Kalila Lin Cen

Security Analysis & Recommendation

ArrowFit Web Application

Project Description

ArrowFit is a web application management tool designed for gyms to manage various trainers and their corresponding trainees in terms of training sessions, credits, and feedback.

Application Outline Mechanism

The ArrowFit application features a three-tier architecture comprising the trainee/user-end, trainer-end, and admin-end, specifically tailored for gym management. The design is optimised for mobile use, the app provides an intuitive interface across all user levels.

1. Trainees access their personal information, credit balance, and session history.
2. Trainers manage session histories and personal information for each trainee, with the ability to add credits to trainee accounts upon receiving payments.
3. The admin-end, utilized by gym administrators, oversees trainer management and overall app functionality, ensuring a smooth operation and interaction between trainees, trainers, and gym management.

Security strengths

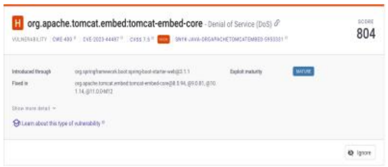
ArrowFit's security includes a double token scheme with access and refresh tokens.

- Access Token: Used for session authentication, it provides short-term access to the user's account and its features.
- Refresh Token: Helps maintain a user's session securely. When the access token expires, the refresh token is used to obtain a new access token without requiring the user to log in again.

Vulnerability	Severity	Frequency	Cause	Exploitation
1.org.yaml:snakeya ml CWE - 20 CVE-2022-1471	NVD 9.8 Critical	High	The org.yaml:snakeyaml package issue is due to its Constructor class not limiting Java object creation from YAML inputs.	Exploitation occurs when a malicious YAML file is deserialized by the SnakeYAML package, leading to arbitrary code execution.
2.org.springframework.security :spring-security-	NVD 9.8 Critical	High	The issue stems from a mismatch between org.springframework.security: spring-security-config and Spring	The mismatch allows exploitation where URL security rules set by Spring Security are not enforced by Spring

config CWE - 284 CVE-2023-34034			WebFlux in the Spring Framework.	WebFlux, leading to unauthorized access control bypass.
3.@babel/traverse CWE-184 CVE-2023-45133	NVD 8.8 High	Low	The @babel/traverse dependency in react-scripts@5.0.1 fails to properly restrict some input types.	The package fails to identify and block certain harmful inputs. This issue can lead to an incomplete list of disallowed inputs, allowing attackers to manipulate how Babel processes and transforms JavaScript code, potentially resulting in code execution, data leakage, and other incidents.
4.org.apache.tomcat.embed:tomcat-emb ed-core CWE - 400 CVE-2023-44487	NVD 7.5 High	High	The vulnerability originates from org.springframework.boot:spring-boot-starter-web@3.1.1, affecting the Tomcat embedded core library 10.1.10 in the Spring Boot framework.	Attackers exploit the vulnerability by initiating and rapidly canceling many streams, causing a DoS attack that leads to resource consumption, performance degradation, or service shutdown.
5.org.springframework.security:spring-security-config CWE - 285 CVE-2023-34035	NVD 7.5 High	High	Caused by improper validation in requestMatchers, is introduced by the org.springframework.security:spring-sec urity-config package in the Spring Framework.	Due to improperly enforced security rules, attackers can exploit this vulnerability to gain unauthorized access to parts of the application.

Screenshots Evidence:



H org.apache.tomcat.embed:tomcat-embed-core - Denial of Service (DoS)

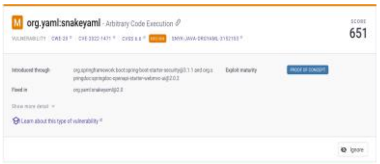
VULNERABILITY CWE-400 CVE-2023-44487 CVSS 7.5 HIGH SNYK-JAVA-ORGAPACHETOMCATEMBED-5953331

SCORE 804

Introduced through: org.springframework.boot:spring-boot-starter-web@3.1.1

Exploit maturity: MATURE

Fixed in: org.apache.tomcat.embed:tomcat-embed-core@8.5.94, @9.0.81, @10.1.14, @11.0.0-M12



M org.yaml:snakeyaml - Arbitrary Code Execution

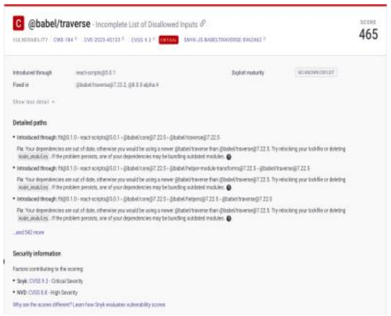
VULNERABILITY CWE-20 CVE-2022-1471 CVSS 8.8 MEDIUM SNYK-JAVA-ORGYAML-3152153

SCORE 651

Introduced through: org.springframework.boot:spring-boot-starter-security@3.1.1 and org.springdoc:springdoc-openapi-starter-webmvc-ui@2.0.2

Exploit maturity: PROOF OF CONCEPT

Fixed in: org.yaml:snakeyaml@2.0



C @babel/traverse - Incomplete List of Disallowed Inputs

VULNERABILITY CWE-184 CVE-2023-45133 CVSS 9.3 CRITICAL SNYK-JS-BABELTRAVERSE-5962462

SCORE 465

Introduced through: react-scripts@5.0.1

Exploit maturity: NO KNOWN EXPLOIT

Fixed in: @babel/traverse@7.23.2, @8.0.0-alpha.4

Detailed paths

- Introduced through: fit@0.1.0 › react-scripts@5.0.1 › @babel/core@7.22.5 › @babel/traverse@7.22.5

  Fix: Your dependencies are out of date, otherwise you would be using a newer @babel/traverse than @babel/traverse@7.22.5. Try relocking your lockfile or deleting node_modules. If the problem persists, one of your dependencies may be bundling outdated modules.
- Introduced through: fit@0.1.0 › react-scripts@5.0.1 › @babel/core@7.22.5 › @babel/helper-module-transforms@7.22.5 › @babel/traverse@7.22.5

  Fix: Your dependencies are out of date, otherwise you would be using a newer @babel/traverse than @babel/traverse@7.22.5. Try relocking your lockfile or deleting node_modules. If the problem persists, one of your dependencies may be bundling outdated modules.
- Introduced through: fit@0.1.0 › react-scripts@5.0.1 › @babel/core@7.22.5 › @babel/helpers@7.22.5 › @babel/traverse@7.22.5

  Fix: Your dependencies are out of date, otherwise you would be using a newer @babel/traverse than @babel/traverse@7.22.5. Try relocking your lockfile or deleting node_modules. If the problem persists, one of your dependencies may be bundling outdated modules.

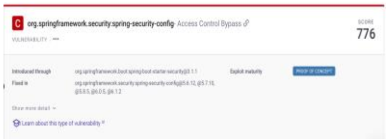
...and 542 more

Security information

Factors contributing to the scoring:

- Snyk: CVSS 9.3 - Critical Severity
- NVD: CVSS 8.8 - High Severity

Why are the scores different? Learn how Snyk evaluates vulnerability scores



C org.springframework.security:spring-security-config- Access Control Bypass

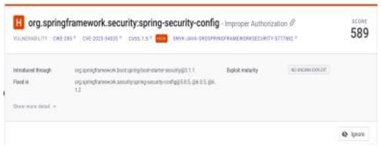
VULNERABILITY ...

SCORE 776

Introduced through: org.springframework.boot:spring-boot-starter-security@3.1.1

Exploit maturity: PROOF OF CONCEPT

Fixed in: org.springframework.security:spring-security-config@5.6.12, @5.7.10, @5.8.5, @6.0.5, @6.1.2



H org.springframework.security:spring-security-config - Improper Authorization

VULNERABILITY CWE-285 CVE-2023-34035 CVSS 7.5 HIGH SNYK-JAVA-ORGSPRINGFRAMEWORKSECURITY-5777892

SCORE 589

Introduced through: org.springframework.boot:spring-boot-starter-security@3.1.1

Exploit maturity: NO KNOWN EXPLOIT

Fixed in: org.springframework.security:spring-security-config@5.8.5, @6.0.5, @6.1.2

I assessed the project's security architecture quality using the CAWE Catalog, which lists 224 flaws across 11 security tactics based on impact, classifying vulnerabilities into architectural and non-architectural categories. Two architectural weaknesses identified are:

Security tactic	**Flaws**	**Cause**
Validate Inputs	Improper input validation	@babel/traverse: This vulnerability involves the failure of web applications using certain Babel plugins to correctly filter inputs, leading to risks like denial of service and code injection. It emphasizes the need for comprehensive input validation to prevent security breaches.
Authorise Actors	Improper Access Control	org.springframework.security:spring-security-config: This vulnerability allows bypassing access controls due to a mismatch in pattern matching. It is an example of an "Access Control" weakness, specifically "Improper Authorization". This can lead to unauthorized access and potential data breaches.

Countermeasures Recommendation

No. Vul	**Vulnerability name**	**Recommendation/Solution**
1	org.yaml:snakeyaml	Upgrade the eorg.yaml:snakeyaml package to version 2.0.
2	org.springframework.security:spring-security-config	Upgrade the package to version 5.6.12, 5.7.10, 5.8.5, 6.0.5 ,or 6.1.2.
3	@babel/traverse	Upgrade @babel/traverse package to a newer versions of, specifically 7.23.2 and 8.0.0-alpha.4.
4	org.apache.tomcat.embed:tomcat-embed-core	Upgrade the Tomcat embedded core 11.0.0-M11-M14.
5	org.springframework.security:spring-security-config	Upgrade the package to version 5.8.5, 6.0.5 or 6.1.2.

Security Principles Breach

No. Vul	Explanation
1	● Principle of Least Privilege: Limit Java object creation from YAML data to only essential types for the application. ● Defence in Depth: Add extra validation to checks to prevent malicious data from being processed.
2	● Fail Securely: In the case of mismatches or errors, the system should default to deny access. ● Separation of Duties: Ensure clear division between components so that a failure in Spring WebFlux doesn't compromise security configured in URL security rules.
3	● Keep Security Simple: Simplify security to effectively block bad inputs. ● Establish Secure Defaults: The default configuration should identify and block potentially harmful inputs. ● Fix Security Issues Correctly: Address the root cause of input handling failures.
4	● Fail Securely: For abnormal stream behaviors, the system should deny further requests or flagging for review. ● Minimise Attack Surface Area: Restrict excessive stream initiation and cancellation.
5	● Avoid Security by Obscurity: Use well-understood practices over obscure configurations.

Critique legal, ethical, and privacy issues

Software issues in org.yaml:snakeyaml and org.springframework.security could break data protection laws like GDPR and risk private data. Despite some safety efforts, their security isn't strong enough, leading to potential legal breaches and a failure to adhere to best software security practices. This results in notable risks to user privacy and data security.