POLARIS INTELLIGENCE

Technical Due Diligence Report

log4j2

03 April 2026

Ref: 9a5ac77e

CONFIDENTIAL

1. Executive Summary

Repository Classification: Production (Confidence: medium)
This repository is assessed as production software. All scanner findings are reported at full weight.

31 known vulnerabilities found in 16 of 264 dependencies. Outdated or vulnerable packages increase attack surface.

Bus factor: Tier 4: Elevated Concentration. Review recommended to assess key-person dependency.

The table below rates risk across 13 dimensions, from Clean (no findings) to Critical (potential deal impact). Together they form the technical risk profile of the target asset.

CATEGORY	RATING	SUMMARY
Secrets & Credentials	CLEAN	No credential exposure detected in scanned files
Dependency Vulnerabilities	CRITICAL RISK	3 critical CVEs
Supply Chain Risk	CLEAN	No known supply chain incidents
Licence & IP	CLEAN	Permissive licence, no IP issues detected
Developer Concentration	MEDIUM RISK	Tier 4: Elevated Concentration
Code Quality	CLEAN	Grade B (67.0/100)
Architecture	MEDIUM RISK	21 circular dependencies (1% of modules)
Malware / Destructive Code	CLEAN	No suspicious patterns detected in scanned files
Test Coverage	CLEAN	Strong testing discipline (42% test ratio, CI gates active)
Infrastructure & Deployment	MEDIUM RISK	Limited deployment infrastructure — significant gaps
Technical Debt	CLEAN	Minimal tech debt (169 markers (0.5/KLOC), 457 suppressions (1.5/KLOC))
Governance & CI/CD	CLEAN	Strong governance (Grade A, 8.4/10)

CATEGORY	RATING	SUMMARY
Engineering Maturity	LOW RISK	Adequate maturity (7.4 releases/yr, 60% community health)

2. Transaction Impact Assessment

This section translates technical findings into their commercial implications for the transaction. Ratings range from Clean (no concern) to Critical (potential deal-breaker), with specific conditions that may need to be met before or after completion.

CATEGORY	ASSESSMENT	DETAIL
Security Exposure	CRITICAL RISK	3 critical CVEs
Operational Risk	MEDIUM RISK	Tier 4: Elevated Concentration; Grade B (67.0/100)
IP & Licence Risk	CLEAN	Permissive licence, no IP issues detected
Integration Complexity	MEDIUM RISK	21 circular dependencies (1% of modules)
Maintenance Burden	MEDIUM RISK	Grade B, 3 quality issues, 169 tech debt markers

Remediation Effort Estimate

3 critical areas requiring immediate action; 15 high-severity areas for short-term remediation; 12 medium-severity items for the integration roadmap.

Estimates assume a senior developer familiar with the technology stack. Actual effort may vary based on codebase familiarity and organisational context.

3. Scope & Methodology

Repository: https://github.com/apache/logging-log4j2

Analysis date: 03 April 2026

Codebase size: 5,723 files, 405,341 lines

LANGUAGE	LINES
Java	346,827
XML	48,471

LANGUAGE	LINES
YAML	4,937
JSON	3,662
Shell	622

Methodology

This report was produced by Polaris Intelligence automated analysis pipeline. The following scanners were applied:

1. **GitHub Enrichment** — project metadata, release cadence, community health
2. **Secret Scanner** — regex pattern matching + context classification
3. **Dependency Scanner** — manifest parsing + OSV vulnerability cross-reference + exploitability analysis
4. **Supply Chain Intelligence** — cross-reference against known supply chain incidents
5. **Licence Auditor** — declared licence + source header contradiction detection
6. **Bus Factor Analysis** — 5-tier developer concentration taxonomy (24-month window)
7. **Code Quality Scorer** — cyclomatic complexity, duplication, security anti-patterns
8. **Architecture Mapper** — import graph, circular dependencies, module coupling
9. **Engineering Maturity** — release discipline, community governance, project signals
10. **Malware Heuristic** — destructive actions, crypto mining, exfiltration, obfuscation
11. **Governance & CI/CD** — OpenSSF Scorecard, branch protection, dependency management

Note: *File counts may vary between sections because each scanner operates on a different subset of files (e.g. quality analysis covers source code files only, while the scope total includes configuration, documentation, and data files).*

This is an automated analysis and does not constitute legal, security, or investment advice. Findings should be verified by qualified professionals.

4. Secrets & Credentials CLEAN

Hardcoded credentials — API keys, database passwords, tokens — are the most common cause of data breaches. Their presence indicates both an immediate security exposure and a gap in the target's engineering practices that transfers with the acquisition.

No hardcoded secrets or credentials detected.

5. Dependency Vulnerabilities CRITICAL RISK

Modern software relies on hundreds of third-party packages. Known vulnerabilities in these dependencies are publicly catalogued and actively exploited. Unpatched critical CVEs represent a quantifiable security liability that transfers to the acquirer.

264 dependencies analysed across 22 manifests (238 runtime, 26 dev/test).

31 dependency vulnerabilities found:

SEVERITY		COUNT
CRITICAL	CRITICAL RISK	3
HIGH	HIGH RISK	15
MEDIUM	MEDIUM RISK	9
LOW	LOW RISK	4

SEVERITY	PACKAGE	VERSION	CVE/ ID	SUMMARY	FIX	EXPOSURE
MEDIUM (3.0)	ch.qos.logback:logback-core	1.3.15	CVE-2025-11226	QOS.CH logback-core is vulnerable to Arbitrary Code Execution through file processing	1.5.19	DECLARED ONLY
LOW (3.0)	ch.qos.logback:logback-core	1.3.15	CVE-2026-1225	Logback allows an attacker to instantiate classes already present on the class path	1.5.25	DECLARED ONLY
HIGH (7.5)	fast-xml-parser	5.0.6	CVE-2026-33036	fast-xml-parser affected by numeric entity expansion bypassing all entity expansion…	5.5.6	DIRECT IMPORT

SEVERITY	PACKAGE	VERSION	CVE/ ID	SUMMARY	FIX	EXPOSURE
LOW (7.5)	fast-xml-parser	5.0.6	CVE-2026-27942	fast-xml-parser has stack overflow in XMLBuilder with preserveOrder	5.3.8	DIRECT IMPORT
HIGH (7.5)	fast-xml-parser	5.0.6	CVE-2026-26278	fast-xml-parser affected by DoS through entity expansion in DOCTYPE (no expansion limit)	5.3.6	DIRECT IMPORT
MEDIUM (5.0)	handlebars	4.7.8	CVE-2026-33916	Handlebars.js has Prototype Pollution Leading to XSS through Partial Template Injection	4.7.9	DIRECT IMPORT
MEDIUM	handlebars	4.7.8	GHSA-2w6w-674q-4c4q			DIRECT IMPORT
HIGH (9.5)	handlebars	4.7.8	CVE-2026-33938	Handlebars.js has JavaScript Injection via AST Type Confusion by tampering @partial-block	4.7.9	DIRECT IMPORT
HIGH (5.0)	org.assertj:assertj-core	3.27.3	CVE-2026-24400	AssertJ has XML External Entity (XXE) vulnerability when parsing untrusted XML via…	3.27.7	DECLARED ONLY
CRITICAL (9.5)	log4j:log4j	1.2.17	CVE-2019-17571	Deserialization of Untrusted Data in Log4j		DECLARED ONLY
CRITICAL (9.5)	log4j:log4j	1.2.17	CVE-2022-23305	SQL Injection in Log4j 1.2.x		DECLARED ONLY

SEVERITY	PACKAGE	VERSION	CVE/ ID	SUMMARY	FIX	EXPOSURE
HIGH (7.5)	log4j:log4j	1.2.17	CVE-2023-26464	Apache Log4j 1.x (EOL) allows Denial of Service (DoS)	2.0	DECLARED ONLY
HIGH	org.codehaus.plexus:plexus-utils	3.6.0	CVE-2025-67030	Plexus-Utils has a Directory Traversal vulnerability in its extractFile method	4.0.3	DECLARED ONLY
HIGH (7.5)	org.springframework.boot:spring-boot	2.7.18	CVE-2025-22235	Spring Boot EndpointRequest.to() creates wrong matcher if actuator endpoint is not exposed	3.4.5	DECLARED ONLY
HIGH (7.5)	org.springframework:spring-core	5.3.39	CVE-2025-41249	Spring Framework annotation detection mechanism may result in improper authorization	6.2.11	DECLARED ONLY
HIGH (9.5)	org.apache.velocity:velocity	1.7	CVE-2020-13936	Sandbox Bypass in Apache Velocity Engine	2.3	DECLARED ONLY
MEDIUM (7.5)	com.fasterxml.jackson.core:jackson-core	2.19.2	GHSA-72hv-8253-57qq	jackson-core: Number Length Constraint Bypass in Async Parser Leads to Potential DoS…	2.21.1	TRANSITIVE
HIGH (7.5)	org.lz4:lz4-java	1.8.0	CVE-2025-66566	yawkat LZ4 Java has a possible information leak in Java safe decompressor	1.10.1	TRANSITIVE
HIGH (7.5)	org.lz4:lz4-java	1.8.0	CVE-2025-12183	LZ4 Java Compression has Out-of-bounds memory operations which can cause DoS	1.8.1	TRANSITIVE

SEVERITY	PACKAGE	VERSION	CVE/ ID	SUMMARY	FIX	EXPOSURE
HIGH (7.5)	io.netty:netty-handler	4.1.94.Final	CVE-2025-24970	SslHandler doesn't correctly validate packets which can lead to native crash when using…	4.1.118.Final	TRANSITIVE
MEDIUM (5.0)	io.netty:netty-common	4.1.94.Final	CVE-2025-25193	Denial of Service attack on windows app using Netty	4.1.118.Final	TRANSITIVE
MEDIUM (7.5)	io.netty:netty-codec	4.1.94.Final	CVE-2025-58057	Netty's decoders vulnerable to DoS via zip bomb style attack	4.2.5.Final	TRANSITIVE
LOW (5.0)	com.google.guava:guava	25.1-jre	CVE-2020-8908	Information Disclosure in Guava	32.0.0-android	TRANSITIVE
MEDIUM (5.0)	com.google.guava:guava	25.1-jre	CVE-2023-2976	Guava vulnerable to insecure use of temporary directory	32.0.0-android	TRANSITIVE
HIGH (7.5)	com.google.code.gson:gson	2.8.2	CVE-2022-25647	Deserialization of Untrusted Data in Gson	2.8.9	TRANSITIVE

First-party packages checked: Maven:org.apache.logging.log4j:log4j-bom, Maven:org.apache.logging.log4j:log4j-jpl, Maven:org.apache.logging.log4j:log4j-slf4j2-impl-fuzz-test. No known CVEs affecting current version.

Historical CVE Record

These CVEs were filed against prior versions of the target's package. While the current version may not be affected, a history of security incidents indicates recurring risk and may inform the buyer's assessment of the development team's security posture.

org.apache.logging.log4j:log4j: 2 CVEs on record

SEVERITY	CVE/ID	SUMMARY
CRITICAL RISK	CVE-2017-5645	Deserialization of Untrusted Data in Log4j
LOW RISK	CVE-2020-9488	Improper validation of certificate with host mismatch in Apache Log4j SMTP appender

org.apache.logging.log4j:log4j-core: 7 CVEs on record

SEVERITY	CVE/ID	SUMMARY
CRITICAL RISK	CVE-2021-45046	Incomplete fix for Apache Log4j vulnerability
HIGH RISK	CVE-2021-45105	Apache Log4j2 vulnerable to Improper Input Validation and Uncontrolled Recursion
CRITICAL RISK	CVE-2017-5645	Deserialization of Untrusted Data in Log4j
CRITICAL RISK	CVE-2021-44228	Remote code injection in Log4j
MEDIUM RISK	CVE-2025-68161	Apache Log4j does not verify the TLS hostname in its Socket Appender
HIGH RISK	CVE-2023-26464	Apache Log4j 1.x (EOL) allows Denial of Service (DoS)
LOW RISK	CVE-2020-9488	Improper validation of certificate with host mismatch in Apache Log4j SMTP appender

Dependency Health

“No known CVEs” does not mean healthy dependencies. Stale or abandoned packages receive no security patches and represent latent risk. Health status is derived from package registry release dates.

STATUS		COUNT
Active (released within 6 months)	CLEAN	36
Stable (released within 1 year)	CLEAN	52
Stale (1–2 years since last release)	MEDIUM RISK	51
Abandoned (2+ years since last release)	HIGH RISK	66
Unknown (registry lookup failed)	LOW RISK	34

Dependency health score: **4.8/10**

Note: *28 dependencies returned implausible release dates from package registries and are excluded from the health score calculation.*

At-Risk Dependencies

PACKAGE	VERSION	STATUS	LAST RELEASE
oro:oro	2.0.8	HIGH RISK abandoned	2005-11-08 (20.4yr ago)
javax.activation:activation	1.1	HIGH RISK abandoned	2009-10-23 (16.4yr ago)
org.apache.velocity:velocity	1.7	HIGH RISK abandoned	2010-11-29 (15.3yr ago)
com.github.jnr:jnr-x86asm	1.0.2	HIGH RISK abandoned	2012-04-07 (14.0yr ago)
log4j:log4j	1.2.17	HIGH RISK abandoned	2012-05-26 (13.8yr ago)
org.osgi:org.osgi.core	6.0.0	HIGH RISK abandoned	2014-07-30 (11.7yr ago)
io.netty:netty-handler	4.1.94.Final	HIGH RISK abandoned	2015-03-03 (11.1yr ago)
io.netty:netty-common	4.1.94.Final	HIGH RISK abandoned	2015-03-03 (11.1yr ago)
io.netty:netty-resolver	4.1.94.Final	HIGH RISK abandoned	2015-03-03 (11.1yr ago)
io.netty:netty-buffer	4.1.94.Final	HIGH RISK abandoned	2015-03-03 (11.1yr ago)
io.netty:netty-transport	4.1.94.Final	HIGH RISK abandoned	2015-03-03 (11.1yr ago)
io.netty:netty-codec	4.1.94.Final	HIGH RISK abandoned	2015-03-03 (11.1yr ago)
javax.jms:javax.jms-api	2.0.1	HIGH RISK abandoned	2015-03-11 (11.0yr ago)
aopalliance:aopalliance	1.0	HIGH RISK abandoned	2015-12-28 (10.3yr ago)
org.apache-extras.beanshell:bsh	2.0b6	HIGH RISK abandoned	2016-02-18 (10.1yr ago)

Dependency Concentration

High concentration of dependencies from a single namespace increases supply chain risk if that maintainer or organisation is compromised.

NAMESPACE / ORG	DEPENDENCIES
org.apache.logging.log4j	38

NAMESPACE / ORG	DEPENDENCIES
org.apache.maven	10
org.mongodb	7
com.github.jnr	7
io.netty	7

6. Developer Concentration (Bus Factor) MEDIUM RISK

"Bus factor" measures how many developers would need to leave before critical knowledge is lost. High concentration in one developer creates key-person dependency — a material operational risk that can delay integration and increase post-acquisition costs.

Tier 4: Elevated Concentration — Bus factor score: **58.6%**

58.6% developer concentration across 14 active contributors is above the recommended threshold. While not critical, proactive knowledge sharing is advised to reduce key-person risk.

Top Contributors

DEVELOPER	COMMITS	OWNED FILES	CORE %	STATUS
Piotr P. Karwasz	403	747	48%	Active
Volkan Yazıcı	238	2454	27%	Active
ASF Logging Services RM	233	16	1%	Active
Christian Grobmeier	44	4	0%	Departed
Gary Gregory	22	21	3%	Active
JWT	7	8	0%	Departed
Alba Herrerías	7	10	1%	Departed
Ralph Goers	7	1	0%	Departed
Ryan Schmitt	6	12	1%	Departed
Kevin Cruz	3	1	0%	Active

Departed Developer Risk

DEVELOPER	MONTHS INACTIVE	CORE FILES OWNED	RISK
Junhyeok Lee	8	46	HIGH RISK
Po Chun Yu	17	12	HIGH RISK
Jay Katariya	14	8	HIGH RISK
Ryan Schmitt	8	4	MEDIUM RISK
Alba Herrerías	15	4	MEDIUM RISK
JWT	10	2	MEDIUM RISK
eldwrjwt	15	2	MEDIUM RISK
Jonas Rutishauser	17	2	MEDIUM RISK
Ninette Adhikari	18	2	MEDIUM RISK
Ramanathan	6	1	MEDIUM RISK

7. Licence & Intellectual Property CLEAN

Copyleft licences (GPL, AGPL) require derivative works to be released under the same open-source terms. If copyleft code is embedded in a proprietary product, the acquirer may face an obligation to open-source their own code — or costly remediation to replace the affected components.

Declared Licences

SPDX ID	RISK	SOURCE	FILE
Apache-2.0	CLEAN	licence_file	`LICENSE.txt`

No licence contamination detected.

8. Code Quality & Technical Debt CLEAN

Code quality directly predicts the cost and speed of post-acquisition development. A low grade signals elevated technical debt — higher bug rates, slower feature delivery, and more expensive onboarding for new developers joining after the transaction.

Quality grade: **B (67.0/100)** — Acceptable

Grade Scale

GRADE	MEANING
A	Excellent maintainability
B	Good engineering quality
C	Moderate technical debt
D	High technical debt
F	Severe structural risk

METRIC	VALUE
Total files	1490
Code lines	127,003
Functions	13648
Classes	1942
Avg function length	7.8 lines
Avg complexity	1.8

Technical Debt Indicators

The following indicators are informational. At this grade level, they represent normal characteristics of a healthy codebase rather than actionable concerns. Duplication above automated thresholds is common in test files and generated code.

- 20 files exceed 500 lines
- 20 functions with high cyclomatic complexity
- Code duplication at 21.4% (threshold: 5.0%)

Complexity Hotspots

SEVERITY	FILE	FUNCTION	COMPLEXITY
HIGH RISK	`log4j-core/src/.../util/CronExpression.java` [F1]	addToSet	77
HIGH RISK	`log4j-core/src/.../util/CronExpression.java` [F1]	storeExpressionVals	71
HIGH RISK	`log4j-core/src/.../util/CronExpression.java` [F1]	getTimeAfter	67
HIGH RISK	`log4j-core/src/...rowableFormatOptions.java` [F2]	newInstance	38
HIGH RISK	`log4j-core/src/...time/FastDatePrinter.java` [F3]	parsePattern	38
HIGH RISK	`log4j-core/src/.../picocli/CommandLine.java` [F4]	Help	33
HIGH RISK	`log4j-core/src/...ookup/StrSubstitutor.java` [F5]	substitute	32
HIGH RISK	`log4j-core/src/.../util/CronExpression.java` [F1]	buildExpression	30
HIGH RISK	`log4j-layout-te...kTraceStringResolver.java` [F6]	findLabeledLineStartIndex	30
HIGH RISK	`log4j-core/src/...bstractConfiguration.java` [F7]	doConfigure	29

Large Files (>500 lines)

FILE	TOTAL LINES	CODE LINES
`log4j-core/src/.../picocli/CommandLine.java` [F4]	5620	3283
`log4j-api/src/m...logging/log4j/Logger.java` [F8]	4881	839
`log4j-api/src/m...j/spi/AbstractLogger.java` [F9]	3916	3158
`log4j-core/src/.../util/CronExpression.java` [F1]	1686	1202
`log4j-core/src/...ookup/StrSubstitutor.java` [F5]	1561	623
`log4j-core/src/...time/FastDatePrinter.java` [F3]	1539	799

FILE	TOTAL LINES	CODE LINES
`log4j-core/src/...bstractConfiguration.java` [F7]	1288	913
`log4j-core/src/...e/impl/Log4jLogEvent.java` [F10]	1277	887
`log4j-core/src/.../config/LoggerConfig.java` [F11]	1245	775
`log4j-core/src/...layout/Rfc5424Layout.java` [F12]	1214	839

Security Anti-Patterns

These patterns represent common security vulnerabilities (OWASP Top 10, CWE). Each warrants developer review to confirm whether it represents an actual risk in context.

SEVERITY	PATTERN	LOCATION	DESCRIPTION
HIGH RISK	ObjectInputStream	`log4j-api/src/m...al/SerializationUtil.java:119` [F13]	Java deserialization via ObjectInputStream — potential RCE if input is untrusted
LOW RISK	SQL string concatenation	`log4j-cassandra.../CassandraAppenderIT.java:84` [F14]	SQL query with string interpolation/ concatenation — potential SQL injection (test file)
LOW RISK	SQL string concatenation	`log4j-cassandra...sandra/CassandraRule.java:94` [F15]	SQL query with string interpolation/ concatenation — potential SQL injection (test file)

9. Architectural Assessment MEDIUM RISK

1486 production modules (plus 1 test files), 5433 internal dependencies, 884 external imports.

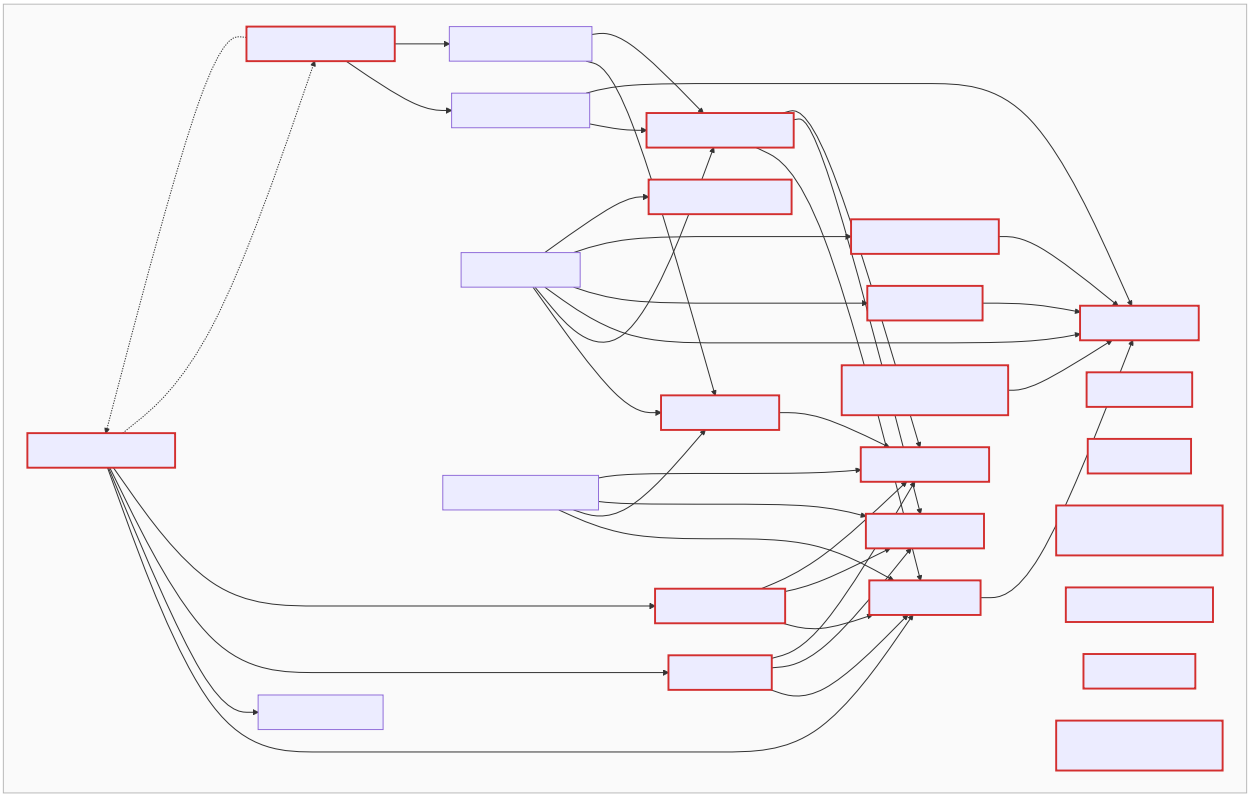
> **21 circular dependencies detected. Circular imports increase coupling and make refactoring more expensive. Breaking these cycles should be prioritised to reduce integration risk.**

Change impact zones: 20 modules have high fan-in (many dependents). Top modules: `log4j-core/src/main/java/org/apache/logging/log4j/core/LogEvent.java`, `log4j-core/src/main/java/org/apache/logging/log4j/core/config/plugins/Plugin.java`, `log4j-api/src/main/java/org/apache/logging/log4j/status/StatusLogger.java`. Changes to these modules carry elevated regression risk.

30 isolated modules detected (no internal imports or dependents). These are typically standalone utilities, examples, platform-specific implementations, or generated code.

Module Dependency Map

Modules are grouped by architectural layer and coloured accordingly. Arrows show import dependencies. Numbers in parentheses indicate how many other modules depend on each file. Thick red borders highlight high fan-in modules (change risk).



Layer Distribution

Modules are grouped by their role in the application: business logic (core functionality), presentation (user-facing), data (storage), and test. A well-structured codebase separates these concerns clearly.

LAYER	MODULES
business_logic	778
infrastructure	445
other	157
presentation	95
data	11

Entry Points (20)

LOCATION		PATTERN
`log4j-1.2-api/s...nfigurationConverter.java:146`	[F16]	`public static void main(`
`src/site/antora...on/Migration1Example.java:25`	[F17]	`public static void main(`
`src/site/antora...on/Migration2Example.java:26`	[F18]	`public static void main(`
`src/site/antora...arkers/MarkerExample.java:37`	[F19]	`public static void main(`
`src/site/antora...glevels/LevelExample.java:34`	[F20]	`public static void main(`
`src/site/antora...CustomMessageExample.java:29`	[F21]	`public static void main(`
`src/site/antora...ages/MessagesExample.java:30`	[F22]	`public static void main(`
`src/site/antora...kups/MainArgsExample.java:27`	[F23]	`public static void main(`
`log4j-perf-test...mh/NanotimeBenchmark.java:33`	[F24]	`public static void main(`
`log4j-perf-test...jmh/VarargsBenchmark.java:33`	[F25]	`public static void main(`

Circular Dependencies (21)

`log4j-api/src/main/java/org/apache/logging/log4j/status/StatusLogger.java → log4j-api/src/main/java/org/apache/logging/log4j/message/ParameterizedNoReferenceMessageFactory.java → log4j-api/src/main/java/org/apache/logging/log4j/status/StatusLogger.java`

`log4j-api/src/main/java/org/apache/logging/log4j/status/StatusLogger.java → log4j-api/src/main/java/org/apache/logging/log4j/spi/AbstractLogger.java → log4j-api/src/main/java/org/apache/logging/log4j/status/StatusLogger.java`

`log4j-api/src/main/java/org/apache/logging/log4j/status/StatusLogger.java → log4j-api/src/main/java/org/apache/logging/log4j/spi/AbstractLogger.java → log4j-api/src/main/java/org/apache/logging/log4j/message/StringFormattedMessage.java → log4j-api/src/main/java/org/apache/logging/log4j/status/StatusLogger.java`

`log4j-api/src/main/java/org/apache/logging/log4j/status/StatusLogger.java → log4j-api/src/main/java/org/apache/logging/log4j/spi/AbstractLogger.java → log4j-api/src/main/java/org/apache/logging/log4j/util/StackLocatorUtil.java → log4j-api/src/main/java/org/apache/logging/log4j/status/StatusLogger.java`

`log4j-api/src/main/java/org/apache/logging/log4j/status/StatusLogger.java → log4j-api/src/main/java/org/apache/logging/log4j/spi/AbstractLogger.java → log4j-api/src/main/java/org/apache/logging/log4j/internal/DefaultLogBuilder.java → log4j-api/src/main/java/org/apache/logging/log4j/status/StatusLogger.java`

16 additional cycles not shown.

High Fan-In Modules (Change Risk)

"Fan-in" measures how many other parts of the codebase depend on a given file. High fan-in files are widely relied upon — changes to them carry elevated risk because they can affect many dependent components.

MODULE		DEPENDENTS	RISK
`log4j-core/src/.../log4j/core/LogEvent.java`	[F26]	261	HIGH RISK
`log4j-core/src/...onfig/plugins/Plugin.java`	[F27]	258	HIGH RISK
`log4j-api/src/m.../status/StatusLogger.java`	[F28]	213	HIGH RISK
`log4j-api/src/m...logging/log4j/Logger.java`	[F8]	181	HIGH RISK
`log4j-api/src/m.../logging/log4j/Level.java`	[F29]	174	HIGH RISK
`log4j-core/src/...config/Configuration.java`	[F30]	143	HIGH RISK
`log4j-api/src/m...g/log4j/util/Strings.java`	[F31]	120	HIGH RISK
`log4j-api/src/m...logging/log4j/Marker.java`	[F32]	108	HIGH RISK
`log4j-api/src/m...og4j/message/Message.java`	[F33]	103	HIGH RISK
`log4j-core/src/...lugins/PluginFactory.java`	[F34]	97	HIGH RISK

Isolated Modules (30)

These files neither use nor are used by any other file in the codebase. They may be unused code, standalone utilities, or components loaded indirectly. Each should be reviewed to confirm it is genuinely needed.

- `log4j-1.2-api/src/main/java/org/apache/log4j/CategoryKey.java`
- `log4j-1.2-api/src/main/java/org/apache/log4j/ProvisionNode.java`
- `log4j-1.2-api/src/main/java/org/apache/log4j/builders/BooleanHolder.java`
- `log4j-1.2-api/src/main/java/org/apache/log4j/builders/Holder.java`
- `log4j-1.2-api/src/main/java/org/apache/log4j/builders/appender/package-info.java`
- `log4j-1.2-api/src/main/java/org/apache/log4j/builders/filter/package-info.java`

- `log4j-1.2-api/src/main/java/org/apache/log4j/builders/layout/package-info.java`
- `log4j-1.2-api/src/main/java/org/apache/log4j/builders/package-info.java`
- `log4j-1.2-api/src/main/java/org/apache/log4j/builders/rewrite/package-info.java`
- `log4j-1.2-api/src/main/java/org/apache/log4j/builders/rolling/package-info.java`
- `log4j-1.2-api/src/main/java/org/apache/log4j/component/helpers/Constants.java`
- `log4j-1.2-api/src/main/java/org/apache/log4j/config/InputStreamWrapper.java`
- `log4j-1.2-api/src/main/java/org/apache/log4j/config/PropertySetterException.java`
- `log4j-1.2-api/src/main/java/org/apache/log4j/config/package-info.java`
- `log4j-1.2-api/src/main/java/org/apache/log4j/helpers/AbsoluteTimeDateFormat.java`

9b. Architecture Topology

This section applies graph-theoretic analysis to the module dependency network identified in the Architectural Assessment. Community detection reveals natural subsystem boundaries; centrality analysis identifies critical bridge modules whose failure or refactoring would disproportionately affect the codebase.

METRIC	VALUE
Modules analysed	1486
Internal dependencies	5433
Subsystems detected	11
Modularity score	0.496 (moderate modularity)
Unclustered modules	259

Subsystem Overview

Communities are groups of modules that are more tightly connected to each other than to the rest of the codebase. Each represents a natural subsystem boundary.

SUBSYSTEM	MODULES
log4j-core/src/ (business_logic, 211 modules)	211
log4j-core/ (infrastructure, 187 modules)	187
log4j-core/ (business_logic, 69 modules)	69
log4j-perf-test/src/ (infrastructure, 2 modules)	2
log4j-core/src/ (business_logic, 185 modules)	185

SUBSYSTEM	MODULES
log4j-core/ (business_logic, 167 modules)	167
log4j-core/src/ (business_logic, 7 modules)	7
log4j-api-java9/ (infrastructure, 2 modules)	2
log4j-1.2-api/src/ (other, 128 modules)	128
log4j-core/ (infrastructure, 240 modules)	240
log4j-core/src/ (business_logic, 29 modules)	29

Bridge Modules (Bottleneck Risk)

Bridge modules sit on critical paths between subsystems. High betweenness centrality means many inter-module communication paths pass through this module — changes here carry elevated risk of cascading failures.

MODULE	CENTRALITY	BRIDGES	FAN-IN / OUT	RISK
`log4j-core/src/...j/core/LoggerContext.java` [F35]	0.0240	6	74 / 29	LOW RISK
`log4j-core/src/...config/Configuration.java` [F30]	0.0222	7	143 / 15	LOW RISK

God Modules (Excessive Coupling)

“God modules” have direct dependencies spanning many subsystems. They violate separation of concerns and make the codebase harder to modify safely — changes risk unintended side effects across multiple subsystems.

MODULE	SUBSYSTEMS	CROSS-EDGES	TOTAL DEGREE	RISK
`log4j-core/src/.../log4j/core/LogEvent.java` [F26]	7	174	268	HIGH RISK
`log4j-api/src/m.../status/StatusLogger.java` [F28]	7	126	221	HIGH RISK
`log4j-core/src/...onfig/plugins/Plugin.java` [F27]	6	120	259	HIGH RISK
`log4j-api/src/m...g/log4j/util/Strings.java` [F31]	7	95	120	HIGH RISK

MODULE	SUBSYSTEMS	CROSS-EDGES	TOTAL DEGREE	RISK
`log4j-api/src/m...logging/log4j/Logger.java` [F8]	7	91	188	HIGH RISK
`log4j-core/src/...config/Configuration.java` [F30]	7	91	158	HIGH RISK
`log4j-api/src/m.../logging/log4j/Level.java` [F29]	7	68	176	HIGH RISK
`log4j-core/src/.../config/LoggerConfig.java` [F11]	6	36	51	HIGH RISK
`log4j-api/src/m.../util/PropertiesUtil.java` [F36]	6	34	56	HIGH RISK
`log4j-core/src/...g4j/core/config/Node.java` [F37]	5	33	78	HIGH RISK

Package Isolation

Independent packages with zero cross-dependencies. In a monorepo / workspace architecture, this indicates clean separation between packages — a positive architectural signal.

PACKAGE A	PACKAGE B	STATUS
log4j-core/src/ (business_logic, 211 modules)	log4j-perf-test/src/ (infrastructure, 2 modules)	Independent
log4j-core/src/ (business_logic, 211 modules)	log4j-api-java9/ (infrastructure, 2 modules)	Independent
log4j-core/ (infrastructure, 187 modules)	log4j-perf-test/src/ (infrastructure, 2 modules)	Independent
log4j-core/ (infrastructure, 187 modules)	log4j-api-java9/ (infrastructure, 2 modules)	Independent
log4j-core/ (business_logic, 69 modules)	log4j-perf-test/src/ (infrastructure, 2 modules)	Independent
log4j-core/ (business_logic, 69 modules)	log4j-core/src/ (business_logic, 7 modules)	Independent
log4j-core/ (business_logic, 69 modules)	log4j-api-java9/ (infrastructure, 2 modules)	Independent
log4j-perf-test/src/ (infrastructure, 2 modules)	log4j-core/src/ (business_logic, 185 modules)	Independent
log4j-perf-test/src/ (infrastructure, 2 modules)	log4j-core/ (business_logic, 167 modules)	Independent

PACKAGE A	PACKAGE B	STATUS
log4j-perf-test/src/ (infrastructure, 2 modules)	log4j-core/src/ (business_logic, 7 modules)	Independent

Structural Gaps

Subsystem pairs with very few connecting dependencies that may indicate missing integration.

SUBSYSTEM A	SUBSYSTEM B	CONNECTING EDGES
log4j-core/ (infrastructure, 187 modules)	log4j-core/src/ (business_logic, 7 modules)	1

Topology Findings

HIGH RISK **God Module:** Module 'log4j-core/src/main/java/org/apache/logging/log4j/core/LogEvent.java' connects 7 subsystems with 174 cross-boundary edges. Changes here risk cascading across the codebase.

HIGH RISK **God Module:** Module 'log4j-api/src/main/java/org/apache/logging/log4j/status/StatusLogger.java' connects 7 subsystems with 126 cross-boundary edges. Changes here risk cascading across the codebase.

HIGH RISK **God Module:** Module 'log4j-core/src/main/java/org/apache/logging/log4j/core/config/plugins/Plugin.java' connects 6 subsystems with 120 cross-boundary edges. Changes here risk cascading across the codebase.

HIGH RISK **God Module:** Module 'log4j-api/src/main/java/org/apache/logging/log4j/util/Strings.java' connects 7 subsystems with 95 cross-boundary edges. Changes here risk cascading across the codebase.

HIGH RISK **God Module:** Module 'log4j-api/src/main/java/org/apache/logging/log4j/Logger.java' connects 7 subsystems with 91 cross-boundary edges. Changes here risk cascading across the codebase.

HIGH RISK **God Module:** Module 'log4j-core/src/main/java/org/apache/logging/log4j/core/config/Configuration.java' connects 7 subsystems with 91 cross-boundary edges. Changes here risk cascading across the codebase.

HIGH RISK **God Module:** Module 'log4j-api/src/main/java/org/apache/logging/log4j/Level.java' connects 7 subsystems with 68 cross-boundary edges. Changes here risk cascading across the codebase.

HIGH RISK **God Module:** Module 'log4j-core/src/main/java/org/apache/logging/log4j/core/config/LoggerConfig.java' connects 6 subsystems with 36 cross-boundary edges. Changes here risk cascading across the codebase.

HIGH RISK **God Module:** Module 'log4j-api/src/main/java/org/apache/logging/log4j/util/PropertiesUtil.java' connects 6 subsystems with 34 cross-boundary edges. Changes here risk cascading across the codebase.

HIGH RISK **God Module:** Module 'log4j-core/src/main/java/org/apache/logging/log4j/core/config/Node.java' connects 5 subsystems with 33 cross-boundary edges. Changes here risk cascading across the codebase.

LOW RISK **Clean Package Isolation:** 25 independent package pairs with zero cross-dependencies — clean monorepo isolation.

10. Test Coverage CLEAN

Automated tests act as safety nets for the codebase. When developers make changes, tests verify nothing else has broken. Strong test coverage means the acquirer's team can modify and extend the code with confidence; weak or absent testing means changes carry a higher risk of introducing undetected problems.

Strong testing discipline — automated tests cover a significant portion of the codebase and run automatically before changes are released.

METRIC	VALUE
Test files	1088
Source files	2576
Test-to-source ratio	42.2%
Test functions / cases	3349

This is a healthy ratio, indicating testing is a routine part of development.

The project uses established testing tools (JUnit), indicating the team has invested in testing infrastructure.

> **Tests run automatically before code changes are accepted. This means new code is verified before it reaches the main codebase, reducing the chance of regressions.**

CI configuration found in: .github/workflows/build.yaml, .github/workflows/labeler.yaml, .github/workflows/close-stale.yaml, .github/workflows/deploy-site.yaml, .github/workflows/develocity-publish-build-scans.yaml, .github/workflows/codeql-analysis.yaml

11. Infrastructure & Deployment MEDIUM RISK

This section assesses whether the deployment process — how the software is built, packaged, and released — is documented in code or relies on undocumented manual steps. Codified deployment means a new team can operate the software independently; undocumented deployment creates dependency on the original developers and increases transition risk.

Limited deployment infrastructure — only basic automation exists. Significant operational knowledge may be undocumented.

Only CI/CD configuration detected. Significant tribal knowledge likely required for full deployment. New team onboarding would require substantial handover.

Infrastructure Found

CATEGORY	TOOLS / CONFIGURATION
CI/CD Automation	github-actions (6)

Gaps Identified

- No containerisation (e.g., Docker) — the application may require manual environment setup on each server
- No orchestration (e.g., Kubernetes, Docker Compose) — multi-service deployment is not automated
- No infrastructure-as-code (e.g., Terraform, CloudFormation) — server provisioning may require manual configuration
- No deployment scripts — the release process may rely on undocumented manual steps

12. Technical Debt CLEAN

Technical debt represents shortcuts, deferred work, and known problems that the development team has acknowledged but not yet fixed. Every codebase carries some debt; what matters is the volume and severity. High technical debt increases the cost of post-acquisition development and the risk that changes introduce new problems.

Minimal technical debt — the codebase is well-maintained with few acknowledged shortcuts or deferred work items.

Developers have left **169** notes in the code flagging work that needs to be done (TODO items, known bugs, temporary workarounds). That is **0.5 markers per 1,000 lines of code**. This density is typical for a well-maintained project.

Debt Markers by Type

TYPE	COUNT
TODO	136
XXX	12
FIXME	7
HACK	5
WORKAROUND	5
TEMP	4

148 markers are planned work items (TODO) while 17 flag known problems (FIXME, HACK, BUG). A high proportion of FIXME/HACK markers is more concerning than TODOs, as they indicate acknowledged broken or fragile code.

457 warning suppressions detected (1.5 per 1,000 lines) — within normal range for a codebase of this size. These represent deliberate exceptions to coding rules, not a material concern.

846 uses of deprecated APIs detected — normal for a mature codebase and not a material risk at current levels.

19 minor error handling patterns detected (empty catch blocks or broad exception handlers) — within normal range and not a material concern at this volume.

13. Governance & CI/CD Security CLEAN

Governance measures the security and maturity of development processes — CI pipeline hardening, release signing, code review enforcement, and dependency management. Weak governance increases post-acquisition remediation costs and ongoing operational risk.

Overall governance: **A** (8.4/10) [OpenSSF Scorecard + local analysis]

OpenSSF Scorecard aggregate: 8.4/10

Individual check scores below reflect Scorecard's automated assessment. Low scores on checks such as branch protection, signed releases, or code review are common for open-source projects and do not necessarily indicate a governance risk — the overall rating accounts for the project's classification and context.

CI Pipeline Security — Grade B

CHECK	SCORE	SOURCE	DETAIL
Token-Permissions	9/10	scorecard	detected GitHub workflow tokens with excessive permissions
Pinned-Dependencies	2/10	scorecard	dependency not pinned by hash detected -- score normalized to 2
Dangerous-Workflow	10/10	scorecard	no dangerous workflow patterns detected
CI-Tests	10/10	scorecard	30 out of 30 merged PRs checked by a CI test -- score normalized to 10
SAST	10/10	scorecard	SAST tool is run on all commits

CI pipelines show adequate security practices with some areas for improvement.

Release Engineering — Grade A

CHECK	SCORE	SOURCE	DETAIL
Signed-Releases	3/10	scorecard	Releases exist (7.4/yr via GitHub tags) but are not cryptographically signed. Score reflects absence of signing, not absence of releases.
Packaging	N/A	scorecard	packaging workflow not detected
Maintained	10/10	scorecard	11 commits and 19 issue activity found in the last 90 days -- score normalized to 10
release_frequency	7/10	enrichment	7.4 releases/year, 0% semver compliant

Release engineering practices are mature.

Governance Posture — Grade A

CHECK	SCORE	SOURCE	DETAIL
Security-Policy	10/10	scorecard	security policy file detected
Contributors	10/10	scorecard	project has 44 contributing companies or organizations
CII-Best-Practices	5/10	scorecard	badge detected: Passing

CHECK	SCORE	SOURCE	DETAIL
License	10/10	scorecard	license file detected

Project demonstrates strong governance practices.

Branch Protection & Code Review — Grade A

CHECK	SCORE	SOURCE	DETAIL
Branch-Protection	8/10	scorecard	branch protection is not maximal on development and all release branches
Code-Review	10/10	scorecard	all changesets reviewed

Branch protection is well configured with enforced code review.

Dependency Management — Grade B

CHECK	SCORE	SOURCE	DETAIL
Dependency-Update-Tool	10/10	scorecard	update tool detected
Vulnerabilities	0/10	scorecard	Scorecard reports 14 via OSV; Polaris found 31 by scanning declared manifest versions. Discrepancies arise from different vulnerability databases and detection methods.
Fuzzing	10/10	scorecard	project is fuzzed

Dependency management practices are adequate.

14. Engineering Maturity LOW RISK

Engineering maturity measures the project's operational health beyond source code quality — release discipline, community governance, and project signals that indicate long-term viability.

Overall maturity: **B** — minor maturity gaps only (risk rating: LOW)

Release Cadence

METRIC	VALUE
Releases per year	7.4
Days since last release	5
Semver compliance	0.0%
Grade	A

Community Health

DOCUMENT	STATUS
SECURITY.md	Present
CONTRIBUTING.md	Missing
CODE_OF_CONDUCT	Present
Issue template	Missing
PR template	Present
Health score	60%
Grade	C

Project Signals

METRIC	VALUE
Stars	3,591
Forks	1,707
Open issues	221
Contributors	100
Repository created on GitHub	2013-06-12
Grade	A

Note: *GitHub API reports 100 contributors while git history analysis (Bus Factor section) identified 30. This discrepancy arises because GitHub counts all commit authors across the full history, while git analysis may use a limited clone depth or different author-deduplication rules.*

15. Malware & Destructive Action Scan CLEAN

This scan searches for code patterns commonly associated with protestware, supply-chain attacks, and sabotage — filesystem wipers, obfuscated payloads, unauthorised network calls, and install-hook abuse. Findings are heuristic and warrant manual review rather than automatic condemnation.

Files scanned: 2,585

> **No suspicious patterns detected.**

16. Risk Summary & Recommendations

Recommendations are prioritised by their potential impact on the transaction. Immediate and Urgent items should be addressed as conditions precedent; Medium items can be scheduled into the post-acquisition integration roadmap.

PRIORITY	CATEGORY	RECOMMENDED ACTION
URGENT	Dependency Vulnerabilities	Patch 3 critical CVEs immediately. Update pinned dependency versions.
MEDIUM RISK	Dependency Health	66 abandoned or deprecated dependencies. Evaluate alternatives to reduce latent supply chain risk.

Appendix A: Raw Data

Dependency Inventory (264 packages)

ECOSYSTEM	PACKAGE	VERSION	VULNS	HEALTH	LAST RELEASE
Maven	org.apache.logging.log4j:log4j-1.2-api	12.1.1	0	unknown	2024-02-17
Maven	org.apache.logging.log4j:log4j-api	12.1.1	0	unknown	2024-02-17
Maven	org.apache.logging.log4j:log4j-api-test	12.1.1	0	unknown	2024-02-17
Maven	org.apache.logging.log4j:log4j-appserver	12.1.1	0	unknown	2023-12-19
Maven	org.apache.logging.log4j:log4j-cassandra	12.1.1	0	unknown	2023-06-18
Maven	org.apache.logging.log4j:log4j-core	12.1.1	0	unknown	2024-11-09
Maven	org.apache.logging.log4j:log4j-core-test	12.1.1	0	unknown	2024-11-09
Maven	org.apache.logging.log4j:log4j-couchdb	12.1.1	0	unknown	2023-06-18
Maven	org.apache.logging.log4j:log4j-docker	12.1.1	0	unknown	2024-11-09
Maven	org.apache.logging.log4j:log4j-flume-ng	2.23.1	0	abandoned	2024-02-17

ECOSYSTEM	PACKAGE	VERSION	VULNS	HEALTH	LAST RELEASE
Maven	org.apache.logging.log4j:log4j-iostreams	12.1.1	0	unknown	2024-02-17
Maven	org.apache.logging.log4j:log4j-jakarta-jms	12.1.1	0	unknown	2025-12-15
Maven	org.apache.logging.log4j:log4j-jakarta-smtp	12.1.1	0	unknown	2023-06-18
Maven	org.apache.logging.log4j:log4j-jakarta-web	12.1.1	0	unknown	2023-06-18
Maven	org.apache.logging.log4j:log4j-jcl	12.1.1	0	unknown	2023-12-19
Maven	org.apache.logging.log4j:log4j-jpa	12.1.1	0	unknown	2023-06-18
Maven	org.apache.logging.log4j:log4j-jpl	12.1.1	0	unknown	2024-02-17
Maven	org.apache.logging.log4j:log4j-jul	12.1.1	0	unknown	2024-11-09
Maven	org.apache.logging.log4j:log4j-layout-template-json	12.1.1	0	unknown	2024-11-09
Maven	org.apache.logging.log4j:log4j-mongodb4	12.1.1	0	unknown	2024-02-17
Maven	org.apache.logging.log4j:log4j-mongodb	12.1.1	0	unknown	2024-11-09
Maven	org.apache.logging.log4j:log4j-slf4j2-impl	12.1.1	0	unknown	2024-02-17
Maven	org.apache.logging.log4j:log4j-slf4j-impl	12.1.1	0	unknown	2024-02-17
Maven	org.apache.logging.log4j:log4j-spring-boot	12.1.1	0	unknown	2023-06-18
Maven	org.apache.logging.log4j:log4j-spring-cloud-config-client	12.1.1	0	unknown	2024-11-09
Maven	org.apache.logging.log4j:log4j-taglib	12.1.1	0	unknown	2023-06-18
Maven	org.apache.logging.log4j:log4j-to-jul	12.1.1	0	unknown	2024-02-17
Maven	org.apache.logging.log4j:log4j-to-slf4j	12.1.1	0	unknown	2024-02-17
Maven	org.apache.logging.log4j:log4j-web	12.1.1	0	unknown	2023-06-18
Maven	org.apache.commons:commons-compress	1.27.1	0	stable	2025-07-26
Maven	org.apache.commons:commons-csv	1.14.0	0	stable	2025-07-27
Maven	commons-logging:commons-logging	1.3.5	0	active	2026-03-04
Maven	com.conversantmedia:disruptor	1.2.21	0	abandoned	2022-10-18
Maven	com.lmax:disruptor	4.0.0	0	abandoned	2023-09-29
Maven	org.apache.flume:flume-ng-embedded-agent	1.11.0	0	abandoned	2022-10-16
Maven	com.fasterxml.jackson:jackson-bom	2.19.2	0	stable	2025-04-25
Maven	com.sun.mail:javax.mail	1.6.2	0	abandoned	2018-08-29
Maven	org.jctools:jctools-core	4.0.5	0	stale	2024-06-01
Maven	com.sleepycat:je	18.3.12	0	abandoned	2018-11-29
Maven	org.zeromq:jeromq	0.6.0	0	abandoned	2024-02-06
Maven	org.apache.kafka:kafka-clients	3.9.1	0	active	2026-02-10
Maven	ch.qos.logback:logback-core	1.3.15	2	stale	2025-03-18
Maven	org.slf4j:slf4j-api	2.0.17	0	abandoned	2024-01-02
Maven	org.openrewrite.recipe:rewrite-migrate-java	3.0.0	0	stable	2025-06-25

ECOSYSTEM	PACKAGE	VERSION	VULNS	HEALTH	LAST RELEASE
Maven	org.openrewrite.recipe:rewrite-logging-frameworks	3.2.0	0	stable	2025-06-25
npm	@antora/cli	3.2.0-alpha.4	0	active	2025-10-03
npm	@antora/site-generator-default	3.2.0-alpha.4	0	active	2025-10-03
npm	@asciidoctor/tabs	1.0.0-beta.6	0	abandoned	2023-08-01
npm	asciidoctor-kroki	0.18.1	0	abandoned	2023-10-11
npm	fast-xml-parser	5.0.6	5	active	2026-03-23
npm	handlebars	4.7.8	7	active	2026-03-26
Maven	org.hamcrest:hamcrest	inherited	0	stale	2024-08-01
Maven	org.junit.jupiter:junit-jupiter-engine	inherited	0	stable	2025-05-02
Maven	org.apache.logging.log4j:log4j-fuzz-test	inherited	0	unknown	
Maven	jakarta.servlet:jakarta.servlet-api	inherited	0	stale	2024-05-24
Maven	org.mockito:mockito-core	inherited	0	stable	2025-05-20
Maven	org.mockito:mockito-junit-jupiter	inherited	0	stable	2025-05-20
Maven	org.json:json	20250517	0	stable	2025-05-17
Maven	javax.servlet:javax.servlet-api	inherited	0	abandoned	2018-04-20
Maven	org.apache.logging.log4j:log4j-layout-template-json-test	inherited	0	abandoned	2024-02-17
Maven	org.apache.groovy:groovy-bom	4.0.27	0	active	2026-01-16
Maven	jakarta.platform:jakarta.jakartaee-bom	9.1.0	0	stale	2025-03-12
Maven	org.junit:junit-bom	5.13.4	0	stable	2025-05-02
Maven	org.mockito:mockito-bom	4.11.0	0	stable	2025-05-20
Maven	org.springframework:spring-framework-bom	5.3.39	0	stable	2025-06-12
Maven	org.apache.logging.log4j:log4j-api-java9	inherited	0	unknown	
Maven	org.apache.logging.log4j:log4j-core-java9	inherited	0	unknown	
Maven	org.apache.logging.log4j:log4j-transform-maven-shade-plugin-extensions	0.2.0	0	stale	2024-10-27
Maven	org.apache.activemq:activemq-broker	6.1.7	0	active	2026-03-20
Maven	org.eclipse.angus:angus-activation	2.0.2	0	abandoned	2024-02-15
Maven	org.assertj:assertj-core	3.27.3	1	stale	2025-03-09
Maven	org.awaitility:awaitility	4.3.0	0	stale	2025-02-21
Maven	com.code-intelligence:jazzer	0.24.0	0	stale	2025-01-29
Maven	com.code-intelligence:jazzer-api	0.24.0	0	stale	2025-01-29
Maven	org.apache-extras.beanshell:bsh	2.0b6	0	abandoned	2016-02-18
Maven	org.apache.cassandra:cassandra-all	3.11.19	0	active	2026-03-17

ECOSYSTEM	PACKAGE	VERSION	VULNS	HEALTH	LAST RELEASE
Maven	com.datastax.cassandra:cassandra-driver-core	3.11.5	0	abandoned	2019-03-18
Maven	org.apache.cassandra:cassandra-thrift	3.11.19	0	stale	2025-02-06
Maven	commons-codec:commons-codec	1.18.0	0	unknown	
Maven	org.apache.commons:commons-dbcp2	2.13.0	0	active	2025-12-11
Maven	commons-io:commons-io	2.19.0	0	unknown	
Maven	org.apache.commons:commons-lang3	3.18.0	0	active	2025-11-12
Maven	org.apache.commons:commons-pool2	2.12.1	0	active	2026-01-06
Maven	org.zapodot:embedded-ldap-junit	0.9.0	0	abandoned	2022-08-12
Maven	com.google.guava:guava	33.4.8-jre	0	stable	2025-04-14
Maven	com.google.guava:guava-testlib	33.4.8-jre	0	stable	2025-04-14
Maven	com.h2database:h2	2.2.224	0	stale	2024-08-12
Maven	org.hamcrest:hamcrest-core	3.0	0	stale	2024-08-01
Maven	org.hamcrest:hamcrest-library	3.0	0	stale	2024-08-01
Maven	org.hdrhistogram:HdrHistogram	2.2.2	0	stale	2024-05-30
Maven	org.hsqldb:hsqldb	2.7.4	0	stale	2024-11-03
Maven	org.apache.httpcomponents:httpclient	4.5.14	0	abandoned	2022-11-30
Maven	org.apache.httpcomponents:httpcore	4.4.16	0	abandoned	2022-11-26
Maven	jakarta.activation:jakarta.activation-api	2.1.3	0	abandoned	2024-02-15
Maven	org.eclipse.angus:jakarta.mail	2.0.3	0	abandoned	2024-02-27
Maven	jakarta.mail:jakarta.mail-api	2.1.3	0	abandoned	2024-02-16
Maven	com.google.code.java-allocation-instrumenter:java-allocation-instrumenter	3.3.4	0	abandoned	2023-12-15
Maven	javax.activation:javax.activation-api	1.2.0	0	abandoned	2017-09-07
Maven	javax.inject:javax.inject	1	0	unknown	
Maven	javax.jms:javax.jms-api	2.0.1	0	abandoned	2015-03-11

Module Dependencies (top 30)

MODULE		FAN-IN	LAYER
`log4j-core/src/.../log4j/core/LogEvent.java`	[F26]	261	business_logic
`log4j-core/src/...onfig/plugins/Plugin.java`	[F27]	258	business_logic
`log4j-api/src/m.../status/StatusLogger.java`	[F28]	213	infrastructure
`log4j-api/src/m...logging/log4j/Logger.java`	[F8]	181	infrastructure
`log4j-api/src/m.../logging/log4j/Level.java`	[F29]	174	infrastructure

MODULE		FAN-IN	LAYER
`log4j-core/src/...config/Configuration.java`	[F30]	143	business_logic
`log4j-api/src/m...g/log4j/util/Strings.java`	[F31]	120	infrastructure
`log4j-api/src/m...logging/log4j/Marker.java`	[F32]	108	infrastructure
`log4j-api/src/m...og4j/message/Message.java`	[F33]	103	infrastructure
`log4j-core/src/...lugins/PluginFactory.java`	[F34]	97	business_logic
`log4j-core/src/...g4j/core/config/Node.java`	[F37]	77	business_logic
`log4j-core/src/...ging/log4j/core/Core.java`	[F38]	75	business_logic
`log4j-core/src/...j/core/LoggerContext.java`	[F35]	74	business_logic
`log4j-core/src/...PluginBuilderFactory.java`	[F39]	73	business_logic
`log4j-core/src/...lugins/PluginElement.java`	[F40]	73	business_logic
`log4j-core/src/...ng/log4j/core/Layout.java`	[F41]	70	business_logic
`log4j-core/src/...uginBuilderAttribute.java`	[F42]	65	business_logic
`log4j-core/src/...ng/log4j/core/Filter.java`	[F43]	65	business_logic
`log4j-api/src/m...PerformanceSensitive.java`	[F44]	62	infrastructure
`log4j-core/src/...gins/PluginAttribute.java`	[F45]	61	business_logic
`log4j-api/src/m...ing/log4j/LogManager.java`	[F46]	58	infrastructure
`log4j-core/src/...core/config/Property.java`	[F47]	57	business_logic
`log4j-api/src/m.../util/PropertiesUtil.java`	[F36]	54	infrastructure
`log4j-core/src/.../log4j/core/Appender.java`	[F48]	45	business_logic
`log4j-api/src/m...j/spi/ExtendedLogger.java`	[F49]	44	infrastructure
`log4j-core/src/...n/SuppressFBWarnings.java`	[F50]	39	business_logic
`log4j-api-java9...og4j/util/LoaderUtil.java`	[F51]	38	infrastructure
`log4j-core/src/.../PluginConfiguration.java`	[F52]	37	business_logic
`log4j-api/src/m...log4j/util/StringMap.java`	[F53]	35	infrastructure
`log4j-api/src/m.../log4j/ThreadContext.java`	[F54]	35	infrastructure

Appendix B: File Reference

Full file paths for truncated references in the report.

REF	FULL PATH
F1	`log4j-core/src/main/java/org/apache/logging/log4j/core/util/CronExpression.java`

REF	FULL PATH
F2	`log4j-core/src/main/java/org/apache/logging/log4j/core/impl/ThrowableFormatOptions.java`
F3	`log4j-core/src/main/java/org/apache/logging/log4j/core/util/datetime/FastDatePrinter.java`
F4	`log4j-core/src/main/java/org/apache/logging/log4j/core/tools/picocli/CommandLine.java`
F5	`log4j-core/src/main/java/org/apache/logging/log4j/core/lookup/StrSubstitutor.java`
F6	`log4j-layout-template-json/src/main/java/org/apache/logging/log4j/layout/template/json/resolver/StackTraceStringResolver.java`
F7	`log4j-core/src/main/java/org/apache/logging/log4j/core/config/AbstractConfiguration.java`
F8	`log4j-api/src/main/java/org/apache/logging/log4j/Logger.java`
F9	`log4j-api/src/main/java/org/apache/logging/log4j/spi/AbstractLogger.java`
F10	`log4j-core/src/main/java/org/apache/logging/log4j/core/impl/Log4jLogEvent.java`
F11	`log4j-core/src/main/java/org/apache/logging/log4j/core/config/LoggerConfig.java`
F12	`log4j-core/src/main/java/org/apache/logging/log4j/core/layout/Rfc5424Layout.java`
F13	`log4j-api/src/main/java/org/apache/logging/log4j/util/internal/SerializationUtil.java`
F14	`log4j-cassandra/src/test/java/org/apache/logging/log4j/cassandra/CassandraAppenderIT.java`
F15	`log4j-cassandra/src/test/java/org/apache/logging/log4j/cassandra/CassandraRule.java`
F16	`log4j-1.2-api/src/main/java/org/apache/log4j/config/Log4j1ConfigurationConverter.java`
F17	`src/site/antora/modules/ROOT/examples/manual/migration/Migration1Example.java`
F18	`src/site/antora/modules/ROOT/examples/manual/migration/Migration2Example.java`
F19	`src/site/antora/modules/ROOT/examples/manual/markers/MarkerExample.java`
F20	`src/site/antora/modules/ROOT/examples/manual/customloglevels/LevelExample.java`
F21	`src/site/antora/modules/ROOT/examples/manual/messages/CustomMessageExample.java`
F22	`src/site/antora/modules/ROOT/examples/manual/messages/MessagesExample.java`
F23	`src/site/antora/modules/ROOT/examples/manual/lookups/MainArgsExample.java`
F24	`log4j-perf-test/src/main/java/org/apache/logging/log4j/perf/jmh/NanotimeBenchmark.java`
F25	`log4j-perf-test/src/main/java/org/apache/logging/log4j/perf/jmh/VarargsBenchmark.java`
F26	`log4j-core/src/main/java/org/apache/logging/log4j/core/LogEvent.java`
F27	`log4j-core/src/main/java/org/apache/logging/log4j/core/config/plugins/Plugin.java`

REF	FULL PATH
F28	`log4j-api/src/main/java/org/apache/logging/log4j/status/StatusLogger.java`
F29	`log4j-api/src/main/java/org/apache/logging/log4j/Level.java`
F30	`log4j-core/src/main/java/org/apache/logging/log4j/core/config/Configuration.java`
F31	`log4j-api/src/main/java/org/apache/logging/log4j/util/Strings.java`
F32	`log4j-api/src/main/java/org/apache/logging/log4j/Marker.java`
F33	`log4j-api/src/main/java/org/apache/logging/log4j/message/Message.java`
F34	`log4j-core/src/main/java/org/apache/logging/log4j/core/config/plugins/PluginFactory.java`
F35	`log4j-core/src/main/java/org/apache/logging/log4j/core/LoggerContext.java`
F36	`log4j-api/src/main/java/org/apache/logging/log4j/util/PropertiesUtil.java`
F37	`log4j-core/src/main/java/org/apache/logging/log4j/core/config/Node.java`
F38	`log4j-core/src/main/java/org/apache/logging/log4j/core/Core.java`
F39	`log4j-core/src/main/java/org/apache/logging/log4j/core/config/plugins/PluginBuilderFactory.java`
F40	`log4j-core/src/main/java/org/apache/logging/log4j/core/config/plugins/PluginElement.java`
F41	`log4j-core/src/main/java/org/apache/logging/log4j/core/Layout.java`
F42	`log4j-core/src/main/java/org/apache/logging/log4j/core/config/plugins/PluginBuilderAttribute.java`
F43	`log4j-core/src/main/java/org/apache/logging/log4j/core/Filter.java`
F44	`log4j-api/src/main/java/org/apache/logging/log4j/util/PerformanceSensitive.java`
F45	`log4j-core/src/main/java/org/apache/logging/log4j/core/config/plugins/PluginAttribute.java`
F46	`log4j-api/src/main/java/org/apache/logging/log4j/LogManager.java`
F47	`log4j-core/src/main/java/org/apache/logging/log4j/core/config/Property.java`
F48	`log4j-core/src/main/java/org/apache/logging/log4j/core/Appender.java`
F49	`log4j-api/src/main/java/org/apache/logging/log4j/spi/ExtendedLogger.java`
F50	`log4j-core/src/main/java/org/apache/logging/log4j/core/internal/annotation/SuppressFBWarnings.java`
F51	`log4j-api-java9/src/main/java/org/apache/logging/log4j/util/LoaderUtil.java`
F52	`log4j-core/src/main/java/org/apache/logging/log4j/core/config/plugins/PluginConfiguration.java`
F53	`log4j-api/src/main/java/org/apache/logging/log4j/util/StringMap.java`
F54	`log4j-api/src/main/java/org/apache/logging/log4j/ThreadContext.java`