep our client’s software secure and protected, we will use the Maven Dependency-Check to perform a static test, known as a dependency check. This test checks a project’s dependencies and any additional related dependency for any known published vulnerability.

A dependency check was performed on our client’s, Artemis Financial, software. As seen above in the dependency check report, we have 13 vulnerable dependencies found. We will define and describe them in a vulnerability assessment report.

**Vulnerability Assessment Report**

**Dependency**

[bcprov-jdk15on-1.46.jar](file:///C:\Users\310130710\OneDrive%20-%20Philips\Documents\Education\CS-305%20Software%20Security\Module%203\project%201\dependency-check-report.html#l1_991c96a4e31e6c19e2b9136c8955bd423f2dc4c7)

**Description**

The Bouncy Castle Crypto package is a Java implementation of cryptographic algorithms. This jar contains JCE provider and lightweight API for the Bouncy Castle Cryptography APIs for JDK 1.5 to JDK 1.7.

**Severity**

High

**Attribution**

CVE-2015-6644

Bouncy Castle in Android before 5.1.1 LMY49F and 6.0 before 2016-01-01 allows attackers to obtain sensitive information via a crafted application, aka internal bug 24106146.

CVE-2016-1000346

In the Bouncy Castle JCE Provider version 1.55 and earlier the other party DH public key is not fully validated. This can cause issues as invalid keys can be used to reveal details about the other party's private key where static Diffie-Hellman is in use. As of release 1.56 the key parameters are checked on agreement calculation.

CVE-2013-1624

The TLS implementation in the Bouncy Castle Java library before 1.48 and C# library before 1.8 does not properly consider timing side-channel attacks on a noncompliant MAC check operation during the processing of malformed CBC padding, which allows remote attackers to conduct distinguishing attacks and plaintext-recovery attacks via statistical analysis of timing data for crafted packets, a related issue to CVE-2013-0169.

CVE-2018-5382

The default BKS keystore use an HMAC that is only 16 bits long, which can allow an attacker to compromise the integrity of a BKS keystore. Bouncy Castle release 1.47 changes the BKS format to a format which uses a 160 bit HMAC instead. This applies to any BKS keystore generated prior to BC 1.47. For situations where people need to create the files for legacy reasons a specific keystore type "BKS-V1" was introduced in 1.49. It should be noted that the use of "BKS-V1" is discouraged by the library authors and should only be used where it is otherwise safe to do so, as in where the use of a 16 bit checksum for the file integrity check is not going to cause a security issue in itself.

CVE-2015-7940

The Bouncy Castle Java library before 1.51 does not validate a point is withing the elliptic curve, which makes it easier for remote attackers to obtain private keys via a series of crafted elliptic curve Diffie Hellman (ECDH) key exchanges, aka an "invalid curve attack."

CVE-2023-33201

Bouncy Castle For Java before 1.74 is affected by an LDAP injection vulnerability. The vulnerability only affects applications that use an LDAP CertStore from Bouncy Castle to validate X.509 certificates. During the certificate validation process, Bouncy Castle inserts the certificate's Subject Name into an LDAP search filter without any escaping, which leads to an LDAP injection vulnerability.

CVE-2020-26939

In Legion of the Bouncy Castle BC before 1.61 and BC-FJA before 1.0.1.2, attackers can obtain sensitive information about a private exponent because of Observable Differences in Behavior to Error Inputs. This occurs in org.bouncycastle.crypto.encodings.OAEPEncoding. Sending invalid ciphertext that decrypts to a short payload in the OAEP Decoder could result in the throwing of an early exception, potentially leaking some information about the private exponent of the RSA private key performing the encryption.

CVE-2016-1000339

In the Bouncy Castle JCE Provider version 1.55 and earlier the primary engine class used for AES was AESFastEngine. Due to the highly table driven approach used in the algorithm it turns out that if the data channel on the CPU can be monitored the lookup table accesses are sufficient to leak information on the AES key being used. There was also a leak in AESEngine although it was substantially less. AESEngine has been modified to remove any signs of leakage (testing carried out on Intel X86-64) and is now the primary AES class for the BC JCE provider from 1.56. Use of AESFastEngine is now only recommended where otherwise deemed appropriate.

CVE-2020-0187

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

CVE-2020-15522

Bouncy Castle BC Java before 1.66, BC C# .NET before 1.8.7, BC-FJA before 1.0.1.2, 1.0.2.1, and BC-FNA before 1.0.1.1 have a timing issue within the EC math library that can expose information about the private key when an attacker is able to observe timing information for the generation of multiple deterministic ECDSA signatures.

CVE-2017-13098

BouncyCastle TLS prior to version 1.0.3, when configured to use the JCE (Java Cryptography Extension) for cryptographic functions, provides a weak Bleichenbacher oracle when any TLS cipher suite using RSA key exchange is negotiated. An attacker can recover the private key from a vulnerable application. This vulnerability is referred to as "ROBOT."

CVE-2016-1000345

In the Bouncy Castle JCE Provider version 1.55 and earlier the DHIES/ECIES CBC mode vulnerable to padding oracle attack. For BC 1.55 and older, in an environment where timings can be easily observed, it is possible with enough observations to identify when the decryption is failing due to padding.

CVE-2016-1000341

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

CVE-2016-1000352

In the Bouncy Castle JCE Provider version 1.55 and earlier the ECIES implementation allowed the use of ECB mode. This mode is regarded as unsafe and support for it has been removed from the provider.

CVE-2016-1000344

In the Bouncy Castle JCE Provider version 1.55 and earlier the DHIES implementation allowed the use of ECB mode. This mode is regarded as unsafe and support for it has been removed from the provider.

CVE-2016-1000343

In the Bouncy Castle JCE Provider version 1.55 and earlier the DSA key pair generator generates a weak private key if used with default values. If the JCA key pair generator is not explicitly initialised with DSA parameters, 1.55 and earlier generates a private value assuming a 1024 bit key size. In earlier releases this can be dealt with by explicitly passing parameters to the key pair generator.

CVE-2016-1000342

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

CVE-2016-1000338

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

**Dependency**

[hibernate-validator-6.0.18.Final.jar](file:///C:\Users\310130710\OneDrive%20-%20Philips\Documents\Education\CS-305%20Software%20Security\Module%203\project%201\dependency-check-report.html#l3_7fd00bcd87e14b6ba66279282ef15efa30dd2492)

**Description**

Hibernate's Bean Validation (JSR-380) reference implementation.

**Severity**

Medium

**Attribution**

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.

**Dependency**

[jackson-databind-2.10.2.jar](file:///C:\Users\310130710\OneDrive%20-%20Philips\Documents\Education\CS-305%20Software%20Security\Module%203\project%201\dependency-check-report.html#l5_0528de95f198afafbcfb0c09d2e43b6e0ea663ec)

**Description**

General data-binding functionality for Jackson: works on core streaming API

**Severity**

High

**Attribution**

CVE-2023-35116

\*\* DISPUTED \*\* 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.

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-2022-42003

In FasterXML jackson-databind before 2.14.0-rc1, 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. Additional fix version in 2.13.4.1 and 2.12.17.1

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-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-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.

**Dependency**

[log4j-api-2.12.1.jar](file:///C:\Users\310130710\OneDrive%20-%20Philips\Documents\Education\CS-305%20Software%20Security\Module%203\project%201\dependency-check-report.html#l10_a55e6d987f50a515c9260b0451b4fa217dc539cb)

**Description**

The Apache Log4j API

**Severity**

Low

**Dependency**

[logback-core-1.2.3.jar](file:///C:\Users\310130710\OneDrive%20-%20Philips\Documents\Education\CS-305%20Software%20Security\Module%203\project%201\dependency-check-report.html#l12_864344400c3d4d92dfeb0a305dc87d953677c03c)

**Description**

logback-core module

**Severity**

Medium

**Attribution**

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.

**Dependency**

[snakeyaml-1.25.jar](file:///C:\Users\310130710\OneDrive%20-%20Philips\Documents\Education\CS-305%20Software%20Security\Module%203\project%201\dependency-check-report.html#l14_8b6e01ef661d8378ae6dd7b511a7f2a33fae1421)

**Description**

YAML 1.1 parser and emitter for Java

**Severity**

Critical

**Attribution**

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.

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-2022-38752

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-2022-38751

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-2022-38749

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-2022-25857

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-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-2022-1471

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.

**Dependency**

[spring-boot-2.2.4.RELEASE.jar](file:///C:\Users\310130710\OneDrive%20-%20Philips\Documents\Education\CS-305%20Software%20Security\Module%203\project%201\dependency-check-report.html#l15_225a4fd31156c254e3bb92adb42ee8c6de812714)

**Description**

Spring Boot

**Severity**

Critical

**Attribution**

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.

CVE-2022-27772

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

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+.

**Dependency**

[spring-boot-starter-web-2.2.4.RELEASE.jar](file:///C:\Users\310130710\OneDrive%20-%20Philips\Documents\Education\CS-305%20Software%20Security\Module%203\project%201\dependency-check-report.html#l16_ec75d01d212b5229c16d872fb127744c0ed46ed8)

**Description**

Starter for building web, including RESTful, applications using Spring MVC. Uses Tomcat as the default embedded container

**Severity**

Critical

**Attribution**

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.

CVE-2022-27772

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

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+.

**Dependency**

[spring-core-5.2.3.RELEASE.jar](file:///C:\Users\310130710\OneDrive%20-%20Philips\Documents\Education\CS-305%20Software%20Security\Module%203\project%201\dependency-check-report.html#l17_3734223040040e8c3fecd5faa3ae8a1ed6da146b)

**Description**

Spring Core

**Severity**

Critical

**Attribution**

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-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-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-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-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-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-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-2022-22950

n 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-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-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-2022-22965

**CISA Known Exploited Vulnerability:**

* Product: VMware Spring Framework
* Name: Spring Framework JDK 9+ Remote Code Execution Vulnerability
* Date Added: 2022-04-04
* Description: Spring MVC or Spring WebFlux application running on JDK 9+ may be vulnerable to remote code execution (RCE) via data binding.
* Required Action: Apply updates per vendor instructions.
* Due Date: 2022-04-25

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.

**Dependency**

[spring-web-5.2.3.RELEASE.jar](file:///C:\Users\310130710\OneDrive%20-%20Philips\Documents\Education\CS-305%20Software%20Security\Module%203\project%201\dependency-check-report.html#l18_dd386a02e40b915ab400a3bf9f586d2dc4c0852c)

**Description**

Spring Web

**Severity**

Critical

**Attribution**

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-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-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-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-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-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-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-2022-22950

n 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-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-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-2022-22965

**CISA Known Exploited Vulnerability:**

* Product: VMware Spring Framework
* Name: Spring Framework JDK 9+ Remote Code Execution Vulnerability
* Date Added: 2022-04-04
* Description: Spring MVC or Spring WebFlux application running on JDK 9+ may be vulnerable to remote code execution (RCE) via data binding.
* Required Action: Apply updates per vendor instructions.
* Due Date: 2022-04-25

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-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.

**Dependency**

[spring-webmvc-5.2.3.RELEASE.jar](file:///C:\Users\310130710\OneDrive%20-%20Philips\Documents\Education\CS-305%20Software%20Security\Module%203\project%201\dependency-check-report.html#l19_745a62502023d2496b565b7fe102bb1ee229d6b7)

**Description**

Spring Web MVC

**Severity**

Critical

**Attribution**

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-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-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-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-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-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-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-2022-22950

n 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-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-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-2022-22965

**CISA Known Exploited Vulnerability:**

* Product: VMware Spring Framework
* Name: Spring Framework JDK 9+ Remote Code Execution Vulnerability
* Date Added: 2022-04-04
* Description: Spring MVC or Spring WebFlux application running on JDK 9+ may be vulnerable to remote code execution (RCE) via data binding.
* Required Action: Apply updates per vendor instructions.
* Due Date: 2022-04-25

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.

**Dependency**

[tomcat-embed-core-9.0.30.jar](file:///C:\Users\310130710\OneDrive%20-%20Philips\Documents\Education\CS-305%20Software%20Security\Module%203\project%201\dependency-check-report.html#l20_ad32909314fe2ba02cec036434c0addd19bcc580)

**Description**

Core Tomcat implementation

**Severity**

Critical

**Attribution**

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.

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-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-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-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-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-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-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.

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-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-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-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" (the default unless a SecurityManager is used) or a sufficiently lax filter to allow the attacker provided object to be deserialized; and d) the attacker knows the relative file path from the storage location used by FileStore to the file the attacker has control over; then, using a specifically crafted request, the attacker will be able to trigger remote code execution via deserialization of the file under their control. Note that all of conditions a) to d) must be true for the attack to succeed.

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-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-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-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-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-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-2020-13934

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-2020-11996

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-2020-1938

**CISA Known Exploited Vulnerability:**

* Product: Apache Tomcat
* Name: Apache Tomcat Improper Privilege Management Vulnerability
* Date Added: 2022-03-03
* Description: Apache Tomcat treats Apache JServ Protocol (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.
* Required Action: Apply updates per vendor instructions.
* Due Date: 2022-03-17

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

**Dependency**

[tomcat-embed-websocket-9.0.30.jar](file:///C:\Users\310130710\OneDrive%20-%20Philips\Documents\Education\CS-305%20Software%20Security\Module%203\project%201\dependency-check-report.html#l22_33157f6bc5bfd03380ebb5ac476db0600a04168d)

**Description**

Core Tomcat implementation

**Severity**

Critical

**Attribution**

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.

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-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-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-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-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-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-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.

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-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-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-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" (the default unless a SecurityManager is used) or a sufficiently lax filter to allow the attacker provided object to be deserialized; and d) the attacker knows the relative file path from the storage location used by FileStore to the file the attacker has control over; then, using a specifically crafted request, the attacker will be able to trigger remote code execution via deserialization of the file under their control. Note that all of conditions a) to d) must be true for the attack to succeed.

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-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-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-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-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-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-2020-13934

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-2020-11996

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-2020-8022

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

CVE-2020-1938

**CISA Known Exploited Vulnerability:**

* Product: Apache Tomcat
* Name: Apache Tomcat Improper Privilege Management Vulnerability
* Date Added: 2022-03-03
* Description: Apache Tomcat treats Apache JServ Protocol (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.
* Required Action: Apply updates per vendor instructions.
* Due Date: 2022-03-17

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

## Mitigation Plan

Interpret the results from the manual review and static testing report. Identify steps to mitigate the identified security vulnerabilities by creating an action list that documents how to fix each vulnerability in your vulnerability assessment report.

Action list

**Dependency**

[bcprov-jdk15on-1.46.jar](file:///C:\Users\310130710\OneDrive%20-%20Philips\Documents\Education\CS-305%20Software%20Security\Module%203\project%201\dependency-check-report.html#l1_991c96a4e31e6c19e2b9136c8955bd423f2dc4c7)

**Description**

The Bouncy Castle Crypto package is a Java implementation of cryptographic algorithms. This jar contains JCE provider and lightweight API for the Bouncy Castle Cryptography APIs for JDK 1.5 to JDK 1.7.

**Remediation**

Red Hat Fuse, based on Apache ServiceMix, provides a small-footprint, flexible, open source enterprise service bus and integration platform. This release of Red Hat Fuse 7.1 serves as a replacement for Red Hat Fuse 7.0, and includes bug fixes and enhancements, which are documented in the Release Notes document linked to in the References.

**Dependency**

[hibernate-validator-6.0.18.Final.jar](file:///C:\Users\310130710\OneDrive%20-%20Philips\Documents\Education\CS-305%20Software%20Security\Module%203\project%201\dependency-check-report.html#l3_7fd00bcd87e14b6ba66279282ef15efa30dd2492)

**Description**

Hibernate's Bean Validation (JSR-380) reference implementation.

**Problem**

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.

**Remediation**

Fixed In Version: hibernate-validator 7.0.0.Alpha2, hibernate-validator 6.1.5.Final, hibernate-validator 6.0.20.Final

**Dependency**

[jackson-databind-2.10.2.jar](file:///C:\Users\310130710\OneDrive%20-%20Philips\Documents\Education\CS-305%20Software%20Security\Module%203\project%201\dependency-check-report.html#l5_0528de95f198afafbcfb0c09d2e43b6e0ea663ec)

**Description**

General data-binding functionality for Jackson: works on core streaming API

**Problem**

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.

**Remediation**

Fixed In Version: jackson-databind-2.11.0, jackson-databind-2.10.5.1

**Dependency**

[log4j-api-2.12.1.jar](file:///C:\Users\310130710\OneDrive%20-%20Philips\Documents\Education\CS-305%20Software%20Security\Module%203\project%201\dependency-check-report.html#l10_a55e6d987f50a515c9260b0451b4fa217dc539cb)

**Description**

The Apache Log4j API

**Problem**

The SmtpAppender should be able to use an SSL configuration element to specify a trust store, host name verification, and a key store, so that smtps connections can be further configured. This should re-use the same <SSL/> configuration element that's used elsewhere like HttpAppender.

**Remediation**

Upgrade to 2.13.2 which supports this feature. Previous versions can set the system property mail.smtp.ssl.checkserveridentity to true to globally enable hostname verification for SMTPS connections.

**Dependency**

[logback-core-1.2.3.jar](file:///C:\Users\310130710\OneDrive%20-%20Philips\Documents\Education\CS-305%20Software%20Security\Module%203\project%201\dependency-check-report.html#l12_864344400c3d4d92dfeb0a305dc87d953677c03c)

**Description**

logback-core module

**Remediation**

This issue is fixed and auto-remediated. Cloud Manager instances with the Auto-upgrade setting enabled should have updated on 20220102 to the fixed version 3.9.14 build4. No further action isrequired. Cloud Manager instances with the Auto-upgrade setting disabled should follow the steps below to enable it to receive the fixed version. 1. Connect to the Cloud Manager command line using SSH. 2. Run "sudo -s". 3. Navigate to the app.conf location - note that the file could be in varying locations per Cloud Manager version. Use “find . -name app.conf” to locate it. 4. Create a backup of the app.conf file. 5. Edit the app.conf file and change the "auto-upgrade" flag from "false" to "true". 6. Save the file and then restart the Cloud Manager instance. 7. Once the instance is back up visit the Cloud Manager UI to confirm that the version is now 3.9.14 or higher.

**Dependency**

[snakeyaml-1.25.jar](file:///C:\Users\310130710\OneDrive%20-%20Philips\Documents\Education\CS-305%20Software%20Security\Module%203\project%201\dependency-check-report.html#l14_8b6e01ef661d8378ae6dd7b511a7f2a33fae1421)

**Description**

YAML 1.1 parser and emitter for Java

**Problem**

Multiple NetApp products incorporate SnakeYAML. SnakeYAML versions prior to 2.0 are susceptible to a vulnerability which when successfully exploited could lead to disclosure of sensitive information, addition or modification of data, or Denial of Service (DoS).

Successful exploitation of this vulnerability could lead to disclosure of sensitive information, addition or modification of data, or Denial of Service (DoS).

**Remediation**

None at this moment according to netapp

**Dependency**

[spring-boot-2.2.4.RELEASE.jar](file:///C:\Users\310130710\OneDrive%20-%20Philips\Documents\Education\CS-305%20Software%20Security\Module%203\project%201\dependency-check-report.html#l15_225a4fd31156c254e3bb92adb42ee8c6de812714)

**Description**

Spring Boot

**Problem**

Multiple NetApp products incorporate Spring Boot. Spring Boot versions 3.0.0 through 3.0.5, 2.7.0 through 2.7.10 and older unsupported versions are susceptible to a vulnerability which when successfully exploited could lead to disclosure of sensitive information, addition or modification of data, or Denial of Service (DoS).

Successful exploitation of this vulnerability could lead to disclosure of sensitive information, addition or modification of data, or Denial of Service (DoS).

**Remediation**

None at the moment according to netapp

**Dependency**

[spring-boot-starter-web-2.2.4.RELEASE.jar](file:///C:\Users\310130710\OneDrive%20-%20Philips\Documents\Education\CS-305%20Software%20Security\Module%203\project%201\dependency-check-report.html#l16_ec75d01d212b5229c16d872fb127744c0ed46ed8)

**Description**

Starter for building web, including RESTful, applications using Spring MVC. Uses Tomcat as the default embedded container

**Problem**

Multiple NetApp products incorporate Spring Boot. Spring Boot versions 3.0.0 through 3.0.5, 2.7.0 through 2.7.10 and older unsupported versions are susceptible to a vulnerability which when successfully exploited could lead to disclosure of sensitive information, addition or modification of data, or Denial of Service (DoS).

Successful exploitation of this vulnerability could lead to disclosure of sensitive information, addition or modification of data, or Denial of Service (DoS).

**Remediation**

None at the moment according to netapp

**Dependency**

[spring-core-5.2.3.RELEASE.jar](file:///C:\Users\310130710\OneDrive%20-%20Philips\Documents\Education\CS-305%20Software%20Security\Module%203\project%201\dependency-check-report.html#l17_3734223040040e8c3fecd5faa3ae8a1ed6da146b)

**Description**

Spring Core

**Problem**

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.

These are the prerequisites for the exploit:

JDK 9 or higher

Apache Tomcat as the Servlet container

Packaged as WAR

spring-webmvc or spring-webflux dependency

**Remediation**

Users of affected versions should apply the following mitigation: 5.3.x users should upgrade to5.3.18+, 5.2.x users should upgrade to 5.2.20+. No other steps are necessary. There are other mitigation steps for applications that cannot upgrade to the above versions. Those are described in the early announcement blog post, listed under the Resources section. Releases that have fixed this issue include:

Spring Framework

5.3.18+

5.2.20+

**Dependency**

[spring-web-5.2.3.RELEASE.jar](file:///C:\Users\310130710\OneDrive%20-%20Philips\Documents\Education\CS-305%20Software%20Security\Module%203\project%201\dependency-check-report.html#l18_dd386a02e40b915ab400a3bf9f586d2dc4c0852c)

**Description**

Spring Web

**Remediation**

Spring Framework 5.3.20 includes 14 fixes and improvements. Spring Framework 5.2.22 includes 2 backports. In addition, these releases include fixes for 2 vulnerabilities.

**Dependency**

[spring-webmvc-5.2.3.RELEASE.jar](file:///C:\Users\310130710\OneDrive%20-%20Philips\Documents\Education\CS-305%20Software%20Security\Module%203\project%201\dependency-check-report.html#l19_745a62502023d2496b565b7fe102bb1ee229d6b7)

**Description**

Spring Web MVC

**Remediation**

Users of affected versions should apply the following mitigation: 5.3.x users should upgrade to 5.3.18+, 5.2.x users should upgrade to 5.2.20+. No other steps are necessary. There are other mitigation steps for applications that cannot upgrade to the above versions. Those are described in the early announcement blog post, listed under the Resources section. Releases that have fixed this issue include:

Spring Framework

5.3.18+

5.2.20+

**Dependency**

[tomcat-embed-core-9.0.30.jar](file:///C:\Users\310130710\OneDrive%20-%20Philips\Documents\Education\CS-305%20Software%20Security\Module%203\project%201\dependency-check-report.html#l20_ad32909314fe2ba02cec036434c0addd19bcc580)

**Description**

Core Tomcat implementation

**Problem**

A specially crafted sequence of HTTP/2 requests 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.

**Remediation**

Fixed in Apache Tomcat 10.0.0-M5

**Dependency**

[tomcat-embed-websocket-9.0.30.jar](file:///C:\Users\310130710\OneDrive%20-%20Philips\Documents\Education\CS-305%20Software%20Security\Module%203\project%201\dependency-check-report.html#l22_33157f6bc5bfd03380ebb5ac476db0600a04168d)

**Description**

Core Tomcat implementation

**Remediation**

This update for tomcat to version 9.0.31 fixes the following issues:

Security issues fixed:

- CVE-2019-17569: Fixed a regression in the handling of Transfer-Encoding

headers that would have allowed HTTP Request Smuggling (bsc#1164825).

- CVE-2020-1935: Fixed an HTTP Request Smuggling issue (bsc#1164860).

- CVE-2020-1938: Fixed a file contents disclosure vulnerability

(bsc#1164692).