OWASP Top 10 Security Report: WiredSpace

This report summarises the security posture of WiredSpace projects against the OWASP Top 10 (2021) vulnerabilities, based on a scan conducted on 20th June 2025, using Snyk CLI and ChatGPT. It provides an overview of scan results and details on a previously identified and remediated vulnerability.

Scan Overview and Tools

The security scan for WiredSpace projects was performed on 20th June 2025. The analysis tool utilised were: Snyk CLI and ChatGPT

The projects included in the scope of this security assessment were:

- Backend (Spring Boot / Gradle)
- Frontend (React / npm)

Overall Scan Results

The comprehensive security scan across all WiredSpace projects yielded positive results, with no new vulnerabilities detected.

Project	Vulnerabilities	Threat Level	Note
wiredspace-frontend	0	—	All dependencies are secure
WiredSpace-API	0	—	Overall, application is secure

This indicates a strong current security posture for the project's dependencies.

OWASP Top 10 Mapping (2021)

The scan mapped the project's security against the OWASP Top 10 (2021) categories. The results indicate no active vulnerabilities across most critical categories.

Category	Vulnerabilities Found	Note
A01: Broken Access Control	✅ Previously	Admin service class was changed.
A02: Cryptographic Failures	❌ No	—
A03: Injection	❌ No	Due to using JPA and parameterized queries
A04: Insecure Design	❌ No	—
A05: Security Misconfiguration	❌ No	—

OWASP Top 10 Mapping (Continued)

The assessment continued to evaluate the remaining OWASP Top 10 categories, with a focus on previously identified issues.

Category	Vulnerabilities Found	Note
A06: Vulnerable and Outdated Components	✅ Previously	mysql-connector-j updated to 9.3.0
A07: Identification and Authentication Failures	❌ No	—
A08: Software and Data Integrity Failures	❌ No	—
A09: Security Logging and Monitoring Failures	❌ No	—
A10: Server-Side Request Forgery	❌ No	—

Security Risks: OWASP Top 10

	Likelihood	Impact	Risk	Actions possible	Planned
A01: Broken Access Control	High	Severe	High	Implement PBAC for all controllers; improve self-access checks	✅ Yes
A02: Cryptographic Failures	Medium	Severe	Moderate	JWT should be sufficient, but consider using https	⚠️ Partial
A03: Injection	Low	Severe	Moderate	Use of JPA (ORM) mitigates raw SQL injection	✅ Yes
A04: Insecure Design	Medium	Moderate	Moderate	Continue regular code reviews and threat modeling	✅ Yes
A05: Security Misconfiguration	Medium	Moderate	Moderate	Harden default settings; add strict security headers	✅ Yes
A06: Outdated Components	Medium	High	High	Use tools to check for dependency vulnerabilities (e.g., Snyk, OWASP Dependency Check)	✅ Yes
A07: Auth Failures	Low	High	Moderate	Implement automatic token expiry check in frontend	✅ Yes
A08: Software/Data Integrity Failures	Low	Severe	Moderate	Avoid unverified modules from public registries/CDNs	✅ Yes
A09: Logging & Monitoring Failures	Medium	Moderate	Moderate	Add authentication/audit logging, possibly ELK stack	❌ No
A10: SSRF	Very Low	Low	Low	No external calls from backend, low exposure	⚠️ Partial

Remediated Issues: Vulnerable Component

One significant issue that was previously identified and has since been successfully remediated involved a vulnerable component within the system.

> **Vulnerable Component:** `com.mysql:mysql-connector-j:9.1.0`

This component was identified as a security risk, falling under the OWASP category **A06: Vulnerable and Outdated Components**.

Also, during internal code review, it was identified that the AdminService layer lacked internal authorization checks, relying solely on controller-level annotations.

Remediation Action and Details

To address the identified vulnerability, a critical update was performed.

> **Updated to:** `9.3.0`

The specific problem associated with this component was an "Incorrect Default Permissions" issue, identified by SNYK-JAVA-COMMYSQL-9725315. This remediation ensures that the system is no longer exposed to the risks associated with this particular vulnerability.

Remediation Action and Details

The following actions were taken regarding the Admin layer issue

- Introduced defensive access checks using AuthenticatedUserProvider in AdminServiceImpl
- Blocked self-demotion and self-deletion of administrators
- Enforced role-based constraints before sensitive actions (promote, demote, delete)
- Shifted access control into service layer for better resilience

These changes mitigate the risk of privilege escalation or unauthorized access via broken access control patterns.