Evaluating the State of Provenance Verification on Maven Central

Johan Nilsson <ni5@kth.se> Supervision: Vivi Andersson, Christofer Vikström, Markus Kilås

Increasing Security Risks in Maven Central

The risk of software supply chain attacks is increasing across package ecosystems [1]. As security mechanisms improve in ecosystems such as PyPI and npm, equal, if not greater **effort must be dedicated to securing Maven Central**, given its critical role in enterprise software development.

Evidence suggests that software supply chain risks in Maven Central should not be underestimated:

- Sophisticated typosquatting attack on Jackson [2]
- MavenGate, malicious dependency injection [3]
- Maven-Hijack, package order exploitation [4]

Challenges in Artifact Verification in Maven Central

Although Sigstore has been supported in Maven Central for over a year, and it contains the largest amount of healthy PGP-signed artifacts among registries [5], **consumers still lack reliable access to signing metadata** (e.g., fingerprints, issuer, subject). As a result, verification via PGP or Sigstore is manual and impractical, and metadata distribution remains largely unexplored.

Questions that should be answered...

1 Security Guarantees
What are the security guarantees of provenance and signature verification in a Maven context?

2 Barriers
What barriers hinder adoption of artifact signing verification in Maven Central?

3 Accessibility
How can tooling and standards make verification the default rather than the exception?

Methology

Simulate supply chain attacks and evaluate signing verification impact

<1%

Investigate the current adoption of Sigstore signing on Maven Central

Investigate the feasibility of obtaining signing metadata for project dependencies

Results to be continued

[1] Sonatype (2026). 2026 *State of the Software Supply Chain Report.*
[2] Charlie Eriksen (2025). *First Sophisticated Malware Discovered on Maven Central via Typosquatting Attack on Jackson*. Aikido Security.
[3] Oversecured (2024). *Introducing MavenGate: A Supply Chain Attack Method for Java and Android Applications*.
[4] Frank Reyes et al. (2025). *Maven-Hijack: Software Supply Chain Attack Exploiting Packaging Order*.
[5] Taylor R. Schorlemmer et al. (2024). *Signing in Four Public Software Package Registries: Quantity, Quality, and Influencing Factors.*