TEA supporting a Java OSS library

P. P. Karwasz, V. P. Apache Logging Services

KoalaCon 2024

TEA supporting a Java OSS library

P. P. Karwasz, V. P. Apache Logging Services

Apache Log4j

Apache Log4j is a popular OSS logging library for Java split into API and logging backend.

Position in the ecosystem:

Log4j API:

- Used by many other **libraries** and applications.
- Can be a deeply nested transitive dependency
- No external dependencies.

Log4j Core:

- Used by **applications**.
- Usually a direct runtime dependency.
- Only optional external dependencies.

Log4j Core extensions (plugins)

- Over 100 external dependencies (including test dependencies).
- Daily Dependabot PRs.

SBOMs in the Maven ecosystem

Do we need SBOMs in Maven?

- All Maven artifacts have a Project Object Model (POM) file.
- The POM file lists authors and contact information.
- The POM file lists **direct** dependencies.
- **All** dependencies can be resolved using Maven Central.
- Dependencies are immutable.

Yes, we do for:

- **Interoperability purposes.**
- **More control on the content.**

TEA for the Maven ecosystem

Currently:

- Apache Log4j publishes SBOMs using the CycloneDX Maven Plugin
- Artifacts are published to Maven Central under well-known GAV coordinates.
- Artifacts are immutable.

Is this enough?

No, it is not, because:

- **We are deploying ecosystem-independent SBOMs in an ecosystem-dependent way.**

SBOMs for a Java library

Do we need SBOMs for a Java library?

- A Java library does not embed its dependencies (usually).
- Libraries can only produce Source SBOMs.
- Source SBOMs are unsuitable for vulnerability analysis (regardless of the security scanners warnings).

Yes, we do:

- **SBOMs can contain additional links, such as VDR and VEX.**
- **We could add links to lifecycle information.**
- **We could add links to recommended (and tested) dependency versions.**

TEA for a Java library

Currently:

- Log4j publishes a link to its self-hosted CycloneDX VDR:

  https://logging.apache.org/cyclonedx/vdr.xml

- We are considering a link to a VEX and CLE file in the future.

Is this enough?

No, because:

- **The VDR file has almost no downloads.**
- **The way to find the VDR link depends on the SBOM format.**

VEX for Open Source libraries

Do we need VEX-es in OSS libraries?

- VEX-es constitute additional work for OSS maintainers.
- Users can just evaluate the source code themselves or ask us.
- We can always say the vulnerability is exploitable and advise users to upgrade the dependencies.
- We can make releases with only dependency updates.

I believe we do:

- **Many of our consumers are OSS applications.**
- **Applications embed their dependencies, so they need to release a new version for exploitable CVEs.**

TEA for VEX-es

Under which conditions are OSS maintainers willing to produce real VEX-es?

Option 1 (unlikely):

- We are paid for releasing VEX-es, proportionally to the number of our dependencies.

Option 2:

- **All our direct dependencies potentially affected publish detailed VEX-es specifying the affected feature.**
- **We have an automatic system to gather and summarize that information.**

TEA for a Java OSS library

What features does TEA provide to a Java OSS library?

- It acts as an open exchange of security information between the Maven ecosystem and the rest of the world.
- It provides an alternative to the National Vulnerability Database for high quality vulnerability reports.
- It puts us back in control of the information we give about our projects.
- It could allow us to make **fast** and **precise** assessments of the exploitability of vulnerabilities in our dependencies.